Cara Mudah dan Cepat Belajar Pengembangan Sistem Informasi

Budi Hartono. M.Kom

YAYASAN PRIMA AGUS TEKNIK

Cara Mudah dan Cepat Belajar Pengembangan Sistem Informasi

Budi Hartono. M.Kom

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Cara mudah dan Cepat Belajar Pengembangan Sistem Informasi

Penulis :
Budi Hartono, M.Kom

ISBN : 9 786336 141076

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang
Diarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin tertulis dari penerbit

Kata Pengantar

Assalamualaikum Wr.Wb

Puji Syukur ke Hadirat Allah SWT karena berkat rahmat dan hidayah-Nya Penulis dapat menyelesaikan pembuatan buku ini. Materi yang terkandung dalam buku ini dibuat dengan sesederhana mungkin sehingga pembaca mudah untuk mempelajari dan mempraktekkannya. Setelah mempelajari materi ini pembaca diharapkan dapan membuat suatu perancangan sistem yang berfungsi untuk pedoman pengembangan sistem baru yang akan di kembangkankan. Penulis menyadari buku ini masih jauh dari sempurna dan mungkin juga banyak terdapat kesalahan. Untuk itu penulis sangat mengharapkan adanya saran dari semua pihak untuk dapat lebih menyempurnakan buku ini.

Sebelum mengakhiri kata pengantar ini, ijinkanlah penulis mengucapkan terima kasih pada pihak-pihak yang memungkinkan diterbitkannya buku ini, yaitu semua staf diUniversitas STEKOM Semarang. Untuk teman teman tercinta, terima kasih atas dorongan serta cintamu.

Wassalamualaikum Wr.Wb

Penulis

Daftar Isi

Halaman Judul. i
Kata Pengantar. cii
Daftar Isi..... iii
Identitas dan Deskripsi Matakuliah.....viii
Capaian Pembelajaran dan Kemampuan akhir viii
Materi. viii

BAB-1 Konsep Dasar **Sistem**.....1
1. Difinisi Sistem. 1
2. Tujuan Sistem.....5
3. Karakterıstık Sistem.....6
4. Klasıfıkası Sistem 9
5. Jenis-jenis Sistem.....12
6. Pelaku Sistem.....13
7. Ciri-ciri Sistem.....16
8. Pengendalian Sistem.....18

BAB-2 Konsep **Sistem** Informasi. 23
i. Pengertian Data dan Informasi 24
2. Siklus Informasi. 27
3. Kualitas Informasi. 29
4. Nilai Informasi 30
5. Kesalahan Informasi 33
6. Konsep Sistem Informasi.....34
7. Perkembangan Sistem Informasi.....36
8. Tujuan Sistem Informasi 38
9. Komponen Sistem Informasi.....41
10. Elemen Sistem Informasi.....44

BAB-3 **Pengembangan Sistem Informasi**.....46
1. Pengertian Pengembangan S I 47
2. Perlunya Pengembangan S I.....47
3. Tim Pengembang S I 51
4. Prinsip Pengembangan S I.....52
5. Siklus Hidup Pengembangan S I. 52

BAB-4 Metode Pengembangan Sistem Informasi.....61
1. SDLC. 62
2. Waterfall.....68
3. Prototyping.....71

4. RAD 73
S. Model Spiral. 76
6. Model V 78
7. End-User Development 81
8. Metode Outsourcing. 83

BAB-5 Tahapan Pengembangan Sistem 86
1. Perencanaan 87
2. Analisis 94
3. Perancangan 113

BAB-6 Tahapan Pengembangan **Sistem.....** 123
1. Testing and Implementasi. 123
2. Tahapan Penerapan..... 128
3. Pemeliharaan. 142

BAB-7 Metode RND 152
1. Pengertian RND. 153
2. Karakteristik RND.....155
3. Langkah Langkah RND 157
4. Kelebihan dan Kekurangan 167
S. Prosedure Pengembangan. 168
6. Laporan RND. 169
7. Sistematika RND 169

BAB-8 Metode SDLC, PROTOTYPE, RAD 172
1. SDLC.....172
2. Prototype.....191
3. RAD.....204

BAB-9 DFD.....209
1. Konsep Pengembangan Tersetruktur. 209
2. DFD.....210
3. Pengertian DFD. 211
4. Latar Belakang 211
S. Tujuan 211
6. Simbol DFD.....212
7. Syarat-syarat.....217
8. Langkah-langkah DFD.....219
9. Kesalahan-kesalahan DFD.....221
10. Bentuk Umum DFD 222

BAB-10 Normalisasi. 223
1. Pengertian Normalisasi 223
2. Tujuan Normalisasi 225
3. Tahapan. 226
4. Defendenci. 232
S. Denormalisasi 233
6. Keuntungan dan Kerugian 234

BAB-11 ERD 235
1. Pengertian ERD 235
2. Komponen ERD 236
3. Derajat Relasi 238
4. Cara Membuat 239
S. Tipe Entitas. 243

BAB-12 Desain I/O 246
1. Pengertian Desain I/O. 246
2. Perancangan Dialog. 250
3. Perancangan Database 260
4. Desain Output. 262

BAB-13 UML. 277
1. Pendahuluan. 278
2. Apa Itu UML 278
3. Konsep Dasar UML. 280
4. Use Case Diagram. 281
S. Class Diagram. 283
6. Hubungan Antar Class. 284
7. Statechart Diagram 285
8. Activity Diagram 286
9. Squense Diagram 287
10. Componen 288
11. Development Diagram 288
12. Langkah-langkah UML 288
13. Tool yang mendukung UML 290

BAB-14 Easy Case. 291
1. Easy Case. 291
2. Project dan Contex Diagram 292
3. DFD Leveled-0. 302
4. Child DFD (DFD Leveled-1) 302
5. Rule Check dan Rule Balance 304
6. User Acces Control 306

7. Hapus User. 308

BAB-15 Rational Rose..... 309
1. Function Requirement..... 311
2. Domain Model. 312
3. Use Case. 314
4. Requirement Review..... 317
S. Robustness Analysis..... 318
6. Preliminary Design Review. 322
7. Tecknical Arsitektur..... 323
8. Squence Diagram. 324
9. Critical Desain Review. 329
10. Coding..... 331

Daftar Pustaka..... 333

JUDUL BUKU

CARA MUDAH DAN CEPAT BELAJAR PENGEMBANGAN SISTEM INFORMASI

DISKRIPSI BUKU

Buku ini berisikan pengenalan pengertian dasar, peran sistem analis, tahapan dan hal-hal yang harus dilakukan dalam melakukan analisis pengembangan sistem informasi. Fokus analisis dilakukan pada tahap investigasi awal, Analisis masalah, Analisis kebutuhan, analisis keputusan dan analisis kelayakan dan sistem proposal.

Buku ini memaparkan pengetahuan tentang konsep analisis dan perancangan dalam sistem informasi. Ruang lingkup yang menjadi kajian matakuliah ini meliputi (1) konsep dasar data dan sistem informasi, (2) Pengembangan sistem informasi, (3) Metode pengembangan sistem, (4) Tahapan sistem, (5) Metode pengembangan dan penelitian, (6) konsep perancangan sistem informasi dengan model terstuktur, (7) konsep perancangan sistem informasi berorientasi objek.

CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menganalisis proses sistem yang sedang berjalan dan mampu melakukan perancangan sistem informasi menggunakan pemodelan tersetruktur dan pemodelan berorientasi objek.

2. Mahasiswa mampu dan dapat memahami konsep-konsep Analisa dan Perancangan Sistem Informasi kemudian mampu menyusun proposal proyek sistem informasi dan membuat desain/ rancangan sistem informasi baik dengan pendekatan tersetruktur maupun dengan pendekatan beorientasi objek

KEMAMPUAN AKHIR YANG DIHARAPKAN

1. Mahasiswa mampu menganalisis proses bisnis yang berjalan pada organisasi atau perusahaan
2. Mahasiswa mampu menjelaskan konsep perancangan sistem informasi berbasis data maupun objek
3. Mahasiswa mampu menjelaskan langkah-langkah perancangan Sistem Informasi berorientasi Objek
4. Mahasiswa mampu melakukan perancangan sistem informasi menggunakan DFD dan UML

MATERI

Kajian matakuliah ini meliputi (1) konsep dasar sistem informasi, (2) Gambaran umum sistem informasi , (3) analisa perancangan sistem, (4) Sistem develompment live cycle (5) Prototype dan RAD, (6) pedoman-pedoman dalam *flowchart*, (7) konsep perancangan sistem informasi dengan model terstuktur(DFD) , (8) normalisasi, (9), entity realtion diagram – ERD, (10) desain input dan output , (11) konsep perancangan sistem informasi berorientasi objek –UML, (12) pratikum pembuatan model tersetrutur DFD dengan tools easy case dan (13-14) pratikum pembuatan model UML dengan rational rose.

BAB-1

KONSEP DASAR SISTEM

Overview

Pada bab ini menjelaskan mengnai konsep dasar sistem antara lain : definisi sistem, subsistem, supersistem / suprasistem, karakteristik sistem, klasifikasi sistem, ciri ciri sistem serta pengendalian sistem.

Tujuan

1. Mahasiswa mengetahui definisi dan sejarah perkembangan sistem, perbedaan pendekataan sistem baik secara prosedure maupun yang menekanan pada komponen
2. Mahasiswa mempunyai gambaran umum mengenai Komponen dan karakteristik sistem.
3. Mahasiswa mempunyai gambaran umum mengenai pengendalian dari sebuah sistem.

Materi

1. Definisi Sistem

Sistem adalah kumpulan / komponen apapun baik phisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

Sistem adalah sebagai satu kesatuan yang terdiri dari komponen - komponen atau subsistem yang tertata dengan teratur, saling interaksi, saling ketergantungan satu dengan yang lainnya, dan tidak dapat dipisahkan (integratif) untuk mewujudkan suatu tujuan.

Sistem adalah kumpulan elemen elemen atau komponen komponen atau subsistem subsistem yang saling berintegrasi dan berinteraksi untuk mencapai tujuan tertentu. Sebuah sistem memiliki sistem yang lebih besar yang dinamakan supersistem / suprasistem , supersistem / suprasistem dimungkinkan memiliki sistem yang lebih besar, sehingga dinamakan super / supra dari supersistem / suprasistem.

Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut ini :

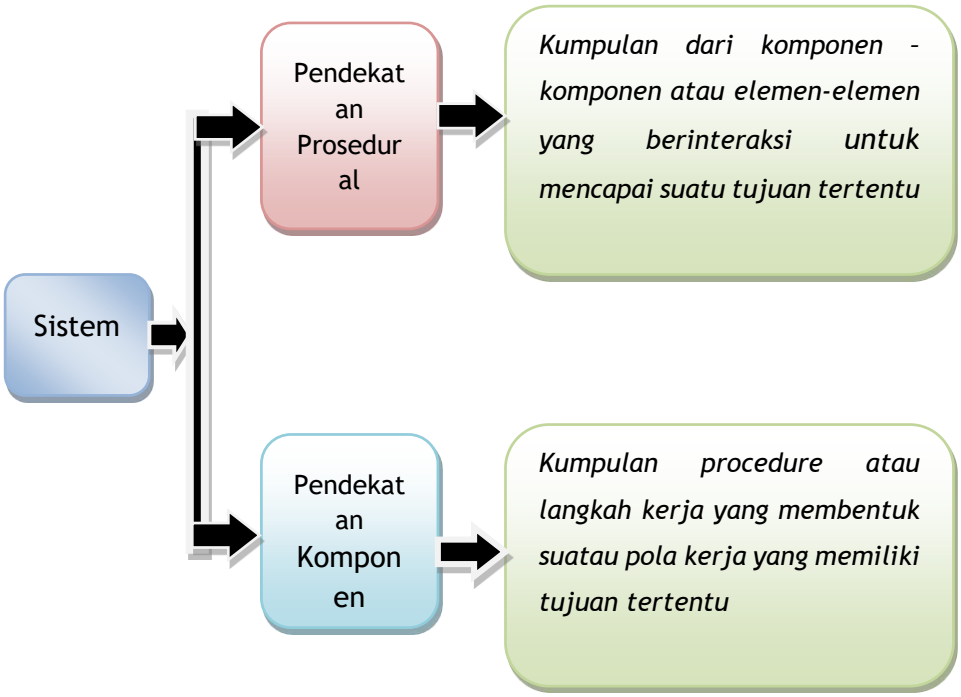
> *Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu*

Prosedur dapat diartikan sebagai : *Suatu prosedur adalah urut-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan Apa (What) yang harus dikerjakan, Siapa (Who) yang mengerjakannya, Kapan (When) dikerjakan dan Bagaimana (How) mengerjakannya*

Pendekatan system yang lebih menekankan pada elemen atau komponennya mendefinisikan system sebagai berikut ini :

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu

Dua pendekatan ini menunjukkan bahwa setiap sistem terdiri dari struktur sistem dan proses sistem. Struktur sistem adalah komponen - komponen yang membentuk sistem tersebut, sedangkan proses sistem adalah yang menjelaskan tata kerja setiap komponen tersebut untuk mencapai tujuan. Jadi, di dalam suatu sistem selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem memiliki sifat-sifat dari sistem dan melakukan suatu fungsi tertentu serta mempengaruhi proses dari sistem secara keseluruhan. Apabila suatu komponen atau suatu subsistem tersebut tidak melakukan fungsinya sesuai sifat dari sistem itu maka kegiatan dari sistem tersebut akan terganggu dan tidak efektif dalam mencapai tujuannya..



Gambar 1.1. Pendekatan Konsep Dasar Sistem

Apa itu Subsistem ?

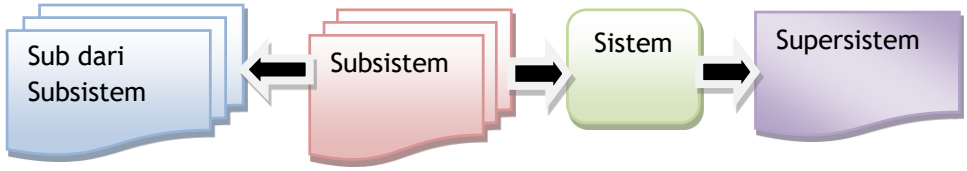
Subsistem adalah serangkaian kegiatan yang dapat ditentukan identitasnya yang berhubungan dalam suatu sistem, jadi subsistem sebenarnya hanyalah sistem di dalam suatu sistem, ini berarti bahwa sistem berada pada lebih dari satu tingkat. Pemisalan lainnya, mobil adalah suatu sistem yang terdiri dari sistem-sistem bawahan seperti sistem mesin, sistem badan mobil dan sistem rangka. Masing-masing sistem ini terdiri dari sistem tingkat yang lebih rendah lagi. Misalnya : Keluarga itu sendiri merupakan subsistem dari sistem Rukun Tetangga (RT), RT merupakan subsistem dari Rukun Warga (RW), RW subsistem dari suatu Kelurahan, Kelurahan subsistem dari suatu Kecamatan, dan demikian seterusnya

Apa itu Supersistem / Suprasistem?

Walaupun istilah supersistem / suprasistem jarang digunakan, sistem seperti ini ada. Jika suatu sistem adalah bagian dari sistem yang lebih besar, sistem yang lebih besar itu adalah supersistem. Contohnya, pemerintahan kota adalah suatu sistem, tetapi ia juga merupakan bagian dari sistem yang lebih besar – pemerintahan propinsi. Pemerintahan propinsi adalah supersistem dari pemerintahan kota dan juga merupakan subsistem dari pemerintahan nasional.

Dari definisi dan penjelasan diatas dapatlah diambil kesimpulan, suatu sistem terdiri dari elemen yang bisa berbentuk individu atau bagian-bagian yang terpisah, kemudian berinteraksi satu sama lain untuk mencapai tujuan. Keluarga, pertama kali terdiri dari 2 individu yang terpisah yang mana individu itu sendiri merupakan suatu sistem yang terdiri dari subsistem-subsistem, kemudian bersatu membentuk keluarga untuk mencapai suatu

tujuan. Keluarga itu sendiri merupakan subsistem dari sistem Rukun Tetangga (RT), RT merupakan subsistem dari Rukun Warga (RW), RW subsistem dari suatu Kelurahan, Kelurahan subsistem dari suatu Kecamatan, dan demikian seterusnya.



Gambar 1.2. Subsistem , Sistem , Supersistem

Dari pengertian di atas, dapat disimpulkan bahwa ciri-ciri dari pengertian sistem terdiri dari sebagai berikut ini.

- ✓ Setiap sistem terdiri dari unsur-unsur atau subsistem-subsistem.
- ✓ Subsistem-subsistem tersebut merupakan bagian terpadu dari sistem itu sendiri.
- ✓ Subsistem saling berhubungan dan saling ketergantungan untuk mencapai tujuan dari sistem.
- ✓ Suatu sistem merupakan bagian dari sistem yang lebih besar.

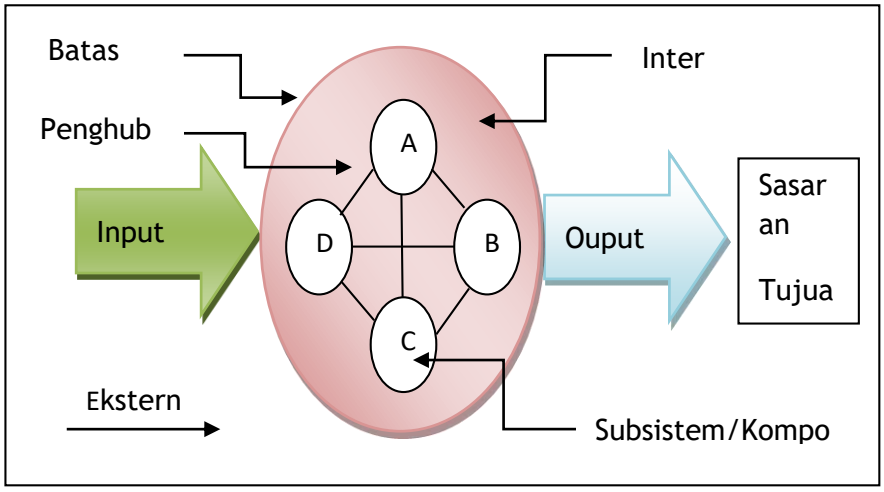
2. Tujuan Sistem

- ✓ Suatu sistem yang dibuat tentunya memiliki maksud tertentu. Sistem dibuat untuk mencapai suatu tujuan (goal) dan sasaran (objective) agar operasi sistem mempunyai arti / berguna.
- ✓ Sasaran akan menentukan input yang diperlukan serta output yang diharapkan untuk memenuhi sasaran / tujuan tersebut
- ✓ Tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran biasanya dalam ruang lingkup yang lebih sempit.

- ✓ Tiga tujuan utama yang berhubungan dengan Sistem Informasi:
  - untuk mendukung fungsi kepengurusan manajemen
  - untuk mendukung pengambilan keputusan manajemen
  - untuk mendukung kegiatan operasi perusahaan
- ✓ Secara lebih spesifik, tujuan sistem informasi bergantung pada kegiatan yang ditangani.
  - Bank?
  - Pasar Swalayan?
  - Toko Online?

3. Karakteristik Sistem

Karakteristik system dapatlah digambarkan sebagai berikut :



Gambar 1.3. Karakteristik Sistem

Komponen Sistem (Components)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-

bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan/berfungsi sebagaimana mestinya. Tentunya sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

Batas Sistem (Boundary)

Batas sistem (boundary) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

Lingkungan Luar Sistem (Environments)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

Penghubung (Interface) Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke yang lainnya. Keluaran (output) dari satu subsistem akan menjadi masukan (input) untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

Masukan (Input) Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

Keluaran (Output) Sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supersistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

Pengolah (Process) Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

Sasaran (Objectives) atau Tujuan (Goal)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempnyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Perbedaan suatu sasaran (objectives) dan suatu tujuan (goal) adalah, goal biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem utama, seperti misalnya sistem bisnis perusahaan, maka istilah goal lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem lainnya yang merupakan bagian atau subsistem dari sistem bisnis, maka istilah objectives yang lebih tepat. Jadi tergantung dari ruang lingkup mana memandang sistem tersebut. Seringkali tujuan (goal) dan sasaran (objectives) digunakan bergantian dan tidak dibedakan.

4. Klasifikasi Sistem

Sesuatu dikatakan sebagai suatu sistem apabila memiliki sifat-sifat tertentu seperti dikemukakan oleh Jogiyanto, sistem memiliki karakteristik atau sifat-sifat tertentu, yakni berikut ini.

Sistem Abstrak (abstract system) dan Sistem Fisik (physical system)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan.

Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.

Sistem Alamiah (natural system) dan sistem Buatan Manusia (human made system)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.

Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan human-machine system atau ada yang menyebut dengan man-machine system. Sistem informasi merupakan contoh man-machine system, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

Sistem Tertentu (deterministic system) dan Sistem tak Tentu (probabilistic system)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah

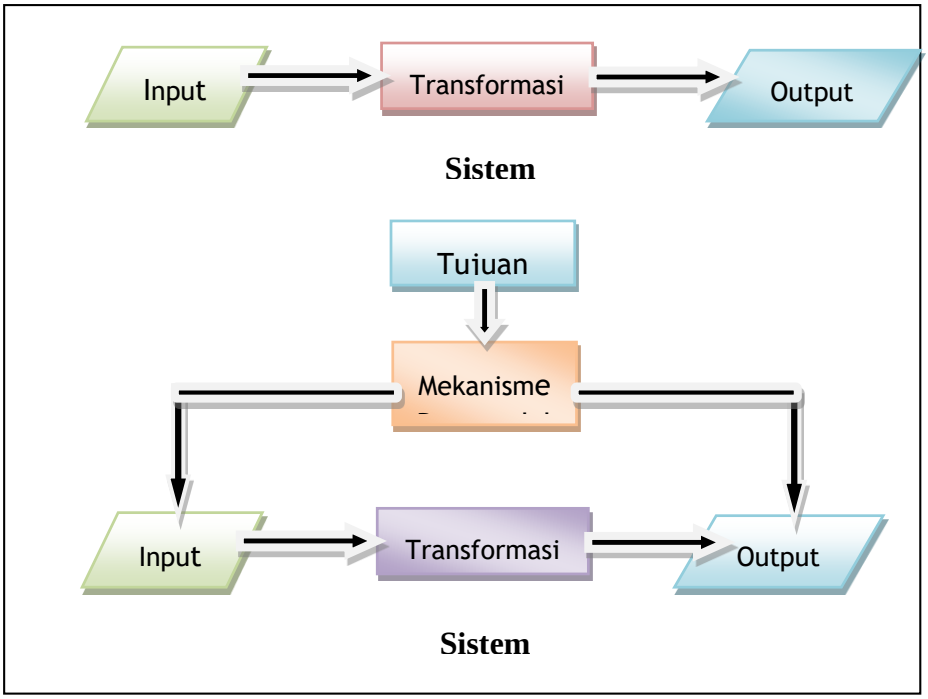
contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan.

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem Tertutup (closed system) dan Sistem Terbuka (open system)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah relatively closed system (secara relatif tertutup, tidak benar-benar tertutup).

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.Klasifikasi sistem terbuka dan tertutup dapat digambarkan sebagai berikut :



Gambar 1.4. Sistem Terbuka dan Tertutup

5. Jenis Jenis Sistem

- ***On-Line System***

  On-line systems adalah *sistem yang menerima secara langsung masukan* pada area dimana mereka dimasukan dan menghasilkan keluaran (yang dapat berupa hasil komputasi) di area, dimana mereka diperlukan.

  Area sendiri dapat terpisah-pisah dalam skala misalnya kilometer, biasanya digunakan bagi reservasi angkutan udara, perbankan dan lain-lain.

- ***Real-time systems***

  Real-time systems adalah sistem yang mekanisme pengontrolan, perekaman data dan pemrosesan yang sangat cepat dengan hasil

yang dapat diterima dalam waktu yang relatif sama. Perbedaannya dengan sistem on line adalah satuan waktu yang digunakan real-time biasanya seperseratus atau seperseribu detik sedangkan on-line masih dalam skala detik. Digunakan untuk sistem airport traffic controller, peluru kendali dan lain-lain.

Perbedaan lainnya; on line biasanya berinteraksi dengan pemakai, sedangkan realtime berinteraksi dengan pemakai dan lingkungan yang dipetakan

- ***Deccission Support System***

  Sistem yang memproses transaksi organisasi seara harian, membantu para manajer mengambil keputusan, mengevaluasi dan menganalisis tujuan organisasi. Digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi dan sistem produksi.

- ***Knowledge-based Systems***

  Program komputer yang dibuat mendekati kemampuan dan pengetahuan seorang pakar. Umumnya menggunkan perangkat keras dan perangkat lunak khusus seperti LISP dan PROLOG

6. Pelaku Sistem

- ***Pemakai***

  Pemakai adalah pelaku yang terpenting karena sistem diciptkaan untuk pemakai dan melalui komunikasi dengan pemekai sistem dibuat dan dirancang hingga mencapai bentuk terakhir.

  Pada umumnya pemakai dibagi menjadi tiga jenis, yaitu :

- ✓ Operasional, yang penting adalah bagaimana warnanya, seperti apa perangktaa masukannya dan hal spesifik lainnya (umumnya user tidak akrab dengan sistem secara keseluruhan dan cenderung lebih keraah sistem secara detil)
- ✓ Pengawas, yang penting bagaimana mereka dapat mengawasi sistem bekerja dan bagaimana performansi pemakai operasionl mengoperasikan sistem.
- ✓ Eksekutif, secara umum mereka tidak terlalu akrab dan terkait secara langsung dengan pengembanan sistem. Biasanya lebel ini lebih berminat pada ynaga sifatnya strategic dan konsep sistem secara global, misalnya kesempatan untuk berkompetisi, produk baru yang bisa mereka dapatkan dari sistem baru.

- ➢ ***Manajemen***

  Umumnya terdiri dari tiga jenis manajemen, yaitu
  manajemen pemakai (menangani pemakaian dimana sistem baru diterapkan),
  manajemen sistem (terlibat dalam pengembangan sistem itu sendiri) dan manajemen umum (yang terlibat dalam strategi perencanaan sistem dan sistem pendukung pengambilan keputusan)

- ➢ ***Pemeriksa***

  Ukuran sistem yang dikerjakan dan bentuk alami organisasi dimana sistem tersebut diimplementasikan dapat menentukan kesimpulan perlu tidaknya pemeriksa. Pemerika biasanya

menentukan segala sesuatunya berdasarkan ukuran standar yang dikembangan pda banyak perusahaan sejenis

- ***Penganalisa Sistem***

  Fungsi-fungsinya adalah: sebagai arkeolog (yang menelusuri bagaimana sebenarnya sistem lama berjalan, bagaimana sistem tersebut dijalankan dan segala hal yang menyangkut sistem lama), inovator (membantu mengembangkan dan membuka wawasan pemakai bagi kemungkinan-kemungkinan lain), mediator (menjalankan fungsi komunikasi dari semua level yaitu pemakai, manajer, pemrogram, pemeriksa dan pelaku sistem lainnya yang mungkin belum mempunyai sikap dan cara pandang yang sama), pimpinan proyek (sebagai personil yang lebih berpengalaman dari pemrogram dan ditunjuk dalam pekerjaan lebih dulu sebelum pemrogram bekerja)

- ***Programer***

  Setelah penganalisa sistem memberikan hasil kerjanya dan kemudian diolah oleh pendesain sistem baru, pemrograman dapat mulai bekerja.

- ***Pendesaian Sistem***

  Pendesain sistem menerima hasil penganalisa sistem (kebutuhan pemakai tidak berorientasi pada teknologi tertentu), yang kemudian ditransformasikan ke desain arsitektur tingkat tinggi dan tidak diformulasikan oleh pemrograman.

- ***Personil Pengoperasian***

  Bertugas dan bertanggung jawab di pusat komputer misanya jaringan, keamanan perangkat keras, keamanan perangkat lunak, pencetakan dan back-up.

  Kadang-kadang pelaku sistem ini memang tidak diperlukan karena sistem yang berjalan tidak besar dan tidak membutuhkan kualifikasi khusus untuk menjalankan sistem.

7. Ciri-ciri Sistem

Untuk mengetahui sesuatu itu sebuah sistem atau bukan, antara lain dapat dilihat dari ciri-cirinya. Ada beberapa rumusan mengenai ciri-ciri sistem ini yang pada dasarnya satu sama lainnya saling melengkapi. Pada umumnya ciri-ciri sistem itu adalah: bertujuan, punya batas, terbuka, tersusun dari subsistem, ada saling keterikatan dan saling tergantung, merupakan satu kebulatan yang utuh, melakukan kegiatan transformasi, ada mekanisme kontrol, dan memiliki kemampuan mengatur dan menyesuaikan diri sendiri.

Kita ketahui ada unsur-unsur sistem yang berpadu jadi satu, ada tujuan yang hendak dicapai oleh sistem, ada kegiatan yang dilakukan sistem, ada sesuatu yang diolah atau diproses sistem, dan ada sesuatu hasil yang dibuat sistem.

Ciri-ciri sistem antara lain: bertujuan, punya batas, terbuka, tersusun dari subsistem, ada saling keterikatan dan saling tergantung, merupakan satu kebulatan yang utuh, ada mekanisme kontrol, dan memiliki kemampuan mengatur dan menyesuaikan diri sendiri.

Untuk lebih jelasnya rumusan mengenai ciri-ciri pokok sistem itu adalah sebagai berikut:

- ✓ Setiap sistem mempunyai tujuan.
- ✓ Setiap sistem mempunyai batas (boundaries) yang memisahkannya dari lingkungannya.
- ✓ Walau sistem itu mempunyai "batas", akan tetapi sistem itu bersifatterbuka, dalam arti berinteraksi juga dengan lingkungannya.
- ✓ Suatu sistem terdiri dari beberapa subsistem yang biasa pula disebut bagian, unsur, atau komponen.
- ✓ Walau sistem itu terdiri dari berbagai bagian, unsur-unsur atau komponen, tidak berarti bahwa sistem itu merupakan sekedar kumpulan dari bagian, unsur atau komponen tersebut, melainkan merupakan suatu kebulatan yang utuh dan padu, atau mempunyai sifat wholism atau di dalam lingkungan.
- ✓ Terdapat saling hubungan dan saling ketergantungan baik di dalam (intern) sistem, maupun antara sistem dengan lingkungannya.
- ✓ Setiap sistem melakukan kegiatan atau proses transformasi atau proses mengubah masukan menjadi keluaran. Karena itu maka sistem sering disebut sebagai "processor" atau "transformator".
- ✓ Di dalam setiap sistem terdapat mekanisme kontrol dengan memanfaatkan tersedianya umpan balik.
- ✓ Karena adanya mekanisme kontrol itu maka sistem mempunyai kemampuan mengatur diri sendiri dan menyesuaikan diri dengan lingkungannya atau keadaan secara otomatik (dengan sendirinya).

8. Pengendalian Sistem

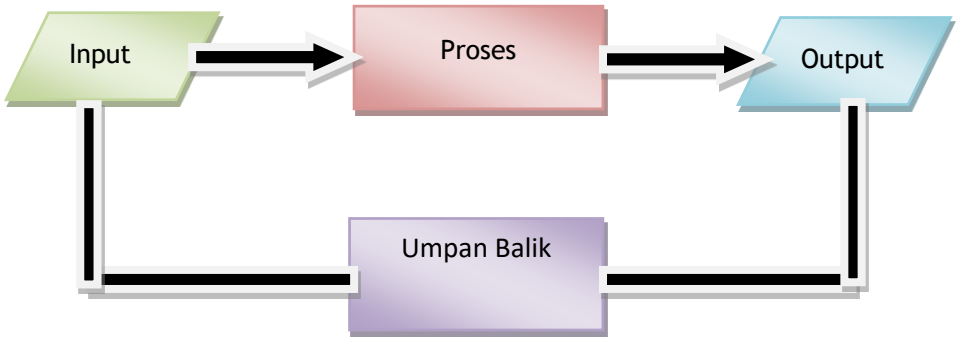
Karena suatu sistem tidak ada yang tertutup, supaya sistem dapat terus melangsungkan hidupnya, maka sistem harus mempunyai daya membela diri/ atau sistem harus mempunyai sistem pengendalian. Pengendalian dari suatu sistem dapat berupa : pengendalian umpan balik (*feedback control system*), dan pengendalian umpan maju (*feedforward control system*) dan pengendalian pencegahan (*preventive control system*).

Pengendalian Umpan Balik

Bentuk dari sistem yang sederhana :

Gambar 1.5. Sistem Sederhana

Untuk maksud pengendalian, dapat ditambahkan suatu sistem pengendalian umpan balik :

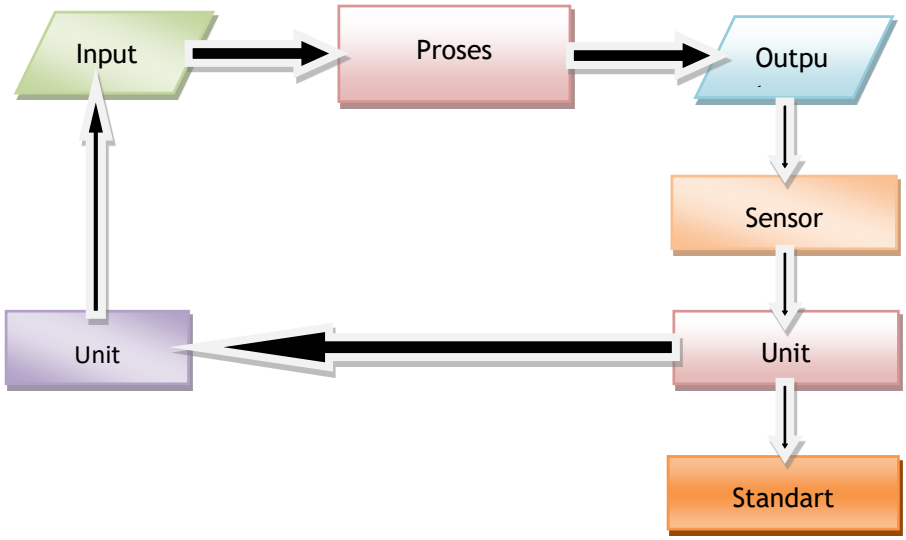


Gambar 1.6. Sistem Sederhana dengan Umpan Balik

Pengendalian umpan balik merupakan proses mengukur keluaran dari sistem yang dibandingkan dengan suatu sistem standar tertentu. Bilamana terjadi perbedaan-perbedaan akan dikoreksi untuk memperbaiki masukan sistem selanjutnya. Studi teoritis tentang sistem pengendalian umpan balik.

Sistem pengendalian umpan balik mempunyai 4 komponen dasar yaitu :

- Suatu karakteristik/kondisi yang dikendalikan/diukur dari keluarannya
- Suatu sensor yang mengukur karakeristik/kondisi tersebut
- Suatu unit pengendali (*control unit*) yang membandingkan hasil ukuran sensor dengan suatu standar
- Suatu unit pengatur (*activating unit*) yang menghasilkan tindakan penyesuaian untuk masukan proses selanjutnya.



Gambar 1.7. Komponen Umpan Balik

Sistem pengendalian umpan balik :

istilah negative feedback, karena hasil balik yang negatif akan dikendalikan supaya menjadi baik untuk masukan proses selanjutnya.

Contoh : sistem thermostat pada AC. Kondisi temperatur yang dihasilkan oleh alat pendingin akan diukur oleh suatu sensor dan dibandingkan dengan standar temperatur yang tidak menyebabkan ruangan menjadi lembab dengan standar temperatur yang tidak menyebabkan ruangan menjadi lembab. Bila temperatur terlalu dingin maka tungku pemanas sebagai unit pengatur dalam thermostat akan dihidupkan. Bila temperatur terlalu panas, maka tungku dimastikan dan alat pendingin akan bekerja kembali.

Pengendalian Umpan Maju

Disebut juga *positive feedback. Positive feedback* mencoba mendorong proses dari sistem supaya menghasilkan hasil balik yang positif. Sistem ini merupakan pengembangan dari sistem pengendalian umpan balik. Di dalam sistem umpan balik, pengendalian dilakukan setelah keluaran dihasilkan. Pengendalian seperti ini dianggap mempunyai kelemahan bilamana penyimpangan dari keluaran dengan standar sangat besar. Padahal keluaran ini merupakan hasil yang sudah terlanjur terjadi dan dapat mengakibatkan hal yang sangat fatal Supaya keluaran dapat dihasilkan umpan balik yang positif, maka pengendalian tidak boleh diukur dari keluarannya, tetapi diukur dan dikendalikan dari prosesnya. Selama proses terjadi di dalam sistem, selalu dilakukan pengamatan dan cepat-cepat diatasi bila mulai terjadi penyimpangan sebelum terlanjur fatal pada keluarannya

Contoh : sistem perencanaan kas.

Bila menggunakan sistem pengendalian umpan balik, akan sangat berbahaya dan bisa berakibat fatal bilamana saldo kas suatu ketika menjadi sangat kecil/defisit, sehingga aktifitas perusahaan akan terganggu. Untuk mengatasi hal itu, maka penerapan sistem pengendalian umpan maju dilakukan dengan meramalkan arus dari saldo kas di masa mendatang dengan membuat sistem anggaran kas. Kondisi yang dikendalikan : semua proses kegiatan perusahaan dan keputusan-keputusan yang mempengaruhi tingkat saldo kas. Proses ini diukur dari arus kas yang diperbolehkan dalam anggaran kas yang telah dibuat. Bila saldo kas yang terjadi berada di luar batas saldo kas yang telah dianggarkan, maka cepat-cepat dilakukan tindakan pengaturan untuk memperbaiki saldo kas. Sebagai pengatur : manajer keuangan yang akan melakukan tindakan penyesuaian terhadap saldo kas.

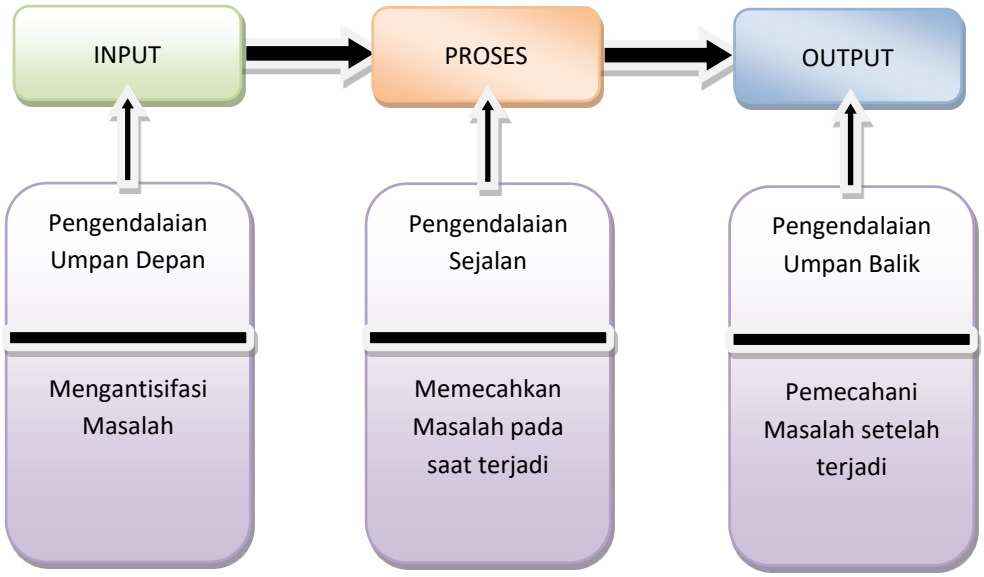
Pengendalian Pencegahan

Pengendalian pencegahan digunakan untuk mencegah terjadinya suatu masalah dalam pencapaian sebuah tujuan. Pengendalian ini dirancang untuk mencegah hasil yang tidak diinginkan dalam pencapaian sebelum kejadian tersebut terjadi dengan kata lain pengendalian pencegahaan mencoba untuk mengendalikan sistem di muka sebelum proses dimulai dengan mencegah hal-hal yang merugikan untuk masuk ke dalam sistem. Pengendalian pencegahan ini akan berjalan dengan efektif apabila fungsi dan personel menjalankan sesuai dengan perannya.

Ketika dirancang ke dalam sistem, pengendalian pencegahan memperkirakan kesalahan yang mungkin terjadi sehingga mengurangi biaya perbaikannya. Namun, pengendalian pencegahan tidak dapat menjamin tidak terjadinya kesalahan atau kecurangan sehingga masih dibutuhkan pengendalian lain untuk melengkapinya

Contoh

sistem pengendalian intern (*internal control)*' dimana penerapan kebijaksanaan kebijaksanaan metode - metode dan prosedur prosedure didalam sistem pengendalian intern dimaksudkan untuk mencega hal-hal yang tidak baik yang mengganggu masukan proses dan hasil dari sistem supaya sistem dapat beroperasi seperti yang disarankan.



Gambar 1.8. Model Pengendalian Sistem

BAB-II

KONSEP SISTEM INFORMASI

Overview

Pada bab ini menjelaskan konsep dasar sistem informasi yang terdiri atas Data, Informasi, Konsep Sistem Informasi, Sikulus informasi, Kualitas informasi, Nilai Informasi, Kesalahan Informasi, Sejarah perkembangan Informasi, Tujuan Informasi, Komponen Sistem Informasi, elemen Sistem Informasi dan arsitektur Sistem Informasi.

Tujuan

1. Mahasiswa mampu membedakan perbedaan data dan informasi, Siklus informasi, kualitas dari sistem informasi dan Nilai dari sebuah sistem informasi.
2. Mahasiswa mempunyai kempuan untuk menjelaskan mengenai Konsep sistem informasi, komponen sistem informasi, kesalah pada sistem informasi, perekembangan sistem informasi.
3. Mahasiswa mampu memahami mengenai elemen dan arsitektur sistem Informasi.

Materi

1. Pengertian Data dan Informasi

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh. Keadaan dari sistem dalam hubungannya dengan keberakhirannya disebut dengan istilah entropy. Informasi yang berguna bagi sistem akan menghindari proses entropy tersebut yang disebut dengan negative entropy atau negentropy. Apakah sebenarnya informasi itu, sehingga sangat penting artinya bagi suatu sistem?. Untuk memahami mengenai informasi, kita bahas mengenai data terlebih dahulu sebab sumber dari informasi adalah data.

Pengertian Data

Data merupakan bentuk jamak dari data-item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (event) adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi. Misalnya penjualan adalah transaksi perubahan nilai barang menjadi nilai uang atau nilai piutang dagang. Kesatuan nyata (fact) adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

> *Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.(Jogiyanto, 2011),* Kejadian-kejadian (event) adalah sesuatu yang terjadi pada saat tertentu.

Kamus Umum Bahasa Indonesia (KUBI), "Data adalah bukti yang ditemukan dari hasil penelitian yang dapat dijadikan dasar kajian atau pendapat".

Data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau pada lingkungan fisik sebelum diolah ke dalam suatu format yang dapat dipahami dan digunakan orang. (Kenneth C. Laudon ,2005). Informasi adalah *data yang sudah dibentuk ke dalam format yang memiliki arti bagi manusia*.

Pengertian Informasi

Informasi adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan. (Jogiyanto, 2011)

Informasi adalah data yang sudah dibentuk ke dalam format yang memiliki arti bagi manusia. (Kenneth C. Laudon, 2005)

Terdapat beberapa definisi Informasi , antara lain :

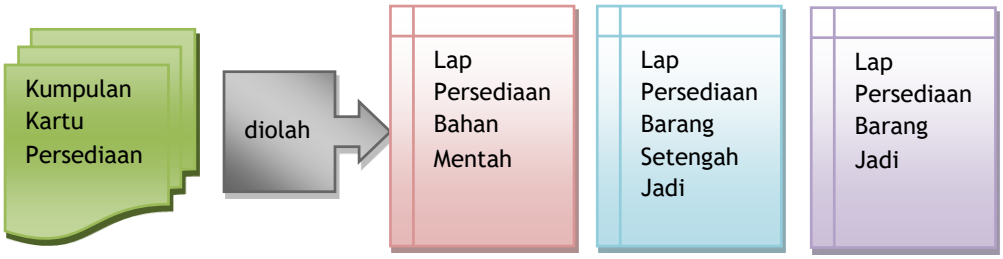
- ✓ Data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.
- ✓ Sesuatu yang nyata atau setengah nyata yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian, sebagai contoh, informasi yang menyatakan bahwa nilai rupiah akan naik. Akan mengurangi ketidakpastian mengenai jadi tidaknya sebuah investasi.

- ✓ Data yang terorganisir untuk membantu memilih beberapa tindakan yang akan dilakukan atau tidak dilakukan.

Jadi, istilah data dan informasi berbeda, data berupa bahan mentah sehingga perlu diolah dan relatif belum memberikan manfaat bagi penggunanya sehingga perlu diolah lebih lanjut melalui media-media suatu model untuk dihasilkan menjadi suatu informasi yang berguna. Misalnya, data persediaan setiap barang di gudang berupa sejumlah lembaran kartu persediaan. Kartu persediaan tersebut belum memberikan laporan yang jelas mengenai kondisi persediaan setiap barang di gudang. Maka setiap kartu persediaan barang perlu diolah lebih lanjut sehingga menghasilkan suatu informasi yang berguna bagi manajer. Dengan menggunakan media suatu model maka kartu persediaan setiap barang dapat menghasilkan berbagai informasi, seperti berikut ini.

- ✓ Informasi laporan jumlah persediaan bahan mentah berguna bagi manajemen dalam penyediaan bahan baku yang harus dibeli.
- ✓ Informasi laporan jumlah persediaan barang setengah jadi berguna bagi manajemen dalam menetapkan skala prioritas produksi.
- ✓ Informasi laporan jumlah persediaan barang jadi yang berguna bagi manajemen untuk menentukan kebijakan promosi penjualan.
- ✓ Informasi laporan persediaan bahan mentah, setengah jadi, dan barang jadi berguna bagi manajemen dalam menyusun laporan keuangan perusahaan bagi akhir tahun tutup buku.

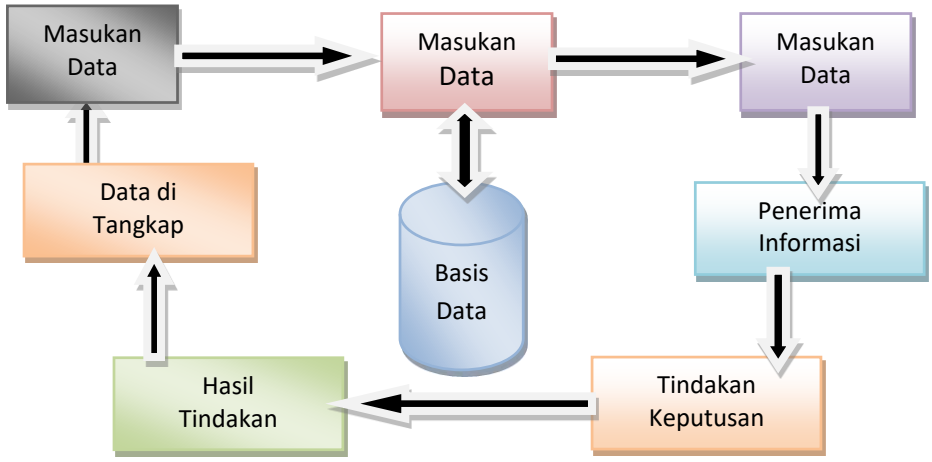
Jadi, apabila digambarkan secara sederhana pemrosesan data menjadi informasi dapat dilihat di bawah ini:



Gambar 2.1 Pemrosesan Data menjadi Informasi

2. Siklus Informasi

Data dapat sangat sederhana, tetapi data juga dapat sangat rumit. Oleh karena itu, data perlu diolah melalui suatu model tertentu untuk menjadi informasi. Informasi akan diterima oleh Pemakai dan Pemakai akan membuat suatu keputusan dan tindakan, hal ini berarti akan menghasilkan tindakan yang lain dan akan menghasilkan data baru yang lain. Data baru akan digunakan kembali oleh Pemakai dan akan menjadi input, selanjutnya akan diolah kembali. Demikian terus sehingga membentuk siklus yang disebut dengan siklus informasi *(information cycle*) atau siklus pengolahan data.



Gambar 2.2 Siklus Informasi

Data yang memiliki nilai akan menghasilkan kualitas informasi. Data yang berkualitas harus memenuhi 3 ketentuan, yakni (a) ketelitian data (*precesion*), (b) Komparabilitas data (c*omparability*), dan (c) validitas data (*validity*).

- ✓ Ketelitian data (precesion)

  Ketelitian data dapat ditentukan oleh kecilnya perbedaan jika observasi yang menghasilkan data itu diulangi atau sumber data yang sama digunakan dalam observasi terhadap kasus yang sama.

- ✓ Komparabilitas data (comparability)

  Data yang memiliki standarisasi yang jelas dan dapat dipertanggungjawabkan sehingga tidak menyulitkan dalam mengambil suatu keputusan atau tidak menimbulkan keraguan.

- ✓ Validitas data (validity)

  Data yang memiliki kegunaan yang tepat dengan kebutuhan dari suatu tujuan yang ingin dicapai pemakai, sebab data yang berkualitas belum tentu valid jika tidak menunjang tujuan pemakai.

Data dapat diklasifikasi menurut jenis, sifat, dan sumbernya. Menurut jenisnya data dibagi menjadi dua, yakni data hitung dan data ukur. Jika dilihat dari segi sifatnya dapat dibagi dua diklasifikasi, yakni data kualitatif dan data kuantitatif. Dilihat dari sumbernya data dapat diklasifikasi menjadi data internal dan data eksternal.

3. Kualitas Informasi

Informasi ibarat darah yang mengalir dalam tubuh suatu organisasi sehingga begitu penting posisinya, sebab dapat digunakan dalam proses pengambilan keputusan dan berhubungan erat dengan nilai keputusan itu sendiri. Fungsi utama dari informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Oleh karenanya kualitas informasi menjadi sangat penting. Kualitas informasi tergantung pada empat hal, yaitu:

- ✓ **Akurat,**

  Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan bagi orang yang menerima informasi tersebut. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Dalam prakteknya, mungkin dalam penyampaian suatu informasi banyak terjadi gangguan yang dapat merubah atau merusak isi dari informasi tersebut. Komponen akurat meliputi :

  - *Completeness*, berarti informasi yang dihasikan atau dibutuhkan harus memiliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian akan mempengaruhi dalam pengambilan keputusan.
  - *Correctness*, berarti informasi yang dihasilkan atau dibutuhkan harus memiliki kebenaran.
  - *Security*, berarti informasi yang dihasilkan atau dibutuhkan harus memiliki keamanan.

- ✓ **Tepat waktu,**

  Informasi yang diterima harus tepat pada waktunya, sebab informasi yang usang (terlambat) tidak mempunyai nilai yang baik, sehingga bila digunakan sebagai dasar dalam pengambilan keputusan akan

dapat berakibat fatal. Saat ini mahalnya nilai informasi disebabkan harus secepatnya informasi itu didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

- ✓ **Relevan,**

  Informasi harus mempunyai manfaat bagi si penerima. Relevansi informasi untuk tiap-tiap orang satu dengan lainnya berbeda. Misalnya informasi sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan.

- ✓ **Ekonomis,**

  Informasi yang dihasilkan mempunyai manfaat yang lebih besar dibandingkan dengan biaya mendapatkannya dan sebagian besar informasi tidak dapat tepat ditaksir keuntungannya dengan satuan nilai uang tetapi dapat ditaksir niai efektivitasnya.

4. Nilai Informasi

Pada umumnya, nilai informasi ditentukan oleh 2 hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya.

Kegunaan informasi adalah untuk mengurangi ketidakpastian di dalam proses pengambilan keputusan tentang keadaan. Namun, perlu dipahami bahwa informasi yang digunakan di dalam suatu sistem informasi pada umumnya digunakan untuk beberapa kegunaan sehingga sulit untuk membandingkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk

mendapatkannya sebab sebagian besar informasi dinikmati tidak hanya oleh satu pihak di dalam manajemen.

Pengukuran nilai informasi pada umumnya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*. Menurut Tata Sutabri bahwa nilai suatu informasi dapat ditentukan berdasarkan sifatnya. Tentang 10 sifat yang dapat menentukan nilai informasi, yaitu sebagai berikut :

- **Kemudahan dalam memperoleh**, Informasi memperoleh nilai yang lebih sempurna apabila dapat diperoleh secara mudah. Informasi yang penting dan sangat dibutuhkan menjadi tidak bernilai jika sulit diperoleh.
- **Sifat luas dan kelengkapannya**, Informasi mempunyai nilai yang lebih sempurna apabila mempunyai lingkup/ cakupan yang luas dan lengkap. Informasi sepotong dan tidak lengkap menjadi tidak bernilai, karena tidak dapat digunakan secara baik.
- **Ketelitian (accuracy**), Informasi mempunyai nilai yang lebih sempurna apabila mempunyai ketelitian yang tinggi/akurat. Informasi menjadi tidak bernilai jika tidak akurat, karena akan mengakibatkan kesalahan pengambilan keputusan.
- **Kecocokan dengan pengguna (relevance**), Informasi mempunyai nilai yang lebih sempurna apabila sesuai dengan kebutuhan penggunanya. Informasi berharga dan penting menjadi tidak bernilai jika tidak sesuai dengan kebutuhan penggunanya, karena tidak dapat dimanfaatkan untuk pengambilan keputusan.
- **Ketepatan waktu,** Informasi mempunyai nilai yang lebih sempurna apabila dapat diterima oleh pengguna pada saat yang tepat. Informasi berharga dan penting menjadi tidak bernilai jika terlambat diterima/

usang, karena tidak dapat dimanfaatkan pada saat pengambilan keputusan.

- **Kejelasan (clarity**), Informasi yang jelas akan meningkatkan kesempurnaan nilai informasi. Kejelasan informasi dipengaruhi oleh bentuk dan format informasi.
- **Fleksibilitas/ keluwesannya**, Nilai informasi semakin sempurna apabila memiliki fleksibilitas tinggi. Fleksibilitas informasi diperlukan oleh para manajer/pimpinan pada saat pengambilan keputusan.
- **Dapat dibuktikan**, Nilai informasi semakin sempurna apabila informasi tersebut dapat dibuktikan kebenarannya. Kebenaran informasi bergantung pada validitas data sumber yang diolah.
- **Tidak ada prasangka**, Nilai informasi semakin sempurna apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.
- **Dapat diukur**, Informasi untuk pengambilan keputusan seharusnya dapat diukur agar dapat mencapai nilai yang sempurna.

Menurut Sutarman (2012), Nilai dari informasi ditentukan oleh lima hal yaitu:

- ✓ Untuk memperoleh pemahaman dan manfaat.
- ✓ Untuk mendapatkan pengalaman.
- ✓ Pembelajaran yang terakumulasi sehingga dapat diaplikasikan dalam pemecahan masalah atau proses bisnis tertentu.
- ✓ Untuk mengekstrak inplikasi kritis dan merfleksikan pengalaman masa lampau yang menyedikan pengetahuan yang terorganisasi dengan nilai yang tinggi. Nilai ini bisa menghindari seorang menajer

darimembuat kesalahan yang sama yang dilakukan oleh manajer lain sebelumnya.

- ✓ Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Sebagian besar informasi tidak dapat ditaksir keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir nilai efektivitasnya.

Informasi dapat dikelompokkan menjadi 3 bagian, yaitu :

- ✓ Informasi strategis

  Informasi ini digunakan untuk mengambil keputusan jangka panjang, mencakup informasi eksternal, rencana perluasan perusahaan dan sebagainya.
- ✓ Informasi Taktis

  Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi trend penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
- ✓ Informasi Teknis

  Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stock, return penjualan dan laporan kas harian.

5. Kesalahan Informasi

Menurut Gordon B. Davis, kesalahan informasi adalah antara lain disebabkan oleh hal-hal sebagai berikut :

- ✓ Metode pengumpulan dan pengukuran data yang tidak tepat.
- ✓ Tidak dapat mengikuti prosedur pengolahan yang benar.
- ✓ Hilang/tidak terolahnya sebagian data.

- ✓ Pemeriksaan/pencatatan data yang salah.
- ✓ Dokumen induk yang salah.
- ✓ Kesalahan dalam prosedur pengolahan (contoh: kesalahan program aplikasi komputer yang digunakan).
- ✓ Kesalahan yang dilakukan secara sengaja.

Penyebab kesalahan tersebut dapat diatasi dengan cara-cara sebagai berikut:

- ✓ Kontrol sistem untuk menemukan kesalahan.
- ✓ Pemeriksaan internal dan eksternal.
- ✓ Penambahan batas ketelitian data.
- ✓ Instruksi dari pemakai yang terprogram secara baik dan dapat menilai adanya kesalahan-kesalahan yang mungkin terjadi.

6. Konsep Sistem Informasi

Definisi Sistem Informasi

Informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Inforamsi diperoleh dari sistem informasi (informastion systems) atau disebut juga dengan processing system atau information processing systems atau information-generating systems. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manejerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Dan berikut ini adalah definisi Sistem Informasi yang lainnya :

- ✓ Sistem informasi adalah sekumpulan hardware, software, brainware, prosedur dan atau aturan yang diorganisasikan secara integral untuk

mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan.

- ✓ Sistem informasi adalah satu kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan output baik dalam bentuk gambar, suara maupun tulisan.
- ✓ Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.
- ✓ Sistem informasi adalah kumpulan antara sub-sub sistem yang salings berhubungan yang membentuk suatu komponen yang didalamnya mencakup input-proses-output yang berhubungan dengan pengolaan informasi (data yang telah dioleh sehingga lebih berguna bagi user).
- ✓ Suatu sistem informasi (SI) atau information system (IS) merupakan aransemen dari orang, data, proses-proses, dan antar-muka yang berinteraksi mendukung dan memperbaiki beberapa operasi sehari-hari dalam suatu bisnis termasuk mendukung memecahkan soal dan kebutuhan pembuat-keputusan manejemen dan para pengguna.

Sifat dari Sistem Informasi

Sistem informasi harus mempunyai beberapa sifat seperti:

- ✓ Pemrosesan informasi yang efektif. Hal ini berhubungan dengan pengujian terhadap data yang masuk, pemakaian perangkat keras dan perangkat lunak yang sesuai.

- ✓ Manajemen informasi yang efektif. Dengan kata lain, operasi manajemen, keamanan dan keutuhan data yang ada harus diperhatikan.
- ✓ Keluwesan. Sistem informasi hendaknya cukup luwes untuk menangani suatu macam operasi.
- ✓ Kepuasan pemakai. Hal yang paling penting adalah pemakai mendapatkan manfaat dan puas terhadap sistem

Kemampuan dari Sistem Informasi

Sistem informasi tentunya memiliki kemampuan sebagai berikut:

- ✓ Memiliki kecepatan akses tinggi, high-volume, komputasi numerik.
- ✓ Menyediakan kecepatan, komunikasi yang akurat dan kolaborasi dengan dan di antara organisasi.
- ✓ Menyimpan informasi dalam jumlah besar dan mudah untuk digunakan.
- ✓ Akses yang cepat dan tidak mahal untuk mendapatkan informasi, dan mendunia.
- ✓ Fasilitas untuk menginterpretasikan sejumlah data yang besar.
- ✓ Meningkatkan efektifitas dan efisiensi dari orang-orang yang bekerja dalam kelompok dalam satu tempat atau dalam lokasi yang berbeda, dimana saja.
- ✓ Mengotomatisasi proses bisnis dan pekerjaan manual.

7. Perkembangan Sistem Informasi

Pada awalnya sistem informasi tidak harus dikaitkan dengan teknologi informasi, namun seiring perkembangan jaman, saat ini suatu sistem informasi tidak dapat lepas dari penggunaan teknologi informasi.

Penggunaan teknologi informasi pada suatu sistem informasi mulai berkembang sekitar tahun 1960an. Pada periode tersebut, sistem informasi yang digunakan masih sangat terbatas. Hal ini disebabkan teknologi perangkat keras maupun perangkat lunak masih sangat jauh jika dibandingkan dengan kondisi sekarang. Tujuan utama sistem informasi pada saat itu adalah untuk melakukan otomatisasi proses bisnis yang berjalan pada organisasi.

Pada periode sekitar tahun 1970an, sistem informasi sudah lebih berkembang. Perkembangan sistem informasi saat itu didominasi dari sudut pandang data. Teknologi basis data saat itu berkembang cukup pesat. Jadi, fokus utama sistem informasi saat itu adalah penyimpanan dan pengaksesan data. Pada saat itu sistem informasi biasanya masih digunakan pada suatu bagian organisasi, khususnya bagian keuangan. Oleh karena itu, kita sekarang sering kali melihat pada suatu organisasi, departemen/bagian sistem informasi (kadang juga disebut bagian teknologi informasi) berada di bawah departemen keuangan.

Pada periode tahun 1980an, sistem informasi berkembang lebih ke arah CSCW (Computer Support Cooperative Work). CSCW adalah aplikasi yang mendukung kerjasama dalam organisasi, misalnya pemanfaatan email, dokumen editor, dan lain-lain. Pada periode ini, sistem informasi mulai mengarah ke bentuk client server. Selain itu, pada periode ini pemanfaatan sistem informasi sudah mulai bertambah luas. Sistem informasi sudah dimanfaatkan pada bermacam-macam bagian organisasi, misalnya bagian keuangan, sumber daya manusia, pemasaran, dan lain-lain.

Pada tahun 1990an, internet berkembang sangat cepat. Perkembangan tersebut juga mendorong perkembangan sistem informasi. Sistem informasi

mulai dimanfaatkan teknologi internet maupun teknologi web. Pada saat itu usaha untuk membuat suatu sistem informasi yang terintegrasi untuk seluruh organisasi sudah mulai dilakukan. Perusahaan-perusahaan perangkat lunak besar di dunia juga mulai mengembangkan sistem informasi yang disesuaikan dengan best practice yang ada, misalnya aplikasi ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), SCM (Supply Chain Management), dan lain-lain.

Pada tahun 2000an, sistem informasi berkembang semakin pesat. Perkembangan ini didorong dengan semakin berkembangan teknologi internet, dengan kapasitas semakin besar dan harga yang semakin murah. Sudah banyak organisasi yang telah mengintegrasikan sistem informasi mereka dengan sistem informasi organisasi lain untuk mendukung kegiatan organisasi tersebut.

Pada masa mendatang, sistem informasi akan semakin berkembang lagi. Perkembangan teknologi dan perubahan dunia usaha yang sangat cepat, mendorong organisasi untuk mengembangkan suatu sistem informasi yang mampu beradaptasi dengan cepat menghadapi perubahan tersebut. Sistem informasi tersebut juga harus dapat diintegrasikan dengan bermacam-macam sistem yang lain agar kinerja organisasi menjadi lebih efisien.

8. Tujuan Sistem Informasi

Sistem informasi dikembangkan untuk tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis. Skema Sistem Informasi Berbasis Komputer di organisasi, dapat dibagi menjadi beberapa bagian:

- ✓ **Sistem Pemrosesan Transaksi (*Transaction Processing Systems*).** Merupakan sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi rutin seperti penggajian, keuangan, inventarisasi dan sebagainya. Sistem ini berfungsi pada level organisasi yang memungkinkan organisasi bisa berinteraksi dengan lingkungan eksternal.

- ✓ **Sistem Otomastisasi Kantor (*Office Automation Systems*)** dan Sistem Kerja Pengetahuan (*Knowledge Work Systems*) Kedua sistem ini bekerja pada level knowledge. Sistem Otomastisasi Kantor (*Office Automation Systems*) mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk transformasikan data atau memanipulasikannya dengan cara-cara tertentu sebelum menyebarkannya secara keseluruhan dengan organisasi dan kadang-kadang diluar organisasi. Aspek-aspek Sistem Otomastisasi Kantor (*Office Automation Systems*) seperti *word processing, spreadsheets*, presentasi.
- ✓ **Sistem Informasi Manajemen (*Management Information System*)** tidak menggantikan Sistem Pemrosesan Transaksi (Transaction Processing Systems), tetapi mendukung spektrum tugas-tugas organisasional yang lebih luas dari Sistem Pemrosesan Transaksi (*Transaction Processing Systems*) termasuk analisis keputusan dan pembuat keputusan. Sistem Informasi Manajemen (*Management Information System*) menghasilkan informasi yang digunakan untuk membuat keputusan, dan juga dapat membatu menyatukan beberapa fungsi informasi bisnis yang sudah terkomputerisasi (basis data).

- ✓ **Sistem Pendukung Keputusan (*Decision Support Systems*)**. Sistem ini hampir sama dengan Sistem Informasi Manajemen (Management Information System) karena menggunakan basis data sebagai sumber data. Sistem ini bermula dari Sistem Informasi Manajemen (*Management Information System*) karena menekankan pada fungsi mendukung pembuat keputusan diseluruh tahap-tahapnya, meskipun keputusan aktual tetap wewenang eksklusif pembuat keputusan.

- ✓ **Sistem Informasi Manajemen (*Management Information System*)**. Sistem yang mendukung spektrum tugas-tugas organisasional yang lebih luas dari Sistem Pemrosesan Transaksi (Transaction Processing Systems) termasuk analisis keputusan dan pembuat keputusan. Juga menghasilkan informasi yang digunakan untuk membuat keputusan, serta dapat membatu menyatukan beberapa fungsi informasi bisnis yang sudah terkomputerisasi (basis data).

- ✓ **Sistem Ahli (*Expert System*) dan Kecerdasan Buatan (*Artificial Intelegent*)** Kecerdasan buatan dimaksudkan untuk mengembangkan mesin-mesin yang berfungsi secara cerdas.Dua cara untuk melakukan riset kecerdasan buatan adalah memahami bahasa alamiahnya dan menganalisis kemampuannya untuk berfikir melalui problem sampai kesimpulan logiknya. Sistem ahli menggunakan pendekatan-pendekatan pemikiran kecerdasan buatan untuk menyelesaikan masalah serta memberikannya lewat pengguna bisnis. Sistem ahli yang disebut juga dengan sistem berbasis

pengetahuan (knowledge based systems) secara efektif menangkap dan menggunakan pengetahuan seorang ahli untuk menyelesaikan masalah yang dialami dalam suatu organisasi. Berbeda dengan sistem pendukung keputusan (decision support systems), sistem ini meninggalkan keputusan terakhir bagi pembuat keputusan sedangkan sistem ahli menyeleksi solusi terbaik terhadap suatu masalah khusus.

- ✓ *Sistem Pendukung Keputusan Kelompok (Group Decision Support Systems*) dan Sistem Kerja Kolaborasi Dukungan Komputer (*Computer-Support Collaborative Work Systems*). Bila kelompok, perlu bekerja bersama-sama untuk membuat keputusan semi-terstruktur dan tak terstruktur, maka group Decision support systems (DSS) menjadi suatu solusinya.
- ✓ **Sistem Pendukung Eksekutif (*Executive Support Systems*).** Sistem tergantung pada informasi yang dihasilkan oleh Sistem Pengolahan Transaksi. Sistem ini membantu para eksekutif mengatur interaksinya dengan lingkungan eksternal dengan menyediakan grafik-grafik dan pendukung komunikasi di tempat-tempat yang bisa diakses seperti kantor.

9. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen komponen yang disebut blok bangunan (building blok), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen hardware, komponen software, komponen basis data, dan komponen kontrol. Semua

komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

- ✓ Komponen Input

  Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dokumen dasar.

- ✓ Komponen Model

  Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yag sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

- ✓ Komponen Output

  Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

- ✓ Komponen Teknologi

  Teknologi merupakan “tool box” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. 5. Komponen perangkat keras (hardware) Perangkat keras berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi, yang berfungsi sebagai tempat untuk menampung basis

data atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

- ✓ Komponen Perangkat Lunak (software)

  Perangkat lunak berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari perangkat keras untuk menciptakan suatu informasi.

- ✓ Komponen Basis Data

  Basis data (database) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan paket perangkat lunak yang disebut DBMS (Database Management System).

- ✓ Komponen Kontrol

  Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidakefisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat

dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

10. Elemen Sistem Informasi

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

- ✓ Orang

  Orang atau personil yang dimaksudkan yaitu operator komputer, analis sistem, programmer, personil data entry, dan manajer sistem informasi/EDP

- ✓ Prosedur

  Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

- ✓ Perangkat keras

  Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data, dan terminal masukan/keluaran.

- ✓ Perangkat lunak

  Perangkat lunak dapat dibagi dalam 3 jenis utama : a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem

manajemen data yang memungkinkan pengoperasian sistem komputer. b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan. c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

- ✓ Basis data

  File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti disket, harddisk, magnetic tape, dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film, dan lain sebagainya.

- ✓ Jaringan komputer

  Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

- ✓ Komunikasi data

  Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

BAB-III

PENGEMBANGAN SISTEM INFORMASI

Overview

Pada bab ini menjelaskan pengembangan sistem informasi, perluanya pengembangan, Tim pengembang sistem informasi, prinsip pengembangan , sampai pada tahap siklus pengembangan sistem informasi.

Tujuan

1. Mahasiswa mampu menjelaskan mengencai pengembangan sistem informasi, perlunya pengembangan, siapa saja yang terlibat dalam pengembangan sistem informasi.
2. Mahasiswa mampu memahami dan menjelaskan prinsip pengembangan sampai pada tahap siklus pengembangan sistem informasi.

Materi

1. Pengertian Pengembangan Sistem

Pengembangan sistem informasi ialah satu set aktivitas, metode, praktik terbaik, siap dikirimkan, dan peralatan terotomasi yang digunakan oleh stakeholder untuk mengembangkan dan memelihara sistem informasi dan perangkat lunak.

Biasanya pengembangan sistem dilakukan apabila sistem yang lama sudah tidak bisa mengimbangi kebutuhan atau pun perkembangan perusahaan atau organisasi.

Pengembangan sistem didefinisikan sebagai :

- ✓ Aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk menyelesaikan persoalan *(problem*) organisasi atau memanfaatkan kesempatan (*opportunities*) yang timbul.
- ✓ Menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.
- ✓ Suatu proses pengaplikasian teknologi informasi untuk suatu tujuan tertentu atau menyelesaikan suatu masalah
- ✓ Memilah suatu masalah yang besar dan kompleks menjadi beberapa bagian kecil yang dapat dikelola.

2. Perlunya Pengembangan Sistem

Sistem lama yang perlu diperbaiki atau diganti disebabkan karena beberapa hal :

- ✓ Adanya permasalahan-permasalahan yang timbul di sistem yang lama. Permasalahan yang timbul dapat berupa :
  - Ketidakberesan sistem yang lama : ketidakberesan dalam sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan. Ketidakberesan ini dapat berupa :
    - ❖ Kecurangan-kecurangan disengaja yang menyebabkan tidak amannya harta kekayaan perusahaan dan kebenaran dari data menjadi kurang terjamin.
    - ❖ Kesalahan-kesalahan yang tidak disengaja yang juga dapat menyebabkan kebenaran dari data kurang terjamin.
    - ❖ Tidak efisiensinya operasi.
    - ❖ Tidak ditaatinya kebijaksanaan manajemen yang telah ditetapkan
  - Pertumbuhan organisasi : kebutuhan informasi yang semakin luas, volume pengolahan data semakin meningkat, perubahan prinsip akuntansi yang baru, menyebabkan harus disusunnya sistem yang baru, karena sistem yang lama tidak efektif lagi dan tidak dapat memenuhi semua kebutuhan informasi yang dibutuhkan manajemen.
- ✓ Untuk meraih kesempatan-kesempatan Dalam keadaan persaingan pasar yang ketat, kecepatan informasi atau efisiensi waktu sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan dan peluang pasar, sehingga teknologi informasi perlu digunakan

untuk meningkatkan penyediaan informasi agar dapat mendukung proses pengambilan keputusan yang dilakukan oleh manajemen.

- ✓ Adanya instruksi dari pimpinan atau adanya peraturan pemerintah

  Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi dari pimpinan atau dari luar organisasi, seperti misalnya peraturan pemerintah

Pengembangan sistem informasi dilakukan untuk mendukung kegiatan bisnis dalam organisasi, tahapannya terdiri dari inisialisasi, analisis, desain, dan implementasi. Pengembangan sistem informasi dapat berupa pembuatan suatu sistem baru maupun penambahan atau perubahan modul pada sistem yang sudah ada. Secara umum, alur pengembangan suatu sistem informasi mempunyai beberapa tahapan. Tahapan pengembangan sistem informasi sering kali disebut juga sebagai System Development Life Cycle (SDLC).

Dalam pengembangan sistem informasi, terdapat 2 (dua) hal utama yang harus diperhatikan.

- ✓ **Produk**. Produk adalah produk yang harus dihasilkan pada setiap tahap pengembangan sistem informasi. Kesalahan dalam pembuatan produk dalam setiap tahap akan menyebabkan kesalahan yang semakin besar pada produk akhir

- ✓ **Proses.** Proses adalah proses pengembangan sistem informasi. Proses ini meliputi tahapan pengembangan mulai dari tahap feasibility sampai implementation. Jika proses tersebut tidak

dilaksanakan sesuai dengan jadwal maka kemungkinan kegagalan proyek menjadi semakin besar

Dengan telah dikembangkannya system yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di system yang baru. Peningkatan-peningkatan ini berhubungan dengan PIECES (merupakan singkatan untuk memudahkan mengingatnya), yaitu sebagai berikut:

- ✓ ***Performance*** **(kinerja**), peningkatan terhadap kinerja (hasil kerja) system yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari throughput dan response time. Throughtput adalah jumlah dari pekerjaan yang dapat dilakukan suatu saat tertentu.
- ✓ ***Response time*** adalah rata-rata waktu yang tertunda diantara dua transaksi atau pekerjaan ditambah dengan waktu response untuk menanggapi pekerjaan terebut.
- ✓ ***Information*** **(informasi**), peningkatan terhadap kualitas informasi yang disajikan.
- ✓ ***Economy*** **(ekonomis**), peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi.
- ✓ ***Control*** **(Pengendalian**), peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan-kecurangan yang dan akan terjadi.
- ✓ ***Efficiency*** **(efisiensi**), peningkatan terhadap efisiensi operasi. Efisiensi berbeda dengan ekonomis. Bila ekonomis berhubungan dengan jumlah sumber daya yang digunakan, efisiensi berhubungan dengan bagaimana sumber daya tersebut digunakan

dengan pemborosan yang paling minimum. Efisiensi dapat diukur dari outputnya dibagi dengan inputnya.

- ✓ ***Services*** **(pelayanan**), peningkatan terhadap pelayanan yang diberikan oleh system.

3. Tim Pengembang Sistem Informasi

Suatu proyek pengembangan sistem informasi biasanya dikembangkan oleh sebuah tim. Tim tersebut biasanya terdiri dari beberapa posisi sebagai berikut:

- ✓ ***Project Leader*** yaitu penanggung jawab utama proyek pengembangan sistem informasi. Seorang project leader harus mampu mengatur waktu dan sumber daya agar sistem informasi dapat diselesaikan sesuai dengan target yang telah ditetapkan. Dalam sebuah proyek pengembangan sistem informasi, seorang project leader sebaiknya tidak merangkap jabatan lain untuk menghindari adanya konflik kepentingan
- ✓ ***System Analyst*** yaitu orang yang bertugas untuk melakukan analisis terhadap kebutuhan user dan kemudian mendokumentasikan kebutuhan user tersebut dalam suatu dokumen teknis yang mudah dipahami oleh anggota tim pengembangan sistem informasi. Seorang system analyst yang baik sebaiknya mempunyai pengetahuan dibidang sistem informasi dan pengembangan perangkat lunak sehingga dia mampu merepresentasikan kebutuhan user dengan baik dalam suatu dokumen. Selain itu, system analyst juga dituntut untuk mempunyai pengetahuan umum

yang luas agar mempermudah dalam memahami kebutuhan user.

- ✓ ***System Designer*** yaitu orang yang bertugas untuk mendesain sistem berdasarkan dokumen kebutuhan user.
- ✓ ***Programmer*** yaitu orang yang bertugas untuk mengimplementasikan desain tersebut menjadi kode program.
- ✓ ***Software Quality Assurance (SQA)*** yaitu orang yang bertugas untuk memastikan semua proses pengembangan sistem informasi berjalan dengan baik dan memastikan produk yang dihasilkan sesuai yang diharapkan

4. Prinsip Pengembangan Sistem Informasi

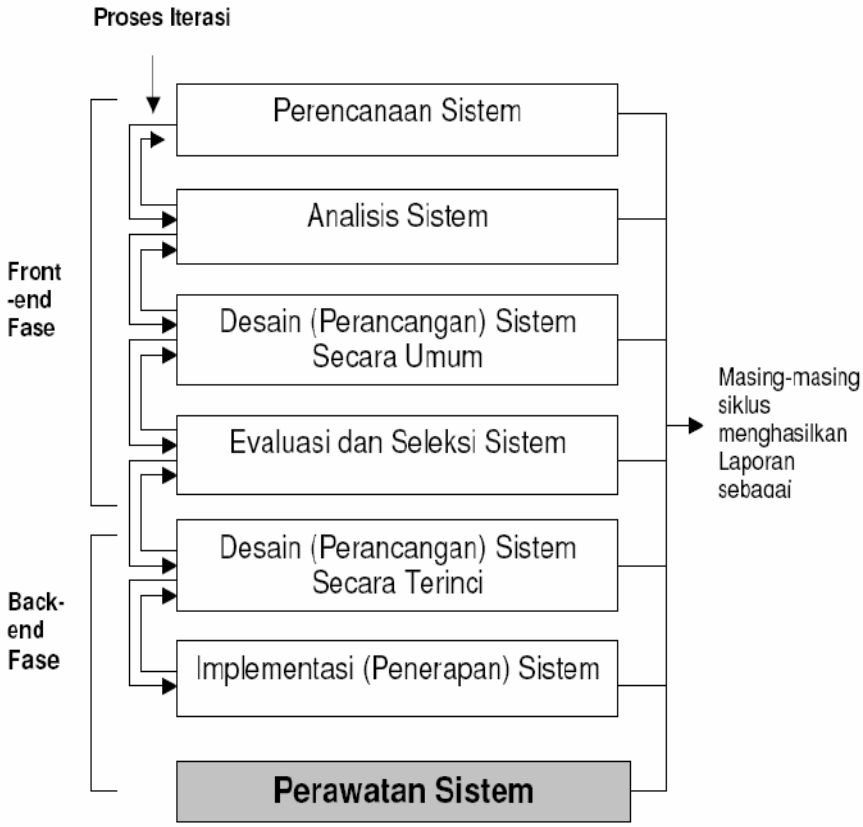
Prinsip pengembangan sistem :

- ✓ Sistem yang dikembangkan adalah untuk manajemen
- ✓ Sistem yang dikembangkan adalah investasi modal yang besar
- ✓ Sistem yang dikembangkan memerlukan orang-orang yang terdidik
- ✓ Proses pengembangan sistem tidak harus urut
- ✓ Jangan takut membatalkan proyek
- ✓ Dokumentasi harus ada untuk pedoman dalam pengembangan sistem

5. Siklus Hidup Pengembangan Sistem Informasi

Siklus hidup pengembangan sistem dapat didefinisikan sebagai serangkaian aktivitas yang dilaksanakan oleh professional dan pemakai sistem informasi untuk mengembangkan dan mengimplementasikan sistem informasi.

Siklus hidup pengembangan sistem dibagi menjadi tujuh fase, yaitu : **Perencanaan, Analisis, Desain sistem secara umum, Evalusi dan Seleksi sistem, Desain sistem terinci, Implementasi dan Pemeliharaan / Perawatan Sistem**. Untuk lebih jelasnya dapat di lihat pada gambar beikut.



Gambar 3.1 Fase Pengembangan Sistem

Perencanaan Sistem

Dalam fase perencanaan sistem dibentuk suatu struktur kerja strategis yang luas dan pandangan sistem informasi baru yang jelas akan memenuhi kebutuhan-kebutuhan pemakai informasi. Selama fase perencanaan sistem, harus dipertimbangkan :

- Faktor – faktor kelayakan yang berkaitan dengan kemungkinan berhasilnya sistem informasi yang dikembangkan dan digunakan.
- Faktor – faktor strategis yang berkaitan dengan pendukung sistem informasi dari sasaran bisnis dipertimbangkan untuk setiap proyek yang diusulkan. Nilai-nilai yang dihasilkan dievaluasi untuk menentukan proyek sistem mana yang akan menerima prioritas yang tertinggi.

Suatu sistem yang diusulkan harus layak, yaitu sistem ini harus memenuhi kriteria-kriteria sebagai berikut :

- ✓ **Kelayakan teknis** untuk melihat apakah sistem yang diusulkan dapat dikembangkan dan diimplementasikan dengan menggunakan teknologi yang ada atau apakah teknologi yang baru dibutuhkan.
- ✓ **Kelayakan ekonomis** untuk melihat apakah dana yang tersedia cukup untuk mendukung estimasi biaya untuk sistem yang diusulkan.
- ✓ **Kelayakan legal** untuk melihat apakah ada konflik antara sistem yang sedang dipertimbangkan dengan kemampuan perusahaan untuk melaksanakan kewajibannya secara legal.

- ✓ **Kelayakan operasional** untuk melihat apakah prosedur dan keahlian pegawai yang ada cukup untuk mengoperasikan sistem yang diusulkan atau apakah diperlukan penambahan/pengurangan prosedur dan keahlian.
- ✓ **Kelayakan rencana** berarti bahwa sistem yang diusulkan harus telah beroperasi dalam waktu yang telah ditetapkan.
- ✓ **Produktivitas** mengukur jumlah output yang dihasilkan oleh input yang tersedia. Tujuan produktivitas adalah mengurangi atau menghilangkan biaya tambahan yang tidak berarti. Produktivitas ini dapat diukur dengan rasio antara biaya yang dikeluarkan dengan jumlah unit yang dihasilkan.
- ✓ **Diferensiasi** mengukur bagaimana suatu perusahaan dapat menawarkan produk atau pelayanan yang sangat berbeda dengan produk dan pelayanan dari saingannya. Diferensiasi dapat dicapai dengan meningkatkan kualitas, variasi, penanganan khusus, pelayanan yang lebih cepat, dan biaya yang lebih rendah.
- ✓ **Manajemen** melihat bagaimana sistem informasi menyediakan informasi untuk menolong manajer dalam merencanakan, mengendalikan dan membuat keputusan. Manajemen ini dapat dilihat dengan adanya laporan-laporan tentang efisiensi produktivitas setiap hari.

Analisis Sistem

Dalam tahap ini dilakukan proses

- ✓ Penilaian, identifikasi dan evaluasi komponen dan hubungan timbale balik yang terkait dalam pengembangan sistem, definisi masalah, tujuan, kebutuhan, prioritas dan kendala-kendala sistem.
- ✓ Fase analisis sistem adalah fase professional sistem melakukan kegiatan analisis sistem.
- ✓ Laporan yang dihasilkan menyediakan suatu landasan untuk membentuk suatu tim proyek sistem dan memulai fase analisis sistem.
- ✓ Tim proyek sistem memperoleh pengertian yang lebih jelas tentang alasan untuk mengembangkan suatu sistem baru.
- ✓ Ruang lingkup analisis sistem ditentukan pada fase ini. Profesional sistem mewawancarai calon pemakai dan bekerja dengan pemakai yang bersangkutan untuk mencari penyelesaian masalah dan menentukan kebutuhan pemakai.
- ✓ Beberapa aspek sistem yang sedang dikembangkan mungkin tidak diketahui secara penuh pada fase ini, jadi asumsi kritis dibuat untuk memungkinkan berlanjutnya siklus hidup pengembangan sistem.
- ✓ Pada akhir fase analisis sistem, laporan analisis sistem disiapkan. Laporan ini berisi penemuan-penemuan dan rekomendasi. Bila laporan ini disetujui, tim proyek sistem siap untuk memulai fase perancangan sistem secara umum. Bila laporan tidak disetujui, tim proyek sistem harus menjalankan analisis tambahan sampai semua peserta setuju.

Perancangan Sistem Secara Umum

Dalam tahap ini hal yang dilakukan yaitu :

- ✓ Dibentuk alternatif-alternatif perancangan konseptual untuk pandangan pemakai. Alternatif ini merupakan perluasan kebutuhan pemakai. Alternatif perancangan konseptual memungkinkan manajer dan pemakai untuk memilih rancangan terbaik yang cocok untuk kebutuhan mereka.
- ✓ Pada fase ini analis sistem mulai merancang proses dengan mengidentifikasikan laporan-laporan dan output yang akan dihasilkan oleh sistem yang diusulkan. Data masing-masing laporan ditentukan. Biasanya, perancang sistem membuat sketsa form atau tampilan yang mereka harapkan bila sistem telah selesai dibentuk. Sketsa ini dilakukan pada kertas atau pada tampilan komputer.

Evaluasi dan Seleksi Sistem

Akhir fase perancangan sistem secara umum menyediakan point utama untuk keputusan investasi. Oleh sebab itu dalam fase evaluasi dan seleksi sistem ini nilai kualitas sistem dan biaya/keuntungan dari laporan dengan proyek sistem dinilai secara hati-hati dan diuraikan dalam laporan evaluasi dan seleksi sistem.

Jika tak satupun altenatif perancangan konseptual yang dihasilkan pada fase perancangan sistem secara umum terbukti dapat dibenarkan, maka semua altenatif akan dibuang. Biasanya, beberapa alternatif harus terbukti dapat dibenarkan, dan salah satunya dengan nilai tertinggi dipilih untuk pekerjaan akhir. Bila satu alternatif perancangan sudah dipilih, maka akan dibuatkan

rekomendasi untuk sistem ini dan dibuatkan jadwal untuk perancangan detailnya.

Perancangan Sistem Secara Detail

Fase perancangan sistem secara detail menyediakan spesifikasi untuk perancangan secara konseptual. Pada fase ini semua komponen dirancang dan dijelaskan secara detail. Perencanaan output (layout) dirancang untuk semua layar, form-form tertentu dan laporan-laporan yang dicetak. Semua output direview dan disetujui oleh pemakai dan didokumentasikan. Semua input ditentukan dan format input baik untuk layar dan form-form biasa direview dan disetujui oleh pemakai dan didokumentasikan.

Berdasarkan perancangan output dan input, proses-proses dirancang untuk mengubah input menjadi output. Transaksi-transaksi dicatat dan dimasukkan secara online atau batch. Macam-macam model dikembangkan untuk mengubah data menjadi informasi. Prosedur ditulis untuk membimbing pemakai dan personil operasi agar dapat bekerja dengan sistem yang sedang dikembangkan.

Database dirancang untuk menyimpan dan mengakses data. Kendali-kendali yang dibutuhkan untuk melindungi sistem baru dari macam-macam ancaman dan error ditentukan. Pada beberapa proyek sistem, teknologi baru dan berbeda dibutuhkan untuk merancang kemampuan tambahan macam-macam komputer, peralatan dan jaringan telekomunikasi.

Pada akhir fase ini, laporan rancangan sistem secara detail dihasilkan. Laporan ini mungkin berisi beribu-ribu dokumen dengan semua spesifikasi untuk masing-masing rancangan sistem yang terintegrasi menjadi satu kesatuan. Laporan ini dapat juga dijadikan sebagai buku pedoman yang

lengkap untuk merancang, membuat kode dan menguji sistem; instalasi peralatan; pelatihan; dan tugas-tugas implementasi lainnya.

Implementasi Sistem

Pada fase ini

- ✓ Sistem siap untuk dibuat dan di instalasi
- ✓ Sejumlah tugas harus dikoordinasi dan dilaksanakan untuk implementasi sistem baru
- ✓ Laporan implementasi yang dibuat pada fase ini ada dua bagian, yaitu
  - Rencana implementasi dalam bentuk Grantt Chart atau (Program and Evaluation Review Technique) PERT Chart
  - Penjadwalan proyek dan tehnik manajemen. Bagian ini merupakan laporan yang menerangkan tugas penting untuk melaksanakan implementasi sistem, seperti pengembangan software, persiapan lokasi peletakan sistem, instalasi peralatan yang digunakan, pengujian sistem, pelatihan untuk para pemakai sistem dan persiapan dokumentasi.

Pemeliharaan/Perawatan Sistem

Tahap pemeliharaan dilakukan setelah tahap implementasi. Sistem baru yang berjalan digunakan sesuai dengan keperluan organisasi. Selama masa hidupnya, sistem secara periodik akan ditinjau. Perubahan dilakukan jika muncul masalah atau jika ternyata ada kebutuhan baru. Selanjutnya, organisasi akan menggunakan sistem yang telah diperbaiki tersebut. Langkah-langkah pemeliharaan sistem terdiri atas:

- ✓ Penggunaan Sistem , yaitu menggunakan sistem sesuai dengan fungsi tugasnya masing-masing untuk operasi rutin atau sehari-hari.
- ✓ Audit sistem, yaitu melakukan penggunaan dan penelitian formal untuk menentukan seberapa baik sistem baru dapat memenuhi criteria kinerja.
- ✓ Penjagaan sistem, yaitu melakukan pemantauan untuk pemeriksaan rutin sehingga sistem tetap beroperasi dengan baik.
- ✓ Perbaikan sistem, yaitu melakukan perbaikan jika dalam operasi terjadi kesalahan (bug) dalam program atau kelemahan rancangan yang tidak terdeteksi saat pengujian sistem.
- ✓ Peningkatan sistem, yaitu melakukan modifikasi terhadap sistem ketika terdapat potensi peningkatan sistem setelah sistem berjalan beberapa waktu.

BAB-IV

METODE PENGEMBANGAN SISTEM INFORMASI

Overview

Pada bab ini menjelaskan mengenai beberapa metode pengembangan sistem(SDLC), Siklus hidup sistem informasi, tahapan tahapan dalam SDLC, kelebihan dan kekurangan SDLC, metode waterfall, tahapan dalam waterfall , kelebihan dan kekurangan waterfall, prototyping, tahapan , kelebihan dan kekurangan , metode RAD, Metode spiral, metode v model dan metode end user.

Tujuan

1. Mahasiswa mengetahui beberapa jenis metode yang dalam pengembangan sistem informasi
2. Mahasiswa mampu menerapakan salah metode pengembangan yang ada saat melaksanakan pembuatan proyek sistem informasi.
3. Mahasiswa mampu melaksanakan metode yang ada dan dapat menerapakan tahap-tahapan yang ada dengan benar.

Materi

Metodologi pengembangan sistem adalah langkah-langkah yang dilalui oleh analisis sistem dalam mengembangkan sistem informasi.

Metodologi adalah : Kesatuan metode-metode, procedure-prosedure, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya.

Metode adalah : Suatu cara/ teknik yang sisematik untuk mengerjakan sesuatu

Terdapat macam-macam representasi metodologi pengembangan sistem, yaitu:

1. **System Development Life Cycle (SDLC)**

Model SDLC atau Sekuensial Linier sering disebut juga model air terjun. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematik dan sekunsial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan

Model ini disusun bertingkat, setiap tahap dalam model ini dilakukan berurutan, satu sebelum yang lainnya. Model ini biasanya digunakan untuk membuat sebuah software dalam skala besar dan yang akan dipakai dalam waktu yang lama. Sangat cocok untuk pengembangan sistem yang besar. Tidak sesuai atau tidak terlalu disarankan untuk *small scale project* karena:

- ✓ *Resource intensive*
- ✓ Tidak fleksibel
- ✓ Sulit untuk aplikasi dengan perubahan cara pengambilan keputusan yang cepat

Gambar 4.1 Metode SDLC

Tahapan-tahapan (SDLC)

- ✓ Fase Perencanaan Sistem

  Dalam tahapan ini dibentuk suatu struktur kerja strategis yang luas dan pandangan sistem informasi baru yang jelas yang akan memenuhi kebutuhan-kebutuhan pemakai informasi. Proyek sistem dievaluasi dan dipisahkan berdasarkan prioritasnya. Proyek dengan prioritas tertinggi akan dipilih untuk pengembangan. Penyediaan sumber daya baru dan penyediaan dana untuk pengembangan sistem. Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Hasil dari tahapan ini adalah : Langkah-langkah detail rencana kerja dan penugasan untuk anggota tim.

- ✓ Fase Analisis Sistem

  - Dilakukan proses penilaian, identifikasi dan evaluasi komponen dan hubungan timbal-balik yang terkait dalam pengembangan system: definisi masalah, tujuan, kebutuhan, prioritas dan kendala-kendala system, ditambah

identifikasi biaya, keuntungan dan estimasi jadwal untuk solusi yang berpotensi.

- Fase analisis sistem adalah fase profesional sistem melakukan kegiatan analisis sistem.
- Laporan yang dihasilkan menyediakan suatu landasan untuk membentuk suatu tim proyek sistem dan memulai fase analisis sistem.
- Tim proyek sistem memperoleh pengertian yang lebih jelas tentang alasan untuk mengembangkan suatu sistem baru.
- Ruang lingkup analisis sistem ditentukan pada fase ini. Profesional sistem mewawancarai calon pemakai dan bekerja dengan pemakai yang bersangkutan untuk mencari penyelesaian masalah dan menentukan kebutuhan pemakai.
- Beberapa aspek sistem yang sedang dikembangkan mungkin tidak diketahui secara penuh pada fase ini, jadi asumsi kritis dibuat untuk memungkinkan berlanjutnya siklus hidup pengembangan sistem.
- Pada akhir fase analisis sistem, laporan analisis sistem disiapkan. Laporan ini berisi penemuan-penemuan dan rekomendasi. Bila laporan ini disetujui,tim proyek sistem siap untuk memulai fase perancangan sistem secara umum. Bila laporan tidak disetujui, tim proyek sistem harus menjalankan analisis tambahan sampai semua peserta setuju.

✓ Fase Perancangan Sistem secara Umum

- Dibentuk alternatif-alternatif perancangan konseptual untuk pandangan pemakai. Alternatif ini merupakan

perluasan kebutuhan pemakai. Alternatif perancangan konseptual memungkinkan manajer dan pemakai untuk memilih rancangan terbaik yang cocok untuk kebutuhan mereka.

- Pada fase ini analis sistem mulai merancang proses dengan mengidentifikasikan laporan-laporan dan output yang akan dihasilkan oleh sistem yang diusulkan. Data masing-masing laporan ditentukan. Biasanya, perancang sistem membuat sketsa form atau tampilan yang mereka harapkan bila sistem telah selesai dibentuk. Sketsa ini dilakukan pada kertas atau pada tampilan komputer.

✓ Fase Evaluasi dan Seleksi Sistem

- Akhir fase perancangan sistem secara umum menyediakan point utama untuk keputusan investasi. Oleh sebab itu dalam fase evaluasi dan seleksi sistem ini nilai kualitas sistem dan biaya/keuntungan dari laporan dengan proyek system dinilai secara hati-hati dan diuraikan dalam laporan evaluasi dan seleksi sistem.
- Jika tak satupun altenatif perancangan konseptual yang dihasilkan pada fase perancangan sistem secara umum terbukti dapat dibenarkan, maka semua altenatif akan dibuang. Biasanya, beberapa alternatif harus terbukti dapat dibenarkan, dan salah satunya dengan nilai tertinggi dipilih untuk pekerjaan akhir. Bila satu alternatif perancangan sudah dipilih, maka akan dibuatkan rekomendasi untuk sistem ini dan dibuatkan jadwal untuk perancangan detailnya.

✓ Fase Perancangan Sistem secara Detail

- Pada fase ini semua komponen dirancang dan dijelaskan secara detail. Perencanaan output (layout) dirancang untuk semua layar, form-form tertentu dan laporan-laporan yang dicetak. Semua output direview dan disetujui oleh pemakai dan didokumentasikan.
- Berdasarkan perancangan output dan input, proses-proses dirancang untuk mengubah input menjadi output. Transaksi-transaksi dicatat dan dimasukkan secara online atau batch. Macam-macam model dikembangkan untuk mengubah data menjadi informasi. Prosedur ditulis untuk membimbing pemakai dan pesonel operasi agar dapat bekerja dengan sistem yang sedang dikembangkan.
- Database dirancang untuk menyimpan dan mengakses data. Kendali-kendali yang dibutuhkan untuk melindungi sistem baru dari macam-macam ancaman dan error ditentukan.
- Pada akhir fase ini, laporan rancangan sistem secara detail dihasilkan. Laporan ini mungkin berisi beribu-ribu dokumen dengan semua spesifikasi untuk masing-masing rancangan sistem yang terintegrasi menjadi satu kesatuan. Laporan ini dapat juga dijadikan sebagai buku pedoman yang lengkap untuk merancang, membuat kode dan menguji sistem; instalasi peralatan; pelatihan; dan tugas-tugas implementasi lainnya.

✓ Fase Implementasi Sistem dan Pemeliharaan Sistem

- Sistem siap untuk dibuat dan diinstalasi.

- Sejumlah tugas harus dikoordinasi dan dilaksanakan untuk implementasi sistem baru.
- Rencana implementasi dalam bentuk *Gantt Chart atau Program and Evaluation Review Technique (PERT) Chart*
- Pengembangan perangkat lunak
- Persiapan lokasi peletakkan system
- Instalasi peralatan yang digunakan
- Pengujian Sistem

Kelebihan dan Kekurangan

✓ Kelebihan

- Mudah diaplikasikan.
- Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.

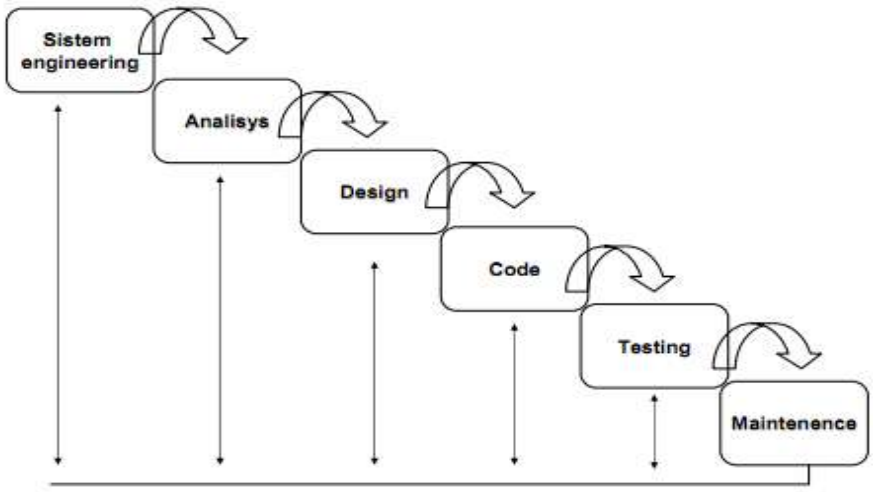
✓ Kekurangan

- Jarang sekali proyek riil mengikuti aliran sekuensial yang dianjurkan model karena model ini bisa melakukan itersi tidak langsung.
- Pelanggan sulit untuk menyatakan kebutuhan secara eksplisit sehingga sulit untuk megakomodasi ketidakpastian pada saat awal proyek.
- Pelanggan harus bersikap sabar karena harus menunggu sampai akhir proyrk dilalui. Sebuah kesalahan jika tidak diketahui dari awal akan menjadi masalah besar karena harus mengulang dari awal.
- Pengembang sering malakukan penundaan yang tidak perlu karena anggota tim proyek harus menunggu tim lain untuk

melengkapi tugas karena memiliki ketergantungan hal ini menyebabkan penggunaan waktu tidak efesien.

2. Waterfall

waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.



Gambar 4.2 Metode SDLC

Secara garis besar metode waterfall mempunyai langkah-langkah sebagai berikut : Analisa, Design, Code dan Testing, Penerapan dan Pemeliharaan.

Tahapan Waterfall

- ✓ Analisa

  Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah

penelitian, wawancara atau study literature.Tahapan ini akan menghasilkan dokumen user requirment atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam bahasa pemprogram.

- ✓ Design

  Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirment. Dokumen inilah yang akan digunakan proggrammer untuk melakukan aktivitas pembuatan sistemnya.

- ✓ Coding & Testing

  Coding merupan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh user. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

- ✓ Penerapan

  Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

- ✓ Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

Keuntungan Metode Waterfall

- ✓ Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- ✓ Document pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

Kelemahan waterfall

- ✓ Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- ✓ Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.
- ✓ Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.

3. Model Prototyping

Prototyping adalah proses iterative dalam pengembangan sistem dimana requirement diubah ke dalam sistem yang bekerja (*working system*) yang secara terus menerus diperbaiki melalui kerjasama antara user dan analis. Prototype juga bisa dibangun melalui beberapa tool pengembangan untuk menyederhanakan proses.

Tahapan-tahapan Model Prototyping

- ✓ Identifikasi Kebutuhan

  Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

- ✓ Mengembangkan Prototyping

  Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

- ✓ Menggunakan Sistem

  Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan.

- ✓ Mengkodekan Sistem

  Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

- ✓ Menguji Sistem

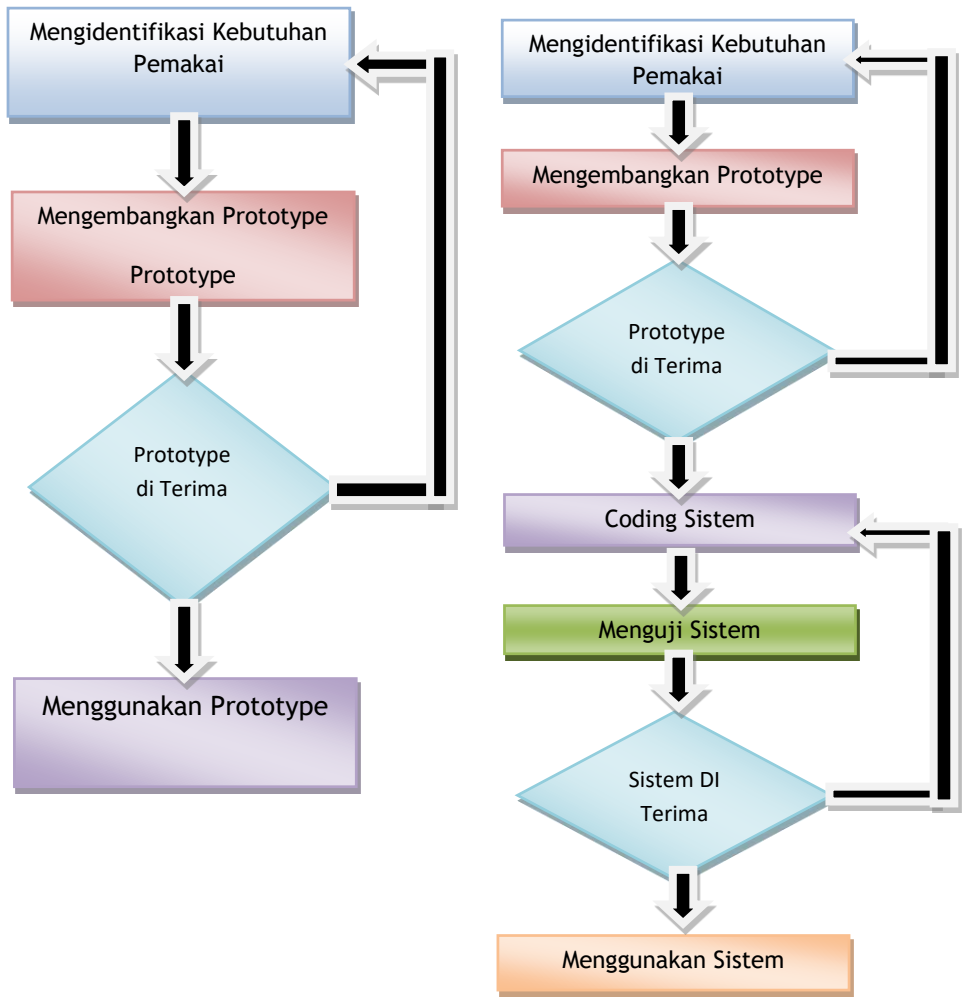
  Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

✓ Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan.

✓ Evaluasi Protoptyping

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.



Jenis-1 Jenis-2

Gambar 4.3 Metode Prototype Jenis 1 dan 2

b. Kelebihan dan Kekurangan

- ✓ Kelebihan
  - Prototype melibatkan user dalam analisa dan desain.
  - Punya kemampuan menangkap requirement secara konkret.
  - Digunakan untuk memperluas SDLC.
- ✓ Kekurangan
  - Proses analisis dan perancangan terlalu singkat.
  - Mengesampingkan alternatif pemecahan masalah.
  - Bisanya kurang fleksible dalam mengahdapi perubahan.
  - Protitype yang dihasilkan tidak selamanya mudah dirubah dan cepat selesai.

4. Model RAD (*Rapid Application Development*)

RAD adalah penggabungan beberapa metode atau teknik terstruktur. RAD menggunakan metode prototyping dan teknik terstruktur lainnya untuk menentukan kebutuhan user dan perancangan sistem informasi selain itu RAD menekankan siklus perkembangan dalam waktu yang singkat (60 sampai 90 hari) dengan pendekatan konstruksi berbasis komponen.

Tahapan-tahapan Model RAD

- ✓ Bussiness Modelling

  Fase ini untuk mencari aliran informasi seperti: informasi mengendalikan proses bisnis, di mana informasi digunakan, siapa yang memprosenya, dan informasi apa yang dimunculkan.
- ✓ Testing and Turnover

Karena menggunakan kembali komponen yang telah ada, maka akan mengurangi waktu pengujian. Tetapi komponen baru harus diuji dan semua interface harus dilatih secara penuh..

- ✓ Aplication Generation

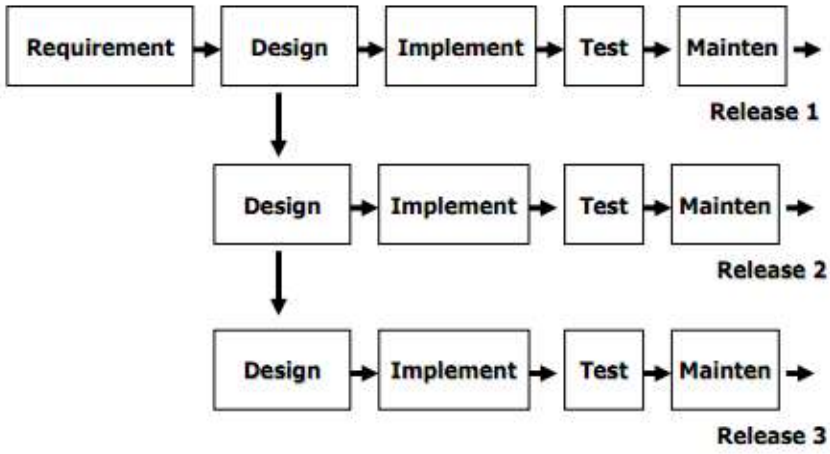
  Selain menggunakan bahasa pemrograman generasi ketiga, RAD juga memakai komponen program yang telah ada atau menciptakan komponen yang bisa dipakai lagi. Alat-alat baantu bisa dipakai untuk memfasilitasi konstruksi perangkat lunak.

- ✓ Process Modelling

  Aliran informasi pada fase data modelling ditransformasikan untuk mendapatkan aliran informasi yang diperlukan pada implementasi fungsi bisnis. Pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atu mendapatkan kembali objek data tertentu

- ✓ Data Modelling

  Fase ini menjelaskan objek data yang dibutuhkan dalam proyek. Karakteristik (atribut) masing-masing data diidentifikasikan dan hubungan antar objek didefinisikan.



Gambar 4.4 Metode RAD

Kelebihan dan Kekurangan

- ✓ Kelebihan
  - RAD mengikuti tahapan pengembangan sistem sepeti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (reusable object).
  - Setiap fungsi dapat dimodulkan dalam waktu tertentu dan dapat dibicarakan oleh tim RAD yang terpisah dan kemudian diintegrasikan sehingga waktunya lebih efesien.

- ✓ Kekurangan
  - Tidak cocok untuk proyek skala besar
  - Proyek bisa gagal karena waktu yang disepakati tidak dipenuhi.
  - Sistem yang tidak bisa dimodularisasi tidak cocok untuk model ini.

- Resiko teknis yang tinggi juga kurang cocok untuk model ini

5. Model Spiral

Model spiral pada awalnya diusulkan oleh Boehm, adalah model proses perangkat lunak evolusioner yang merangkai sifat iteratif dari prototype dengan cara kontrol dan aspek sistematis model sequensial linier. Model iteratif ditandai dengan tingkah laku yang memungkinkan pengembang mengembangkan versi perangkat lunak yang lebih lengkap secara bertahap.

a. Tahapan-tahapan Model Spiral

- ✓ Komunikasi Pelanggan

  Yaitu tugas-tugas untuk membangun komunikasi antara pelanggan dan kebutuhan- kebutuhan yang diinginkan oleh pelanggan.

- ✓ Perencanaan

  Yaitu tugas-tugas untuk mendefinisikan sumber daya, ketepatan waktu, dan proyek informasi lain yg berhubungan.

- ✓ Analisis Resiko

  Yaitu tugas-tugas yang dibutuhkan untuk menaksir resikomanajemen dan teknis.

- ✓ Perekayasaan

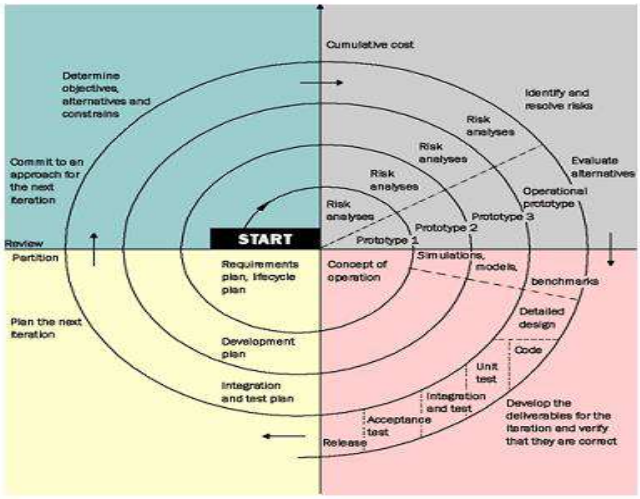
  Yaitu tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari apikasi tersebut.

- ✓ Konstruksi dan Peluncuran

Yaitu tugas-tugas yang dibutuhkan untuk mengkonstruksi, menguji, memasang, dan memberi pelayanan kepada pemakai.

✓ Evaluasi Pelanggan

Yaitu tugas-tugas untuk mendapatkan umpan balik dari pelanggan.



Gambar 4.5 Metode Spirral

Kelebihan dan Kekurangan

✓ Kelebihan

- Dapat disesuaikan agar perangkat lunak bisa dipakai selama hidup perangkat lunak komputer.
- Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar
- Pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses
- Menggunakan prototipe sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk.

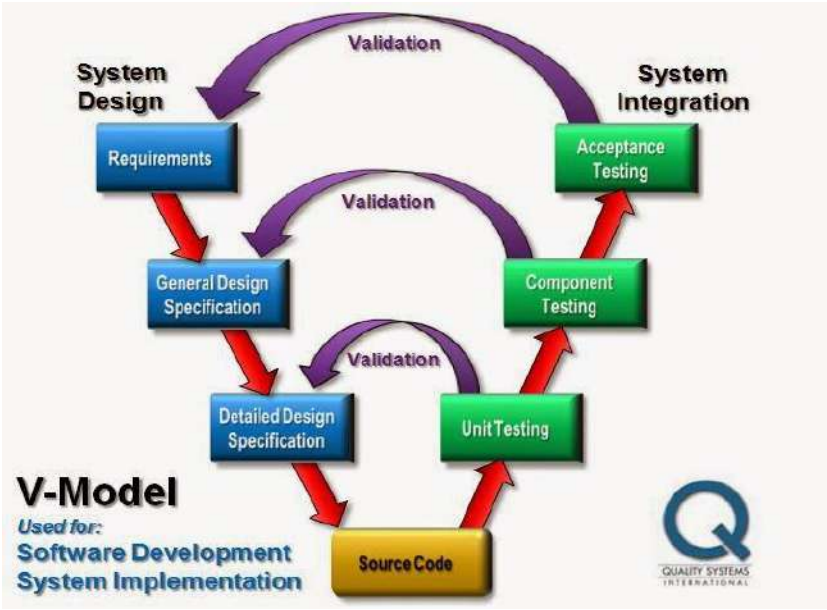
- Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja iteratif .
- Membutuhkan pertimbangan langsung terhadp resiko teknis sehingga mengurangi resiko sebelum menjadi permaslahan yang serius.

✓ Kekurangan

- Sulit untuk menyakinkan pelanggan bahwa pendekatan evolusioner ini bisa dikontrol.
- Memerlukan penaksiran resiko yang masuk akal dan akan menjadi masalah yang serius jika resiko mayor tidak ditemukan dan diatur.
- Butuh waktu lama untuk menerapkan paradigma ini menuju kepastian yang absolute.

6. Model V

Model ini merupakan perluasan dari model waterfall. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model waterfall. Jika dalam model waterfall proses dijalankan secara linear, maka dalam model V proses dilakukan bercabang.



Gambar 4.6 Metode V Model

Tahapan-Tahapan Model V

- ✓ Requirement Analysis & Acceptance Testing

  Tahap Requirement Analysis sama seperti yang terdapat dalam model waterfall. Keluaran dari tahap ini adalah dokumentasi kebutuhan pengguna. Acceptance Testing merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak

- ✓ System Design & System Testing

  Dalam tahap ini analis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi software yang meliputi organisasi sistem secara umum, struktur data, dan yang lain. Selain itu tahap ini juga

menghasilkan contoh tampilan window dan juga dokumentasi teknik yang lain seperti Entity Diagram dan Data Dictionary.

✓ Architecture Design & Integration Testing

Sering juga disebut High Level Design. Dasar dari pemilihan arsitektur yang akan digunakan berdasar kepada beberapa hal seperti: pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar interface, detail teknologi yang dipakai.

✓ Module Design & Unit Testing

Sering juga disebut sebagai Low Level Design. Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan programmer melakukan coding. Tahap ini menghasilkan spesifikasi program seperti: fungsi dan logika tiap modul, pesan kesalahan, proses input-output untuk tiap modul, dan lain-lain.

✓ Coding

Dalam tahap ini dilakukan pemrograman terhadap setiap modul yang sudah dibentuk.

Kelebihan dan Kekurangan

✓ Kelebihan

- V Model sangat fleksibel. V Model mendukung project tailoring dan penambahan dan pengurangan method dan tool secara dinamik. Akibatnya sangat mudah untuk melakukan tailoring pada V Model agar sesuai dengan suatu proyek tertentu dan sangat mudah untuk

menambahkan method dan tool baru atau menghilangkan method dan tool yang dianggap sudah obsolete.

- V Model dikembangkan dan di-maintain oleh publik. User dari V Model berpartisipasi dalam change control board yang memproses semua change request terhadap V Model.

✓ Kekurangan

- V Model adalah model yang project oriented sehingga hanya bisa digunakan sekali dalam suatu proyek.
- V Model adalah model yang project oriented sehingga hanya bisa digunakan sekali dalam suatu proyek.

Penggunaan

V Model digunakan dalam proyek teknologi informasi di negara Jerman. Hal ini berlaku terutama untuk proyek teknologi informasi pada pada sektor pertahanan negara Jerman. Selain itu, V Model juga digunakan oleh *software developer* negara Jerman untuk proyek teknologi informasi lain.

7. Metode End-user Development

Disini pengembangan dilakukan langsung oleh end-user. Keterlibatan langsung end-user sangat menguntungkan, karena memahami benar bagaimana sistem bekerja. Artinya tahap analisis sistem dapat dilakukan lebih cepat. Kelemahan adalah pada pengendalian mutu dan kecenderungan tumbuhnya "*private*" sistem informasi. Integrasi dengan sistem yang lain menjadi sulit.

Tahapan-tahapan EUD

- ✓ Tahap inisasi (*initiation*)

  Yaitu tahap dimana organisasi(perusahaan) mulai pertama kali mngenal teknologi informasi.

- ✓ Tahap ketularan (*contagion*)

  Yaitu tahap diamana organisasi (perusahaan) sudah mulai banyak yang menggunakan teknologi informasi meskipun ini dilakukan atau tidak terlalu mempertimbangkan untung ruginya dari penggunaan teknologi informasi ini.

- ✓ Tahap kendali (*control)*

  Pada tahap ini organisasi (perusahaa) sudah mulai selektif di dalam penggunaan teknologi informasi. Ada hal yang dijadikan pertimbangan sebelum memutuskan penggunaan teknolgi informasi seperti pertimbangan untung rugi.

- ✓ Tahap matang (*mature*)

  Pada tahap ini organisasi (perusahaan) menggunakan teknologi informasi tidak hanya mempertimbangakan keuntungan (benfit) yang akan didapatkan serta berapa biaya (cost) yang harus dikeluarkan tetapi lebih dari itu bagaimana teknologi informasi yang digunakan dapat dijadikan sebagai alat keunggulan di dalam bersaing.

Kelebihan dan Kekurangan

- ✓ Kelebihan
  - Dapat menghindari permasalahan kemacetan di departemen sistem informasi.

- Kebutuhan pemakai sistem dapat lebih terpenuhi karena dapat dikembangkan sendiri oleh pemakai.
- Menambah atau meningkatkan partisifasi aktif pemakai dalam proses pengembangan sistemnya sehingga akan ada kepuasan sendiri dari pemakai sistem.
- Dapat menambah kualitas pemahaman pemakai terhadap aplikasi yang dikembangkan serta teknollogi yang digunakan dalam sistem.

✓ Kekurangan

- Karena pemakai sistem harus mengembangkan aplikasinya sendiri,maka dalam haln ini pemakai sekaligus pengembang sistem dituntut untuk memiliki pemahaman mengenai teknologi informasi (computer literacy) serta pemahaman tentang pengembangan sistem infomasi.
- End user computing memiliki resiko dapat menggangu bahkan merusak sisteminformasi di luar yang dikembangkan oleh pemakai sistem.
- End user computing pasti akan berhadapan dengan maslah kemampuan teknispemakai sekaligus pengembang sistem.

8. Metode Outsourcing

Outsourcing merupakan salah satu metode pengelolaan teknologi informasi dengan cara memindahkan pengelolaannya pada pihak lain, yang tujuan akhirnya adalah efektivitas dan efisiensi kerja. Metode ini seringkali juga disamakan dengan metode lain seperti : sub kontrak, *supplier*, proyek atau istilah lain yang berbeda-beda dilapangan, namun pada dasarnya adalah sama, yaitu pemindahan layanan kepada pihak lain.

Kelebihan dan Kekurangan

- ✓ Kelebihan
  - Manajemen TI yang lebih baik, TI dikelola oleh pihak luar yang telah berpengalaman dalam bidangnya, dengan prosedur dan standar operasi yang terus menerus dikembangkan.
  - Fleksibiltas untuk meresponse perubahan TI yang cepat, perubahan arsitektur TI berikut sumberdayanya lebih mudah dilakukan
  - Akses pada pakar TI yang lebih baik
  - Fokus pada inti bisnis, perusahaan tidak perlu memikirkan bagaimana sistem TI-nya bekerja
- ✓ Kekurangan
  - Terdapat kekhawatiran tentang keamanan sistem informasi karena adanya peluang penyalahgunaan sistem informasi oleh vendor, misalnya pembajakan atau pembocoran informasi perusahaan
  - Ada peluang sistem informasi yang dikembangkan tidak sesuai dengan kebutuhan perusahaan dikarenakan vendor tidak memahami kebutuhan sistem dalam perusahaan tersebut.
  - Transfer knowledge terbatas karena pengembangan sistem informasi sepenuhnya dilakukan oleh vendor.
  - Relatif sulit melakukan perbaikan dan pengembangan sistem informasi karena pengembangan perangkat lunak dilakukan oleh vendor, sedangkan perusahaan umumnya hanya terlibat sampai rancangan kebutuhan sistem.

- Dapat terjadi ketergantungan kepada konsultan.
- Resiko tidak kembalinya investasi yang telah dikeluarkan apabila terjadi

BAB-V

TAHAPAN PENGEMBANGAN SISTEM

(Perencanaan, Analisis, Perancangan Sistem)

Overview

Pada bab ini menjelaskan mengenai difinisi , perluanya , proses dan model Perencanaan, analisis sistem, batasan, mendefinisikan kebutuhan terkait sistem yang akan dikembangkan, perancangan / desain sistem informasi , perancangan sistem dengan menggunakan pendekatan berorientasi objek dan pendekataan tersetruktur. Perbandingan metode desain sistem berorientasi objek dengan pendekatan terstruktur

Tujuan

1. Mahasiswa mampu menerapakan tahapan tahapan dalam pengembangan sistem informasi pada tahapan *front-end*

2. Mahasiswa mampu menjelaskan dan menerapakan tahapan awal dari pengembangan sistem mulai dari perencanaan sampai pada tahap peracnagan sistem secara umum.

3. Mahasiswa mampu menjelaskan dan mampu menganalisis sebuah persoalan dan dapat mencarikan solusi pada tahapan analisis sistem.

Materi

1. Perencanaan

Definisi Perencanaan Sistem

Perencanaan sistem atau *feasibility* adalah tahap pertama yang harus dilakukan sebelum mulai melakukan pengembangan sistem informasi. Terdapat beberapa hal yang sebaiknya dilakukan pada tahap ini, antara lain adalah mendefinisikan proyek, memodelkan proyek, membuat perkiraan anggaran dan penjadwalan proyek, menyeimbangkan rencana proyek dan menyetujui rencana proyek.

Perlunya Perencanaan Sistem

Perencanaan sistem merupakan suatu aktivitas yang harus dilaksanakan sebelum dikembangkannnya sebuah sistem. Perencanaan sistem perlu dilakukan agar pembangunan pengembangan sistem sesuai *blueprint* yang ada, yang sesuai dengan visi, misi, tujuan dan sasaran organisasi. Biasanya pengembangan sistem dilaksanakan dalam lingkup proyek. Sebelum pelaksanaan proyek pengembangan sistem informasi dimulai, maka proyek tersebut harus mendapatkan persetujuan dari pengambil keputusan. Pengambil keputusan pada suatu organisasi yaitu manajemen tingkat atas *(executive*). Namun, kadang-kadang manajemen akan meminta pendapat bawahannya, manajer level menengah (*middle manager*) maupun calon pengguna aplikasi *(functional user*), dalam melakukan pengambilan keputusan pelaksanaan proyek.

Oleh karena itu, dalam melakukan pendefinisian proyek, anda harus memahami karakteristik kebutuhan para pengambil keputusan. Berikut ini adalah karakteristik umum mengenai orang-orang yang terlibat pengambilan keputusan tersebut.

- ✓ *Executive (manajemen tingkat atas*

  Prioritas utama *executive* adalah ROI (*Return On Invesment*). Jadi agar proyek dapat disetujui, maka anda harus mampu meyakinkan mereka bahwa proyek tersebut dapat meningkatkan ROI

- ✓ *Executive (manajemen tingkat atas)*

  Prioritas utama executive adalah ROI (Return On Invesment). Jadi agar proyek dapat disetujui, maka anda harus mampu meyakinkan mereka bahwa proyek tersebut dapat meningkatkan ROI

- ✓ *Middle manager (manajer level menengah*

  Prioritas utama middle manager biasanya adalah bagaimana meningkatkan produktivitas kerja. Jadi sistem informasi yang akan dikembangkan tersebut harus mampu menunjukkan seberapa besar produktivitas kerja akan meningkat dengan adanya sistem baru tersebut

- ✓ *Functional user (pengguna aplikasi langsung*

  Kebutuhan utama functional user adalah suatu aplikasi yang akan mempermudah pekerjaan mereka. Jadi jika functional user dilibatkan dalam pengambilan keputusan, maka anda harus mampu menunjukkan kemudahan-kemudahan apa yang akan diperoleh functional user dengan adanya sistem informasi yang akan dikembangkan tersebut

Pada tahap ini, dokumen yang dihasilkan adalah dokumen proposal proyek. Sebuah dokumen proposal proyek tersebut minimal terdiri dari hal-hal sebagai berikut:

- ✓ Keuntungan yang akan diperoleh calon pengguna dengan adanya sistem informasi yang akan dikembangkan tersebut. Anda sebaiknya mengetahui siapa yang mengambil keputusan pengadaan sistem baru dan tunjukkan kelebihan sistem baru tersebut sesuai dengan karakteristik kebutuhannya
- ✓ Rencana biaya yang dibutuhkan untuk pengembangan, jika anda menjual sistem informasi tersebut ke pihak lain, berarti rencana biaya pengembangan di sini diganti dengan harga sistem informasi yang anda jual
- ✓ Waktu yang dibutuhkan untuk pengembangan sistem

Proses Perencanaan Sistem

Proses perencanaan sistem dilakukan dengan:

- ✓ Menetapkan suatu kerangka kerja strategi menyeluruh untuk memenuhi kebutuhan informasi pemakai
- ✓ Membagi tugas dan tanggung jawab pada orang yang merencanakan sistem
  - Steering Committee (SC),Chief Information Officer (CIO), Chief Executive Officer (CEO), Chief Financial Officer (CFO) dan Eksekutif Senior.
  - Tugas SC : merupakan penghubung antara tujuan bisnis dan sistem informasi yang membantu untuk mencapai tujuan tersebut
- ✓ Membuat komponen laporan
  - Komponen keseluruhan berhubungan dengan sumber daya yg akan diperoleh (3-5 tahun), meliputi : personil baru,

hardware, software, peralatan telekomunikasi, lokasi computer dan keamanan.

- Komponen aplikasi: suatu portfolio yang disetujui dari proposal proyek sistem, secara luas menyatakan apa saja yang termasuk dalam komponen keseluruhan

✓ Melakukan komunikasi dengan analis sistem

- Keduanya berhubungan dengan proses mendefinisikan kebutuhan pemakai
- Perbedaannya pada cakupan dan tahap rinci

Memastikan bahwa pada perencanaan sistem, suatu sistem yang diusulkan harus layak dan mendukung faktor strategik. Untuk menilai kedua kemungkinan tersebut maka harus diadakan evaluasi terhadap faktor kelayakan dan faktor strategi

Pemodelan Proyek

Pemodelan proyek mempunyai fokus pada pembuatan simulasi mengenai usaha yang dibutuhkan untuk mencapai tujuan proyek. Pemodelan ini menghasilkan sebuah WBS (*Work Breakdown Structure*) yang digunakan untuk menentukan semua usaha yang dibutuhkan untuk menyelesaikan proyek dengan sukses. WBS adalah daftar semua pekerjaan yang harus dilakukan untuk menghasilkan produk yang diinginkan.

Dalam sebuah proyek, ada banyak pekerjaan yang harus dilakukan. Sebuah pekerjaan yang kompleks, sebaiknya dipecah lagi menjadi beberapa sub-pekerjaan. Dan beberapa pekerjaan yang terlalu kecil dan detail sebaiknya digabungkan menjadi sebuah pekerjaan. Dalam menentukan pekerjaan apa saja yang harus dilakukan dalam sebuah proyek, agar tidak ada

yang terlalu kompleks maupun terlalu detail, sangat tergantung dari pengalaman seseorang dan besar atau kecilnya proyek.

Pembuatan WBS membutuhkan kontribusi dari anggota tim yang akan terlibat dalam proyek tersebut. Sebuah metode yang efektif dalam pembuatan WBS adalah membuat sebuah sesi diskusi yang melibatkan semua anggota tim dan memberikan kesempatan bagi mereka untuk memberikan ide-ide yang mereka miliki. Setelah WBS selesai dibuat, tim tersebut kemudian harus menggambarkan keterhubungan antara setiap tugas pekerjaan, menentukan

Tugas apa yang harus sudah selesai sebelum tugas lain dilakukan. Keterhubungan antar pekerjaan ini nanti dibutuhkan dalam melakukan proses penjadwalan.

Perkiraan dan penjadwalan proyek ini fokus kepada penentuan waktu, biaya, dan sumber daya yang dibutuhkan dalam pelaksanaan proyek. Kebanyakan orang yang melakukan estimasi, biasanya mulai melakukan estimasi dengan cara menentukan seberapa besar man-hours atau man-days yang dibutuhkan untuk menyelesaikan pekerjaan. Angka ini nanti juga dibutuhkan dalam menentukan waktu dan biaya yang dibutuhkan.

Berikut ini adalah tujuh tahapan proses estimasi.

- ✓ Langkah 1: Membuat estimasi pekerjaan

  Estimasi pekerjaan seharusnya melibatkan anggota tim yang menjalankan pekerjaan tersebut. Sehingga estimasi tersebut akan realistis dan anggota tim akan punya komitmen dan termotivasi untuk mencapai estimasi tersebut. Estimasi ini kemudian dapat dimodifikasi untuk menyesuaikan dengan jadwal dan sumber daya yang ada.

- ✓ Langkah 2: Membuat perencanaan awal

  Perencanaan awal proyek berisi sebuah jadwal yang dibuat berdasarkan ketergantungan antar pekerjaan (task) dan estimasi pekerjaan tersebut. Jadwal tersebut berisi kapan pekerjaan dimulai, berapa lama, dan kapan pekerjaan tersebut harus sudah selesai. Biaya dapat dihitung dari pekerjaan apa saja yang harus dilakukan dan biaya untuk pembelian barang.

- ✓ Langkah 3: Membandingkan perencanaan awal dengan tujuan

  Tahap selanjutnya adalah pembandingkan antara tujuan awal proyek dengan estimasi rencana jadwal dan biaya yang sudah dilakukan. Tujuan awal proyek biasanya merupakan hal yang konstan dan telah disetujui oleh *executive*. Negosiasi ini tidak diperlukan jika tujuan awal telah sesuai dengan rencana jadwal dan biaya yang dilakukan. Tetapi jika tidak sesuai, maka langkah 4, 5, 6 harus dilakukan.

- ✓ Langkah 4 : Negosiasi perubahan untuk estimasi

  Anda melakukan perubahan estimasi mengenai rencana waktu dan anggaran agar sesuai dengan tujuan awal. Langkah ini mengandung risiko sangat besar apabila anda melakukannya tanpa persetujuan anggota tim yang lain, maka anda akan kehilangan komitmen dan motivasi anggota tim. Anggota tim akan beranggapan jadwal dan anggarannya tidak realistis, sehingga kemungkinan proyek gagal menjadi sangat besar.

- ✓ Langkah 5 : Negosiasi perubahan untuk tujuan proyek

  Langkah ini adalah melakukan negosiasi dengan executive karena dengan perubahan estimasi yang telah anda lakukan, rencana awal

tersebut tidak realistis. Perubahan rencana tersebut dapat berupa penambahan waktu dan anggaran maupun pengurangan kompleksitas sistem. Usahakan agar sebisa mungkin rencana yang anda lakukan telah disetujui oleh anda, anggota tim anda, dan executive.

- ✓ Langkah 6 : Membuat keputusan terus/berhenti

  Setelah melakukan langkah 4 dan 5, anda harus mengambil keputusan apakah akan meneruskan proyek tersebut maupun tidak.

- ✓ Langkah 7 : Mempersiapkan jadwal dan anggaran

  Rencana awal pengembangan sistem informasi telah siap. Rencana ini terdiri dari tiga hal, yaitu jadwal kegiatan (waktu mulai, durasi, dan waktu selesai), alokasi sumber daya manusia terhadap kegiatan, dan rencana anggaran.

Setelah rencana anggaran dan jadwal selesai dibuat, hal yang harus dilakukan selanjutnya adalah menyeimbangkan rencana proyek tersebut dengan kondisi organisasi. Biasanya sebuah organisasi akan menjalankan beberapa proyek. Dan dalam organisasi tersebut uang dan sumber daya manusia merupakan hal yang terbatas. Jadi hal yang harus dilakukan adalah mengatur jadwal dan anggaran agar semua proyek yang sedang dikerjakan dapat berjalan dengan baik. Terdapat bermacam-macam perangkat lunak yang dapat digunakan untuk membantu mengelola sumber daya tersebut. Dengan pengelolaan yang baik, maka sumber daya uang dan manusia dapat tersedia pada saat dibutuhkan.

Setelah rencana anggaran dan jadwal selesai dibuat, hal yang harus dilakukan selanjutnya adalah menyeimbangkan rencana proyek tersebut dengan kondisi organisasi. Biasanya sebuah organisasi akan menjalankan beberapa proyek. Dan dalam organisasi tersebut uang dan sumber daya manusia merupakan hal yang terbatas. Jadi hal yang harus dilakukan adalah mengatur jadwal dan anggaran agar semua proyek yang sedang dikerjakan dapat berjalan dengan baik.

Terdapat bermacam-macam perangkat lunak yang dapat digunakan untuk membantu mengelola sumber daya tersebut. Dengan pengelolaan yang baik, maka sumber daya uang dan manusia dapat tersedia pada saat dibutuhkan.

Tahap terakhir adalah persetujuan rencana, dokumen terkait rencana target (target tanggal selesai, target biaya, target rencana penggunaan sumber daya). Dokumen ini merupakan dokumen persetujuan antara project leader, executive, dan client yang akan digunakan sebagai acuan jika selama proyek berlangsung terjadi perubahan ruang lingkup proyek dan juga digunakan sebagai acuan mengukur performa tim.

2. Pengertian Analisis Sistem :

Analisis sistem adalah Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem. Tahap ini merupakan tahap yang kritis dan sangat penting, karena kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya.

Misalnya anda dihadapkan pada suatu sistem untuk menentukan seberapa jauh sistem tersebut telah mencapai sasarannya. Jika sistem mempunyai beberapa kelemahan, anda harus dapat menemukannya. Tugas ini yang disebut sebagai analisis sistem.

Tugas utama dari menganalisis sistem meliputi :

- ✓ Menentukan lingkup sistem
- ✓ Mengumpulkan fakta
- ✓ Menganalisis fakta
- ✓ Mengkomunikasikan temuan-temuan tersebut melalui laporan analisis sistem

Fakta merupakan bagian dari informasi yang menunjukkan realita, situasi dan relasi yang menjamin analisis dan pemodelan.

Langkah-Langkah Di Dalam Analisis Sistem

Langkah-langkah dasar yang harus dilakukan oleh analis sistem adalah :

- ✓ *Identify*, mengidentifikasi masalah
- ✓ *Understand*, memahami kerja sistem yang ada
- ✓ *Analyze*, menganalisis sistem
- ✓ *Report*, membuat laporan hasil analisis

Untuk masing-masing langkah ini, beberapa tugas perlu dilakukan oleh analis sistem. Supaya memudahkan untuk melakukan koordinasi dan pengawasan, koordinator tim analis dapat membuat suatu kertas kerja yang memuat tugas-tugas yang harus dikerjakan untuk masing-masing langkah analisis sistem ini.

Mengidentifikasi Masalah

Mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itu langkah pertama yang harus dilakukan oleh analis sistem adalah mengidentifikasi terlebih dahulu masalah-masalah yang terjadi.

Tugas yang harus dilakukan analis sistem adalah :

- ✓ Mengidentifikasi penyebab masalah
- ✓ Mengidentifikasi titik keputusan
- ✓ Mengidentifikasi personil-personil kunci

Mengidentifikasi Penyebab Masalah

Analis sistem harus mempunyai pengetahuan yang cukup tentang aplikasi yang sedang dianalisisnya. Untuk aplikasi bisnis, analis sistem perlu mempunyai pengetahuan tentang sistem bisnis yang diterapkan di organisasi, sehingga dapat mengidentifikasi penyebab terjadinya masalah ini.

Tugas mengidentifikasi penyebab masalah dimulai dengan mengkaji ulang terlebih dahulu subyek permasalahan yang telah diutarakan oleh

manajemen atau yang telah ditemukan oleh analis sistem di tahap perencanaan sistem.

Mengidentifikasi Titik Keputusan

Setelah penyebab terjadinya masalah dapat diidentifikasi, selanjutnya juga harus mengidentifikasikan titik keputusan penyebab masalah tersebut. Titik keputusan menunjukkan suatu kondisi yang menyebabkan sesuatu terjadi.

Analis sistem bila telah dapat mengidentifikasi terlebih dahulu **titik-titik keputusan penyebab masalah,** maka dapat memulai penelitiannya di titik-titik keputusan tersebut. Sebagai dasar identifikasi titik-titik keputusan ini, dapat digunakan dokumen paperwork flow atau form flowchart bila dokumentasi ini dimiliki oleh perusahaan.

Mengidentifikasi Personil-personil Kunci

Setelah titik-titik keputusan penyebab masalah dapat diidentifikasi beserta lokasi terjadinya, maka selanjutnya yang perlu diidentifikasi adalah personil-personil kunci baik yang langsung maupun yang tidak langsung dapat menyebabkan terjadinya masalah tersebut. Identifikasi personil-personil kunci ini dapat dilakukan dengan mengacu pada **bagan alir dokumen perusahaan serta dokumen deskripsi kerja** (*job description*).

Memahami Kerja Sistem

Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Diperlukan data yang dapat diperoleh dengan cara melakukan penelitian. Bila di tahap perencanaan sudah pernah diadakan penelitian, sifatnya masih penelitian pendahuluan (preliminary

survey). Sedangkan pada tahap analisis sistem, penelitiannya bersifat penelitian terinci (detailed survey).

Analis sistem perlu mempelajari apa dan bagaimana operasi dari sistem yang ada sebelum mencoba untuk menganalisis permasalahan, kelemahan dan kebutuhan pemakai sistem untuk dapat memberikan rekomendasi pemecahannya. Sejumlah data perlu dikumpulkan, dengan menggunakan teknik pengumpulan data yang ada, yaitu **wawancara, oberservasi, daftar pertanyaan dan pengambilan sampel.**

Tugas yang perlu dilakukan di langkah ini adalah :

- ✓ Menentukan jenis penelitian
- ✓ Merencanakan jadual penelitian
  - Mengatur jadual wawancara
  - Mengatur jadual observasi
  - Mengatur jadual pengambilan sampel
- ✓ Membuat penugasan penelitian
- ✓ Membuat agenda wawancara
- ✓ Mengumpulkan hasil penelitian

Menentukan Jenis Penelitian

Jenis penelitian perlu ditentukan untuk masing-masing titik keputusan yang akan diteliti. Jenis penelitian tergantung dari jenis data yang diperoleh, dapat berupa data tentang operasi sistem, data tentang perlengkapan sistem, pengendalian sistem, atau I/O yang digunakan oleh sistem.

Merencanakan Jadual Penelitian

Supaya penelitian dapat dilakukan secara efisien dan efektif, maka jadual penelitian harus direncanakan terlebih dahulu yang meliputi :

- ✓ Dimana penelitian akan dilakukan
- ✓ Apa dan siapa yang akan diteliti
- ✓ Siapa yang akan meneliti
- ✓ Kapan penelitian dilakukan

Membuat Penugasan Penelitian

Setelah rencana jadual penelitian dibuat, maka tugas dilanjutkan dengan menentukan tugas dari masing-masing anggota tim analis sistem, yang ditentukan oleh koordinator analis sistem melalui surat penugasan dengan menyertakan lampiran kegiatan penelitian yang harus dilakukan.

Membuat Agenda Wawancara

Sebelum wawancara dilakukan, waktu dan materi wawancara perlu didiskusikan. Rencana ini dapat ditulis di agenda wawancara dan dibawa selama wawancara berlangsung. Tujuannya adalah supaya wawancara dapat diselesaikan tepat pada waktunya dan tidak ada materi yang terlewatkan.

Mengumpulkan Hasil Penelitian

Fakta atau data yang diperoleh dari hasil penelitian harus dikumpulkan sebagai suatu dokumentasi sistem lama, yaitu :

- ✓ Waktu untuk melakukan suatu kegiatan
- ✓ Kesalahan melakukan kegiatan di sistem yang lama
- ✓ Pengambilan sampel

- ✓ Formulir dan laporan yang dihasilkan oleh sistem lama
- ✓ Elemen-elemen data
- ✓ Teknologi yang digunakan di sistem lama
- ✓ Kebutuhan informasi pemakai sistem / manajemen

Menganalisis Hasil Penelitian

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.

Menganalisis Kelemahan Sistem

Penelitian dilakukan untuk menjawab pertanyaan :

- ✓ apa yang dikerjakan ?
- ✓ bagaimana mengerjakannya ?
- ✓ siapa yang mengerjakan ?
- ✓ dimana dikerjakan ?

Menganalisis kelemahan sistem sebaliknya dilakukan untuk menjawab pertanyaan :

- ✓ mengapa dikerjakan ?
- ✓ perlukah dikerjakan ?
- ✓ apakah telah dikerjakan dengan baik ?

Sasaran yang diinginkan oleh sistem yang baru ditentukan oleh kriteria penilaian sebagai berikut : relevance, capacity, efficiency, timeliness, accessibility, flexibility, accuracy, reliability, security, economy, simplicity

ANALISA	DAFTAR PERTANYAAN
Distribusi pekerjaan	Apakah tugas dan tanggungjawab telah didefinisikan dan diterapkan dengan jelas ? Apakah telah didistribusikan dengan efektif untuk masing-masing personil dan unit organisasi ?
Pengukuran pekerjaan	Apakah kebijakan dan prosedur telah dipahami dan diikuti ? Apakah produktivitas karyawan memuaskan ? Apakah unit-unit organisasi telah bekerja sama dan terkoordinasi dengan baik menjadi arus data dengan lancar ? Apakah terjadi operasi yang tumpah tindih ? Seberapa perlu hasil dari tiap-tiap operasi ? Apakah terdapat operasi yang menghambat arus data ? Apakah volume puncak dari data dapat ditangani dengan baik ? Apakah terdapat standar kinerja yang baik dan selalu mutakhir ?
Keandalan	Apakah jumlah kesalahan yang terjadi di masing-masing operasi diminimumkan ? Apakah operasi-operasi telah direncanakan dengan baik dan terkendali ?
Dokumen	Seberapa perlu dokumen-dokumen yang ada ?

	Apakah masing-masing dokumen telah dirancang untuk penggunaan yang efektif ? Apakah tembusan dari dokumen perlu ?
Laporan	Dapatkah laporan dipersiapkan dengan mudah dari file dan dokumen yang ada ? Apakah terdapat duplikasi di file, catatan dan laporan ?
Teknologi	Apakah fasilitas dari sistem informasi (personil, peralatan dan fasilitas lain) cukup untuk menangani volume rata-rata data tanpa terjadi penundaan yang berarti ?

Berdasarkan pertanyaan dan kriteria ini, selanjutnya analis sistem akan dapat melakukan analis dari hasil penelitian dengan baik untuk menemukan kelemahan dan permasalahan dari sistem yang ada.

Analisa yang dilakukan meliputi :

Menganalisis Kebutuhan Informasi Pemakai / Manajemen

Tugas lain dari analis sistem yang diperlukan sehubungan dengan sasaran utama sistem informasi, yaitu menyediakan informasi yang dibutuhkan bagi para pemakainya perlu dianalisis.

Membuat Laporan Hasil Analisis

Laporan hasil analisis diserahkan ke Panitia Pengarah (Steering Committee) yang nantinya akan diteruskan ke manajemen. Pihak manajemen

bersama-sama dengan panitia pengarah dan pemakai sistem akan mempelajari temuan-temuan dan analis yang telah dilakukan oleh analis sistem yang disajikan dalam laporan ini.

Tujuan utama dari penyerahan laporan ini kepada manajemen adalah :

- ✓ Analisis telah selesai dilakukan
- ✓ Meluruskan kesalah-pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analis sistem tetapi tidak sesuai menurut manajemen
- ✓ Meminta pendapat dan saran dari pihak manajemen
- ✓ Meminta persetujuan kepada pihak manajemen untuk melakukan tindakan selanjutnya (dapat berupa meneruskan ke tahap disain sistem atau menghentikan proyek bila dipandang tidak layak lagi)

Semua hasil yang didapat dari penelitian perlu dilampirkan pada laporan hasil analisis ini, sehingga manajemen dan user dapat memeriksa kembali kebenaran data yang telah diperoleh.

Mendefinisikan Lingkup Sistem Baru Dan Pengumpulan Informasi

Untuk melaksanakan pekerjaan ini, analis sistem perlu menentukan lingkup sistem dari sistem yang baru dan mendapatkan informasi yang banyak.

Ada tiga sumber dari fakta studi, yaitu :

- ✓ Sistem yang berjalan
- ✓ Sumber internal lainnya
- ✓ Sumber eksternal

Apa yang termasuk ke dalam sistem baru ?

Untuk menjawab pertanyaan ini secara umum, analis sistem memerlukan beberapa pertanyaan khusus berikut ini :

- ✓ Informasi apa yang dibutuhkan ?
- ✓ Siapa yang membutuhkan ?
- ✓ Kapan dibutuhkan ?
- ✓ Dalam bentuk apa dibutuhkannya ?
- ✓ Dari mana asalnya informasi ?
- ✓ Kapan dan bagaimana dikumpulkannya ?

Masalah utama bagi profesional sistem, baik yang baru mau pun yang berpengalaman adalah mengubah / menterjemahkan sebuah instruksi dari : "*I want a daily purchasing report*" ke dalam "*Develop a new purchasing and inventory managemet system*". DFD merupakan alat yang cocok untuk mendefinisikan lingkup analisis sistem.

Batasan-batasan yang ada dalam mendefinisikan sistem membatasi penyelesaian sistem dan rekomendasi yang dihasilkan dari analisis. Definisi awal dari lingkupnya merupakan pokok untuk mendefinisikan ulang pada saat dimana temuan-temuan terjadi saat analisis. Biasanya banyak menghabiskan waktu dan uang, sehingga perlu komitmen waktu dari user untuk ikut berpartisipasi.

Keuntungan dari pemodelan sistem yang berjalan

Menyediakan kesempatan untuk menentukan apakah sistem memuaskan, perlu sedikit perbaikan, membutuhkan pemeriksaan yang besar,

atau diganti. Juga menyediakan sumber ide perancangan untuk membantu analis mengidentifikasikan sumber yang ada bagi sistem yang baru.

Saat sistem baru diimplementasikan, analis bertanggungjawab atas kapan tugas dan kegiatan akan dibutuhkan untuk menghapus pertahap sistem yang berjalan dan memulai mengoperasikan sistem yang baru. Saat dikonversi, analis harus mengetahui tidak hanya kegiatan apa yang dilaksanakan tetapi juga kegiatan-kegiatan yang sudah dilaksanakan. Dengan mempelajari dan memodelkan sistem yang berjalan, memberi jawaban terhadap analis.

Kerugian dari pemodelan sistem yang berjalan

Dalam banyak situasi dimana sistem baru unik atau berbeda sekali dengan sistem yang berjalan, mempelajari sistem yang berjalan kadang-kadang menyesatkan. Jeleknya, sistem yang berjalan menjadi tidak relevan dan menganalisisnya menjadi menghabiskan waktu dan uang. Lebih menyenangkan jika mengidentifikasikan apa yang dibutuhkan dari sistem baru dibandingkan membicarakan kembali apa yang terjadi dengan sistem yang berjalan.

Pengumpulan informasi dari orang yang menggunakan sistem

Sumber yang utama adalah orang yang akan menggunakan sistem yang baru. Pengetahuan teknis user dari user baru hingga yang ahli disebut pengetahuan sintaksis. Untuk bisnis dan pekerjaan merupakan pengetahuan semantik dan merupakan keahliannya.

Untuk membangun sistem baru, analis sistem mengumpulkan data semantik dari user. Data semantik ini yang akan mengarahkan analis sistem selama menganalis dan merancang. Analis sistem dan perancang kemudian

akan mendiskusikan data teknis dengan ahli teknisi sistem seperti programmer untuk mendapatkan spesifikasi sistem (mikro atau mini spec). Jadi analis sistem perlu berkomunikasi dengan user yang memiliki pengetahuan semantik di satu sisi, dan di sisi lain dengan teknisi sistem yang mempunyai pengetahuan sintaksis.

Sumber kedua didapat dari dokumen kerja yang ada dalam organisasi. Dokumen dapat diklasifikasikan misalnya struktur organisasi, apa yang sudah dilakukan organisasi, dan rencana apa yang akan dilakukan organisasi.

Pengumpulan informasi dari sumber di luar organisasi user

Informasi yang berasal dari luar organisasi membuka cakrawala ide dan teknik. Banyak industri dari kelompok dan seminar memberikan pengalaman sistem informasi dan merekomendasikan cara yang lebih baik.

Teknik-Teknik Pengumpulan Informasi Tambahan

Interview (Wawancara)

- ✓ Merupakan teknik yang efektif digunakan selama pengembangan sistem.
- ✓ Merupakan sebuah pertukaran informasi antara pewawancara (analis sistem) dengan yang diwawancarai (user).
- ✓ Perlu ada perencanaan, dan perlu ada tujuan khusus.
- ✓ Terdiri dari pertanyaan dan menjawab pertanyaan.
- ✓ Analis sistem menggunakan mekanisme feedback dan cara utama untuk mengumpulkan fakta lapangan dan melihat gap yang ada.

Ada dua tipe pertanyaan dalam wawancara, yaitu ***Open-ended*** dan ***Closed-ended***.

Pertanyaan Open-Ended

Pertanyaan ini adalah netral dan tidak dibatasi. Pewawancara mengijinkan secara bebas orang yang diwawancarai dalam menjawab pertanyaan, dan pewawancara menganjurkan yang diwawancarai untuk memberikan informasi yang tidak diketahui sebelumnya kepada pewawancara.

Pertanyaan Closed-Ended

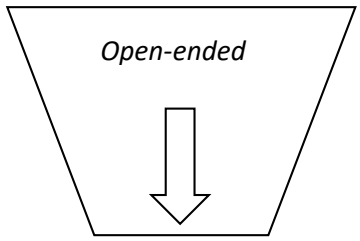
Dalam closed-ended, pewawancara lebih mudah mengontrol yang diwawancarai, karena apa yang akan ditanyakan sudah pasti dan menghindari yang diwawancarai menjawab bebas. Cara yang terbaik adalah menhindari pertanyaan yang berasumsi, dan pernyataan dengan akhir "could you ?" atau "isn't it ?"

Pertanyaan dapat dibagi dalam 2 kategori , yaitu :

- ✓ Primer : untuk topik khusus, direncanakan dan netral.
- ✓ Sekunder : merupakan kelanjutan dari pertanyaan primer untuk mendapatkan tambahan informasi, pertanyaannya bersifat tidak terencana.

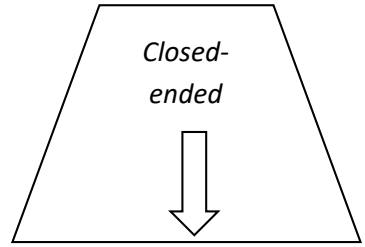
Merencanakan Urutan Pertanyaan :

Funnel format



Closed-ended

Inverted funnel format



Open-ended

***Funnel Format* :**

Dengan format ini, pewawancara mulai dengan pertanyaan open-ended, kemudian menggunakan pertanyaan closed-ended, secara berangsur-angsur wawancara berakhir sampai informasi khusus yang dibutuhkan.

***Inverted Funnel Format* :**

Pewawancara mulai dengan pertanyaan closed-ended khusus dan secara berangsur-angsur yang diwawancarai memberikan gambaran keluar dari titik dimana dia akan menjawab dan memperluas jawaban untuk pertanyaan open-ended.

Psikologi Wawancara

Berkenaan dengan hubungan antar manusia. Berikut ini adalah pedoman-pedoman yang dapat dilakukan oleh pewawancara untuk melakukan tindak lanjut terhadap tingkah laku dari orang yang diwawancarai.

Ada dua sumber dasar informasi, sumber pertama yaitu apa yang orang katakan mengenai calon yang diwawancarai, dan sumber kedua adalah catatan atau dokumen organisasi seperti struktur organisasi atau resume organisasi.

Setelah selesai wawancara, pewawancara sebaiknya menyimpan atau mencatat dan mengevaluasi hasilnya. Dan perlu dibuat ringkasan dan laporan ke manajemen dan ke yang diwawancarai untuk menunjukkan bahwa apa yang dikatakan mereka cukup penting untuk dicatat.

Sampling

Sampling adalah aplikasi dari prosedur tertentu yang kurang dari 100% item dalam suatu survey untuk mengevaluasi atau estimasi beberapa

karakteristik dari populasi. Sampling berguna untuk menentukan karakteristik atau nilai dari seluruh item yang akan dipergunakan hingga selesai.

Sampling Statistik dan Non Statistik

Probabilitas statistik digunakan untuk mengontrol resiko sampling dalam sampling statistik. Dalam sampling non statistik, analis menentukan ukuran sampling dan mengevaluasi hasil seluruh sample berdasarkan pertimbangan dan pengalaman. Analis perlu mengetahui dimana harus dipergunakan jumlah sample yang besar, dan dimana untuk jumlah sample yang kecil.

Tahapan dalam Perencanaan Sampling

Sampling statistik mau pun non statistik tahapannya sama, hanya dalam sampling statistik tidak menggunakan formula matematika dan tabel statistik.

Tahapannya adalah :

- ✓ Menentukan tujuan sampling.
- ✓ Mendefinisikan populasi dan unit sampling.
- ✓ Menspesifikasikan karakteristik.
- ✓ Menentukan ukuran sample.
- ✓ Menentukan metode pemilihan sample dan melaksanakannya.
- ✓ Ada tiga metode sampling : random number, sampling sistematis dan sampling blok.
- ✓ Mengevaluasi hasil sample dan membuat suatu kesimpulan.

Observasi

Observasi memiliki banyak tujuan. Analis dapat menentukan apa yang akan dilakukan, bagaimana dilakukan, siapa yang mengerjakan, kapan dikerjakan, berapa lama dikerjakan, dimana dikerjakan, dan mengapa dikerjakan. Analis juga dapat berpartisipasi dalam pelaksanaan prosedur yang dilakukan oleh pegawai.

Beberapa cara mengobservasi :

Persiapan

Sebelum observasi dimulai, analis akan :

- ✓ Mengidentifikasi dan menentukan apa yang akan diobservasi
- ✓ Mengestimasi waktu untuk observasi
- ✓ Menjamin persetujuan manajemen untuk pelaksanaan observasi
- ✓ Menjelaskan ke organisasi yang diobservasi, apa yang akan dilakukan dan mengapa

Pelaksanaan Observasi

Observasi akan dikelola secara efektif oleh analis dengan mengikuti aturan berikut ini :

- ✓ Analis perlu terbiasa dengan lingkungan sekitar dan komponen dalam area yang diobservasi. Membiasakan dengan pekerjaan yang berjalan pada tempat tersebut.
- ✓ Selama observasi, analis secara berkala melakukan pencatatan.
- ✓ Analis perlu mencatat hal-hal yang khusus. Deskripsi yang umum dan samar sebaiknya dihindari.

- ✓ Jika analis berinteraksi dengan orang-orang yang diobservasi, analis sebaiknya berulang-ulang membuat komentar kualitatif dan penilaian.
- ✓ Analis perlu menunjukkan kebaikan dan keamanan selama observasi.

Analis kemungkinan bisa tidak memiliki banyak waktu dalam mengobservasi. Untuk itu, teknik sampling dapat digunakan secara efektif untuk mengurangi waktu yang dibutuhkan dalam mengobservasi dan tetap dapat mengumpulkan fakta / hasil yang pasti.

Menyimpulkan Analisis Sistem Dan Mengkomunikasi-Kan Temuan

Selama fase analisis sistem, analis perlu tetap memelihara komunikasi yang ekstensif dengan user, manajer proyek dan orang-orang proyek lainnya. Komunikasi tersebut meliputi :

- ✓ Feedback ke orang yang diwawancarai, atau diobservasi menurut apa yang analis ketahui.
- ✓ Verifikasi dengan user untuk menemukan keterhubungan fungsi dan aktivitas yang perlu analis indentifikasi.
- ✓ Melakukan pertemuan secara periodik untuk memberitahu manajer proyek dan personal proyek lainnya tentang progress, status dan ketepatan waktu.

Menyiapkan Laporan Analisis Sistem

Laporan Analisis Sistem disiapkan secara professional dengan menggunakan kertas, PC atau teknologi CASE. Beberapa aturan dalam

membuat professional documented deliverables sama seperti membuat laporan-laporan hasil penulisan.

Beberapa isi laporan analisis sistem yang perlu ada adalah :

- ✓ Alasan dan lingkup dari analisis sistem
- ✓ List dari masalah utama yang diidentifikasi
- ✓ Pernyataan yang lengkap dan definisi dari kebutuhan user
- ✓ List dari asumsi kritis
- ✓ Rekomendasi

Mempresentasikan secara oral

Penyampaian presentasi secara oral dari setiap dokumen analisis sistem memerlukan komunikasi yang lengkap dan jelas. Ada tiga metode yang dapat digunakan untuk prensentasi secara oral ini,yaitu:

- ✓ Mengingat
- ✓ Membaca
- ✓ Tanpa persiapan

Dapat pula mempresentasikan dengan bantuan teknologi audio visual. Ada empat hal yang dapat dihasilkan dari analisis sistem :

Proyek dilepas.

Proyek sistem yang dilepas dapat berasal dari masalah utama yang tidak dapat diselesaikan. Alasan lain, adanya perubahan dari prioritas sistem oleh pihak manajemen atau komite pengarah, yang mengakibatkan proyek sistem yang sekarang dilepas.

Proyek ditunda.

Pada saat ini, pihak manajemen akan menentukan sumber daya untuk proyek sistem lain dengan prioritas yang lebih tinggi. Maka instalasi untuk *backbone* telekomunikasi akan tertunda, sehingga proyek sistem yang sekarang ditunda. Beberapa pemakai kunci mungkin sedang berlibur atau tidak masuk untuk beberapa minggu, sehingga menyebabkan proyek ditunda sementara waktu dari SDLC.

Proyek diganti.

Hasil ini berarti bahwa aspek penting dari proposal sistem yang asli mempunyai perubahan yang berarti. Seperti perubahan yang melibatkan perluasan utama dan penyusunan dari lingkup sistem. Atau mungkin pemakai membutuhkan perubahan yang berarti dari perkiraan yang lebih cepat itu, menyebabkan kebutuhan sumber daya yang lebih banyak atau sedikit.

Proyek dilanjutkan.

Proyek sistem akan diteruskan seperti rencana dalam laporan analisis sistem.

3. Perancangan (*Desain*) Sistem

Desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas perangkat lunak

biasanya dinilai dari segi kepuasan pengguna perangkat lunak terhadap perangkat lunak yang digunakan.

Konsep Dasar Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak. sistem informasi, atau sistem lainnva). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nvata. Ada banvak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek mi, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennva dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya. dan dapat berinteraksi satu sama lain.

Karakteristik atau sifat-sifat yang dipunyai sebuah sistem berorientasi objek adalah sebagai berikut:

- ✓ *Abstraksi*

  Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan

- ✓ *Enkapsulasi*

  pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek. untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya

- ✓ *Pewarisan (inheritance)*

  mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya

- ✓ *Reusabilily*

  pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut

- ✓ *Generalisasi dan Spesialisasi*

  menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus

- ✓ *Komunikasi Antar Objek*

  komunikasi antar objek dilakukan lewat pesan (message) yang dikirim dan satu objek ke objek lainnya

- ✓ *Polymorphism*

kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

Metodologi Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnva. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek. perancangan berorientasi objek, pemrograman berorientasi objek. dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis. real-time, utility, dan sebagainva) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

- ✓ meningkatkan produktivitas , karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnva yang melibatkan objek tersebut (reusable)
- ✓ kecepatan pengembangan, karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnva kesalahan pada saat pengkodean
- ✓ kemudahan pemeliharaan, karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah
- ✓ adanya konsistensi, karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
- ✓ meningkatkan kualitas perangkat lunak, karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannva, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Pengertian Objek dan Kelas

Kelas adalah kumpulan dari objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunvai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi maka akan dibuat sebuah objek. Objek dilihat dari segi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehingga sebuah objek hanya ada saat sebuah program dieksekusi, jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas. Ilustrasi kelas dan objek dapat dilihat pada gambar berikut.

Enkapsulasi

Enkapsulasi dapat dianggap sebagai sebuah bungkusan. Enkapsulasi inilah yang diimplementasikan dalam sebuah kelas dimana di dalam sebuah kelas terdiri dari atribut dan metode yang dibungkus dalam suatu kelas. Enkapsulasi pada sebuah kelas bertujuan untuk melindungi atribut dan metode-metode yang ada di dalam kelas agar tidak sembarangan diakses oleh kelas lain.

Atribut

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

Operasi atau Metode (Method)

Operasi atau metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

Metode atau operasi dapat berasal dari : event, aktivitas atau aksi keadaan, fungsi, kelakuan dunia nyata. Contoh metode atau operasi misalnya Read, Write, Move. Copy. dan sebagainya.

Pengertian Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda. Ilustrasi dari sebuah package dapat dilihat pada gambar berikut.

Pengertian Antarmuka (Interface)

Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan

mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu. Metode pada antarmuka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka, misalnya pada antarmuka terdapat deklarasi metode printAnimal() maka pada kelas yang mengimplementasikan metode itu harus ditulis sama. Antarmuka atau interface biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.

Sekilas Pendekatan terstruktur

Teknik terstruktur, merupakan pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan untuk kemudian dapat disatukan kembali menjadi satu kesatuan yang dapat dipergunakan untuk memecahkan masalah.

Dalam hubungannya dengan pengembangan sistem informasi dan software aplikasi sistem informasi, pemrograman terstruktur adalah proses yang berorientasi kepada teknik yang digunakan untuk merancang dan menulis program secara jelas dan konsisten. Desain terstruktur merupakan salah satu proses yang berorientasi teknik yang digunakan untuk memilah-milah program besar ke dalam hirarki modul-modul yang menghasilkan program komputer yang lebih kecil agar mudah untuk diimplementasikan dan dipelihara (diubah). Analisis Terstruktur Modern merupakan teknik yang berorientasi kepada proses yang paling populer dan banyak digunakan dewasa ini. Pemodelan data merupakan suatu teknik yang berorientasi kepada data dengan menunjukkan sistem hanya datanya saja terlepas dari bagaimana data tersebut akan diproses atau digunakan untuk menghasilkan informasi.

Rekayasa Informasi merupakan perpaduan dari pemodelan data dan proses, juga memberikan penekanan baru terhadap pentingnya perencanaan sistem informasi.

Ciri-ciri utama teknik terstruktur adalah sebagai berikut:

- ✓ Merancang berdasar modul
- ✓ Modularisasi adalah proses yang membagi suatu sistem menjadi beberapa modul yang dapat beroperasi secara independen
- ✓ Bekerja dengan pendekatan top-down
- ✓ Dimulai dari level atas (secara global) kemudian diuraikan sampai ke tingkat modul (rinci)
- ✓ Dilakukan secara iterasi, dengan iterasi akan didapat hasil yang lebih baik, terlalu banyak iterasi juga akan menurunkan hasilnya dan menunjukkan bahwa tahap sebelumnya tidak dilakukan dengan baik kegiatan dilakukan secara paralel
- ✓ Pengembangan subsistem-subsistem dapat dilakukan secara paralel, sehingga akan memperpendek waktu pengembangan sistem

Secara teknis berikut adalah gambar ilustrasi teknik terstruktur di dalam program:

Perbandingan Pendekatan OO dan Terstruktur

Perbedaan yang paling dasar dari pendekatan terstruktur dan pendekatan OO (Object Oriented) atau berorientasi objek adalah pada metode berorientasi fungsi atau aliran data (Data Flow Diagram (DFD)) (pendekatan terstruktur), dekomposisi permasalahan dilakukan berdasarkan fungsi atau

proses secara hirarki, mulai dan konteks sampai proses-proses yang paling kecil, sementara pada metode berorientasi objek. dekomposisi permasalahan dilakukan berdasarkan objek-objek yang ada dalam sistem.

BAB-VI

TAHAPAN PENGEMBANGAN SISTEM

(Penerapan dan Pemeliharaan Sistem)

Overview

Pada bab ini menjelaskan mengenai difinisi penerapan dan pemeliharaan sistem, testing dan implementasi, tujuan pengujian, evaluasi , sasaran sistem , pemeliharaan sistem, prosedure pemeliharaan, Jenis pemeliharaan, tools pemeliharaan, dan mengembangkan perubahan sistem.

Tujuan

1. Mahasiswa mampu menerapakan tahapan tahapan dalam pengembangan sistem informasi pada tahapan *back-end*
2. Mahasiswa mampu menjelaskan dan menerapakan tahapan akhir dari pengembangan sistem mulai dari penerapaan sampai pada tahap pemeliharaan sistem.
3. Mahasiswa mampu menjelaskan dan mampu menerapakan atau menggunakan sistem baik secara langsung maupun bertahap serta mampu memrawat sistem dengan benar.

Materi

1. Testing dan Implementasi System Informasi

Setelah dilakukan penyusunan system maka dilakukan pengujian system dengan tujuan untuk mengetahui masih ada atau tidak kesalahan program,

kekurangan atau system yang disusun, kemudian dilakukan implementasi system. Langkah langkah dalam testing dan implementasi sistem :

- Melakukan pengujian terhadap system informasi / program aplikasi/aplikasi sebelum digunakan
- Menguji dan membandingan dengan system sebelumnya, untuk memunculkan keunggulan pada system yang lama dan mengurangi kesalahan pada system yang baru. System yang baru lebih baik dari pada system yang lama.
- Mervisi system yang diuji, sampai system benar benar dapat menyelesaikan maslah pada system / organisasi (revisi system sebelum system digunakan)
- Sestem yang sudah digunakan, berarti sudah melalui pengujian system dan system layak dioperasikan / digunakan.

Definisi Pengujian Sistem

- Suatu proses yang dilakukan untuk menilai apakah yang dirancang telah sesuai dengan apa yang diharapkan
- Suatu kegian untuk mengevaluasi keunggulan dan kelermahan terhadap sesuatu yang diuji (kwalitas produk)
- Mengevaluasi terhadap urutan kegiatan yang sistematis dalam mencapai tujuan system
- Mengevaluasi keseimbangan jumlah pelaksanaan kegiatan dengan beban kerja dalam sesuatu prosedur kegiatan

Pengujian dan Mengevaluasi

Hala hal yang terlibat dalam suatu kegiatan untuk mencapai tujuan yang diharapkan untuk sipengguna

Testing Proses yang dibuat sedemikian rupa untuk mengidentifikasikan adanya ketidak sesuaian hasil sebuah system informasi dengan apa yang diharapkan.

Berdasarkan pengertian diatas testing mempunyai beberapa tujuan :

- ✓ Testing dilakukan untuk memastikan mutu dari suatu produk yaitu menguji apakah produk (dalam hal ini system informasi) yang dihasilkan telah sesuai dengan mutu yang dipersyaratkan. Testing dilakukan untuk memastikan atau menjaga mutu suatu produk
- ✓ Testing merupakan proses analisa dan entitas software, pada testing ini bertujuan untuk mendeteksi adanya perbedaan antrar kondisi software yang ada dengan kondisi yang diinginkan, untuk melihat kerusakan suatu produk melakukan evaluasi fitur fitur dari software.

Pengujian Sistem

- Melakukan proses evaluasi terhadap system yang sudah ada apakah system sudah sesuai yang dharapkan user
- Menilai dan mengevaluasi terhadap output atau ahasil system
- Menguji terhadpa input, pengelolaan (proses)dan output system
- Melakukan penilaian dan evaluasi terhadpat komponen system prosedur pelaksanaan kegiatan dan mutu atau kwalitas hasil system

Pengujian terhadap system

Personil

- Personil ditempakan sudah sesuai dengan skill atau kemampuan yang dimilikinya
- Beban kerja yang optimum untuk masing masing personil

- Loyalitas atau kemamuan bekerja sama untuk menyelesaikan suatu kegiatan
- Kemampuan personil dalam menyelesaikan masalah

Pengujian kegiatan

- Prosedur dan system kerja yang sistematis
- Perencanaan yang terkontrol dan terjadwal
- Arah tujuan atau target ang dapat dilaksanakan sesuai dengan perencanaan
- Hasil kegiatan yang terukur
- Kesemimbangan kegiatan dengan bersarnya biaya yang digunakan

Pengujian misi atau tujuan

- Adanya integrasi antara personil yang terlibat dengan kegiatan yang dilaksanakn dalam mencapai target system
- Kwaliatas dari kegiatan yang mewujudkan tujuan system

Tujuan Testing dan Implementasi

Melakukan pengujian terhadap system informasi apakah sudah memenuhi kebutuhan user atau system informasi sudah layak digunakan dengan melalui :

- Uji analisis
- Uji perancangan
- Uji implementasi

Testing dan Sistem Informasi

Elemen kritis dari jaminan kwlitas dan mempresentasikan kajian pokok dari spesifkasi, desain dan pengkodean atau pngujian terhadap aplikasi system informasi berdasrkan komonen. System informasi yaitu pengujian model, pengujian output, database, teknologi dan pengujian kendali terhadap aplikasi system informasi. Sehingga memberikan system informasi yang berkwalitas terhadap user.

Testing system informasi atau pengujian system informasi à pengujian terhadap aplikasi system informasi (*software application*)

Program aplikasi

- Pengujian sintax
- Pengujian logika dan proses
- Pengujian output , yaitu program bebas dari kesalahan, keraguan dan kegagalan.

Pengujian terhadap komponen system informasi

- ✓ Model / proses

  Menguji hasil rancangan tentang proses atau moder dari suatu system informasi yang akan dbuat dengan program aplikasi. Contoh menguji flow chart KRS
- ✓ Output

  Menguji hasil rancangan dengan laporan yang disajikan aplikasi, apakah sudah memberikan informasi sesuai dengan kebutuhan user.
- ✓ Input

  Pengujian terhadap jumlah atau variable data yang dibutuhkan output.

- ✓ Database

  Menguji atribtu data , relasi antar file atau hubungan elemen data, julah database yang sesuai untuk menghaslkan output dan menampung data sesuai dengan form entri
- ✓ Teknologi

  Pengujian terhadap kemampuan peralatan yang digunakan, kemapuan software dalam menjalankan aplikasi, sehingga mampu memberikan informasi yang iteraktiv kepada user.contoh SS, SO, SA
- ✓ Control atau kendali

  Menguji terhadap keamanan dan hak akses dari aplikasi dan dapat memonitor

Konsep pengujian testing dan implementasi system informasi

- Pengujian terhadap mosel system informasi
- Pengujian terhadap model system
- Pengujian terhadap lgica, output, relasi data dalam menghasilkan laporan informasi kepada user.
- Pengujian terhadap tahapan analisis
- Pengujian tehadap tahapan perancangan
- Pengjian terhadap keunggulan dan kelemahan software dan hardware yang digunakan
- Pengujian terhadap kasus yang dibahas.

2. Tahap Penerapan Sistem

Penerapan merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Adapun tahapannya yaitu :

- ✓ **Merencanakan Penerapan;**

  Manajer dan spesialis informasi harus memahami dengan baik pekerjaan yang diperlukan untuk menerapkan rancangan sistem dan untuk mengembangkan rencana penerapan yang sangat rinci.

- ✓ **Mengumumkan Penerapan;**

  Proyek penerapan diumumkan kepada para pegawai dengan cara yang sama pada penelitian sistem. Tujuannya adalah untuk menginformasikan kepada para pegawai mengenai keputusan untuk menerapkan sistem baru dan meminta kerjasama mereka.

- ✓ **Mendapatkan Sumber Daya Perangkat Keras;**

  Rancangan sistem disediakan bagi para pemasok berbagai jenis perangkat keras yang terdapat pada konfigurasi sistem yang disetujui. Setiap pemasok diberikan *request for proposal (RFP)*, yang berisi antara lain :

  - Surat yang ditransmisikan
  - Tujuan dan kendala sistem
  - Rancangan sistem : deskripsi ringkasan, kriteria kerja, konfigurasi peralatan, dokumentasi sistem ringkasan, perkiraan volume transaksi, perkiraan ukuran file.
  - Jadual pemasangan

Selanjutnya mereka membuat usulan tertulis, bagaimana peralatan yang diusulkan akan membuat sistem mencapai kriteria kinerjanya. Ketika semua usulan telah diterima dan dianalisis, *Commite* memilih satu pemasok atau lebih. Spesialis informasi memberi dukungan bagi keputusan tersebut dengan mempelajari usulan dan membuat rekomendasi. Setelah disetujui, perusahaan melakukan pemesanan.

Contoh Kebutuh perangkat kerasnya adalah :

- Processor Pentium Dual Core, Recommended: 2.2 GHz atau yang lebih tinggi
- Harddisk terpasang 500 GB, rekomendasi minimum adalah 10 GB
- Memori (RAM) terpasang 2 GB, rekomendasi minimum 512 MB
- Mouse dan Keyboard
- Resolusi Layar Monitor 1024 x 800
- Printer inkjet Canon MP270

✓ **Mendapatkan Sumber Daya Perangkat Lunak;**

Ketika perusahaan memutuskan untuk menciptakan sendiri perangkat lunak aplikasinya, programmer menggunakan dokumentasi yang disiapkan oleh analis sistem sebagai titik awal. Programmer dapat menyiapkan dokumentasi yang lebih rinci seperti flowchart atau bahasa semu (*psedudo code*) yang terstruktur, dilakukan pengkodean, dan pengujian program. Hasil

akhirnya adalah software library dari program aplikasi. Jika peangkat lunak aplikasi jadi (prewritten application software) dibeli, pemilihan pemasok perangkat lunak dapat mengikuti prosedur yang sama seperti yang digunakan untuk memilih pemasok perangkat keras, yaitu RFP dan Usulan.

Contoh Kebutuh perangkat lunaknya adalah :

- Sistem operasi : Microsoft Windows 7 Ultimate
- Bahasa Pemrograman : Borland Delphi 7
- Software Report : Quick Report (Bawaan Borland Delphi 7)
- Software DBMS : Microsoft Accsess 2003

✓ **Menyiapkan database;**

Pengelola database (*database administrator – DBA*) bertanggung jawab untuks emua kegiatan ynag berhubungan dengan data, dan mencakup persiapan database. Hal tersebut memerlukan pengumpulan data baru atau data yang telah ada perlu dibentuk kembali sehingga sesuai dengan rancangan sistem baru dan menggunakan sistem manajemen basis data (database management sistem – DBMS).

✓ **Menyiapkan fasilitas fisik;**

Jika perangkat keras dan sistem baru tidak sesuai dengan fasilitas yang ada, perlu dilakukan konstruksi baru atau perombakan. Sehingga pembangunan fasilitas tersebut merupakan tugas berat

dan harus dijadualkan sehingga sesuai dengan keseluruhan rencana proyek.

- ✓ **Mendidik peserta dan pemakai;**

  Sistem baru kemungkinan besar akan mempengaruhi banyak orang. Beberapa orang akan membuat sistem bekerja. Mereka disebut dengan peserta, yang meliputi operator entry data, pegawai coding, dan pegawai administrasi lainnya. Semuanya harus dididik tentang peran mereka dalam sistem. Pendidikan harus dijadualkan jauh setelah siklus hidup dimulai, tepat sebelum bahan-bahan yang dipelajari mulai diterapkan.

- ✓ **Menyiapkan usulan cutover;**

  Proses menghentikan penggunaan sistem lama dan memulai menggunakan sistem baru disebut cutover. Ketika seluruh pekerjaan pengembangan hampir selesai , tim proyek merekomendasikan kepada manajer agar dilaksanakan cutover (dalam memo atau laporan lisan)

- ✓ **Menyetujui atau menolak masuk ke sistem baru;**

  Manajer dan SC MIS menelaah status proyek dan menyetujui atau menolak rekomendasi tersebut. Bila manajemen menyetujui maka manajemen menentukan tanggal cutover. Namun, bila manajemen menolak maka manajemen menentukan tindakan yang harus diambil dan tugas yang harus diselesaikan sebelum cutover akan dipertimbangkan kembali, kemudian manajemen menjadualkan tanggal baru.

✓ **Masuk ke Sistem Baru.**

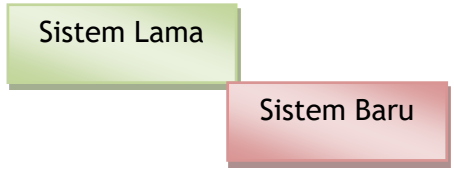
- **Konversi Langsung (*Direct Conversion*)**

  Konversi ini dilakukan dengan cara menghentikan sistem lama dan menggantikannya dengan sistem baru.. Konversi langsung adalah pengimplementasian sistem baru dan pemutusan jembatan sistem lama, yang kadang-kadang disebut pendekatan cold turkey. Apabila konversi telah dilakukan, maka tak ada cara untuk balik ke sistem lama.

  Asumsi dari penggunaan sistem ini diantarnya

  - ❖ Data sistem yang lama bias digantikan sistem yang baru
  - ❖ Sistem yang lama sepenuhnya tidak bernilai.
  - ❖ Sistem yang barn bersifat kecil atau sederhana atau keduanya.
  - ❖ Rancangan sistem baru sangat berbeda dari sistem lama, dan perbandingan antara sistem – sistem tersebut tidak berarti.

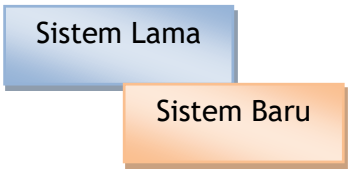
  Kelebihan dari sistem konversi langsung ini yaitu sistem ini relatif murah, namun memiliki resiko kegagalan yang cukup besar.



Gambar 1. Konversi Langsung

- **Konversi Paralel (*Parallel Conversion*)**

Konversi Paralel adalah suatu pendekatan dimana baik sistem lama dan baru beroperasi secara serentak untuk beberapa période waktu. Pada konversi ini, sistem baru dan sistem lama sama-sama dijalankan. Setelah melalui masa tertentu, jika sistem baru telah bisa diterima untuk menggantikan sistem lama, maka sistem lama segera dihentikan. Kelebihan dari sistem ini yaitu merupakan pendekatan yang paling aman sedangkan kelemahan dari sistem ini adalah merupakan cara yang paling mahal, karena pemakai harus menjalankan dua sistem sekaligus. Besarnya biaya dikeluarkan untuk penduplikasian fasilitas-fasilitas dan biaya personel yang memelihara sistem rangkap tersebut.



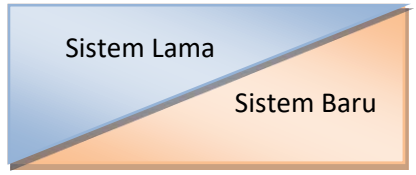
Gambar 2. Konversi Paralel

- **Konversi Bertahap (*Phase-In Conversion*)**

  Konversi ditakukan dengan menggantikan suatu bagian dari sistem lama dengan sistem baru. Jika terjadi sesuatu, bagian yang baru tersebut akan diganti kembali dengan yang lama. Jika tak terjadi masalah, modul-modul baru akan dipasangkan lagi untuk mengganti modul-modul lama yang lain. Dengan pendekatan seperti ini, akhirnya semua sistem lama akan tergantikan oleh sistem baru. Kelebihan dari sistem konversi ini yaitu kecepatan perubahan dalam

organisasi tertentu bisa diminimasi, dan sumber-sumber pemrosesan data dapat diperoleh sedikit demi sedikit selama période waktu yang luas. Sedangkan kelemahannya yitu keperluan biaya yang harus diadakan untuk mengembangkan interface temporer dengan sistem lama, daya terapnya terbatas, dan terjadi kemunduran semangat di organisasi, sebab orang-orang tidak pernah merasa menyelesaikan sistem.

Sistem konversi ini dianggap lebih aman daripada konversi langsung. Dengan metode Konversi Phase-in, sistem baru diimplementasikan beberapa kali, yang secara sedikit demi sedikit mengganti yang lama. la menghindarkan dari risiko yang ditimbulkan oleh konversi langsung dan memberikan waktu yang banyak kepada pemakai untuk mengasimilasi perubahan. Untuk menggunakan metode phase-in, sistem harus disegmentasi.

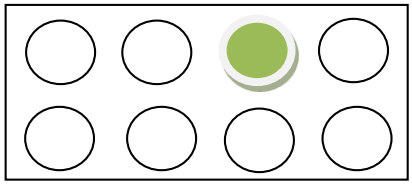


Gambar 3. Konversi Bertahap

- **Konversi Pilot (*Pilot Conversion*)**

  Pendekatan ini dilakukan dengan cara menerapkan sistem baru hanya pada lokasi tertentu yang diperlakukan sebagai pelopor. Jika konversi ini dianggap berhasil, maka akan diperluas ke tempat-tempat yang lain. Ini merupakan

pendekatan dengan biaya dan risiko yang rendah. Dengan metode Konversi Pilot, hanya sebagian dari organisasilah yang mencoba mengembangkan sistem baru. Kalau metode phase-in mensegmentasi sistem, sedangkan metode pilot mensegmentasi organisasi.



Gambar 4. Konversi Pilot

Cara untuk mengkonversi file adalah dengan cara :

- **Konversi File Gradual**

  Konversi file gradual adalah konversi yang dilakukan dengan beberapa tahapan. Biasanya digunakan dengan metode paralel dan phase-in. Cara kerja metode ini adalah :

  - ❖ Suatu transaksi diterima dan dimasukkan ke dalam sistem.
  - ❖ Program mencari file master baru.
  - ❖ Jika record tersebut tidak ditemukan dalam file master baru, file master lama diakses untuk record yang tepat, dan ditambahkan ke file master baru dan di update.
  - ❖ Jika transaksi tersebut adalah record baru maka record baru disiapkan dan ditambahkan ke file master baru.

- **Konversi File Total**

Konversi file total adalah konversi file dengan cara memformat sistem lama ke sistem baru dengan suatu software tertentu. Rancangan file baru hampir selalu mempunyai field-field record tambahan, struktur pengkodean baru, dan cara baru perelasian item-item data (misalnya, file-file relasional). Dengan menggunakan klasifikasi file perlu diperhatikan jenis prosedur kendali yang digunakan selama konversi.

Rencana Implementasi

Implementasi sistem lama ke sistem baru akan melibatkan :

- Tim implementasi terdiri dari :
  - ❖ Profesional sistem yang merancang sistem
  - ❖ Para manajer dan beberapa staff
  - ❖ Perwakilan Vendor
  - ❖ Pemakai Primer
  - ❖ Pengcode
  - ❖ Teknisi
- Keperluan implementasi sistem
  - ❖ Persiapan tempat
  - ❖ Pelatihan personil
  - ❖ Persiapan/pembuatan dokumentasi
  - ❖ Konversi file dan sistem
  - ❖ Peninjauan Pasca Implementasi

  Hal-hal yang dapat dilakukan untuk menghindari kegagalan penerapan sistem yaitu :

- ❖ Adanya keteraturan dan saling terkait agar terjadi kerjasama yang baik.
- ❖ Menerapkan sistem dengan memperhatikan semua aspek yang terlibat di dalamnya
- ❖ Kembali ke tujuan awal dari perusahaan dan sistem harus mampu memenuhi kebutuhan-kebutuhan yang telah ditetapkan untuk dikembangkan
- ❖ Memperhatikan sumber daya manusia dengan cara melakukan sosialisasi dan pelatihan.

Perkembangan teknologi yang begitu cepat dan persaingan yang semakin kompetitif menyebabkan perusahaan harus melakukan perubahan pada sistem informasinya. Hal ini dilakukan agar perusahan tidak ketinggalan dengan pesaing lainnya. Akan tetapi kadangkala terjadi kesalahan besar yang dapat berakibat fatal pada suatu organisasi pada saat pengalihan sistem informasi organisasi tersebut dari suatu sistem lama ke sistem yang baru.

Fenomena kesalahan dalam konversi sistem informasi dapat terjadi apabila tidak dilakukan langkah-langkah awal dengan tepat sebelum dilakukan konversi. Adapun hal yang perlu dilakukan sebelum proses konversi yaitu 1) Proses perencanaan dan permodelan, meliputianalisa kebutuhan dan design, 2) konstruksi, meliputi penyusunan kode dan pengujian 3) Pemrograman dan pengetesan perangkat lunak (software), meliputi kegiatan : Developmental (error testing per modul oleh programmer), Alpha testing (error testing ketika sistem digabungkan dengan interface user oleh software tester), dan Beta testing (testing dengan lingkungan dan data sebenarnya).

Langkah langkah yang dilakukan agar kesalahan alih sistem informasi dapat dihindari :

- Lihat kembali dan koreksi visi yang ingin di bangun, pelajari implementasi apa yang belum maksimal dan latih sumber daya manusia agar mampu mengoptimalkan peranti yang sudah dibeli. Hal ini hanya akan mungkin untuk dilaksanakan apabila pimpinan perusahaan mengetahui tentang TI/sedikit tentang TI, sehingga dia paham apa yang ingin dicapai perusahaannya dengan mengaplikasikan TI ini.
- Harus menciptakan sinergisme diantara subsistem-subsistem yang mendukung pengoperasian sistem sehingga akan terjadi kerjasama secara terintegrasi diantara subsistem-subsistem ini. Asumsi hanya akan tercapai apabila para perancang sistem ini mengetahui masalah-masalah informasi apa yang ada di perusahaan dan yang harus segera di selesaikan. Biasanya para perancang sistem ini akan mulai pada tingkat perusahaan, selanjutnya turun ke tingkat-tingkat sistem.

Fenomena penyebab kegagalan ini dapat berasal dari 3 (tiga) stakeholder utamanya dari organisani/perusahaan tersebut, yaitu: management yang mewakili pihak perusahaan, vendors sebagai pihak ketiga yang membantu implementasi sistem baru tersebut, dan user sebagai pihak yang menggunakan sistem tersebut. Management adalah salah satu penyebab dari kesalahan peralihan sistem lama ke sistem baru, hal tersebut dapat terjadi oleh beberapa faktor, antara lain

- Kurangnya dukungan dan komitmen dari pimpinan puncak dan manajemen perusahaan, sehingga inisiatif sistem baru

yang digulirkan berjalan dengan tersendat-sendat, Buruknya perencanaan yang disusun oleh pihak manajemen sehingga ketika ingin dieksekusi mengalami banyak hambatan dan kesulitan.

- Ketidakinginan manajemen dalam "merubah paradigma" berpikir maupun bekerja ebih senang kondisi status quo sehingga berbagai prasyarat utama untuk menjalankan atau mengimplementasikan sistem baru tersebut tidak tercapai
- Ekspektasi yang terlampau berlebihan dari pihak manajemen terhadap sistem baru yang ingin diterapkan tanpa perduli dengan isu-isu terkait dengan cara atau pendekatan atau strategi menerapkan sistem tersebut secara efektif,
- Pendefinisian kebutuhan yang kabur, sehingga ruang lingkup sistem baru yang ingin diterapkan menjadi tidak jelas yang tentu saja mempertinggi resiko kegagalan dalam implementasinya.
- Sosialisasi mengenai sistem baru yang buruk kepada segenap karyawan perusahaan sehingga banyak pihak yang menolak dibandingkan dengan yang mendukung. Pihak berikutnya yang dapat menyebapkan terjadinya kegagalan peralihan sistem informasi lama ke sistem informasi baru adalah pihak ketiga atau vendor.

Faktor faktor kegagalan yang disebapkan oleh vendor :

- Kurangnya pengalaman dari vendor maupun orang yang ditugaskan untuk mengimplementasikan sistem baru tersebut terutama untuk ruang lingkup penugasan serupa di industri yang sejenis.

- Tidak mampu memberikan pemahaman dan penjelasan yang baik dan benar mengenai paradigma yang dipergunakan dalam sistem baru kepada mereka yang berkepentingan sehingga seringkali terjadi kekeliruan dalam cara memandangnya.
- Pemilihan aplikasi yang keliru, atau tidak sesuai dengan situasi dan kondisi perusahaan yang membutuhkannya.
- Salah dalam usaha membantu manajemen dalam mendefinisikan kebutuhannya sehingga ketika sistem baru tersebut diterapkan, tidak memberikan hasil sebagaimana yang diharapkan oleh para stakeholder terkait.
- Tidak memberikan pelatihan yang memadai dan efektif kepada segenap stakeholder sehingga mereka tidak dapat menggunakan dan memanfaatkannya secara baik. Pihak terakhir yang memiliki andil besar dalam dalam penyebap kegagalan sistem informasi lama ke sistem informasi baru adalah user sebagai pihak yang menggunakan sistem tersebut.

Faktor-faktor yang dapat ditimbulkan oleh user adalah :

- Ketidakinginan para user untuk merubah cara kerja dalam beraktivitas sehari-hari sehingga selalu menentang segala bentuk aplikasi sistem baru tersebut, yang pada dasarnya membutuhkan keinginan dan kemampuan untuk bekerja dengan cara yang lebih efektif dan efisien.
- Kurangnya porsi pelatihan bagi para user agar yang bersangkutan memiliki kompetensi dan keahlian yang memadai untuk menjalankan sistem baru tersebut.
- Harapan yang berlebihan dan cenderung keliru terhadap sistem yang baru yang biasanya para user menganggap bahwa

teknologi informasi dan software dapat menyelesaikan segala masalah dan kesulitan yang ada.

Perubahan SI lama ke SI baru dapat mengakibatkan kesalahan yang beresiko dan berakibat fatal bagi jalannya suatu organisasi apabila tidak tepat dalam pelaksanaan SI barunya. Dalam memperkecil resiko yang ada, maka perlu kiranya diperhatikan berbagai cara dalam mengkonversi sistem dan langkah-langkah yang harus dilakukan sebelum pengalihan sistem informasi. Menurut O'Brien (2005) operasi awal dari sistem bisnis yang baru dapat menjadi tugas yang sulit. Hal ini biasanya memerlukan proses konversi dari penggunaan sistem yang ada saat ini (sistem lama) ke operasi aplikasi yang baru atau yang lebih baik.

3. Pemeliharaan Sistem

Pemeliharaan Sistem adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu sistem dalam, atau memperbaikinya sampai, suatu kondisi yang bisa diterima. Pada bulan April 1970 didefinisikan sebuah istilah untuk Teknologi Pemeliharaan yang mencakup pengertian yang lebih luas dari pada pengertian Pemeliharaan diatas. Istilah ini adalah Teroteknologi.

Sistem perlu dipelihara karena beberapa hal, yaitu :

- ✓ Sistem memiliki kesalahan yang dulunya belum terdeteksi, sehingga kesalahan-kesalahan sistem perlu diperbaiki.
- ✓ Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.

- ✓ Sistem mengalami perubahan karena perubahan lingkungan luar (perubahan bisnis).
- ✓ Sistem terinfeksi malware aktif
- ✓ Sistem berkas corrupt
- ✓ Perangkat keras melemah

Pemeliharaan sebuah sistem teknologi informasi dapat dikatakan seba gian besar bergantung pada ketersediaan sumber daya manusia di bidang IT yang memiliki kemampuan yang sesuai dengan kebutuhan dan kompleksitas sistim yang dimiliki. Alternatif yang terbaik bila perusahaan tidak ingin direpotkan dengan permasalahan di bidang IT yang hanya dianggap sebagai bidang penunjang adalah dengan menjalin kontrak pemeliharaan dengan perusahaan di bidang jasa teknologi informasi.

Tahap pemeliharaan dilakukan setelah tahap implementasi. Sistem baru yang berjalan digunakan sesuai dengan keperluan organisasi. Selama masa hidupnya, sistem secara periodik akan ditinjau. Perubahan dilakukan jika muncul masalah atau jika ternyata ada kebutuhan baru. Selanjutnya, organisasi akan menggunakan sistem yang telah diperbaiki tersebut.

Hal dalam Pemeliharaan Sistem meliputi :

- ✓ *System Back-Up*

  Membuat Salinan/copy untuk data-data penting perusahaan yang ada pada computer user maupun server ke dalam backup storage (External Disk).

- ✓ *System Optimization*

  Melakukan Defragmentasi data dan membuang sampah-sampah yang ada pada computer, serta memperbaiki kesalahan setting sehingga computer dapat berjalan normal.

- ✓ *System Rebuild*

  Membangun dan menata ulang kembali system yang rusak oleh faktor yang tidak disengaja, agar system dapat bekerja normal kembali.

- ✓ *System Upgrade*

  Menambah fungsi, memperbaharui system yang ada sesuai dengan kebutuhan pelanggan, serta melakukan testing stabilitas untuk hardware dan software.

- ✓ *Training dan Pelatihan*

  Memberikan Pengarahan dan konsultasi kepada operator computer, sehingga operator dapat mengoperasikan computer sesuai dengan prosedur pengoperasian komputer yang baik dan benar.

- ✓ *Update Anti Virus & Pembersihan Virus*

  Melakukan Update Definition file Anti Virus sehingga anti virus yang ada dapat memproteksi komputer dari serangan virus baik virus lama amaupun baru, dan juga melakukan scaning virus serta membersihkan komputer dari Virus.

- ✓ *System Security*

  Pemasangan Firewall dan sistem authentifikasi untuk pengamanan system dan data penting perusahaan dari orang luar yang tidak berkepentingan.

a. Prosedur Pemeliharaan Sistem

- ✓ *Langkah-Langkah Pemeliharaan Sistem*

- Penggunaan Sistem, yaitu menggunakan sistem sesuai dengan fungsi tugasnya masing-masing untuk operasi rutin atau sehari-hari.
- Audit Sistem, yaitu melakukan penggunaan dan penelitian formal untuk menentukan seberapa baik sistem baru dapat memenuhi kriteria kinerja. Hal semacam ini disebut penelaahan setelah penerapan dan dapat dilakukan oleh seorang auditor internal.
- Penjagaan Sistem, yaitu melakukan pemantauan untuk pemeriksaan rutin sehingga sistem tetap beroperasi dengan baik. Selain itu juga untuk menjaga kemutakhiran sistem jika sewaktu-waktu terjadi perubahan lingkungan sistem atau modifikasi rancangan software.
- Perbaikan Sistem, yaitu melakukan perbaikan jika dalam operasi terjadi kesalahan (bugs) dalam program atau kelemahan rancangan yang tidak terdeteksi saat tahap pengujian sistem.
- Peningkatan Sistem, yaitu melakukan modifikasi terhadap sistem ketika terdapat potensi peningkatan sistem setelah sistem berjalan beberapa waktu, biasanya adanya potensi peningkatan sistem tersebut terlihat oleh manajer kemudian diteruskan kepada spesialis informasi untuk dilakukan modifikasi sesuai keinginan manajer.

✓ *Tahapan-Tahapan Siklus Hidup Pemeliharaan Sistem (SMLC)*

Tahap pemeliharaan dilakukan setelah tahap implementasi. Sistem baru yang berjalan digunakan sesuai dengan keperluan organisasi. Selama masa hidupnya, sistem secara periodik akan ditinjau. Perubahan dilakukan jika

muncul masalah atau jika ternyata ada kebutuhan baru. Selanjutnya, organisasi akan menggunakan sistem yang telah diperbaiki tersebut.

Pemeliharaan sistem dilaksanakan untuk 5 alasan, yaitu :

- ✓ Memperbaiki kesalahan
- ✓ Menjaga kemutakhiran sistem
- ✓ Meningkatkan sistem
- ✓ Menyiapkan usulan rekayasa ulang
- ✓ Menyetujui atau menolak rekayasa ulang sistem

Siklus terakhir dari SDLC Pemeriksaan periodik, audit dan permintaan pengguna akan menjadi source untuk melakukan perawatan sistem diseluruh masa hidup sistem.

Siklus Hidup Pemeliharaan Sistem (SDLC), yaitu:

- ✓ Permintaan Perubahan
- ✓ Mengubah permohonan pemeliharaan menjadi suatu perubahan
- ✓ Menspesifikasi perubahan
- ✓ Membangun pengganti
- ✓ Menguji pengganti
- ✓ Melatih pengguna dan melakukan tes penerimaan
- ✓ Pengkonversian dan pelepasan ke operasi
- ✓ Mengupdate dokumentasi
- ✓ Melakukan pemeriksaan pasca implementasi

b. Jenis-Jenis Pemeliharaan Sistem

Jenis Pemeliharaan terbagi menjadi 4 jenis pemeliharaan, yaitu :

- ✓ Pemeliharaan Korektif

  Pemeliharaan korektif adalah bagian pemeliharaan sistem yang tidak begitu tinggi nilainya dan lebih membebani, karena pemeliharaan ini mengkoreksi kesalahan-kesalahan yang

ditemukan pada saat sistem berjalan. Umumnya pemeliharaan korektif ini mencakup kondisi penting atau bahaya yang memerlukan tindakan segera. Kemampuan untuk mendiagnosa atau memperbaiki kesalahan atau malfungsi dengan cepat sangatlah berharga bagi perusahaan.

- ✓ Pemeliharaan Adaptif

  Pemeliharaan adaptif dilakukan untuk menyesuaikan perubahan dalam lingkungan data atau pemrosesan dan memenuhi persyaratan pemakai baru. Lingkungan tempat sistem beroperasi adalah dinamik, dengan demikian, sistem harus terus merespon perubahan persyaratan pemakai. Misalnya, Undang-Undang Perpajakan yang baru mungkin memerlukan suatu perubahan dalam kalkulasi pembayaran bersih. Umumnya pemeliharaan adatif ini baik dan tidak dapat dihindari.

- ✓ Pemeliharaan Perfektif (Penyempurnaan)

  Pemeliharaan penyempurnaan mempertinggi cara kerja atau maintainabilitas (kemampuan untuk dipelihara). Tindakan ini juga memungkinkan sistem untuk memenuhi persyaratan pemakai yang sebelumnya tidak dikenal. Ketika membuat perubahan substansial modul apapun, petugas pemeliharaan juga menggunakan kesempatan untuk meng-upgrade kode, mengganti cabang-cabang yang kadaluwarsa, memperbaiki kecerobohan, dan mengembangkan dokumentasi. Sebagai contoh, kegiatan pemeliharaan ini dapat berbentuk perekayasaan ulang atau restrukturisasi perangkat lunak, penulisan ulang dokumentasi, pengubahan format dan isi laporan, penentuan logika pemrosesan

yang lebih efisien, dan pengembangan efisiensi pengoperasian perangkat.

- ✓ Pemeliharaan Preventif

  Pemeliharaan Preventif terdiri atas inspeksi periodik dan pemeriksaan sistem untuk mengungkap dan mengantisipasi permasalahan. Karena personil pemeliharaan sistem bekerja dalam sistem ini, mereka seringkali menemukan cacat-cacat (bukan kesalahan yang sebenarnya) yang menandakan permasalahan potensial. Sementara tidak memerlukan tindakan segera, cacat ini bila tidak dikoreksi di tingkat awal, jelas sekali akan mempengaruhi baik fungsi sistem maupun kemampuan untuk memeliharanya dalam waktu dekat.

c. Prosedur Pemeliharaan Sistem

Prosedur Pemeliharaan Sistem meliputi hal sebagai berikut :

- ✓ SDLC dan SWDLC
- ✓ Definisi data standar
- ✓ Bahasa pemrograman standar
- ✓ Rancangan Moduler
- ✓ Model yang dapat digunakan kembali
- ✓ Dokumentasi standar
- ✓ Kontrol sentral

d. Alat-Alat Untuk Pemeliharaan Sistem

Case Tools yang digunakan untuk membantu pemeliharaan sistem dari sistem lama ke dalam sistem baru adalah:

- ✓ Rekayasa Maju (Forward engineering)

Proses tradisional perpindahan dari logika dan abstraksi dari tingkat tinggi, perancangan implementasi yang independen untuk implementasi fisik dari sistem. Rekayasa maju mengikuti urutan kebutuhan melalui perancangan implementasinya.

- ✓ Rekayasa Mundur (Reverse Engineering)

  Proses menganalisa suatu sistem untuk mengidentifikasi elemen-elemen sistem dan antar hubungannya berdasarkan sistem yang ada, serta untuk menciptakan dokumentasi dalam tingkat abstraksi yang lebih tinggi dari sekarang. Untuk mendapatkan gambarannya proses pengembangan sistem dari awalnya. Tidak mengubah fungsinalitas sistem yang ada. Diterapkan untuk yang tidak ada sistem dokumentasinya.
- ✓ Rekayasa Ulang (Reengineering)

  Rekayasa ulang merupakan analisis yang bersifat menyeluruh dan lengkap dari proses bisnis dan sistem informasi guna mencapai peningkatan kinerja secara dramatis.
- ✓ Restrukturisasi (Restrukturing)

  Transformasi suatu sistem menjadi sistem baru atau bentuk lain tanpa mengubah fungsionalitasnya.

e. Mengatur Pemeliharaan Sistem

- ✓ Tentukan jadwal maintenance pada system yang kita miliki
- ✓ Update software yang compatible terhadap system kita
- ✓ Gunakan tenaga ahli yang terpercaya untuk

f. Mengembangkan Perubahan Sistem Manajemen

Salah satu konsep yang dibahas dan di analisis paling sering dalam beberapa tahun terakhir telah terjadi perubahan organisasi dan konsep terkait

resistensi terhadap perubahan dan manajemen perubahan. Perubahan telah banyak didefinisikan sebagai membuat perbedaan materi dalam sesuatu dibandingkan dengan keadaan sebelumnya, atau mengubah sesuatu, atau hanya menjadi berbeda. Semua definisi ini dapat diterapkan untuk mengubah seperti itu terjadi dalam organisasi dan bisnis. Perubahan organisasi bisa berarti perubahan teknologi infrastruktur (misalnya, bergerak dari lingkungan mainframe untuk komputasi terdistribusi), strategi pemasaran (target basis pelanggan baru), atau manajemen dan praktek pengambilan keputusan. Perubahan organisasi bukanlah hal baru dengan lanskap bisnis Amerika.

Sejak abad kesembilan belas dan Revolusi Industri, perusahaan harus berurusan dengan perubahan dalam skala yang semakin cepat. Semakin besar perkembangan teknologi dan semakin besar jumlah produk dan informasi yang dihasilkan, semakin diperlukan menjadi bagi perusahaan untuk memberikan manajemen yang efektif dan mengembangkan praktek organisasi yang solid Para profesional bisnis yang paling dihormati di Amerika Serikat telah orang-orang yang paling mampu memanfaatkan perubahan dalam bisnis dan perekonomian. Sebagai contoh, pada akhir abad kesembilan belas, Andrew Carnegie sangat memperluas kerajaannya dengan membeli usaha yang sangat ia bergantung pada untuk bisnis baja-nya, membuat satu perusahaannya contoh sukses pertama dari integrasi vertikal.

Dimulai pada 1990-an, perubahan datang pada tingkat yang secara eksponensial lebih cepat karena faktor-faktor seperti persaingan yang meningkat dalam ekonomi global, memperluas pasar, cara-cara baru melakukan bisnis (seperti e-commerce), dan tugas di mana-mana menjaga dengan yang terbaru teknologi. Guru manajemen Peter F. Drucker mengabdikan bukunya Manajemen Tantangan dari 21 abad ke topik yang

sangat. Akibatnya, perusahaan harus merevisi (atau menyusun) misi perusahaan dan tujuan, praktek manajemen, dan fungsi bisnis sehari-hari. Perusahaan secara rutin mulai merancang ulang strategi bisnis, sering menggantikan bagan organisasi tradisional hirarki dengan struktur datar berpusat di sekitar “diberdayakan” tim.

Tujuan utama bagi kebanyakan organisasi adalah untuk perubahan iklim dan budaya perusahaan. iklim Suatu organisasi dapat didefinisikan oleh bagaimana karyawan melihat alasan mendasar organisasi. karena, khususnya, misi perusahaan secara keseluruhan dan tujuan dan bagaimana pentingnya pengertian karyawan kesejahteraan adalah tujuan-tujuan tersebut. Iklim perusahaan kemudian melahirkan budaya organisasi yang terdiri dari apa karyawan lihat sebagai kepercayaan manajemen dan sistem nilai. Kedua hal, iklim dan budaya, kemudian menentukan bagaimana setiap manajer dan karyawan bentuk kinerja nya sendiri, biasanya dalam rangka paling berhasil mencapai tujuan perusahaan dan mudah-mudahan memastikan keberhasilan sendiri serta perusahaan. Faktor-faktor ini mempengaruhi setiap aspek dari pekerjaan setiap orang termasuk proses pengambilan keputusan, pola komunikasi dalam organisasi, dan tanggung jawab individu dan tanggung jawab perusahaan.

BAB-VII

METODE PENELITIAN & PENGEMBANGAN

(*Research and Development*)

Overview

Pada bab ini menjelaskan mengenai metode pengembangan sistem dengan model *Research and Development*, dimana metode penelitian dan pengembangan ini dengan menggunakan model Borg and Gall dan Model Sugiyono dimana kedua model ini menggunakan sepuluh langkah yang di mulai dari identifikasi kebutuhan sampai pada tahap *deseminasi* atau penyebaran produk.

Tujuan

1. Mahasiswa mengetahui definisi penelitian pengembangan, serta dapat memahami langkah-langkah dalam penelitian pengembangan
2. Mahasiswa mampu mengaplikasikan model penelitian pengembangan ini, dengan mengikuti langkah langkah yang sudah ada dalam penelitiannya baik untuk skripsi maupun untuk penelitian diluar
3. Mahasiswa mampu membedakan penelitian pengembangan yang ada baik dari Borg and Gall , Sugiyono maupun yang lainnya.

Materi

1. Pengertian Penelitian dan Pengembangan

Metode penelitian dan pengembangan atau dalam bahasa inggrisnya *Research and Development* adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut.

Untuk dapat menghasilkan produk tertentu digunakan penelitian yang bersifat analisis kebutuhan dan untuk menguji keefektifan produk tersebut supaya dapat berfungsi di masyarakat luas, maka diperlukan penelitian untuk menguji keefektifan produk tersebut.

Pada mulanya penelitian R&D ini diaplikasikan di dunia industri, dan merupakan ujung tombak dari dunia industri dalam menghasilkan suatu poduk baru, yang mana produk ini benar-benar dibutuhkan oleh pasar. Menurut Borg and Gall (1989), hampir 4% biaya dipakai untuk melakukan penelitian R&D ini, bahkan pada bidang - bidang tertentu seperti pada bidang komputer dan farmasi alokasi dananya bisa melebihi dari 4%. Sedangkan, untuk bidang – bidang pendidikan dan sosial peranan R&D masihlah amat kecil yaitu kurang dari 1% dari biaya pendidikan secara keseluruhan. Hal inilah yang dianggap sebagai salah satu alasan utama mengapa kemajuan di dalam bidang pendidikan sedikit tertinggal bila dibandingkan dengan bidang-bidang yang lain.

Borg and Gall (1989) menyatakan ***educational research and development is a process used to develop and validate educational product*.** Yang mana artinya bahwa penelitian pengembangan pendidikan (R&D) merupakan sebuah proses yang dipakai untuk mengembangkan dan

memvalidasi produk pendidikan. Hasil dari penelitian pengembangan ini tidak hanya untuk pengembangan sebuah produk yang sudah ada saja, melainkan juga untuk menemukan suatu pengetahuan atau jawaban atas permasalahan praktis. Menurut Sugiyono (2009) metode penelitian dan pengembangan (R&D) merupakan **metode penelitian yang dipakai untuk menghasilkan produk tertentu, dan menguji keektifan produk itu**. Agar bisa menghasilkan suatu produk tertentu yang dipakai untuk penelitian yang bersifat analisis kebutuhan (digunakan metode *survey atau kualitatif*) dan untuk menguji keefektifan produk tersebut supaya bisa berfungsi di masyarakat luas, maka diperlukan penelitian guna menguji keektifan produk tersebut (digunakan metode eksperimen).

Penelitian pengembangan atau *research and development* (R&D) adalah sebuah strategi atau metode penelitian yang cukup ampuh untuk memperbaiki praktik (Sukmadinata, 2009). Penelitian Pengembangan juga diartikan sebagai suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada yang dapat dipertanggungjawabkan (Sujadi, 2003:164).

Lebih lanjut lagi, Borg and Gall (1989) menyatakan bahwa untuk melakukan penelitian analisis kebutuhan sehingga bisa dihasilkan suatu produk yang bersifat hipotetik, tidak jarang memakai metode penelitian dasar (basic research). Berikutnya guna menguji produk yang masih bersifat hipotetik itu, menggunakan eksperimen atau action research. Sesudah produk teruji, maka bisa diaplikasikan secara luas. Proses pengujian produk dengan menggunakan penelitian eksperimen ini disebut dengan penelitian terapan (*applied research*). Penelitian R & D itu sendiri, bertujuan untuk menemukan, mengembangkan, dan memvalidasi suatu produk, dengan begitu penelitian R & D bersifat longitudinal.

Menurut Sugiyono (2009), produk - produk pendidikan yang dihasilkan bisa berupa kurikulum yang spesifik untuk keperluan pendidikan tertentu, media pendidikan, metode mengajar, buku ajar, modul, model uji kompetensi, sistem evaluasi, kompetensi tenaga kependidikan, penataan ruang kelas untuk penerapan model pembelajar tertentu, model manajemen, model unit produksi, sistem pembinaan sistem penggajian, pegawai, dan lain-lain. Sejalan dengan hal ini, Sukmadinata (2008) menyatakan bahwa penelitian dan pengembangan (R&D) adalah suatu pendekatan penelitian untuk menghasilkan suatu produk baru atau menyempurnakan produk yang sudah ada. Produk yang dihasilkan dapat berbentuk hardware maupun software. Produk software misalnya seperti program untuk pengolahan data, perpustakaan atau laboratorium, pembelajaran di kelas, ataupun model-model pendidikan, pembelajaran pelatihan, evaluasi, bimbingan, manajemen, dan lain sebagainya. Sedangkan untuk produk hardware misalnya seperti modul, buku, paket, alat bantu pembelajaran yang ada di kelas dan laboratorium, atau program pembelajaran. Penelitian dan pengembangan ini, tidak sama dengan penelitian biasa yang hanya menghasilkan saran - saran bagi perbaikan, penelitian dan pengembangan ini menghasilkan suatu produk yang bisa langsung digunakan.

2. Karakteristik atau Ciri Utama R&D

Terkait karakteristik dengan penelitian R & D, Borg and Gall (1989) menjelaskan mengenai 4 iri utama di dalam penelitian R & D, yaitu

1. *Studying research findings pertinent to the product to be develop*
   Artinya, : melakukan studi atau penelitian awal (pendahuluan) guna

mencari temuan - temuan penelitian yang berhubungan dengan produk yang hendak dikembangkan.

2. *Developing the product base on this findings* Artinya : , mengembangkan produk berdasarkan pada hasil temuan penelitian awal (pendahuluan) itu.
3. *Field testing it in the setting where it will be used eventually* , Artinya, dilakukan pengujian lapangan dalam seting atau situasi senyata mungkin di mana produk tersebut nantinya akan dipakai.
4. *Revising it to correct the deficiencies found in the field-testing stage*. Dapat diartikan bahwa melakukan revisi guna memperbaiki kelemahan - kelemahan yang ditemukan pada tahap - tahap pengujian lapangan.

Dari 4 ciri utama penelitian R & D tersebut, dapat memberikan suatu gambaran bahwa ciri utama R & D yaitu adanya langkah - langkah penelitian awal (pendahuluan) tekait dengan produk yang hendak dikembangkan. Berdasarkan hasil penelitian awal tersebut kemudian baru dibuat produk yang cocok.

Kajian penelitian R & D mencakup organisasi dan manajemen, pembiayaan, mutu/pembelajaran, dan guru. Masalah - masalah yang berhubungan dengan bidang organisasi dan manajemen yakni terkait metodologi pengubahan perilaku sistemik organisasi pendidikan, meliputi demokratisasi kultural, debirokratisasi struktural, dan profesionalisme interaksional. Sedangkan, untuk masalah - masalah yang berkaitan dengan bidang pembiayaan pendidikan yang meliputi model-model pembiayaan sekolah atau satuan penddikan, model pendanaan masyarakat, model penggajian guru, dan model beasiswa.

Sementara masalah-masalah yang berhubungan dengan bidang mutu pembelajaran antara lain meliputi efektivitas kinerja program (akselerasi, pengembangan kultur, unggulan, ujian akhir), implementasi kurikulum, dan model-model pembelajaran aktif dan berbasis budaya. Sedangkan, untuk masalah-masalah di bidang guru antara lain meliputi model LPMP yang efektif sebagai pusat pengembangan karier guru, advokasi profesi, model diklat yang efektif, dan model peningkatan insentive atau kesejahteraan guru.

3. Langkah-Langkah R & D

Terdapat beberapa model penelitian R & D yang ada di dalam bidang pendidikan, antara lain yaitu model Borg and Gall dan model Sugiyono. Secara ringkas kedua model tersebut dapat diuraikan sebagaimana berikut ini.

Model Borg and Gall

Borg dan Gall (1989) menyatakan bahwa penelitian R & D dalam dunia pendidikan meliputi 10 langkah, yakni:

(1) Research and Information colletion;

(2) Planning;

(3) Develop Preliminary form of Product;

(4) Preliminary Field Testing;

(5) Main Product Revision;

(6) Main Field Testing;

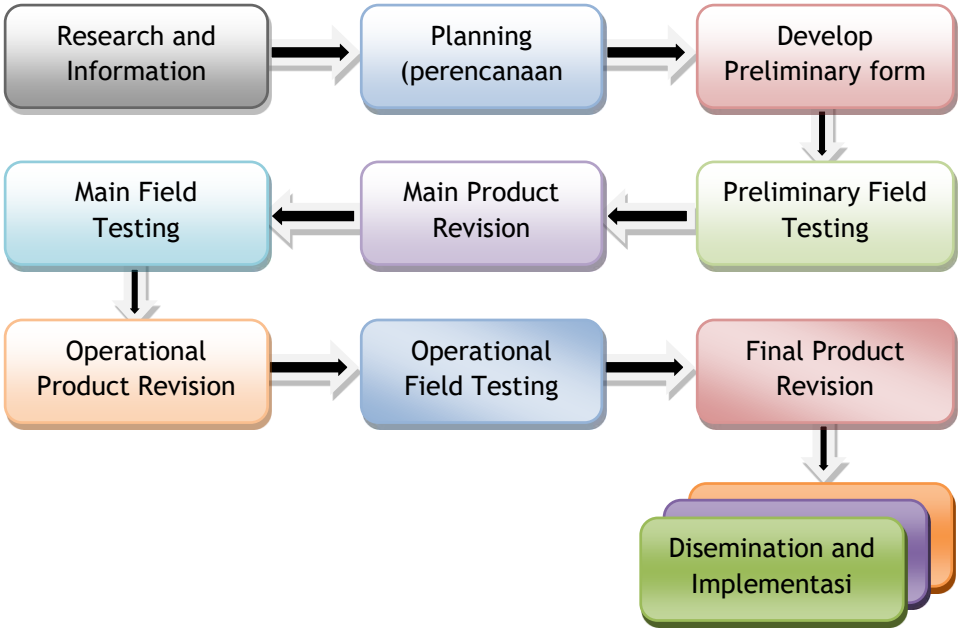
(7) Operational Product Revision;

(8) Operational Field Testing;

(9) Final Product Revision; dan

(10) Disemination and Implementasi.

Skema dari langkah - langkah tersebut dapat dilihat pada gambar berikut.



Gambar 7.1 Skema langkah – langkah Borg dan Gall

Langkah-langkah Penelitian R & D menurut Borg dan Gall

Secara ringkas langkah - langkah penelitian R & D menurut Borg dan Gall dapat diuraikan sebagai berikut.

1. Research and Information colletion (penelitian dan pengumpulan data)

Langkah pertama ini mencakup analisis kebutuhan, penelitian pustaka, penelitian literatur, penelitian skala kecil dan standar laporan yang diperlukan. Untuk menjalankan analisis kebutuhan terdapat beberapa kriteria yang berhubungan dengan urgensi pengembangan produk dan pengembangan produk itu sendiri, juga ketersediaan SDM yang kompeten dan kecukupan waktu guna mengembangkannya. Adapun studi literatur dijalankan untuk pengenalan sementara terhadap produk yang hendak dikembangkan, dan hal

ini dilaksanakan untuk mengumpulkan temuan riset dan informasi lain yang berkaitan dengan pengembangan produk yang telah direncanakan. Sedangkan riset skala kecil perlu dijalankan supaya peneliti mengetahui beberapa hal tentang produk yang hendak dikembangkan.

2. Planning (perencanaan)

Merupakan proses penyusunan rencana penelitian, yang meliputi kemampuan - kemampuan yang dibutuhkan dalam melaksanakan penelitian, rumusan tujuan yang akan dicapai melalui penelitian tersebut, desain atau langkah - langkah penelitian, serta kemungkinan pengujian dalam ruang lingkup yang terbatas.

3. Develop Preliminary form of Product (pengembangan draft produk awal)

Merupakan langkah yang tidak kalah penting, pada langkah ini meliputi kegiatan penentuan desain produk yang hendak dikembangkan (desain hipotetik), penentuan sarana dan prasarana penelitian yang diperlukan selama kegiatan atau proses penelitian dan pengembangan, penentuan tahap - tahap pelaksanaan pengujian desain di lapangan, dan penentuan deskripsi tugas dari pihak - pihak yang ikut terlibat di dalam penelitian ini. Termasuk juga di dalamnya antara lain, pengembangan bahan pembelajaran, proses pembelajaran serta instrumen evaluasi.

4. Preliminary Field Testing (uji coba lapangan awal)

Langkah ke empat ini merupakan langkah pengujian produk yang telah dihasilkan secara terbatas, yakni melakukan uji lapangan awal terhadap desain produk yang sifatnya terbatas, baik itu substansi desainnya maupun pihak - pihak yang ikut terlibat. Uji lapangan awal dilaksanakan secara

berulang - ulang sehingga dapat memperoleh desain yang layak, baik itu substansi ataupun metodologinya. Misalnya uji ini dilaksnakan di satu sampai dengan tiga sekolah, menggunakan enam hingga 12 subjek uji coba (guru). Selama uji coba harus diadakan pengamatan atau observasi, wawancara dan juga pengedaran angket menganai kelayakan desain produk tadi. Hasil dari pengumpulan data melalui kuesioner dan observasi selanjutnya dilakukan analisis.

5. Main Product Revision (revisi hasil uji coba)

Setelah mendapatkan hasil dari uji coba lapangan awal, maka langkah selanjutnya adalah revisi hasil uji coba. Langkah revisi hasil uji coba merupakan langkah perbaikan model atau desain berdasarakan pada hasil uji lapangan terbatas. Penyempurnaan produk awal akan dilaksanakan sesudah dilaksanakan uji coba lapangan secara terbatas. Pada tahap penyempurnaan produk awal tersebut, lebih banyak dilaksanakan dengan pendekatan kualitatif. Evaluasi yang dilaksanakan lebih pada evaluasi terhadap proses, sehingga perbaikan yang dilaksanakan lebih pada hal yang bersifat perbaikan internal.

6. Main Field Testing (uji lapangan produk utama)

Langkah uji lapangan produk utama ini, merupakan uji produk yang dilakukan secara lebih fokus terhadap hal yang meliputi uji efektivitas desain produk, uji efektivitas desain (umumnya langkah memakai teknik eksperimen model penggulangan). Hasil dari pengujian pada tahap ini yaitu diperolehnya desain yang efektif, baik itu dari sisi substansi maupun dari sisi metodologi. Misalnya, uji ini dilakukan di 5 sampai 15 sekolah dengan subjek sebanyak 30 sampai 100. Pengumpulan data mengenai dampak sebelum dan sesudah

implementasi produk memakai kelas khusus, yaitu data kuantitatif penampilan subjek uji coba (guru) sebelum dan sesudah menerapkan model yang diujicobakan. Hasil-hasil dari pengumpulan data ini, selanjutnya dievaluasi dan bila memungkinkan dibandingkan dengan hasil dari kelompok pembanding.

7. Operational Product Revision (revisi produk)

Langkah revisi produk ini, merupakan penyempurnaan produk atas hasil uji lapangan berdasarkan masukan dan hasil uji lapangan utama. Jadi perbaikan kali ini merupakan perbaikan ke dua sesudah dilaksanakannya uji lapangan yang lebih luas dari pada uji lapangan yang pertama. Penyempurnaan produk dari hasil uji lapangan lebih luas ini, akan membuat produk yang dikembangkan menjadi lebih mantap karena pada tahap uji coba lapangan sebelumnya telah dilaksanakan dengan adanya kelompok kontrol. Desain yang dipakai adalah desain pretest dan posttest. Disamping perbaikan yang bersifat internal, penyempurnaan produk ini juga berdasarkan pada evaluasi hasil sehingga pendekatan yang dipakai adalah pendekatan kuantitatif.

8. Operational Field Testing (uji coba lapangan skala luas/uji kelayakan)

Pada langkah ini sebaiknya dilaksanakan dengan skala yang besar, meliputi uji efektivitas dan adaptabilitas desain produk, dan uji efektivitas dan adabtabilitas desain yang melibatkan para calon pemakai produk tersebut. Hasil dari uji lapangan berupa model desain yang sudah siap diterapkan, baik dari sisi substansinya ataupun metodologinya. Misalnya uji ini dilaksanakan di 10 sampai 30 sekolah dengan subjek sebanyak 40 sampai 200. Pengujian

ini dilakukan melalui angket, wawancara, dan observasi dan yang kemudian hasilnya dianalisis.

9. Final Product Revision (revisi produk final)

Langkah revisi produk final ini, merupakan penyempurnaan produk yang sedang dikembangkan. Penyempurnaan produk akhir dipandang perlu guna lebih akuratnya produk yang sedang dikembangkan. Pada tahap revisi produk final ini telah diperoleh suatu produk yang tingkat efektivitasnya bisa dipertanggungjawabkan. Hasil penyempurnaan produk akhir mempunyai nilai "generalisasi" yang bisa diandalkan. Penyempurnaan didasarkan atas masukan atau hasil uji kelayakan dalam skala luas.

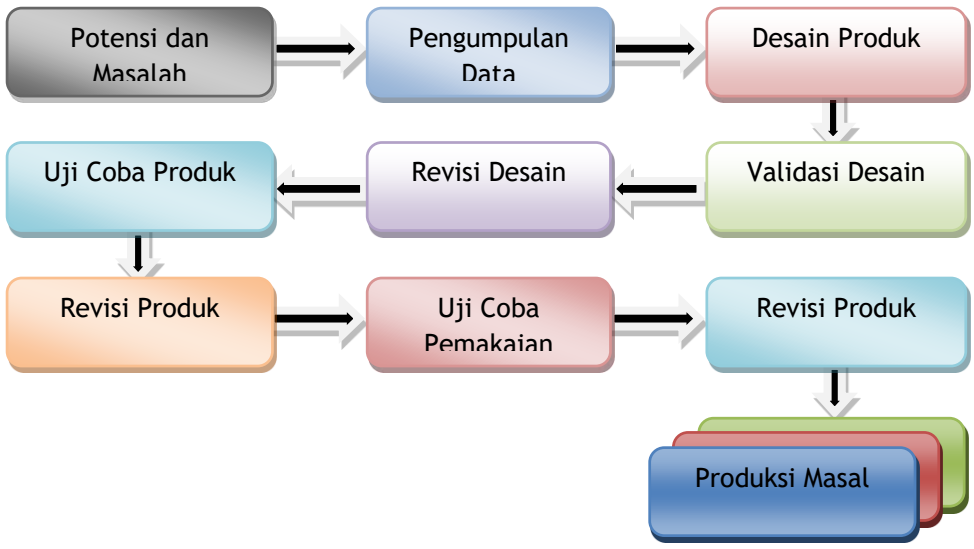
10. Disemination and Implementasi (Desiminasi dan implementasi)

Desiminasi dan implementasi, merupakan tahap pelaporan produk kepada forum - forum profesional di dalam jurnal dan implementasi produk pada praktik pendidikan. Penerbitan produk untuk didistribusikan secara komersial maupun free guna dimanfaatkan oleh publik. Distribusi produk haruslah dilaksanakan sesudah melalui quality control. Selain itu juga harus dilakukan monitoring terhadap pemanfaatan produk oleh publik guna mendapatkan masukan dalam kerangka mengendalikan kualitas produk.

Model Sugiyono

Sugiyono (2009) menyatakan bahwa langkah-langkah dalam penelitian R & D terdiri atas 10 langkah sebagaimana berikut ini: (1) Potensi dan masalah; (2) Pengumpulan data; (3) Desain produk; (4) Validasi desain; (5) Revisi desain; (6) Ujicoba produk; (7) Revisi produk; (8) Ujicoba pemakaian; (9) Revisi produk; dan (10) Produksi masal.

Secara skematik langkah - langkah tersebut dapat ditunjukkan seperti pada gambar berikut ini.



Gambar 7.2 Skema langkah – langkah Borg dan Gall

Langkah-langkah Penelitian R & D menurut Sugiyono

Secara ringkas langkah - langkah tersebut dapat dijelaskan sebagai berikut.

1. Potensi dan Masalah

Penelitian selalu bermula dari adanya potensi atau masalah. Potensi merupakan segala sesuatu yang jika didayagunakan akan mempunyai nilai tambah. Masalah juga dapat diubah menjadi sebagai potensi, apabila peneliti bisa mendayagunakan masalah tersebut. Masalah akan terjadi bila ada penyimpangan, antara yang diharapkan dengan yang keadaan terjadi. Masalah ini bisa diatasi melalui R & D yaitu dengan cara menelitinya, sehingga bisa ditemukan suatu model, sistem atau pola penanganan terpadu

yang efektif yang bisa dipakai untuk mengatasi masalah tersebut. Potensi dan masalah yang dikemukakan dalam suatu penelitian haruslah ditunjukkan dengan data yang empirik. Data tentang potensi dan masalah tidak harus dicari sendiri, akan tetapi bisa juga berdasarkan laporan penelitian orang lain maupun dari dokumentasi laporan kegiatan yang berasal dari perorangan atau instansi tertentu yang masih up to date.

2. Mengumpulkan Informasi

Sesudah potensi dan masalah bisa ditunjukkan secara faktual dan up to date, langkah berikutnya adalah mengumpulkan berbagai informasi dan studi literatur yang bisa dipakai sebagai bahan guna merencanakan membuat produk tertentu yang diharapkan bisa mengatasi masalah tersebut. Studi ini ditujukan guna menemukan konsep - konsep maupun landasan -landasan teoretis yang bisa memperkuat suatu produk, khususnya yang berhubungan dengan produk pendidikan, misal produk yang berbentuk program, model, sistem, software, pendekatan, dan sebagainya. Di lain pihak melalui studi literatur ini akan mengkaji ruang lingkup suatu produk, keluasan penggunaan, kondisi - kondisi pendukung supaya produk bisa dipakai atau diimplementasikan secara optimal, serta keterbatasan dan keunggulan nya. Studi literatur juga dibutuhkan guna mengetahui langkah -langkah yang paling tepat dalam mengembangkan produk tersebut.

3. Desain Produk

Produk yang dihasilkan dari suatu penelitian R & D ini ada banyak sekali jenisnynya. Untuk menghasilkan sistem kerja baru, maka haruslah dibuat rancangan kerja baru berdasarkan penilaian terhadap system kerja lama, sehingga bisa ditemukan kelemahan- kelemahan terhadap sistem

tersebut. Disamping itu, perlu dilakukan penelitian terhadap unit lain yang dipandang sistem kerjanya baik. Selain itu, harus dilakukan pengkajian terhadap referensi mutakhir yang berkaitan dengan sistem kerja yang modern beserta indikator sistem kerja yang bagus. Hasil akhir dari kegiatan ini biasanya berupa desain produk baru yang telah lengkap dengan spesifikasinya. Desain ini masih bersifat hipotetik, karena efektivitasnya masih belum terbukti, dan baru bisa diketahui setelah melewati pengujian - pengujian. Desain produk haruslah diwujudkan kedalam bentuk gambar atau bagan, sehingga bisa dipakai sebagai pegangan guna menilai dan membuatnya, serta akan memudahkan pihak lain untuk lebih memahaminya.

4. Validasi Desain

Validasi desain adalah suatu proses kegiatan yang bertujuan untuk menilai apakah rancangan produk, dalam hal ini sistem kerja baru secara rasional akan lebih efektif dari yang lama atau tidak. Dikatakan secara rasional, karena validasi pada tahap ini masih bersifat penilaian berdasarkan pemikiran rasional, belum berdasarkan pada fakta lapangan. Validasi produk bisa dijalankan dengan cara menghadirkan beberapa tenaga ahli atau pakar yang sudah berpengalaman memberikan penilaian terhadap produk baru yang dirancang tersebut. Setiap pakar diminta untuk memberikan nilai desain baru tersebut, sehingga langkah selanjutnya bisa diketahui kekuatan dan kelemahannya. Validasi desain bisa dijalankan pada sebuah forum diskusi. Sebelum berdiskusi, peneliti mempresentasikan proses penelitian sampai ditemukan desain tersebut, beserta dengan keunggulannya.

5. Perbaikan Desain

Sesudah desain produk jadi, divalidasi melalui diskusi bersama para pakar dan para ahli lainnya. Maka akan bisa diketahui kelemahan-kelemahannya. Kelemahan tersebut kemudian dicoba untuk dikurangi dengan jalan memperbaiki desain tersebut. Yang bertugas memperbaiki desain adalah peneliti yang akan menghasilkan produk tersebut.

6. Uji coba Produk

Desain produk yang sudah dibuat tidak dapat langsung diujicobakan terlebih dahulu. Akan tetapi haruslah dibuat terlebih dahulu, hingga menghasilkan produk, dan produk itulah yang diujicobakan. Pengujian bisa dilaksankan melalui ekperimen, yaitu membandingkan efektivitas dan efesiensi sistem kerja yang lama dengan sistem kerja yang baru.

7. Revisi Produk

Pengujian produk terhadap sampel yang terbatas tersebut dapat menunjukkan bahwa kinerja sistem kerja baru ternyata yang lebih baik bila dibandingkan dengan sistem yang lama. Perbedaan yang sangat signifikan, sehingga sistem kerja baru tersebut bisa diterapkan atau diberlakukan.

8. Ujicoba Pemakaian

Setelah pengujian terhadap produk yang dihasilkan sukses, dan mungkin ada revisi yang tidak begitu penting, maka langkah berikutnya yaitu produk yang berupa sistem kerja baru tersebut diberlakukan atau diterapkan pada kondisi nyata untuk ruang lingkup yang luas. Dalam pengoperasian sistem kerja baru tersebut, tetap harus dinilai hambatan atau kekurangan yang muncul guna dilakukan perbaikan yang lebih lanjut.

9. Revisi Produk

Revisi produk ini dilaksanakan, bila dalam perbaikan pada yang kondisi nyata terdapat kelebihan dan kekurangan. Dalam uji pemakaian produk, sebaiknya pembuat produk selaku peneliti selalu mengevaluasi bagaimana kinerja dari produknya dalam hal ini yaitu sistem kerja.

10. Pembuatan Produk Masal

Pada tahap pembuatan produk masal ini dilaksanakan bila produk yang telah diujicobakan dinyatakan efektif serta layak untuk diproduksi secara masal. Sebagai contoh pembuatan mesin yang dapat mengubah sampah menjadi bahan yang bermanfaat, hendak diproduksi masal bila berdasarkan studi kelayakan baik dari aspek ekonomi, teknologi, dan ligkungan memenuhi. Jadi untuk memproduksi suatu produk, pengusaha dan peneliti harus saling bekerja sama.

4. Kelebihan Dan Kekurangan

Kelebihan :

- ✓ Pendekatan R & D mampu menghasilkan suatu produk/model yang memiliki nilai validasi tinggi, karena produk tersebut dihasilkan melalui serangkaian uji coba di lapangan dan divalidasi oleh ahli.
- ✓ Pendekatan R & D merupakan penghubung antara penelitian yang bersifat teoritis dengan penelitian yang bersifat praktis.
- ✓ Metode penelitian yang ada dalam R & D cukup komprehensif, mulai dari metode deskriptif, evaluatif, dan eksperimen.

- ✓ Mendorong proses inovasi produk/model yang tiada henti sehingga diharapkan akan selalu ditemukan produk/model yang selalu actual dengan tuntutan kekinian.
- ✓ Mampu mengatassi kebutuhan nyata dan mendesak (real needs in the here-and-now) melalui pengembangan solusi atas suatu masalah sembari menghasilkan pengetahuan yang bisa digunakan di masa mendatang.

Kekurangan:

- ✓ Pada prinsipnya memerlukan waktu yang relative panjang, karena prosedur yang harus ditempuh relatif kompleks.
- ✓ Tidak bisa digeneralisasikan secara utuh, karena penelitian R & D ditujukan untuk pemecahan masalah "here and now", dan dibuat berdasar sampel (spesifik), bukan populasi.
- ✓ Penelitian R & D memerlukan sumber dana dan sumber daya yang cukup besar.

5. Prosedur Pengembangan

Penelitian dan pengembangan dalam konteks pendidikan disebut penelitian dan pengembangan pendidikan (educational research and development, (E, R, & D), merupakan proses yang digunakan untuk mengembangkan dan memvalidasi produk-produk pendidikan, seperti buku ajar, strategi/metode/model/program pembelajaran/ pelatihan, dan sebagainya. Tahap-tahap dari proses penelitian dan pengembangan ditunjukkan sebagai siklus penelitian dan pengembangan.

6. Laporan Penelitian & Pengembangan

Seperti yang telah dikemukakan bahwa metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk meneliti sehingga menghasilkan produk baru, dan selanjutnya menguji keefektifan produk tersebut.

Dengan demikian laporan penelitian yang dibuat harus selalu dilampiri dengan produk yang dihasilkan berikut spesifikasi dan penjelasannya. Lampiran berupa produk yang dihasilkan tersebut, dibuat dalam buku tersendiri dan diberikan penjelasan tentang kehebatan produk tersebut berdasarkan hasil uji coba, serta cara menggunakan produk tersebut. Sistematika laporan adalah seperti berikut.

7. Sistematika Laporan Penelitian R & D

Halaman judul

Abstrak

Pengantar

Daftar isi

Daftar gambar

Daftar tabel

Bab I Pendahuluan

A. Latar belakang

B. Rumusan masalah

C. Tujuan

D. Manfaat

Bab II Landasan Teori, Kerangka Berpikir, dan Pengajuan Hipotesis

A. Deskripsi teori

B. Kerangka berpikir

C. Hipotesis (produk yang akan dihasilkan)

Bab III Prosedur Penelitian

A. Langkah-langkah penelitian

B. Metode penelitian tahap i

1. Populasi sampel sumber data
2. Teknik pengumpulan data
3. Instrument penelitian
4. Analisis data
5. Perencanaan desain produk
6. Validasi desain

C. Metode penelitian tahap ii

1. Model rancangan eksperimen untuk menguji produk yg dirancang
2. Populasi dan sampel
3. Teknik pengumpulan data
4. Instrument penelitian
5. Teknik analisis data

Bab IV Hasil Penelitian dan Pembahasan

A. Desain awal produk (gambar dan penjelasan)

B. Hasil pengujian pertama

C. Revisi produk (gambar setelah revisi dan penjelasan)

D. Hasil pengujian kedua

E. Revisi produk (gambar setelah revisi dan penjelasan)

F. Pengujian tahap ketiga (bila perlu)

G. Penyempurnaan produk (gambar terakhir dan penjelasan)

H. Pembahasan produk

Bab v kesimpulan dan saran penggunaannya

A. Kesimpulan

B. Saran penggunaan

Daftar pustaka

Lampiran instrumen

Lampiran data

Lampiran produk yang dihasilkan berikut buku penjelasan

BAB-VIII

METODE SDLC, PROTOTYPE dan RAD

Overview

Pada bab ini menjelaskan mengenai metode pengembangan yang sering dipakai, ada tiga metode yang akan dibahas dalam bab ini antara lain : metode SDLC, metode prototype, dan metode RAD. Pada bab ini juga menjelaskan tahapan tahan yang harus dilewati pada metode ini secara detail.

tar

Tujuan

1. Mahasiswa mempunyai gambaran yang tentang metode pengembangan siklus hidup sistem, prototype, dan rad.
2. Mahasiswa dapat mengaplikasi ke dalam pembuatan aplikasi dengan menggunakan metode yang ada sesuai dengan langkah langkah yang benar.
3. Mahasiswa mampu menerapakan salah satu metode yang ada dalam membuat sebuah aplikasi sistem informasi

Materi

1. Siklus Hidup Sistem

Siklus hidup sistem (system life cycle – SLC) adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis

komputer. SLC sering disebut dengan pendekatan air terjun (waterfall approach) bagi pengembangan dan penggunaan sistem. Dilakukan dengan strategi Top-Down Design.

Tahapan dari siklus hidup sistem yaitu : a. Tahap Perencanaan , b. Tahap Analisis, c. Tahap Rancangan , d. Tahap Penerapan, e. Tahap Penggunaan

Kelima tahap tersebut secara diagram nampak seperti Gambar diatas

Empat tahap pertama dinamakan dengan siklus hidup pengembangan sistem (system development life cycle – SDLC).

Siklus hidup sistem yang pertama dikelola oleh manajer unit jasa informasi, dibantu oleh manajer dari analisis sistem, pemrograman dan operasi. Namun kecenderungan saat ini, meletakkan tanggung jawab pada tingkat yang lebih tinggi dan lebih rendah. Ada tiga tingkatan besar (hirarki) dari manajemen siklus hidup sistem, yaitu :

- **Tanggung Jawab Eksekutif**

  Ketika sistem memiliki nilai strategis atau mempengaruhi seluruh organisasi, direktur utama atau komite eksekutif mungkin

memutuskan untuk mengawasi proyek pengembangannya. Ketika lingkup sistem menyempit dan fokusnya lebih operasional kemungkinan besar kepemimpinan akan dipegang oleh eksekutif tingkat yang lebih rendah, seperti wakil direktur utama, direktur bagian administrasi, dan CIO.

- **Komite Pengarah SIM (steering committee MIS – SC MIS)**

  Banyak perusahaan membuat suatu komite khusus, di bawah tingkat komite eksekutif, yang bertanggung jawab atas pengawasan seluruh proyek sistem. Jika tujuan komiter tersebut adalah memberikan petunjuk, pengarahan dan pengendalian yang berkesinambungan, dalam rangka penggunaan sumber daya komputer perusahaan maka komite tersebut dinamakan Komite Pengarah SIM.

  Komite Pengarah SIM melaksanakan tiga fungsi utama, yaitu :

  - ❖ Menetapkan kebijakan
  - ❖ Menjadi pengendali keuangan
  - ❖ Menyelasaikan pertentangan

  Keuntungan yang dicapai :

  - ❖ Semakin besar kemungkinan komputer akan digunakan untuk mendukung pemakai di seluruh perusahaan.
  - ❖ Semakin besar kemungkinan proyek-proyek komputer akan mempunyai perencanaan dan pengendalian yang baik.

- **Kepemimpinan Proyek**

  Komite pengarah SIM yang terlibat langsung dengan rincian pekerjaan, tanggung jawabnya ada pada Tim Proyek. Tim proyek

mencakup semua orang yang ikut serta dalam pengembangan sistem berbasis komputer. Kegiatan tim tersebut diarahkan oleh seorang Pemimpin Proyek yang memberikan pengarahan selama proyek berlangsung. Tidak seperti komite pengarah SIM, tim proyek tidak berkelanjutan dan biasanya dibubarkan ketika penerapan sistem telah selesai.

✓ **Tahap Perencanaan**

Keuntungan dari merencanakan proyek CBIS, yaitu :

❖ Menentukan lingkup dari proyek

Unit organisasi, kegiatan atau sistem manakah yang terlibat dan mana yang tidak ? Hal tersebut akan memberikan perkiraan awal dari skala sumber daya yang diperlukan.

❖ Mengenali berbagai area permasalahan potensial

Akan menunjukkan hal-hal yang mungkin tidak berjalan dengan semestinya, sehingga hal tersebut dapat dicegah.

❖ Mengatur urutan tugas

Banyak tugas-tugas terpisah yang diperlukan untuk mencapai sistem. Tugas tersebut diatur dalam urutan logis berdasarkan prioritas informasi dan kebutuhan agar efisien.

❖ Memberikan dasar untuk pengendalian

Tingkat kinerja metode pengukuran tertentuharus dispesifikasikan sejak awal.

Langkah-langkahnya

- Menyadari masalah

  Kebutuhan akan proyek CBIS (Computer Based Information System) biasanya dirasakan oleh manajer perusahaan, non manajer, dan elemen-elemen dalam lingkungan perusahaan.

- Mendefinisikan masalah

  Setelah manajer menyadari adanya masalah, ia harus memahaminya dengan baik agar dapat mengatasi permasalah tersebut. Ia melakukan identifikasi dimana letak permasalahannya, penyebabnya dan berusahan mengumpulkan semua informasi. Jika perusahaan mempunyai kebijakan untuk mendukung end user computing, dan manajer ingin memakai pendekatan tersebut untuk pengembangan sistem, maka ia bertanggung jawab untuk membuat definisi. Selain itu, manajer memerlukan bantuan analis sistem yang saling bekerja sama dengan manajer.

- Menentukan tujuan sistem

  Manajer dan analis sistem mengembangkan suatu daftar tujuan sistem yang harus dipenuhi oleh sistem untuk memuaskan pemakai. Sehingga tujuan hanya dinyatakan secara umum, yang nantinya akan dibuat lebih spesifik.

- Mengidentifikasi kendala sistem

  Sistem baru dalam pengoperasiannya tidak bebas dari kendala. Beberapa kendala mungkin ditimbulkan oleh lingkungan, seperti laporan pajak yang diminta oleh pemerintah dan informasi pembayaran yang dibutuhkan oleh konsumen. Kendala lainnya,

seprti keharusan menggunakan perangkat keras yang telah ada atau menyiapkan dan menjalankan sistem pada tanggal tertentu. Kendala-kendala tersebut penting untuk diidentifikasi sebelum sistem benar-benar mulai dikerjakan. Dengan demikian, baik rancangan sistem maupun kegiatan proyek akan berada di antara kendala-kendala tersebut.

- Membuat studi kelayakan

  Studi kelayakan adalah suatu tinjauan seklias pada faktor-faktor utama yang akan mempengaruhi kemampuan sistem untuk mencapai tujuan yang diinginkan. Ada enam dimensi kelayakan, yaitu :

  - Teknis; tersediakan hardware dan software untuk melaksanakan pemrosesan yang diperlukan ?
  - Pengembalian ekonomis; dapatkah sistem yang diajukan dinilai secara keuangan dengan membandingkan kegunaan dan biayanya ?
  - Pengembalian non ekonomis; dapatkah sistem yang diajukan dinilai berdasarkan keuntungan-keuntungan yang tidak dapat diukur dengan uang?
  - Hukum dan etika; akankah sistem yang diajukan beroperasi dalam batasan hukum dan etika ?
  - Operasional; apakah rancangan sistem akan didukung oleh orang-orang yang akan menggunakannya ?
  - Jadual; mungkinkah penerapan sistem dalam kendala waktu yang ditetapkan ? Analis sistem mengumpulkan informasi yang diperlukan untuk menyawab pertanyaan-

pertanyaan tersebut dengan mewawancarai beberapa pegawai penting dalam area pemakai.

- Menyiapkan usulan penelitian sistem

  Jika suatu sistem dan proyek tampak layak, diperlukan penelitian sistem secara menyeluruh. Penelitian sistem (sistem study) akan memberikan dasar yang terinci bagi rancangan sistem baru mengenai apa yang harus dilakukan sistem itu dan bagaimana sistem tersebut melakukannya. Analis akan menyiapkan usulan penelitian sistem yang memberi dasar bagi manajer untuk menentukan perlu tidaknya pengeluaran untuk analisi. Hal penting yang harus diingat tentang usulan tersebut adalah bahwa sebagian besar isinya didasarkan pada perkiraan (perkiraan merupakan informasi terbaik yang tersedia) dan perkiraan jauh lebih baik daripada tanpa informasi sama sekali. Selebihnya akan dipelajari ketika siklus hidup mulai berjalan.

- Menyetujui atau menolak proyek penelitian

  Manajer dan komite pengarah menimbang pro dan kontra proyek dan rancangan sistem yang diusulkan, serta menentukan apakah perlu diteruskan keputusan teruskan / hentikan. Pertimbangan penting yang perlu dilakukan yaitu :

  - Akankah sistem yang diusulkan dapat mencapai tujuannya ?
  - Apakah penelitian proyek yang diusulkan merupakan cara terbaik untuk melakukan analisis sistem ?
  - Jika keputusannya adalah teruskan maka proyek akan berlanjut ke tahap penelitian (analisis). Namun, jika

keputusannya hentikan maka semua pihak mengalihkan perhatiannya ke masalah-masalah lain.

❖ Menetapkan mekanisme pengendalian

Sebelum penelitian sistem dimulai, SC MIS menetapkan pengendalian proyek dengan menentukan apa yang harus dikerjakan, siapa yang melakukannya, dan kapan akan dilaksanakan. Setelah jadual ditetapkan, jadual tersebut harus didokumentasikan dalam bentuk yang memudahkan pengendalian. (misalkan gunakan Microsoft Project).Bentuk dari usulan penelitian sistem secara umum mencakup :

- Ikhtisar eksekutif
- Pendahuluan
- Tujuan dan kendala sistem
- Berbagai alternatif sistem yang mungkin
- Proyek penelitian sistem yang disarankan
- Tugas yang harus dilaksanakan
- Kebutuhan sumber daya manusia
- Jadual kerja
- Perkiraan biaya
- Dampak yang diharapkan dari sistem
  - Dampak pada struktur organisasi perusahaan
  - Dampak pada operasi perusahaan
  - Dampak pada sumber daya perusahaan
- Rencana pengembangan umum (tahap analisis, rancangan dan penerapan)

✓ **Tahap Analisis**

Ketika perencanaan selesai dan mekanisme pengendalian telah berjalan, tim proyek beralih pada analisis sistem yang telah ada. Analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem baru atau diperbarui. Adapun tahapannya yaitu :

❖ Mengumumkan Penelitian Sistem

Manajer khawatir terhadap penerapan aplikasi komputer baru yang mempengaruhi kerja para pegawainya. Sehingga perlu dikomunikasikan kepada para pegawai tentang :

- alasan perusahaan melaksanakan proyek
- bagaimana sistem baru akan menguntungkan perusahaan dan pegawai.

❖ Mengorganisasikan Tim Proyek

Tim proyek yang akan melakukan penelitian sistem dikumpulkan. Agar proyek berhasil, pemakai sangat perlu berperan aktif daripada berperan pasif. Banyak perusahaan mempunyai kebijakan menjadikan pemakai sebagai pemimpin proyek dan bukannya spesialis informasi.

❖ Mendefinisikan Kebutuhan Informasi

Analis mempelajari kebutuhan informasi pemakai dengan terlibat dalam berbagai kegiatan pengumpulan informasi (wawancara, pemgamatan, pencarian catatan, dan survei). Dari semua metode tersebut, wawancara perorangan lebih disukai, dengan alasan :

- Menyediakan komunikasi dua arah dan pengamatan terhadap bahasa tubuh.
- Dapat meningkatkan antusiasme pada proyek baik dari pihak spesialis maupun pihak pemakai.
- Dapat menjalin kepercayaan antara pemakai dan spesialis informasi.
- Memberi kesempatan bagi peserta proyek untuk mengungkapan pandangan yang berbeda bahkan bertentangan.

Dokumentasi dapat berupa flowchart, diagram aliran data (data flow diagram), dan grafik serta penjelasan naratif dari proses dan data. Istilah kamus proyek sering digunakan untuk menggambarkan semua dokumentasi yang menjelaskan suatu sistem.

❖ Mendefinisikan Kriteria Kinerja Sistem

Langkah selanjutnya adalah menspesifikasikan secara tepat apa yang harus dicapai oleh sistem, yaitu kriteria kinerja sistem. Misalkan,

- Laporan harus disiapkan dalam bentuk salinan kertas dan tampilan komputer;
- Laporan harus tersedia tidak lebih dari 3 hari setelah akhir bulan;
- Laporan harus membandingkan pendapatan dan biaya actual dengan anggarannya baik untuk bulan lalu maupun sepanjang tahun hingga sekarang (year to date).

❖ Menyiapkan Usulan Rancangan

Analis sistem memberikan kesempatan bagi manajer untuk membuat keputusan teruskan atau hentikan untuk kedua kalinya. Dalam hal ini manajer harus menyetujui tahap rancangan dan kungan bagi keputusan tersebut termasuk di dalam usulan rancangan.

❖ Menerima atau Menolak Proyek Rancangan

Manajer dan komite pengarah SIM mengevaluasi usulan rancangan dan menentukan apakah akan memberikan persetujuan atau tidak. Dalam beberapa kasus, tim mungkin diminta melakukan analisis lain dan menyerahkannya kembali atau mungkin proyek ditinggalkan. Jika disetujui, proyek maju ke tahap rancangan.

Secara diagram tahapan analisis tampak pada Gambar 3. sedangkan contoh format untuk dokumen usulan rancangan, yaitu sebagai berikut :

- Ikhtisar eksekutif
- Pendahuluan
- Definisi masalah
- Tujuan dan kendala sistem
- Kriteria kinerja
- Berbagai alternatif sistem yang mungkin
- Rancangan proyek yang disarankan
  - Tugas-tugas yang harus dilaksanakan
  - Kebutuhan sumber daya manusia
  - Jadual kerja
  - Perkiraan biaya

- Dampak yang diharapkan dari sistem
  - Dampak pada struktur organisasi perusahaan
  - Dampak pada operasi perusahaan
  - Dampak pada sumber daya perusahaan
- Rencana pengembangan umum (tahap analisis, rancangan dan penerapan)

✓ **Tahap Perancangan**

Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Jika sistem itu berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatann yang akan digunakan. Langkah-langkah tahapan rancangan yaitu :

❖ Menyiapkan rancangan sistem yang terinci

Analis bekerja sama dengan pemakai dan mendokumentasikan rancangan sistem baru dengan alat-alat yang dijelaskan dalam modul teknis. Beberapa alat memudahkan analis untuk menyiapkan dokumentasi secara top-down, dimulai dengan gambaran besar dan secara bertahap mengarah lebih rinci. Pendekatan top-down ini merupakan ciri rancangan terstruktur (structured design), yaitu rancangan bergerak dari tingkat sistem ke tingkat subsistem. Alat-alat dokumentasi yang popular yaitu diagram arus data (data flow diagram), diagram hubungan entitas (entity relationship duagram), kamus data (data dictionary), flowchart, model hubungan objek, dan spesifikasi kelas.

❖ Mengidentifikasi berbagai alternatif konfigurasi sistem

Analis mengidentifikasi konfigurasi – bukan merek atau model – peralatan komputer yang akan memberikan hasil yang terbaik bagi sistem dalam menyelesaikan pemrosesan. Identifikasi merupakan suatu proses berurutan, dimulai dengan berbagai kombinasi yang dapat menyelesaikan setiap tugas.

- Mengevaluasi berbagai alternatif konfigurasi sistem

  Analis bekerja sama dengan manajer mengevaluasi berbagai alternatif. Alternatif yang dipilih adalah yang paling memungkinkan subsistem memenuhi kriteria kinerja, dengan kendala-kendala yang ada.

- Memilih konfigurasi terbaik

  Analis mengevaluasi semua konfigurasi subsistem dan menyesuaikan kombinasi peralatan sehingga semua subsistem menjadi satu konfigurasi tunggal. Setelah selesai, analis membuat rekomendasi kepada manajer untuk disetujui. Bila manajer menyetujui konfigurasi tersebut, persetujuan selanjutnya dilakukan oleh SC MIS.

- Menyiapkan usulan penerapan

  Analis menyiapkan usulan penerapan (implementation proposal) yang mengikhtisarkan tugas-tugas penerapan yang harus dilakukan, keuntungan yang diharapkan, dan biayanya.

- Menyetujui atau menolak penerapan sistem

  Keputusan untuk terus pada tahap penerapan sangatlah penting, karena usaha ini akan sangat meningkatkan jumlah

orang yang telibat. Jika keuntungan yang diharapkan dari sistem melebihi biayanya, maka penerapan akan disetujui. dokumen usulan penerapan, yaitu sebagai berikut :

- Ikhtisar eksekutif
- Pendahuluan
- Definisi masalah
- Tujuan dan kendala sistem
- Kriteria kinerja
- Rancangan sistem
  - Deskripsi ringkasan
  - Konfigurasi peralatan
- Proyek penerapan yang disarankan
  - Tugas-tugas yang harus dilaksanakan
  - Kebutuhan sumber daya manusia
  - Jadual kerja
  - Perkiraan biaya
- Dampak yang diharapkan dari sistem
  - Dampak pada struktur organisasi perusahaan
  - Dampak pada operasi perusahaan
  - Dampak pada sumber daya perusahaan
- Rencana penerapan umum

✓ **Tahap Penerapan**

Penerapan merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang

menghasilkan suatu sistem yang bekerja. Adapun tahapannya yaitu :

- Merencanakan penerapan;

  Manajer dan spesialis informasi harus memahami dengan baik pekerjaan yang diperlukan untuk menerapkan rancangan sistem dan untuk mengembangkan rencana penerapan yang sangat rinci.

- Mengumumkan penerapan;

  Proyek penerapan diumumkan kepada para pegawai dengan cara yang sama pada penelitian sistem. Tujuannya adalah untuk menginformasikan kepada para pegawai mengenai keputusan untuk menerapkan sistem baru dan meminta kerjasama mereka.

- Mendapatkan sumber daya perangkat keras;

  Rancangan sistem disediakan bagi para pemasok berbagai jenis perangkat keras yang terdapat pada konfigurasi sistem yang disetujui. Setiap pemasok diberikan request for proposal (RFP), yang berisi antara lain :

  - Surat yang ditransmisikan
  - Tujuan dan kendala sistem
  - Rancangan sistem : deskripsi ringkasan, kriteria kerja, konfigurasi peralatan,
  - dokumentasi sistem ringkasan, perkiraan volume transaksi, perkiraan ukuran file.
  - Jadual pemasangan

Selanjutnya mereka membuat usulan tertulis, bagaimana peralatan yang diusulkan akan membuat sistem mencapai kriteria kinerjanya. Ketika semua usulan telah diterima dan dianalisis, SC MIS memilih satu pemasok atau lebih. Spesialis informasi memberi dukungan bagi keputusan tersebut dengan mempelajari usulan dan membuat rekomendasi. Setelah disetujui, perusahaan melakukan pemesanan.

- Mendapatkan sumber daya perangkat lunak;

  Ketika perusahaan memutuskan untuk menciptakan sendiri perangkat lunak aplikasinya, programmer menggunakan dokumentasi yang disiapkan oleh analis sistem sebagai titik awal. Programmer dapat menyiapkan dokumentasi yang lebih rinci seperti flowchart atau bahasa semu (psedudo code) yang terstruktur, dilakukan pengkodean, dan pengujian program. Hasil akhirnya adalah software library dari program aplikasi. Jika peangkat lunak aplikasi jadi (prewritten application software) dibeli, pemilihan pemasok perangkat lunak dapat mengikuti prosedur yang sama seperti yang digunakan untuk memilih pemasok

  perangkat keras, yaitu RFP dan Usulan.

- Menyiapkan database;

  Pengelola database (database administrator – DBA) bertanggung jawab untuks emua kegiatan ynag berhubungan dengan data, dan mencakup persiapan database. Hal tersebut memerlukan pengumpulan data baru atau data yang telah ada

perlu dibentuk kembali sehingga sesuai dengan rancangan sistem baru dan menggunakan sistem manajemen basis data (database management sistem – DBMS).

- Menyiapkan fasilitas fisik;

  Jika perangkat keras dan sistem baru tidak sesuai dengan fasilitas yang ada, perlu dilakukan konstruksi baru atau perombakan. Sehingga pembangunan fasilitas tersebut merupakan tugas berat dan harus dijadualkan sehingga sesuai dengan keseluruhan rencana proyek.

- Mendidik peserta dan pemakai;

  Sistem baru kemungkinan besar akan mempengaruhi banyak orang. Beberapa orang akan membuat sistem bekerja. Mereka disebut dengan peserta, yang meliputi operator entry data, pegawai coding, dan pegawai administrasi lainnya. Semuanya harus dididik tentang peran mereka dalam sistem. Pendidikan harus dijadualkan jauh setelah siklus hidup dimulai, tepat sebelum bahan-bahan yang dipelajari mulai diterapkan.

- Menyiapkan usulan cutover;

  Proses menghentikan penggunaan sistem lama dan memulai menggunakan sistem baru disebut cutover. Ketika seluruh pekerjaan pengembangan hampir selesai , tim proyek merekomendasikan kepada manajer agar dilaksanakan cutover (dalam memo atau laporan lisan)

- Menyetujui atau menolak masuk ke sistem baru;

Manajer dan SC MIS menelaah status proyek dan menyetujui atau menolak rekomendasi tersebut. Bila manajemen menyetujui maka manajemen menentukan tanggal cutover. Namun, bila manajemen menolak maka manajemen menentukan tindakan yang harus diambil dan tugas yang harus diselesaikan sebelum cutover akan dipertimbangkan kembali, kemudian manajemen menjadualkan tanggal baru.

- ❖ Masuk ke sistem baru.

Ada 4 pendekatan dasar (cutover), yaitu :

- Percontohan (pilot) yaitu suatu sistem percobaan yang diterapkan dalam satu subset dari keseluruhan operasi.
- Serentak (immediate) merupakan pendekatan yang paling sederhana yakni beralih dari sistem lama ke sistem baru pada saat yang ditentukan.
- Bertahap (phased), sistem baru digunakan berdasarkan bagian per bagian pada suatu waktu.
- Paralel (parallel), mengharuskan sistem lama dipertahankan sampai sistem baru telah diperiksa secara menyeluruh. Akan memberikan pengamanan yang paling baik terhadap kegagalan tetapi yang paling mahal, karena kedua sumber daya harus dipertahankan.

Cutover menandakan berakhirnya bagian pengembangan dari siklus hidup sistem.

Penggunaan sistem dapat dimulai sekarang.

- ✓ **Tahap Penggunaan**

Tahap penggunaan terdiri dari 5 langkah, yaitu :

- Menggunakan sistem

  Pemakai menggunakan sistem untuk mencapai tujuan yang diidentifikasikan pada tahap perencanaan.

- Audit sistem

  Setelah sistem baru mapan, penelitian formal dilakukan untuk menentukan seberapa baik sistem baru itu memenuhi kriteria kinerja. Studi tersebut dikenal dengan istilah penelaahan setelah penerapan (post implementation review). Hasil audit dilaporkan kepada CIO, SC MIS dan pemakai. Proses tersebut diulangi, mungkin setahun sekali, selama penggunaan sistem berlanjut.

- Memelihara sistem

  Selama manajer menggunakan sistem, berbagai modifikasi dibuat sehingga sistem terus memberikan dukungan yang diperlukan. Modifikasinya disebut pemeliharaan sistem (sistem maintenance). Pemeliharaan sistem dilaksakan untuk 3 alasan, yakni :

  - Memperbaiki kesalahan
  - Menjaga kemutakhiran sistem
  - Meningkatkan system

- Menyiapkan usulan rekayasa ulang

  Ketika sudah jelas bagi para pemakai dan spesialis informasi bahwa sistem tersebut tidak dapat lagi digunakan, diusulkan kepada SC MIS bahwa sistem itu perlu direkayasa ulang

(reengineered). Usulan itu dapat berbentuk memo atau laporan yang mencakup dukungan untuk beralih pada suatu siklus hidup sistem baru. Dukungan tersebut mencakup penjelasan tentang kelemahan inheren sistem, statistik mengenai biaya perawatan, dan lain-lain.

- Menyetujui atau menolak rekayasa ulang sistem

  Manajer dan komite pengarah SIM mengevaluasi usulan rekayasa ulang sistem dan menentukan apakah akan memberikan persetujuan atau tidak.

2. Prototype

Pada tahun 1960-an: Teknik-teknik prototyping pertama cepat menjadi diakses pada tahun delapan puluhan kemudian dan mereka digunakan untuk produksi komponen prototipe dan model. Sejarah prototipe cepat dapat ditelusuri sampai akhir tahun enam puluhan, ketika seorang profesor teknik, Herbert Voelcker, mempertanyakan dirinyasendiri tentang kemungkinan melakukan hal-hal menarik dengan alat komputer dikontroldan otomatis mesin. Alat-alat mesin baru saja mulai muncul di lantai pabrik itu. Voelcker berusaha mencari jalan di mana alat-alat mesin otomatis dapat diprogram denganmenggunakan output dari program desain komputer.Kemudian 1970: Voelcker mengembangkan alat dasar matematika yang dengan jelas menggambarkan tiga aspek dimensi dan menghasilkan teori-teori awal teori algoritma dan matematika untuk pemodelan solid. Teori-teori ini membentuk dasar program komputer modern yang digunakan untuk merancang hampir

segala hal mekanis,mulai dari mobil mainan terkecil ke gedung pencakar langit tertinggi.

Teori Volecker berubah metode perancangan pada tahun tujuh puluhan, namun, metode lama untuk merancang masih sangat banyak digunakan. Metode lama terlibat baik alat masinis ataumesin dikendalikan oleh komputer. Para cowok logam dipotong dan bagian yangdibutuhkan tetap sesuai kebutuhan. Namun, pada tahun 1987, Carl Deckard, bentuk peneliti dari University of Texas,datang dengan ide yang revolusioner yang baik.

Dia memelopori manufaktur yang berbasis lapisan, dimana ia memikirkan membangun lapisan model dengan lapisan. Diadicetak model 3D dengan menggunakan sinar laser untuk bedak sekering logam dalam **prototipe** solid, single layer pada suatu waktu. Deckard mengembangkan ide ini menjadisebuah teknik yang disebut "Selective Laser Sintering".

a. Definisi Prototype

Prototype merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Dengan metode prototype ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.Prototype dapat diartikan sebagai proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat.

Model tersebut dapat berupa tiga bentuk:

- ✓ Bentuk prototype di atas kertas/model berbasis komputer yang menggambarkan interaksi manusia yang mungkin terjadi.
- ✓ Working prototype, yang mengimplementasikan sebagian dari fungsi yang ditawarkan perangkat lunak.

- ✓ Program jadi yang melakukan sebagian atau seluruh fungsi yang akan dilakukan, tapi masih ada fitur yang masih dikembangkan.

Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detal output apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efesiensi algoritma, kemampuan sistem operasi dan interface yang menghubungkan manusia dan komputer. Untuk mengatasi ketidakserasian antara pelanggan dan pengembang , maka harus dibutuhakan kerjasama yanga baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalm menyelasaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.

Kunci agar model prototype ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa prototype dibangun untuk mendefinisikan kebutuhan. Prototype akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual aktual direkayasa dengan kualitas dan implementasi yang sudah ditentukan.

Prototyping merupakan Javascript Framework yang dibuat untuk lebih memudahkan proses dalam membangun aplikasi berbasis web. Metode protyping sebagai suatu paradigma baru dalam pengembangan sistem informasi, tidak hanya sekedar suatu evolusi dari metode pengembangan sistem informasi yang sudah ada, tetapi sekaligus merupakan revolusi dalam pengembangan sistem informasi manajemen.

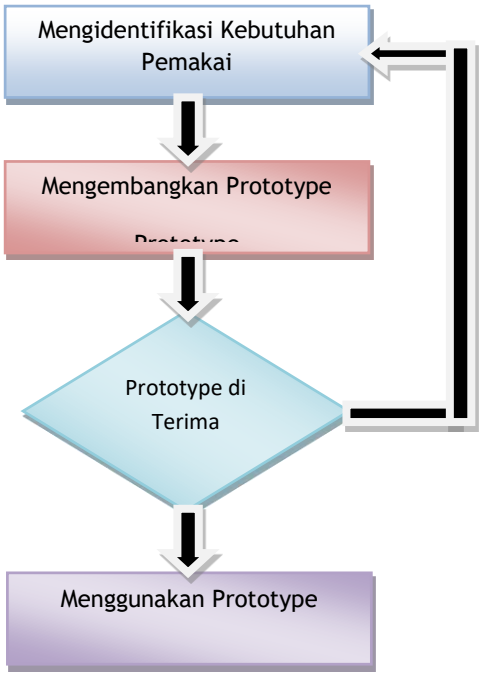
Karakteristik metode prototyping meliputi langkah-langkah :

- ✓ Pemilahan fungsi
- ✓ Penyusunan Sistem Informasi
- ✓ Evaluasi
- ✓ Penggunaan Selanjutnya

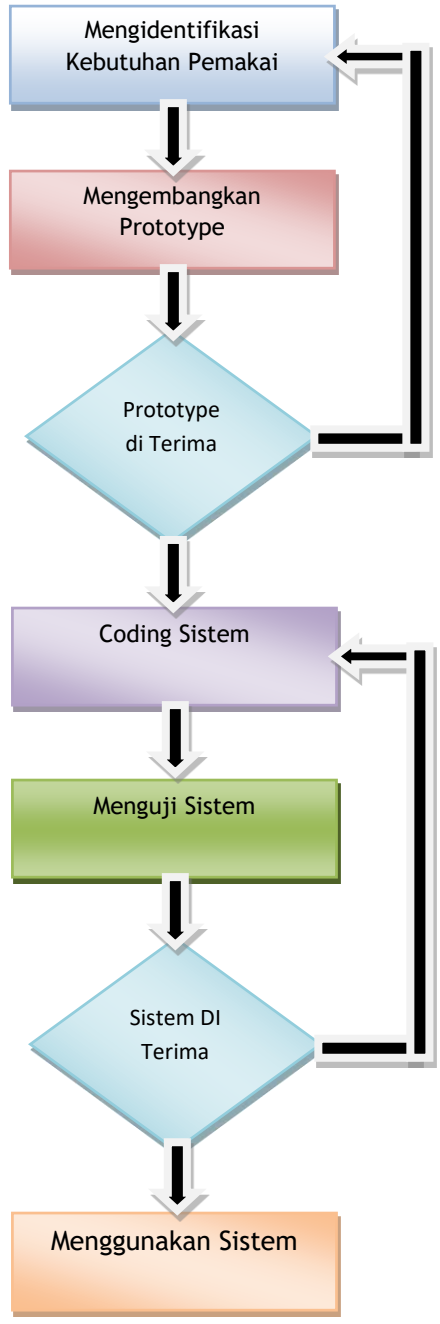
b. Jenis –jenis Prototype

Ada dua jenisprototypeantara lain:

- ✓ Jenis-I : Suatu Sistem yang akan menjadi sistem operasional



Jenis-I



Jenis-II

- ✓ Jenis-II :Suatu model yang dapatdibuang yang berfungsi sebagai cetak biru bagi system operasional.

c. Jenis-jenis prototyping meliputi:

- ✓ *Feasibility prototyping* → digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk system informasi yang akan disusun.
- ✓ *Requirement prototyping* → digunakanuntukmengetahuikebutuhanaktivitasbisnis user.
- ✓ *Desain Prototyping* → digunakanuntukmendorongperancangan system informasi yang akandigunakan.
- ✓ *Implementation prototyping* → merupakan lanjutan dari rancangan prototipe, prototipe ini langsung disusun sebagai suatu system informasi yang akan digunakan.

d. Teknik-teknik prototyping meliputi:

- ✓ Perancangan Model
- ✓ Perancangan Dialog
- ✓ Simulasi

e. Keunggulan dan Kelemahan Prototype:

Keunggulan Prototype

- ✓ Adanya komunikasi yang baik antara pengembang dan pelanggan
- ✓ Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
- ✓ Pelanggan berperan aktif dalam pengembangan system
- ✓ Lebih menghemat waktu dalam pengembangan system
- ✓ Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan Prototype

- ✓ Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitasperangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangja waktu lama.
- ✓ penegmbang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwaprogram tersebut hanya merupakan cetak biru sistem .
- ✓ Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik

Prototyping bekerja dengan baik pada penerapan-penerapan yang berciri sebagai berikut:

- ✓ Resiko tinggi Yaitu untuk maslaha-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
- ✓ Interaksi pemakai penting. Sistem harus menyediakan dialog on-line antara pelanggan dan komputer.
- ✓ Perlunya penyelesaian yang cepat.
- ✓ Perilaku pemakai yang sulit ditebak.
- ✓ Sitem yang inovatif. Sistem tersebut membutuhkan cara penyelesaian masalah dan penggunaan perangkat keras yang mutakhir.
- ✓ Perkiraan tahap penggunaan sistem yang pendek.

f. Tahapan – tahapan prototype

Tahapan-tahapandalam Prototyping adalahsebagaiberikut

- ✓ Pengumpulan kebutuhan

  Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akandibuat.

- ✓ Membangun prototyping

  Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan(misalnya dengan membuat input dan format output).

- ✓ Evaluasi protoptyping

  Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah selanjutnya akan diambil. Jika tidak prototyping direvisi dengan mengulangi langkah dari awal

- ✓ Mengkodekan system

  Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan kedalam bahasa pemrograman yang sesuai.

- ✓ Menguji system

  Setelah system sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

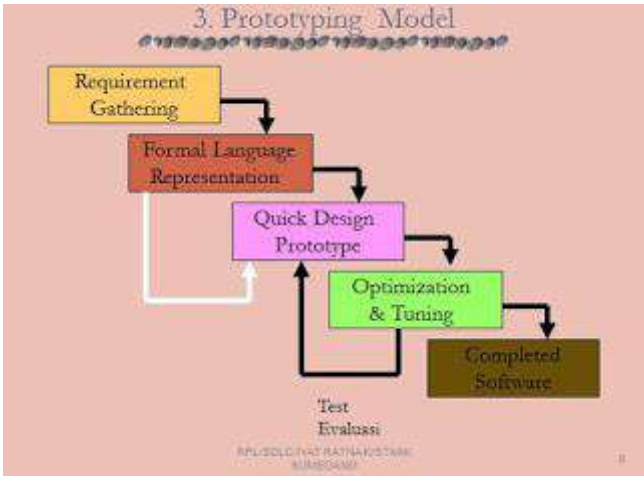
- ✓ Evaluasi Sistem.

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan , langkah selanjutnya dilakukan; jika tidak, ulangi 2 langkah sebelumnya.

✓ Menggunakan system

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

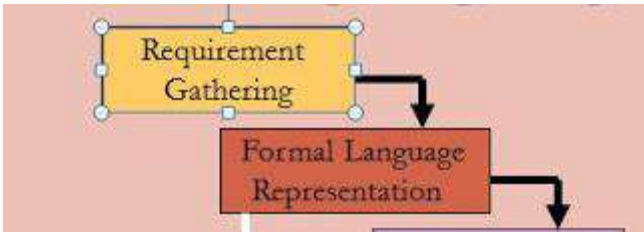
g. Tahapan Metode Prototyping



Gambar 1.0

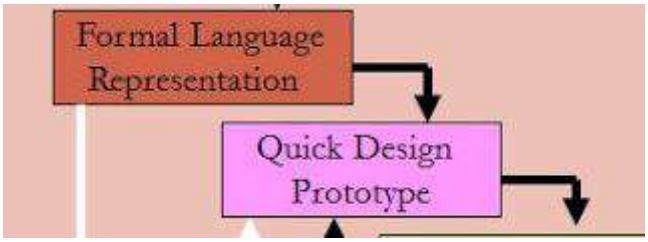
Contoh tahapan metode prototyping

✓ *Requirement Gathering*



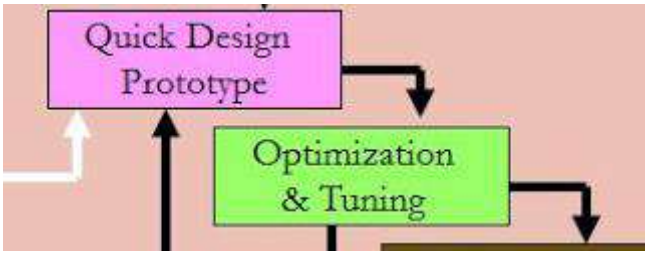
Pada tahapan prototyping atau phase ini adalah menjelaskan bahwasan-Nya user dan analis melakukan pertemuan lalu kemudian melakukan conversation antara kedua-Nya, user mendeskripsikan mengenai spesifikasi kebutuhan dari aktivitas yang user lakukan seperti dalam pekerjaan-Nya atau aktivitas lain-Nya, kemudian analis harus berusaha memahami apa maksud dari deskripsi user yang diajukan-Nya tersebut. Dan kedua belah pihak yaitu tentunya antara user dan analis (pengembang) berusaha untuk setuju pada tahap ini.

- ✓ *Formal Language Representation*



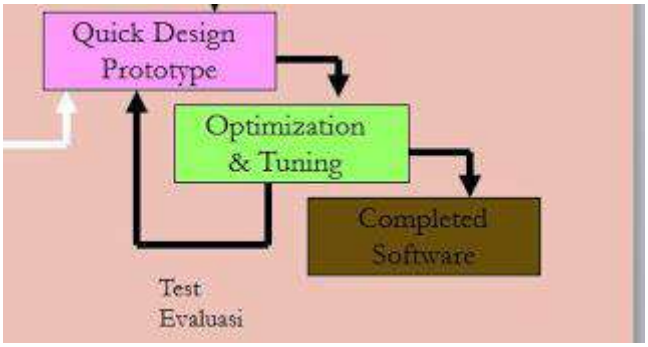
Hasil dari tahap atau phase pertama tersebut dijadikan analis sebagai dasar idea (konsep) pembuatan program atau sistem, kemudian pada tahapan ini juga analis menjadikan spesifikasi yang telah diperoleh menjadi konsep yang mudah di mengerti oleh analis dan tentu-Nya dalam tahapan ini analis sudah mengetahui apa maksud spesifikasi dari deskirpsi user tersebut.

- ✓ Quick Design Prototype



Pada tahapan atau phase ini analis akan dilakukan perencanaan dan pemodelan secara cepat berupa rancangan cepat (quick design) dan kemudian akan memulai konstruksi pembuatan prototype.

- ✓ Optimization and Tuning



Selanjutnya pada tahapan atau phase ini program dibuat berdasarkan prototype yang telah diajukan dan disetujui bersama (Team analis), kemudian pada tahap ini juga sebuag program di uji coba oleh analis dan tentu-Nya user, kemudia user menilai apakah program tersebut dapat di terima atau tidak.
Jika sebuah program tersebut user menolak maka analis harus kembali pada phase atau tahapan sebelum-Nya untuk memperbaiki kesalahan, kekurangan dan ketidak sesuaian program tersebut dengan spesifikasi user.

- ✓ Complete Software

Pada tahapan ini program telah disetujui (berhasil memenuhi spesifikasi yang diajukan) oleh user dan selanjut-Nya user dapat menggunakan program pesanan-nya dengan sukses, tentu-Nya pada tahapan ini program telah diserahkan kepada user.

h. Kekurangan dan Kelebihan Metode Prototyping

Kelebihan Metode Prototyping

- ✓ Adanya komunikasi yang baik antara pengembang dan pelanggan.
- ✓ Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- ✓ Pelanggan berperan aktif dalam pengembangan sistem.
- ✓ Lebih menghemat waktu dalam pengembangan sistem.
- ✓ Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya

Kekuarangan Metode Prototyping

- ✓ Pelanggan yang melihat working version dari model yang dimintanya tidak menyadari, bahwa mungkin saja prototype dibuat terburu-buru dan rancangan tidak tersusun dengan baik
- ✓ Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman sederhana.
- ✓ Pengembang kadang-kadang membuat implementasi sembarang, karena ingin working version bekerja dengan cepat

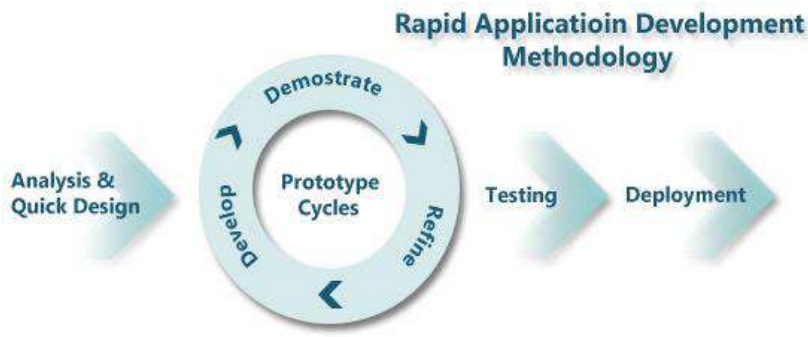
Contoh Aplikasi

Karena metode prototype mengharuskan (interaksi kebutuhan) perangkat lunak yang telah diuji dan diterima pelanggan kemudian siap untuk digunakan. Model Prototyping ini sangat sesuai diterapkan untuk kondisi yang beresiko tinggi di mana masalah-masalah tidak terstruktur dengan baik, terdapat fluktuasi kebutuhan pemakai yang berubah dari waktu ke waktu atau yang tidak terduga, bila interaksi dengan pemakai menjadi syarat mutlak dan waktu yang tersedia sangat terbatas sehingga butuh penyelesaian yang segera. Model ini juga dapat berjalan dengan maksimal pada situasi di mana sistem yang diharapkan adalah yang inovatif dan mutakhir sementara tahap penggunaan sistemnya relatif singkat. Contoh aplikasi dengan metode prototype tersebut adalah sebagai berikut :

- ✓ Aplikasi E-Banking
- ✓ Aplikasi GO-JEK
- ✓ Aplikasi Microsoft (Office, excel, access dan powerpoint)
- ✓ Aplikasi sistem database untuk pendataan pasiennya

Kenapa aplikasi diatas dapat dikategorikan sebagai aplikasi yang sudah menggunakan metode prototype, karena aplikasi diatas telah memenuhi unsur dari metode protoype tersebut yaitu seperti perangkat lunak yang telah diuji dan diterima pelanggan dan tentunya unsur interaksi dengan pemakai yang telah menjadi syarat mutlak.

3. Rapid Application Development (RAD)



Rapid application development (RAD) atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam *teknik incremental* (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid application development* menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) user dan selanjutnya disingkirkan. Working model digunakan kadang-kadang saja sebagai basis desain dan implementasi sistem final.

Penerapan

Model RAD mengadopsi model waterfall dan pembangunan dalam waktu singkat yang dicapai dengan menerapkan :

- ✓ *Component based construction* (pemrograman berbasis komponen bukan prosedural).

- ✓ Penekanan pada penggunaan ulang (*reuse*) komponen perangkat lunak yang telah ada.
- ✓ Pembangkitan kode program otomatis/semi otomatis.
- ✓ Multiple team (banyak tim), tiap tim menyelesaikan satu tugas yang selevel tapi tidak sama. Banyaknya tim tergantung dari area dan kompleksitasnya sistem yang dibangun.

Jika keutuhan yang diinginkan pada tahap analisis kebutuhan telah lengkap dan jelas, maka waktu yang dibutuhkan untuk menyelesaikan secara lengkap perangkat lunak yang dibuat adalah berkisar 60 sampai 90 hari. Model RAD hampir sama dengan model waterfall, bedanya siklus pengembangan yang ditempuh model ini sangat pendek dengan penerapan teknik yang cepat.

Sistem dibagi-bagi menjadi beberapa modul dan dikerjakan beberapa tim dalam waktu yang hampir bersamaan dalam waktu yang sudah ditentukan. Model ini melibatkan banyak tim, dan setiap tim mengerjakan tugas yang selevel, namun berbeda. Sesuai dengan pembagian modul sistem.

Alasan Memilih Metode Rad

Di dalam memilih metode RAD harus memperhatikan alasan-alasan berikut ini:

Alasan yang Buruk

- ✓ Apabila menggunakan RAD hanya untuk menghemat biaya pengembangan suatu sistem. Hal ini disebabkan karena dengan menggunakan metode RAD membutuhkan suatu tim yang mengerti betul mengenai manajemen biaya. Sebab bila tidak, maka biaya yang dikeluarkan akan menjadi lebih besar.

- ✓ Apabila menggunakan RAD hanya untuk menghemat waktu pengembangan suatu sistem. Hal ini disebabkan karena dengan menggunakan metode RAD membutuhkan suatu tim yang mengerti betul mengenai manajemen waktu. Sebab bila tidak maka waktu yang dibutuhkan akan menjadi lebih lama.

Alasan yang Baik

- ✓ Apabila menggunakan RAD untuk mendapatkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah.
- ✓ Apabila menggunakan RAD untuk memberikan batasan-batasan pada suatu system supaya tidak mengalami perubahan.
- ✓ Apabila menggunakan RAD untuk menghemat waktu, dan kalau memungkinkan bisa menghemat biaya serta menghasilkan produk yang berkualitas.

Keuntungan RAD

Beberapa keuntungan dalam menggunakan metode RAD adalah sebagai berikut:

- ✓ Membeli sistem yang baru memungkinkan untuk lebih menghemat biaya ketimbang mengembangkan sendiri.
- ✓ Proses pengiriman menjadi lebih mudah, hal ini dikarenakan proses pembuatan lebih banyak menggunakan potongan-potongan script.
- ✓ Mudah untuk diamati karena menggunakan model prototype, sehingga user lebih mengerti akan sistem yang dikembangkan.
- ✓ Lebih fleksibel karena pengembang dapat melakukan proses desain ulang pada saat yang bersamaan.

- ✓ Bisa mengurangi penulisan kode yang kompleks karena menggunakan wizard.
- ✓ Keterlibatan user semakin meningkat karena merupakan bagian dari tim secara keseluruhan.
- ✓ Mampu meminimalkan kesalahan-kesalahan dengan menggunakan alat-alat bantuan (CASE tools).
- ✓ Mempercepat waktu pengembangan sistem secara keseluruhan karena cenderung mengabaikan kualitas.
- ✓ Tampilan yang lebih standar dan nyaman dengan bantuan software-software pendukung.

Kerugian RAD

Beberapa kerugian dalam menggunakan metode RAD adalah sebagai berikut :

- ✓ Dengan melakukan pembelian belum tentu bisa menghemat biaya dibandingkan dengan mengembangkan sendiri.
- ✓ Membutuhkan biaya tersendiri untuk membeli peralatan-peralatan penunjang seperti misalnya software dan hardware.
- ✓ Kesulitan melakukan pengukuran mengenai kemajuan proses.
- ✓ Kurang efisien karena apabila melakukan pengkodean dengan menggunakan tangan bisa lebih efisien.
- ✓ Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.
- ✓ Lebih banyak terjadi kesalahan apabila hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.
- ✓ Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.

- ✓ Sistem sulit diaplikasikan di tempat yang lain.
- ✓ Fasilitas yang tidak perlu terkadang harus disertakan, karena menggunakan komponen yang sudah jadi, sehingga hal ini membuat biaya semakin meningkat.

BAB-IX

DATA FLOW DIAGRAM (DFD)

Overview

Pada bab ini menjelaskan mengenai salah satu tools perancangan sistem informasi secara teretruktur yaitu dengan menggunakan basis *Data Flow Of Diagram* (DFD), Konsep perancangan secara tersetruktur, sejarah DFD, model DFD, Komponen DFD, Dekompoisi , Contex Diagram, DFD Level, langkah langkah dalam pembuatan DFD.

Tujuan

1. Mahasiswa mampu dan dapat menjelaskan mengenai konsep perancangan secara tersetruktur.
2. Mahasiswa dapat memahami komponen komponen, contex diagram, dekomposisi dalam pendekatan tersetruktur,
3. Mahasiswa mampu mengaplikasikan perencanaan tersetruktur dalam perancangan sistem informasi.

Materi

1. Konsep Perancangan Terstruktur

Pendekatan perancangan terstruktur dimulai dari awal 1970. Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari

sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas.

Melalui pendekatan terstruktur, permasalahan yang komplek di organisasi dapat dipecahkan dan hasil dari sistem akam mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan)

2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan system sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.

Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

3. Pengertian DFD

Diagram Alir Data (DAD) atau ***Data Flow Diagram (DFD*)** adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

4. Latar belakang DFD

Suatu yang lazim bahwa ketika menggambarkan sebuah sistem kontekstual data flow diagram yang akan pertama kali muncul adalah interaksi antara sistem dan entitas luar. DFD didisain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil adan untuk menggarisbawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu "dikembangkan" untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya.

5. Tujuan dan Manfaat DFD

Tujuan DFD adalah :

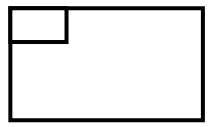
- ✓ Memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem
- ✓ Menggambarkan fungsi-fungsi(dan sub fungsi) yang mentransformasi aliran data

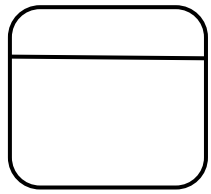
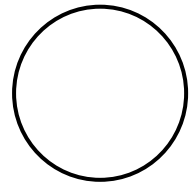
Manfaat DFD adalah :

- ✓ Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
- ✓ DFD ini adalah salah satu alat pembuatan model yang sering digunakan,khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.
- ✓ DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

6. Simbol DFD

Komponen DFD menurut Gane Sarson dan Yourdan dan Marco

Komponen	Gane and Sarson	Yourdon De Marco
ENTITY		

PROSES		
STORE		
ARROW		

Terminator/Kesatuan luar (*External Entity*)

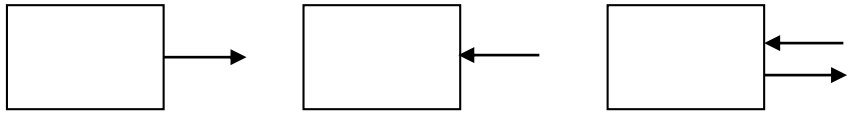
Setiap sistem pasti mempunyai batas sistem (boundary) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (external entity) merupakan kesatuan (entity) di lingkungan luar sistem yang berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan membeikan input atau menerima output dari sistem (Jogiyanto, 1989).

Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

Terdapat 2 jenis Terminator :

- ✓ Terminator Sumber, merupakan terminator yang menjadi sumber.

- ✓ Terminator Tujuan, merupakan terminator yang menjadi tujuan data / informasi sistem.



Terminator Sumber Terminator Tujuan T. Sumber & Tujuan

Terminator dapat berupa orang, sekelompok orang, organisasi, perusahaan/departemen yang berada diluar sistem yang akan dibuat, diberi nama yang berhubungan dengan sistem tsb dan biasanya menggunakan kata benda.

Contoh : Dosen, Mahasiswa.

Hal yang perlu diperhatikan tentang terminator :

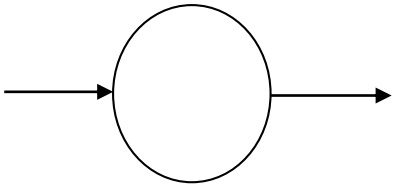
- ✓ Alur data yang menghubungkan terminator dgn sistem, menunjukkan hubungan sistem dgn dunia luar.
- ✓ Profesional sistem tidak dapat mengubah isi/cara kerja, prosedur yang berkaitan dgn Terminator.
- ✓ Hubungan yang ada antar terminator tidak digambarkan dalam DFD.

Proses (process)

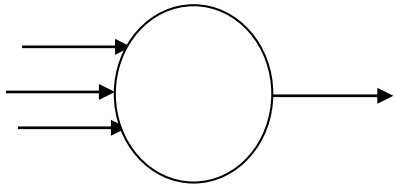
Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dan hasil suatu arus data yang masuk ke dalam proses untuk dilakukan arus data yang akan keluar dari prises. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul.

Penamaan proses disesuaikan dgn proses/kegiatan yang sedang dilakukan.

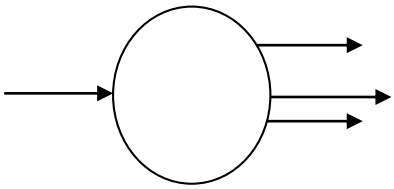
Ada 4 kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



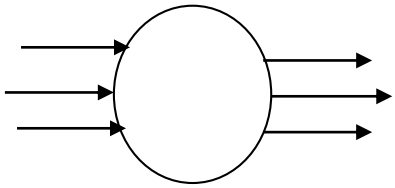
1 Input & 1 Output



M Input & 1 Output



1 Input & M Output



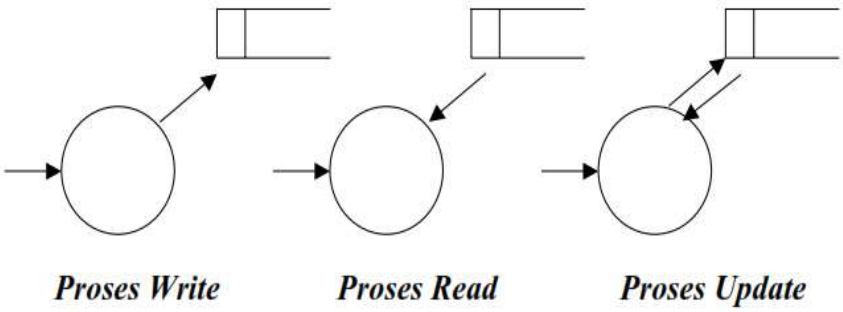
M Input & M Output

Ada beberapa hal yang perlu diperhatikan tentang proses :

- ✓ Proses harus memiliki input dan output.
- ✓ Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
- ✓ Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

Simpanan data (data store)

Simpanan data (data store) merupakan simpanan dari data yang dapat berupa file atau database di sistem komputer, arsip atau catatan manual, kotak tempat data di meja seseorang, tabel acuan manual, agenda atau buku. Simpanan data di DFD dapat disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.



Yang perlu diperhatikan tentang data store :

- ✓ Alur data dari proses menuju data store, hal ini berarti data store berfungsi sebagai tujuan/tempat penyimpanan fari suatu proses (proses write).
- ✓ Alur data dari data store ke proses, hal ini berarti data store berfungsi sebagai sumber/ proses memerlukan data (proses read).
- ✓ Alur data dari proses menuju data store dan sebaliknya berarti berfungsi sebagai sumber dan tujuan.

Arus data (data flow)

Arus data (data flow) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (Process), simpanan data (data store) dan kesatuan luar (external entity). Arus data ini menunjukkan arus data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

Ada 4 konsep tentang alur data :

- ✓ Packets of data

  Apabila ada 2 data / lebih yang mengalir dari 1 sumber yang sama menuju pada tujuan yang sama dan mempunyai hubungan digambarkan dengan 1 alur data.

- ✓ Diverging data flow

  Apabila ada sejumlah paket data yang berasal dari sumber yang sama menuju pada tujuan yang berbeda atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda.

- ✓ Converging data flow

  Apabila ada beberapa alur data yang berbeda sumber menuju ke tujuan yang sama.

- ✓ Sumber dan Tujuan

  Arus data harus dihubungkan pada proses, baik dari maupun yang menuju proses.

7. Syarat Memuat DFD

Syarat-syarat pembuatan DFD ini adalah :

- ✓ Pemberian nama untuk tiap komponen DFD
- ✓ Pemberian nomor pada komponen proses
- ✓ Penggambaran DFD sesering mungkin agar enak dilihat
- ✓ Penghindaran penggambaran DFD yang rumit
- ✓ Pemastian DFD yang dibentuk itu konsiten secara logika

Tips-tips dalam membuat DFD

Berikut ini tips-tips dalam membuat DFD :

- ✓ Pilih notasi sehingga proses yang didekomposisi atau tidak didekomposisi dapat dibaca dengan mudah
- ✓ Nama proses harus terdiri dari kata kerja dan kata benda
- ✓ Nama yang dipakai untuk proses, data store, dataflow harus konsisten (identitas perlu)
- ✓ Setiap level harus konsisten aliran datanya dengan level sebelumnya
- ✓ Usahakan agar external entity pada setiap level konsisten peletakannya
- ✓ Banyaknya proses yang disarankan pada setiap level tidak melebihi 7 proses
- ✓ Dekomposisi berdasarkan kelompok data lebih disarankan (memudahkan aliran data ke storage yang sama)
- ✓ Nama Proses yang umum hanya untuk prose yang masih akan didekomposisi
- ✓ Pada Proses yang sudah tidak didekomposisi, nama Proses dan nama Data harus sudah spesifik
- ✓ Aliran ke storage harus melalui proses, tidak boleh langsung dari external entity
- ✓ Aliran data untuk Proses Report .. : harus ada aliran keluar. Akan ada aliran masuk jika perlu parameter untuk mengaktifkan report
- ✓ Aliran data yang tidak ada datastorenya harus diteliti, apakah memang tidak mencerminkan persisten entity (perlu disimpan

dalam file/tabel), yaitu kelak hanya akan menjadi variabel dalam program.

8. Langkah membuat/menggambar DFD

Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

- ✓ Identifikasi Entitas Luar, Input dan Output

  Identifikasi terlebih dahulu semua entitas luar, input dan ouput yang terlibat di sistem.

- ✓ Buat Diagram Konteks (diagram context)

  Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya.

  Caranya :

  - Tentukan nama sistemnya.
  - Tentukan batasan sistemnya.
  - Tentukan terminator apa saja yang ada dalam sistem.
  - Tentukan apa yang diterima/diberikan external entity dari/ke sistem.
  - Gambarkan diagram konteks.

- ✓ Buat Diagram Level Zero (Overview Diagram)

  Diagram ini adalah dekomposisi dari diagram konteks.

  Caranya :

- Tentukan proses utama yang ada pada sistem.
- Tentukan apa yang diberikan/diterima masing-masing proses ke/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya).
- Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
- Hindari perpotongan arus data
- Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

✓ Buat Diagram Level Satu

Diagram ini merupakan dekomposisi dari diagram level zero. Caranya :

- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
- Tentukan apa yang diberikan/diterima masing-masing sub-proses ke/dari sistem dan perhatikan konsep keseimbangan.
- Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
- Hindari perpotongan arus data.
- Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya.C ontoh : 1.1, 1.2, 2

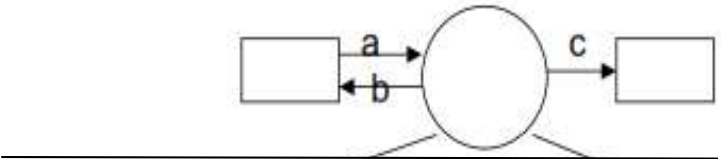
9. Kesalahan dalan pembuatan DFD

Umumnya kesalahan dalam pembuatan DFD adalah :

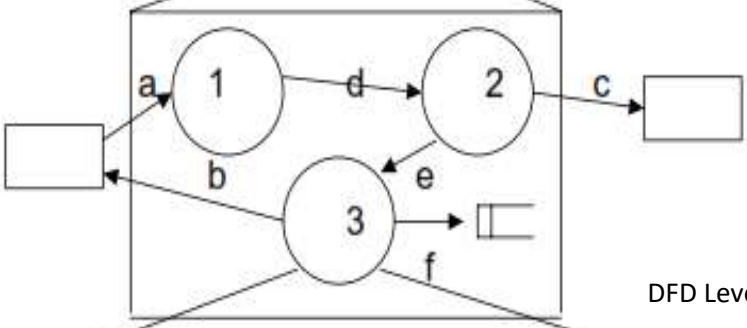
Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan *black hole* (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.

- ✓ Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan miracle (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.
- ✓ Input yang masuk tidak sesuai dengan kebutuhan proses
- ✓ Data Store tidak memiliki keluaran
- ✓ Data Store tidak memiliki masukan
- ✓ Hubungan langsung antar entitas luar
- ✓ Masukan langsung entitas data store
- ✓ Keluaran langsun dari data store ke Entitas luar
- ✓ Hubungan langsung antar data store
- ✓ Data masukan dan keluaran yang tidak bersesuain dalam data store

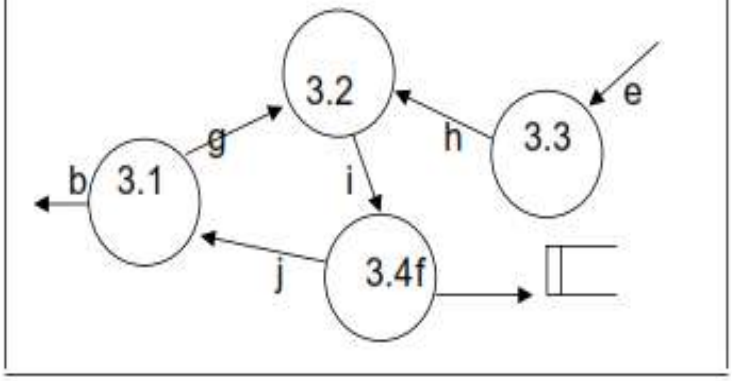
10. Bentuk Umum DFD



Contex Diagram



DFD Level 1



DFD Level 2

BAB-X

NORMALISASI

Overview

Pada bab ini membahas konsep normalisasi database berikut konsep-konsep lain yang mendasarinya. Dalam bab ini juga akan ditampilkan contoh penerapan normalisasi untuk tabel-tabel sederhana dalam kasus database.

Tujuan

1. Mahasiswa mamapu memahami definisi dan tujuan dari normalisasi serta tahapan dalam normalisasi
2. Mahasiswa mampu mengidentifikasi super key, candidate key dan primary ey dari sebuah table
3. Mahasiswa mampu mengidentifikasi functional dependency yang ada pada sebuah tabel, termasuk partial dan transitive FD
4. Mahasiswa mengenal bentuk normal pertama, ke dua, ke tiga dan BCNF serta mampu melakukan normalisasi dengan menerapkan bentuk-bentuk normal tersebut
5. Mahasiswa mengenal sekilas tentang bentuk-bentuk normal lain dan memahami konsep denormalisasi

Materi

1. Pengertian Normalisasi

*"Normalisasi (Normalization) adalah Proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang" (**Jogiyanto, 2011).***

Normalisasi adalah langkah-langkah sistematis untuk menjamin bahwa struktur database memungkinkan untuk general purpose query dan bebas dari insertion, update dan deletion anomalies yang dapat menyebabkan hilangnya integritas data (E.F. Codd, 1970)

Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Bentuk Normal adalah suatu aturan yang dikenakan pada relasi-relasi dalam basis data dan harus dipenuhi oleh relasi tersebut pada level-level normalisasi. Suatu relasi dikatakan dalam bentuk normal tertentu jika memenuhi kondisi-kondisi tertentu.

Proses normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel yang menunjukkan entitas dan relasinya untuk memenuhi kebutuhan pamakai di dalam suatu organisasi atau Proses normalisasi adalah proses untuk memperoleh properti-properti skema relasi yang bagus menjadi bentuk normal lebih tinggi sehingga syarat-syarat dibawah ini terpenuhi:

- ✓ **Mengoptimalisasi *Redudansi*** (pengulangan data yang tidak perlu). Redudansi tidak bisa dihilangkan sama sekali karena berguna untuk integritas referensial, tetapi redudansi bisa dioptimalisasi. Untuk jumlah data yang tidak terlalu banyak mungkin tidak terlalu berpengaruh dalam hal penggunaan harddisk. Tapi bayangkan jika ada ribuan, bahkan jutaan

redudansi, mungkin akan sangat berpengaruh pada penggunaan ruang.

- ✓ **Menghilangkan *Anomali*.** Anomali pada dasarnya adalah ketidak-konsistenan (inkonsistensi). Misalkan ada pergantian nama dari Bank Perkasa menjadi Bank Perkasa Utama sebanyak 4 record. Jika pergantian nama hanya dilakukan pada salah satu record saja, maka terjadi ketidak-konsistenan yaitu satu nomor bank berrelasi dengan 2 nama bank yang berbeda

Anomali adalah proses pada basis data yang mempunyai efek samping yang tidak diharapkan. Misal : ketidakkonsistenan data, suatu data hilang pada saat dihapus, dll

2. Tujuan dari Normalisasi

Pada dasarnya normalisasi dilakukan untuk memperbaiki desain tabel yang kurang baik sehingga
penyimpanan data menjadi lebih efisien dan bebas anomali data. Untuk memperjelas pemahaman tentang proses normalisasi, perhatikan diagram berikut:

Intinya, normalisasi dilakukan terhadap desain tabel yang sudah ada dengan tujuan untuk meminimalkan *redundansi* (pengulangan) data dan menjamin

integritas data dengan cara menghidari 3 Anomali Data: *Update, Insertion dan Deletion Anomaly*.

- ✓ Update Anomali : Anomali ini terjadi bila ada perubahan pada sejumlah data yang mubazir, tetapi tidak seluruhnya diubah.
- ✓ Insertion Anomali ini terjadi pada saat penambahan data ternyata ada elemen yang kosong dan elemen tsb justru menjadi key.
- ✓ Deletion Anomali : Anomali ini terjadi apabila dalam satu baris/ tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang.

3. Tahapan Normalisasi

Bentuk – bentuk Normalisi yang ada dalam mendesain basis data adalah sebagai berikut :

- ✓ Bentuk normal pertama (1NF)
- ✓ Benatuk normal kedua (2NF)
- ✓ Bentuk normal ketiga (3NF)
- ✓ Bentuk normal Boyce-Codd (BCNF)
- ✓ Bentuk normal keempat (4NF)
- ✓ Bentuk normal kelima (5NF)

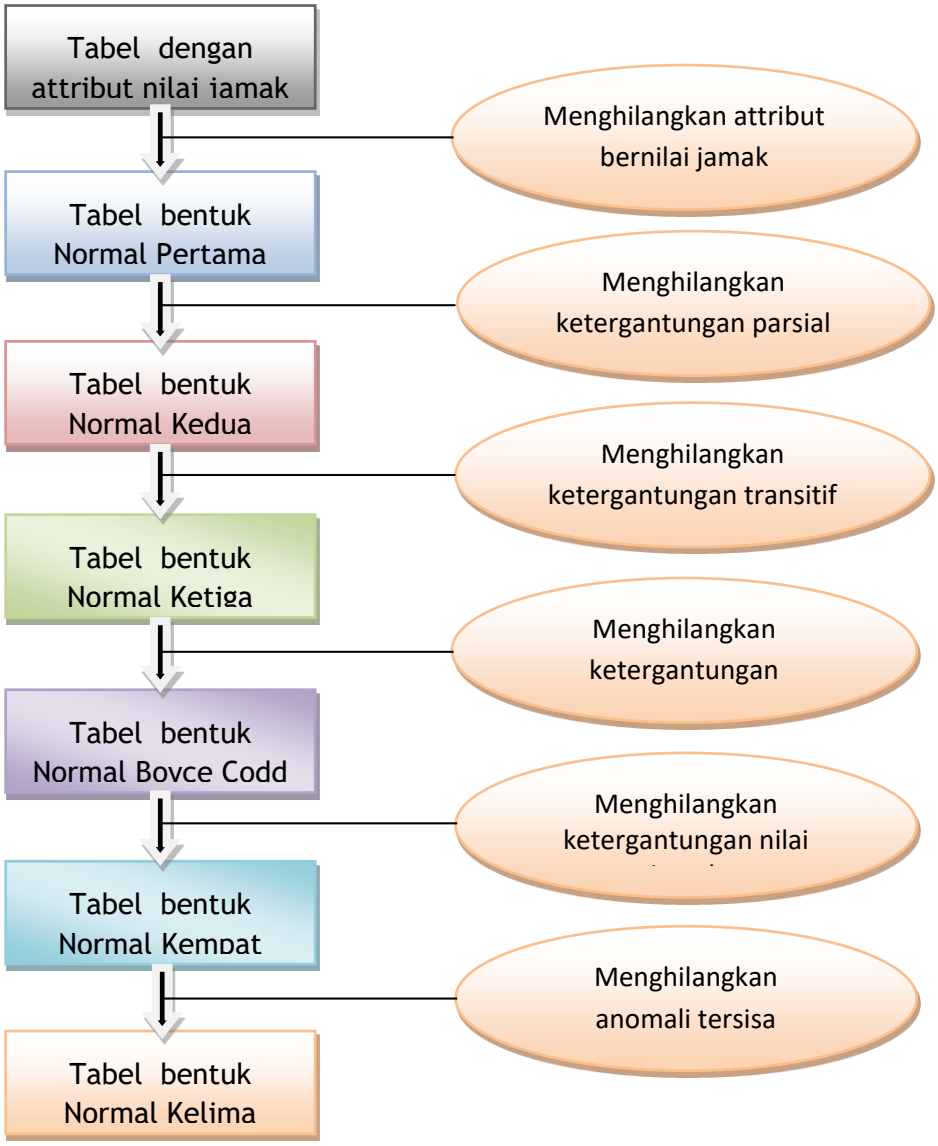
Bentuk normal pertama (1NF) s/d normal ketiga (3NF), merupakan bentuk normal yang umum dipakai. Artinya adalah pada kebanyakan relasi, bila ketiga bentuk normal tersebut telah dipenuhi, maka persoalan anomali tidak akan muncul lagi didalam kita melakukan perancangan database.

Kriteria dalam proses normalisasi adalah kebergantungan fungsional, kebergantungan banyak nilai dan kebergantungan join. Ketiga tipe kebergantungan tersebut digunakan untuk menilai relasi – relasi yang

dihasilkan dari konversi diagram ER menjadi kumpulan relasi – relasi. Proses normalisasi membentuk relasi – relasi bentuk normal menggunakan dekomposisi yang memecah relasi menjadi relasi – relasi berbentuk normal lebih tinggi.

Berikut ini adalah gambar untuk langkah – langkah melakukan normalisasi data, seperti

tampak berikut ini:



✓ **Bentuk Normal Pertama (First Normal Form / 1NF).**

Bentuk normal pertama dikenakan pada tabel yang belum ternormalisasi (masih memiliki atribut yang berulang).

Suatu relasi dikatakan bentuk normal pertama, jika dan hanya jika setiap atribut bernilai tunggal untuk setiap baris. Tiap field hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukanlah pecahan kata – kata sehingga artinya lain. Tidak ada set atribut yang berulang-ulang atau atribut bernilai ganda.

✓ **Bentuk Normal Pertama (First Normal Form / 2NF).**

Definisi Bentuk Normal Kedua (2 NF) adalah :

- Memenuhi bentuk 1 NF (normal pertama).
- Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama /

primary key.

Sehingga untuk membentuk normal kedua tiap tabel / file haruslah ditentukan kunci-kunci atributnya. Kunci atribut haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

Bentuk normal kedua terpenuhi jika pada sebuah tabel, semua atribut yang tidak termasuk dalam key primer memiliki ketergantungan fungsional (KF) pada key primer secara utuh. Sebuah tabel dikatakan memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya bergantung pada sebagian dari key primer).

✓ **Bentuk Normal Ketiga (Third Normal Form /3NF)**

Definisi Bentuk Normal Ketiga (3 NF) adalah:

- Memenuhi bentuk 2 NF (normal kedua).
- Atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci utama / primary key.

Bentuk normal tahap ketiga (3NF) merupakan kriteria alternatif, jika kriteria BCNF yang ketat tidak dapat terpenuhi. Sebuah tabel dikatakan berada dalam Bentuk Normal tahap Ketiga (3NF), jika untuk setiap KF dengan notasi X → A, dimana A mewakili semua atribut tunggal didalam tabel yang tidak ada didalam X. Maka X haruslah superkey pada tabel tersebut, atau A merupakan bagian dari key primer pada tabel tersebut.

✓ **Bentuk Normal Boyce-Codd (BCNF)**

Definisi Bentuk BCNF adalah :

- Memenuhi bentuk 3 NF (normal ketiga).
- Semua penentu (determinan) adalah kunci kandidat (atribut yang bersifat unik).

Setiap atribut harus bergantung fungsi pada atribut superkey. BCNF merupakan bentuk normal sebagai perbaikan terhadap 3 NF. Suatu relasi yang memenuhi BCNF selalu memenuhi 3 NF, tetapi tidak untuk sebaliknya. Suatu relasi yang memenuhi 3 NF belum tentu memenuhi BCNF. Karena bentuk 3 NF masih memungkinkan terjadi anomali.

✓ **Bentuk Normal Keempat (Fourth Normal Form /4NF)**

Suatu relasi dikatakan dalam bentuk normal keempat dengan ketentuan sebagai berikut ini :

- Bila dan hanya bila telah berada dalam bentuk BCNF dan tidak ada multivalued dependency nontrivial.
- Multivalued dependency (MVD) dipakai dalam bentuk normal keempat (4 NF).
- Dependensi ini dipakai untuk menyatakan hubungan satu ke bantak (one to many).

Bentuk Normal tahap Keempat (4NF) berkaitan dengan sifat ketergantungan banyak nilai (Multivalued defendency) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Sedangkan Bentuk Normal tahap Kelima (4NF) (merupakan nama lain dari project-join Normal Form atau PJNF) berkenaan dengan ketergantungan relasi antar tabel (Join Defendency).

✓ **Bentuk normal kelima (5 NF)**

Dependensi gabungan mendasari bentuk normal kelima. Suatu relasi R (X,W,Z) memenuhi dependensi gabungan jika gabungan dari proyeksi A, B, C dengan A, B, C merupakan sub himpunan dari atribut – atribut R. Dependensi gabungan sesuai dengan definisi diatas dinyatakan dengan notasi :

- * (A, B, C)
- dengan A = XY, B = YZ, C = ZX

Bentuk normal Kelima (5 NF) yang terkadang disebut PJ/NF (Projection Join / Normal Form), menggunakan acuan dependensi

gabungan. Suatu relasi berada dalam 5 NF jika dan hanya jika setiap dependensi gabungan dalam R tersirat oleh kunci kandidat relasi R.

4. Defedensi (Ketergantungan)

Depedensi merupakan konsep yang mendasari normalisasi yang menjelaskan hubungan antar atribut atau secara lebih khusus menjelaskan nilai suatu atribut yang menentukan nilai atribut lainnya.

Macam – macam depedensi : Depedensi fungsional, Depedensi fungsional sepenuhnya, Depedensi total, Depedensi transitif

- ✓ **Depedensi fungsional**

  Definisi : Suatu atribut Y mempunyai depedensi fungsional terhadap atribut X jika dan hanya jika setiap nilai X berhubungan dengan sebuah nilai Y.

  Notasi : X Y (X secara fungsional menentukan Y)

- ✓ **Depedensi fungsional sepenuhnya**

  Definisi : Suatu atribut Y mempunyai depedensi fungsional penuh terhadap X jika

  - Y mempunyai depedensi fungsional terhadap X dan /
  - Y tidak memiliki depedensi terhadap bagian dari X

- ✓ **Depedensi Total**

  Definisi : Suatu atribut Y mempunyai depedensi total terhadap atribut X jika

  - Y memiliki depedensi fungsional terhadap X dan
  - X memiliki depedensi fungsional terhadap Y

- Notasi : X Y

✓ **Depedensi Transitif**

Definisi : Atribut Z mempunyai depedensi transitif terhadap X bila :

- Y memiliki depedensi fungsional terhadap X
- Z memiliki depedensi fungsional terhadap Y

5. Denormalisasi

Denormalisasi adalah proses menggandakan data secara sengaja (sehingga menyebabkan edundansi data) untuk meningkatkan performa database, untuk meningkatkan kecepatan akses data atau memperkecil query cost.

Yang perlu diingat tentang denormalisasi adalah bahwa denormalisasi tidak sama dengan tidak melakukan normalisasi. Denormalisasi dilakukan setelah tabel dalam kondisi ‘cukup normal’ mencapai level bentuk normal yang dikehendaki).

Salah satu contoh teknik Denormalisasi adalah Materialized View pada DBMS Oracle. Materialized view adalah teknik menggandakan data dengan cara membuat tabel semu berupa view fisik (yang benar-benar dituliskan di disk, bukan sebatas di memory). Materialized view biasanya dibuat dari hasil join beberapa tabel yang sering diakses tapi jarang diupdate.

Materialized view akan menyebabkan redundansi data, namun sebagai imbalannya kecepatan akses data meningkat drastis sebab data dapat langsung diakses melalui materialized view tanpa harus menunggu query menyelesaikan operasi join dari beberapa tabel.

Ada beberapa alasan melakukan denormalisasi:

- ✓ Mempercepat proses query dengan cara meminimalkan cost yang disebabkan oleh operasi join antar tabel
- ✓ Untuk keperluan Online Analytical Process (OLAP)
- ✓ Dan lain-lain

Adapun konsekuensi denormalisasi adalah sebagai berikut:

- ✓ Perlu ruang ekstra untuk penyimpanan data
- ✓ Memperlambat pada saat proses insert, update dan delete sebab proses-proses tersebut harus dilakukan terhadap data yang redundant (ganda)

Dengan demikian dapat disimpulkan bahwa denormalisasi harus dilakukan dengan bijak sebab walaupun memiliki beberapa keuntungan namun juga memiliki konsekuensi yang patut diperhitungkan.

6. Keuntungan dan Kerugian Normalisasi

Keuntungan normalisasi

- ✓ Mengeliminasi modification anomalies
- ✓ Mengurangi duplikasi data
  - Mengeliminasi masalah data integrity
  - Space penyimpanan data lebih hemat

Kerugian

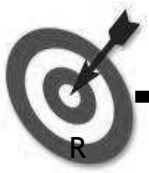
- ✓ SQL query akan lebih rumit, terutama untuk mengakses data dari banya tabel
- ✓ DBMS akan berjalan lebih lambat karena perlu kerja yang ekstra

BAB-XI

ENTITY REALATION DIAGRAM (ERD)

Overview

Pada bab ini membahas konsep entity realtiaon diagram, komponen-komponen dalam entity relasi diagram, hubungan antar entitas, relasi antar tabel yang menghasilkan entity baru dan yang pelru dilebur menjadi entity baru, sampai pada tahap transformasi entitas ke dalam tabel tabel serta contoh penerapan ERD untuk tabel-tabel sederhana dalam kasus database.



Tujuan

1. Mahasiswa mampu menjelaskan fungsi utama dari ER-diagramm derajat dari suatu relasi.
2. Mahasiswa mampu mengidentifikasi komponen komponen dalam ER-diagram, relasi dalam Er-diagram.
3. Mahasiswa mampu menerapkan ER-Diagrgam dalam praktek perancangan tersetruktur sistem informasi
4. Mahasiswa mampu mentrasformasikan entitas-entias serta hasil realsi dalm bentuk tabel-tabel.

Materi

1. Pengertian ERD

Menurut Brady dan Loonam (2010), *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analyst* dalam tahap analisis

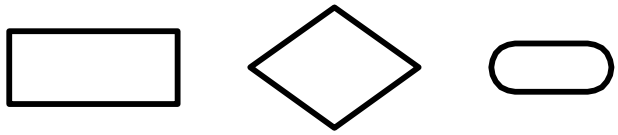
persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database.*

ERD menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika. ERD merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.ERD berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berbeda dengan DFD(*Data Flow Diagram*) yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data.

Pada pengertian sempitnya, ERD adalah sebuah konsep yang mendeskripsikan hubungan antara penyimpanan (*database*) dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek yaitu disebut sebagai *entity* dan hubungan atau relasi antar objek-objek tersebut.

2. Komponen-Komponen ERD

Dalam pembentukan ERD terdapat 3 komponen yang akan dibentuk, yaitu entitas, relasi, dan atribut.



Entitas Relasi Atribut

Gambar 14.1 Komponen-komponen ERD.

✓ Entitas

Entitas menurut Brady dan Loonam (2010), adalah objek yang menarik di bidang organisasi yang dimodelkan.Contoh : Mahasiswa, Kartu Anggota Perpustakaan (KAP), dan Buku.

✓ Relasi

Suatu relasi atau hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.Contoh : Mahasiswa mendaftar sebagai anggota perpustakaan (KAP), relasinya adalah mendaftar.

✓ Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas.Atribut memiliki struktur internal berupa tipe data. Jenis-jenis atribut :

- Atribut Key

  Atribut Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (Row/Record) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan key tidak boleh ada baris data dengan nilai

yang sama.Contoh : Nomor Pokok Mahasiswa (NPM), NIM, dan nomor pokok lainnya

- Atribut simple

  Atribut yang bernilai atomik (tidak dapat dipecah/dipilah lagi). Contoh : Alamat, penerbit, tahun terbit, judul buku.

- Atribut Multivalue

  Nilai dari suatu attribut yang mempunyai lebih dari satu nilai (multivalue) dari atrribut yang bersangkutan. Contoh : dari sebuah buku, yaitu terdapat beberapa pengarang.

- Atribut Composite

  Atribut composite adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisa dipecah lagi atau mempunyai sub attribute.Contoh : dari entitas nama yaitu nama depan, nama tengah, dan nama belakang

- Atribut Derivatif

  Atribut yang tidak harus disimpan dalam database Ex. Total. atau atribut yang dihasilkan dari atribut lain atau dari suatu relationship. Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus.

3. Derajat Relasi atau Kardinalitas Rasio

Selain itu, dalam ERD juga terdapat kardinalitas. Kardinalitas menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya.

- ✓ One to One (1:1)

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

- ✓ One to many (1:M) atau Many to one (M:1)

  Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

- ✓ Many to Many (M:M)

  Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

4. Cara Membuat ERD

- ✓ Menentukan Entity

  Disini kita dituntut untuk menentukan dengan cermat sebuah entity yang ada dalam suatu proyek atau masalah.Entity berguna untuk menentukan peran, kejadian, lokasi, hal nyata, dan konsep penggunaan untuk database.

- ✓ Menentukan Relasi

  Setelah kita berhasil membuat Entity, langkah selanjutnya adalah menentukan relasi antar entity. Relasi apa yang terdapat antara Entity A dan B, apakah entity A dan B memiliki relasi "one to one", "one to many", atau "many to many".

- ✓ Gambar ERD Sementara

  Jika sudah mengetahui Entity beserta Relasinya, sekarang kita buat dulu gambar ERD sementara.Entity digambarkan dengan persegi, relasi digambarkan dengan garis.

- ✓ Isi Kardinalitas

  Kardinalitas menentukan jumlah kejadian satu entitas untuk sebuah kejadian pada entitas yang berhubungan. Contohnya antara Entitas Buku, Distributor dan Pengarang, kardinalitas yang ada berupa:

  - Satu pengarang dapat menulis banyak buku.
  - Satu buku ditulis satu pengarang.
  - Banyak buku didistribusikan oleh satu distributor.

- ✓ Tentukan Primary Key (Kunci Utama)

  Menentukan Primary Key pada masing-masing entity.Primary Key adalah atribut pada entity yang bersifat unik.Jadi, setiap entity hanya memiliki satu Primary Key saja. Contoh: Entity Buku memiliki Primary Key bernama kode buku. Kode Buku ini bersifat unik, karena masing-masing buku memiliki kode yang berbeda-beda.

  Tentukan pula Foreign Key (Kunci Tamu) pada masing-masing Entity. Foreign Key adalah Primary Key yang ada dalam Entity yang lain. Contoh :pada Entity Pengarang misalnya terdapat atribut kode buku, yang mana, kode buku merupakan *Primary Key* dari *Entity* buku.

- ✓ Gambar ERD Berdasarkan Primary Key

  Menghilangkan relasi "many to many" dan memasukkan Primary dan Foreign Key pada masing-masing entitas. Relasi many to many antar entity perlu dihilangkan dengan cara menambah atribut baru antara 2 entity yang memiliki relasi many to many.

- ✓ Menentukan Atribut

Jika sudah melakukan step diatas, sekarang saatnya menentukan atribut pada masing-masing Entitas. Telitilah dalam menentukan atribut.

- ✓ Pemetaan Atribut

  Apabila atribut telah ditentukan, sekarang pasang atribut dengan entitas yang sesuai.

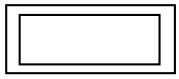
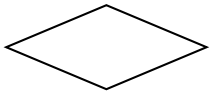
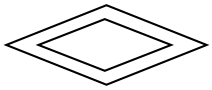
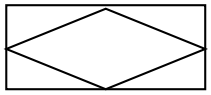
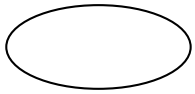
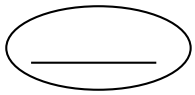
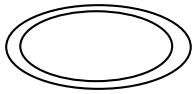
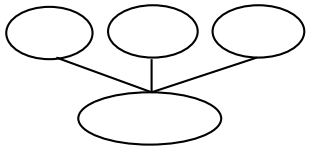
- ✓ Gambar ERD dengan Atribut

  Mengatur ERD seperti langkah 6 dengan menambahkan atribut dan relasi yang ditemukan.

- ✓ Periksa Hasil

  Periksa lagi ERD. Apakah ERD sudah menggambarkan sistem yang akan dibangun? Jika belum, cek kembali dari awal.

Simbol-simbol pada ERD :

NOTASI	KETERANGAN
	Entity
	*Weak Entity*
	*Relationship*
	*Indentity Relationship*
	*Gerund*
	*Attribut*
	*Key Attribut*
	*Weak Attribut*
	*Composite Attribut*

	Derived Attribut

5. Tipe Entitas

Pada umumnya entitas yang digunakan dalam diagram ER adalah strong entity atau entitas yang kuat, dimana keberadaannya tidak tergantung pada entitas lainnya Tetapi pada kenyataannya pembuatan model ER adakalanya melibatkan weak entity (entitas yang lemah) atau Agregasi.

- ✓ **Entitas Kuat**

  Entitas ini tidak memiliki ketergantungan dengan entitas yang lainnya, masing– masing dapat berdiri sendiri. Contohnya Entitas Mahasiswa, Dosen, Matakuliah. Gambar 6.7. berikut ini menggambarkan entitas kuat.

- ✓ **Entitas Lemah**

  Entitas ini muncul karena ketergantungannya pada sebuah relationship terhadap entitas lain. Karena sifat ketergantungannya entitas ini dapat memiliki ataupun tidak memiliki sebuah kunci atribut. Sebuah entitas lemah dapat terbentuk untuk melengkapi entitas kuatnya yang memiliki Multivalue Attribute ataupun karena komposisi dari relationship banyak ke banyak, banyak ke satu ataupun satu ke banyak. Seperti dijelaskan pada gambar 6.8. dan 6.9.

- ✓ **Entitas Agregasi**

Sebuah relasi terbentuk tidak hanya dari entitas tapi terkadang juga mengandung unsur dari relasi yang lain. Jika terjadi hal demikian dapat diakomodasi dengan Agregasi yang menggambarkan sebuah himpunan relasi yang secara langsung menghubungkan sebuah himpunan entitas dengan sebuah himpunan relationship dalam diagram ER yang mengandung unsur agregasi diperlihatkan pada gambar 10 berikut ini.

6. Tipe Atribut

Ada beberapa tipe atribut yang perlu diperhatikan dalam penggambaran model ER, yaitu Atribut kunci, single value (attribute atribut bernilai tunggal), multivalue attribute (atribut bernilai banyak/jamak), composite attribute (atribut komposit) dan derived attribute (atribut turunan).

- ✓ **Atribut Bernilai Banyak dan Bernilai Tunggal**

  Banyak atribut yang memiliki satu nilai tunggal pada suatu entitas tertentu, atribut yang demikian disebut atribut bernilai tunggal (Single Value Attribute). Contohnya atribut Tgl_lhr, TELP, dll pada entitas mahasiswa, atribut KD_MK pada entitas MATAKULIAH. Ada juga atribut yang memiliki sekelompok nilai yang disebut atribut bernilai banyak (Multi Value attribute). Contoh atribut HOBBY pada entitas Mahasiswa .

- ✓ **Atribut Atomik dan Komposit**

  Suatu atribut mungkin terdiri dari beberapa atribut yang lebih kecil atau sub-sub atribut yang lebih sedehana yang disebut dengan atribut komposit. Contoh Atribut NAMA dapat terdiri dari NAMA_DEPAN, NAMA_TENGAH, NAMA_BLK. Tetapi jika dalam penerapan pengolahan datanya tidak ada aktifitas yang

melibatkan sub-sub atribut tersebut, maka atribut tidak perlu diuraikan menjadi atribut komposit.

Sedangkan ada juga atribut yang tidak dapat dibagi kedalam sub-sub atribut disebut dengan atribut atomik atau atribut simple. Contoh dari atribut sederhana ini yaitu Jenis Kelamin , SKS, Sem dan lain-lain

- ✓ **Atribut Turunan**

  Pada beberapa kasus, ada dua atau lebih nilai atribut yang berhubungan misalkan atribut UMUR dan atribut TGL_LHR pada entitas Mahasiswa. Nilai atribut UMUR dapat ditentukan dengan tanggal saat ini dan nilai atribut TGL_LHR mahasiswa yang bersangkutan. Atribut UMUR ini disebut dengan atribut turunan. Penggambaran atribut ini dengan menggunkan notasi pada model ER dapat dilihat pada gambar 14. berikut ini.

BAB-XII

DESAIN INPUT dan OUTPUT

Overview

Pada bab ini membahas konsep normalisasi database berikut konsep-konsep lain yang mendasarinya. Dalam bab ini juga akan ditampilkan contoh penerapan normalisasi untuk tabel-tabel sederhana dalam kasus database.

Tujuan

1. Mahasiswa akan mengetahui konsep desain sistem terinci dan bagaimana melaksanakan desain sistem terinci
2. Mahasiswa akan dapat melakukan desain sistem terinci, mulai dari Desain output, input, Desain dialog terminal sampai desain file
3. Mahasiswa memahami cara untuk melakukan pengendalian input melalui validasi transaksi, cek data dan modifikasi data transaksi

Materi

1. Desain Output

Tujuan Desain Input

- ✓ Membuat penyelesaian input yang mudah dan efisien (*Effectiveness)*
- ✓ Menjamin input akan memenuhi tujuan yang diharapkan (*Accuracy*)

- ✓ Menjamin penyelesaian yang tepat (*Ease of use*)
- ✓ Membuat tampilan layar dan formulir yang menarik (*Consistency*)
- ✓ Membuat input yang tidak rumit (*Simplicity*)
- ✓ Membuat tampilan layar dan formulir yang konsisten (*Attractiveness*)

Syarat-syarat Desain Input

- ✓ Yang diinputkan hanya data-data variabel (bukan konstanta)
- ✓ Tidak perlu menginputkan data yang dapat dihitung atau disimpan dalam program
- ✓ Gunakan kode untuk atribut-atribut yang sesuai ‰ Jika suatu dokumen dirancang untuk mengumpulkan data, gunakan hal-hal berikut:
    - cantumkan instruksi pengisian form (dokumen)
    - minimalkan jumlah tulisan tangan
    - urutkan data yang harus diisi seperti urutan membaca buku (kiri /kanan, atas / bawah)
    - Jika mungkin, gunakan rancangan berdasar pada metafor (misal desain layar input penarikan rekening berdasar desain form standar penarikan rekening)

Kontrol internal dalam desain input:

- ✓ Jumlah input harus dimonitor, jangan sampai ada yang terlewat
- ✓ Pastikan bahwa data yang diinputkan valid : „ eksistensi (harus ada/tidak) „ tipe data domain nilai-nilai kombinasi self-checking digits „ format.

Tipe proses

Input data bisa dilakukan pada waktu yang berbeda. Hal ini ditentukan oleh jenis pemrosesan yang dibedakan menjadi:

- ✓ Batch processing: „ data tidak diproses langsung ketika datang, tetapi dikumpulkan dulu dalam satu batch (kumpulan), baru diproses
- ✓ Online processing:„ Data diinputkan (biasanya melalui workstation) dan langsung diproses
- ✓ Remote batch:„ Kombinasi batch dan online: data dimasukkan secara online atau offline kemudian dikumpulkan dalam batch sebelum akhirnya diproses.

Langkah-langkah Desain Input

- ✓ Langkah-langkah desain input: ‰
  - Identifikasi input sistem dan review kebutuhan Pemakai
  - Pilih kontrol GUI yang sesuai:
  - TextBox
  - Radio Button
  - Check Box
  - List Box
  - Drop-Down List
  - Combination Box (Combo Box)
  - Buttons

Desain, validasi, dan tes input menggunakan beberapa kombinasi alat bantu layout dan prototyping‰ Jika perlu, buat pula desain dokumen sumber (formulir yang dipakai untuk menyimpan data transaksi)

Proses Input dapat melibatkan dua atau tiga tahapan utama, yaitu :

- ✓ Data capture / Penangkapan data
- ✓ Data preparation / Penyiapan data
- ✓ Data entry / Pemasukan data

Input yang menggunakan alat input tidak langsung mempunyai 3 tahapan utama, yaitu *data capture*, *data preparation* dan *data entry*, sedangkan input yang menggunakan alat input langsung terdiri dari 2 tahapan utama, yaitu *data capture* dan *data entry*.

Tipe Input dapat dibedakan :

- ✓ Eksternal , Pada tipe ini pemasukan data berasal dari luar organisasi. Contoh : faktur pembelian, kwitansi-kwitansi dari luar organisasi, dll
- ✓ Internal , Pada tipe ini pemasukan data hasil komunikasi pemakai dengan sistem. Contoh : faktur penjualan, order penjualan, dll

Yang perlu diperhatikan dalam Perancangan Input adalah :

- ✓ Tipe input
- ✓ Fleksibel format
- ✓ Kecepatan
- ✓ Akurat
- ✓ Metode verifikasi

- ✓ Mudah dikoreksi
- ✓ Keamanan
- ✓ Mudah digunakan
- ✓ Kompatibel dengan sistem yang lain
- ✓ Biaya yang ekonomis

Langkah-langkah Perancangan Input Secara Umum :

- ✓ Menentukan kebutuhan Input dari sistem yang baru
- ✓ Input yang akan dirancang dapat ditentukan dari DFD sistem baru yang telah dibuat
- ✓ Menentukan parameter dari Input

2. Perancangan Dialog / Antarmuka

Tujuan dari perancangan Dialog adalah :

- ✓ Untuk menjaga agar pemasukan data benar
- ✓ Untuk menjawab pertanyaan yang sering diajukan oleh pemakai

Tipe Dialog :

- ✓ Dialog Aktif : Pemakai mengajukan pertanyaan atau memasukan data
- ✓ Dialog Pasif : Pada tipe ini pemakai memilih pilihan yang tersedia

Yang perlu diperhatikan dalam Perancangan Dialog adalah :

- ✓ Mudah digunakan
- ✓ Dapat memberikan petunjuk
- ✓ Menggambarkan atau sesuai dengan keinginan pemakai

- ✓ Cepat memberikan respon
- ✓ Dapat menampilkan pesan kesalahan
- ✓ Fleksibel
- ✓ Pemakai
- ✓ Human Engineering
- ✓ Dialog dan istilah

Tipe pemakai komputer:

- ✓ Ahli
- ✓ Pemula

Umumnya desain interface saat ini berasumsi pemakai adalah pemula yang sedang dalam proses menjadi ahli.

Desain antarmuka perlu memperhatikan:

Beberapa masalah yang sering muncul dalam desain antarmuka:

- ✓ Penggunaan jargon atau singkatan-singkatan komputer yang berlebihan
- ✓ Desain yang tidak jelas
- ✓ Ketidakmampuan membedakan tindakan-tindakan alternatif yang harus dipilih pemakai
- ✓ Pendekatan problem-solving yang tidak konsisten
- ✓ Desain yang tidak konsisten

Beberapa hal untuk memecahkan masalah di atas:

- ✓ Pahami user dan tugas mereka

- ✓ Libatkan user dalam desain atarmuka
- ✓ Uji sistem dengan melibatkan user
- ✓ Lakukan proses desain secara iteratif

Syarat-syarat Desain Antarmuka

- ✓ Layar harus diformat agar berbagai info, perintah, pesan muncul di area layar yang konsisten
- ✓ Pesan, perintah, informasi jangan terlalu panjang
- ✓ Jangan terlalu sering memakai atribut display yang mengganggu seperti blinking, highlight, dsb
- ✓ Nilai default dibuat jelas
- ✓ Antisipasi error yang mungkin dibuat user
- ✓ Jika ada error, user mestinya tidak boleh melanjutkan tanpa memperbaiki error tsb
- ✓ Jika user melakukan sesuatu yang membahayakan sistem, keyboard harus terkunci dan pesan utk meminta bantuan teknisi harus dimunculkan

Syarat-syarat dialog:

- ✓ Gunakan kalimat sederhana dan benar
- ✓ Jangan mencoba melucu
- ✓ Jangan menghina

Syarat-syarat istilah:

- ✓ Jangan gunakan jargon komputer

- ✓ Hindari singkatan jika mungkin
- ✓ Konsisten dengan pemilihan istilah
- ✓ Pilihlah istilah yang tepat untuk perintah

Proses perancangan antarmuka:

- ✓ Buatlah bagan dialog antarmuka (misal memakai diagram status/state diagram)
- ✓ Buatlah prototype dialog dan antarmuka
- ✓ Carilah umpan balik dari user
- ✓ Jika perlu, kembali ke langkah 1 atau 2

CONTOH KASUS

Contoh kasus perancangan dialog / antar muka Sistem Informasi Akademik

Entitas Luar Pendukung Sistem

Berdasarkan penagamatan secara langasung atau perancangan sebelumnya terdapat tiga entitas luar yang berhubungan dengan sistem. Setiap entitas luar tersebut mempunyai peran masing – masing yang berhubungan dengan sistem. Peran dari setiap entitas tersebut akan diuraikan sebagai berikut :

No.	Nama Entitas	Keterangan
1	Mahasiswa	✓ Mengisi formulir FRS/KRS untuk melakukan perwalian terhadap sejumlah matakuliah yang akan diambil.

		✓ Mengikuti Perkuliahan sampai dengan dinyatakan lulus semua matakuliah dan aturan lainnya yang telah ditetapkan program studi tersebut. ✓ Mengikuti setiap Ujian (UTS, UAS). ✓ Menerima Transkrip Nilai dan Kartu Hasil Studi (KHS).
	Dosen	✓ Mengajar satu atau lebih mata kuliah tertentu. ✓ Memberikan tugas, quiz, praktikum, ujian atau lainnya sebagai bahan evaluasi belajar. ✓ Mencatat data hadir mahasiswa pada setiap kali pertemuan proses belajar mengajar. ✓ Memberikan penilaian kepada setiap mahasiswa untuk setiap matakuliah yang diajarkannya.
	Prodi	✓ Mencatat data matakuliah ✓ Mencatat data hasil perwalian ✓ Mengolah data nilai akademik setiap mahasiswa. ✓ Membuat atau mencetak Transkrip Nilai, KHS.

Ketentuan – Ketentuan

Adapun ketentuan – ketentuan yang berlaku dalam aktivitas akademik yang berhubungan dengan pengolahan data nilai adalah sebagai berikut :

No.	Item	Ketentuan / Aturan
1	Data Mata Kuliah	Kode mata kuliah mengacu pada pengelompokkan mata kuliah berdasarkan : Mata Kuliah Pengembangan Kepribadian (MPK), Mata Kuliah Keilmuan & Ketrampilan (MKK), Mata Kuliah Keahlian Berkarya (MKB), Mata Kuliah Prilaku Berkarya (MPB), Mata Kuliah Kehidupan Bermasyarakat (MBB). Namun pada intinya keseluruhan mata kuliah tersebut mengacu pada kurikulum standar internasional.
2	Perwalian	Setiap pengambilan matakuliah dilakukan secara normal oleh setiap mahasiswa yaitu berkisar 20 sks sesuai paket sks tiap semester, kecuali semester tahun terakhir. Namun pengambilan jumlah beban sks oleh mahasiswa STEKOM Semarang semester berikutnya ditentukan oleh IP mahasiswa tersebut dengan berpedoman pada ketentuan – ketentuan yang telah dijelaskan pada bab sebelumnya. Proses perwalian harus selesai sebelum perkuliahan semester selanjutnya dimulai.
3	Data Nilai	✓ Data nilai yang telah diolah berupa angka dengan rentang nilai 0 – 100 dan disebut nilai absolut.

		✓ Data nilai kemudian dikonversi menjadi nilai mutu dan nilai lambang. ✓ Pada hasil pengolahan data nilai apabila diperoleh mahasiswa dengan nilai D dengan jumlah enam buah untuk S1 dan empat buah untuk D-3, maka nama mahasiswa tersebut wajib mengulang matakuliah tersebut, jika diperoleh nilai E maka mahasiswa tersebut wajib mengulang, dan jika nilai TD maka mahasiswa tersebut harus melengkapi tugas maximum 2 minggu dan jika tidak melengkapinya nilai E.

Pengkodean

Format pengkodean yang diterapkan pada sistem tersebut adalah sebagai berikut :

Nomor Induk Mahasiswa (NIM)

Identitas mahasiswa yang sekarang diterapkan adalah {99 99 9999}

1 2 3

Keterangan :

Dua (2) digit pertama : Kode Program Studi

Untuk Strata-1

11 : Sistem Komputer

12 : Desain Grafis

13 : Manajemen Informatika

14 : Komputerisasi Akuntansi

Untuk Program Diploma

21 : Teknik Komputer

22 : Komputer Grafis

23 : Teknik Elektronika

24 : Komputerisasi Akuntansi

Dua (2) digit kedua : Kode tahun masuk (16 →2016, 17 → 2017, 18→2018, dst)

Empat 4 digit pertama : Nomor Urut Mahasiswa

Contoh :

- 11160002 berarti mahasiswa program studi S1-sistem komputer, angkatan tahun masuk 2016 dengan nomor urut 0002.
- 22170008 berarti mahasiswa program studi D3-komputer grafis, angkatan tahun masuk 2017 dengan nomor urut 0008.

Kode Mata Kuliah

Kode mata kuliah yang diterapkan adalah {XX-X-X-9-999}

1(xx) 2(x) 3(9) 4(9) 5(9) 6(99)

2 digit pertama(xx) : Kode Program Studi

SK : Sistem Komputer

DG : Desain Grafis

MI : Manajemen Informatika, dst

1 digit kedua (x) : Sebagai kelompok matakuliah

A→ Mata Kuliah Pengembangan Kepribadian (MPK),

B →Mata Kuliah Keilmuan & Ketrampilan (MKK),

C →Mata Kuliah Keahlian Berkarya (MKB),

D → Mata Kuliah Prilaku Berkarya (MPB),

E → Mata Kuliah Kehidupan Bermasyarakat (MBB).

1 digit ketiga (9) : Kode Semester

1 → Semester 1

2 → Semester 2

3 → Semester 3

4 → Semester 4 , dst

1 digit keempat (x) : Jenis Matakuliah

U → Umum

W → Wajib

P → Pilihan

3 digit keempat : Nomor Urut Matakuliah

999 → 001

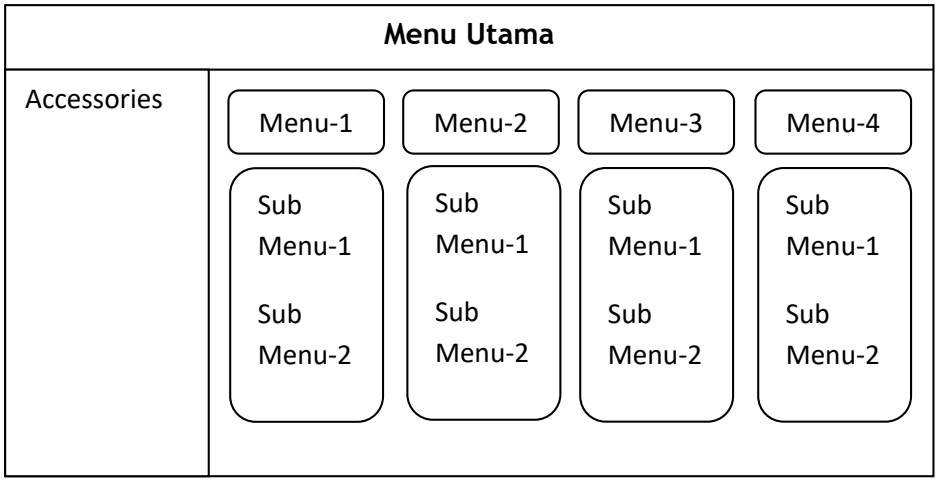
999 → 002, dst

- (min) sebagai tanda pemisah

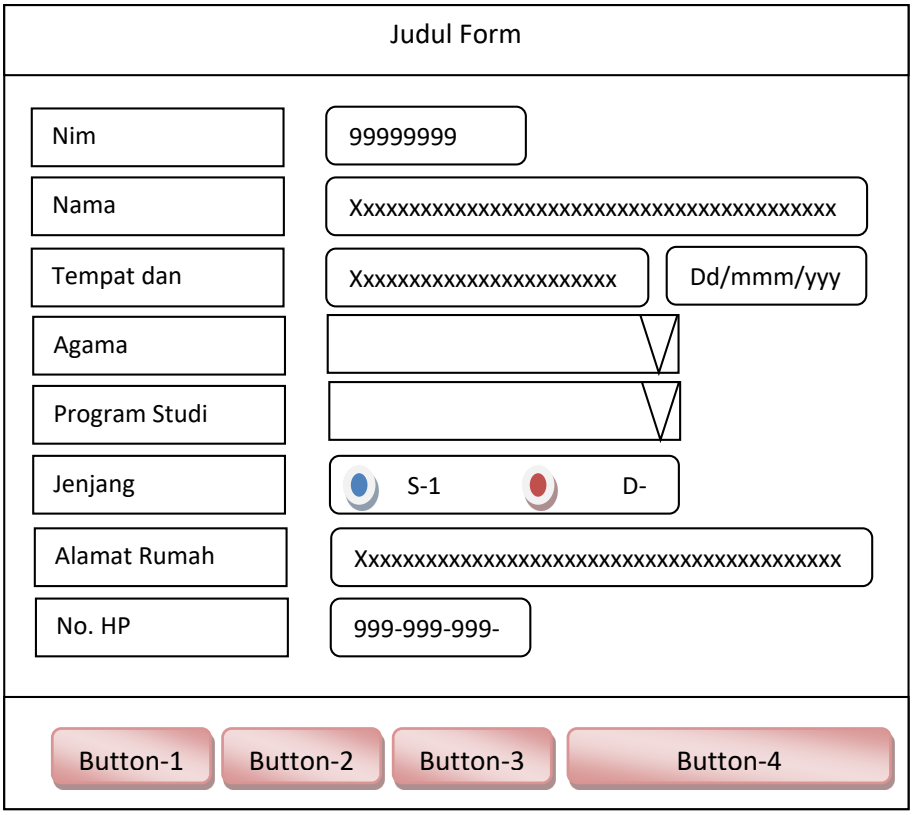
Contoh :

- SK-A-U-1-001 adalah kode mata kuliah dari program sistem komputer , kelompok MPK , merupakan matakuliah umum, pada semester-1 , dengan nomor urut 001.
- DG-B-W-007 adalah kode mata kuliah dari program desain grafis , kelompok MKK , merupakan matakuliah wajib, pada semester-1 , dengan nomor urut 001.

Desain Menu Utama



Desain Form Mahasiswa



3. Perancangan Database

Penerapan database dalam sistem informasi disebut dengan database system. Sistem basis data (database system) ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya.

✓ **Tipe dari File**

- File Master : Berisi data yang tetap dimana pemrosesan terhadap data hanya pada waktu-waktu tertentu. Terdapat 2 tipe file master :
  - ❖ File Referensi , Data yang tetap, dimana pengolahan terhadap data tersebut memerlukan waktu yang lama
  - ❖ File Dinamik , Data yang ada dalam file berubah tergantung transaksi
- File Input / Transaksi : Berisi data masukan yang berupa data transaksi dimana data-data tersebut akan diolah oleh komputer
- File Laporan : Berisi informasi yang akan ditampilkan
- File Sejarah / Arsip : Berisi data masa lalu yang sudah tidak aktif lagi, tetapi disimpan untuk keperluan masa datang
- File Backup / Pelindung : Berisi salinan data-data yang masih aktif di database pada suatu waktu tertentu
- File Kerja / Temporary File : Berisi data-data hasil pemrosesan yang bersifat sementara
- File Library : Berisi program-program aplikasi atau utility program

✓ **Akses File :**

Metode yang menunjukkan bagaimana suatu program komputer akan membaca

record-record dari suatu file. File dapat diakses dengan 2 cara, yaitu : *Sequential* (urut) , *Direct / Random* (langsung)

- ✓ **Organisasi File : Pengaturan dari record secara logika didalam file dihubungkan satu dengan yang lainnya.**
  - File Urut (Sequential File) : Merupakan file dengan organisasi urut dengan pengaksesan urut pula
  - File Urut Berindex (Indexed Sequential File) : Merupakan file dengan organisasi urut dengan pengaksesan langsung
  - File Akses Langsung (Direct Acces File) : Merupakan file dengan organisasi acak dengan pengaksesan langsung
- ✓ **Alat Perancangan Database**
  - ERD
  - Mapping
  - Normalisasi

Langkah-langkah Perancangan Database secara umum :

- Menentukan kebutuhan file database untuk sistem baru
- Menentukan parameter dari file database

4. Desain Output

Pada tahap desain output secara terinci, desain output ini hanya dimaksudkan untuk menentukan kebutuhan output dari sistem baru. Output apa saja yang dibutuhkan untuk sistem yang baru? Desain output secara terinci dimaksudkan untuk menjawab pertanyaan ini.

Bagaimana dan seperti apa bentuk dari output-output tersebut? Desain output terinci dimaksudkan untuk menjawab pertanyaan ini. Desain output yang akan dibahas pada bab ini adalah untuk output berbentuk laporan di media keras. Desain output di media lunak dalam bentuk dialog di layar terminal akan dibahas di bab selanjutnya.

Macam-macam Bentuk Laporan

Bentuk dari laporan yang dihasilkan oleh sistem informasi, yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

Laporan Berbentuk Tabel

Berikut ini adalah macam-macam laporan yang berbentuk tabel yang menekankan kualitas isi serta kegunaannya :

- ✓ ***Notice Report***

  Notice report merupakan bentuk laporan yang memerlukan perhatian khusus. Laporan ini harus dibuat sesederhana mungkin, tetapi jelas, karena dimaksudkan supaya permasalahan-permasalahan yang terjadi tampak dengan jelas sehingga dapat langsung ditangani.

Lapotan Penjualan Kenaikan Barang
Jenis Barang Bulan ..... Tahun .....

Daerah Penjualan	X Kenaikan
Demak	10 x
Kudus	15 x
Pati	5 x
Grobogan	**20 x**

✓ ***Equipoised Report***

Isi dari equipoised report adalah hal-hal yang bertentangan. Laporan ini biasanya digunakan untuk maksud perencanaan. Dengan disajikannya informasi yang berisi hal-hal bertentangan, maka dapat dijadikan sebagai dasar di dalam pengambilan keputusan. Contoh Laporannya adalah :

Laporan Perencananaan Memasuki Pasar Baru
Daerah Semarang Untuk Tahun 2018

	Jelek	Baik
Penjualan	1.000.000	1.600.000
Harga Pokok Penjualan	600.000	1.000.000
Laba Kotor	400.000	600.000
Biaya Penjualan	300.000	350.000
Biaya Administrasi	125.000	150.000
Laba (Rugi)	(25.000)	100.000

✓ ***Variance Report***

Macam laporan ini menunjukkan selisih (variance) antara standar yang sudah ditetapkan dengan hasil kenyataannya atau sesungguhnya. Contoh dari laporan ini adalah :

PT. LANGGENG JAYA ABADI

LAPORAN PENJUALAN BULAN .....TAHUN .....

Barang Terjual	Unit Jual	Harga Beli	Harga Jual	Selisih Harga	Total Selisih
Hardisk Eksternal	20	500,000	650.000	150.000	3.000.000
Flashdisk 16GB	50	120.000	150.000	30.000	1.500.000
Spiker Aktif	10	25.000	40.000	10.000	100.000
Total Selisih					4.600.000

✓ ***Comparative Report***

Isi dari laporan ini adalah membandingkan antara satu hal dengan hal yang lainnya. Misalnya pada laporan rugi/laba atau neraca dapat dibandingkan antara nilai-nilai elemen tahun berjalan dengan tahun-tahun sebclumnya. Contoh dari comparative report adalah sebagai berikut:

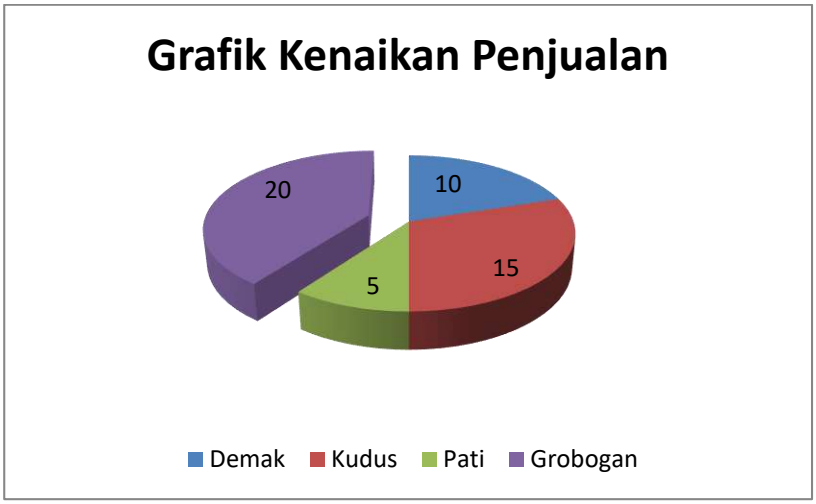
NERACA
Per 31 Desember 20....

Aktiva	31-12-2016	31-12-2017	Selisih	Prosentase
Aktiva Lancar	300.000	450.000	150.000	50.00%
Aktiva Tetap	250.000	300.000	50.000	20.00%
Total Aktiva	550.000	750.000	200.000	36.00%

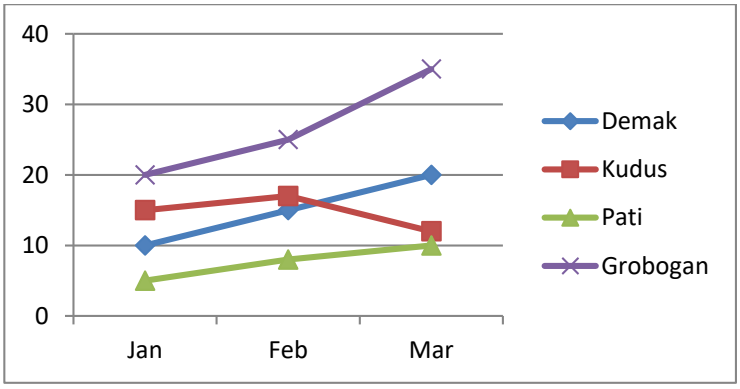
Laporan Berbentuk Grafik

Laporan yang berbentuk grafik atau bagan dapat diklasifikasikan diantaranya sebagai bagan garis (line chart), bagan batang (bar chart) dan bagan pastel (pie chart).

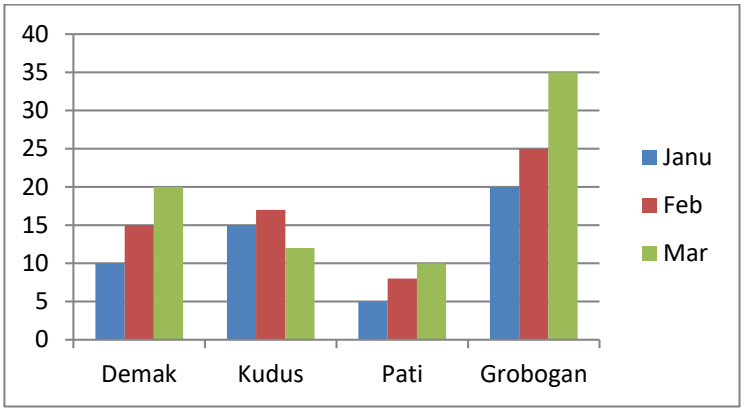
Daerah	Kenaikan Penjualan
Denak	10
Kudus	15
Pati	5
Grobogan	20



Grafik Lingkaran



Grafik Garis



Grafik Batang

✓ ***Bagan Garis***

Pada bagan garis (line chart), variasi dari data ditunjukkan dengan suatu garis atau kurva.

Bagan garis mempunyai beberapa kebaikan, yaitu:

- Dapat menunjukkan hubungan antara nilai dengan baik.
- Dapat menunjukkan beberapa titik.
- Tingkat ketepatannya dapat diatur sesuai dengan skalanya.
- Mudah dimengerti.

Disamping kebaikannya, bagan garis mempunyai beberapa kelemahan, yaitu:

- Bila terlalu banyak garis atau kurva (sekitar lebih dari 4 buah garis atau kurva), maka akan tampak ruwet.
- Hanya terbatas pada 2 dimensi.
- Spasi dapat menyesatkan.

✓ **Bagan Batang**

Nilai-nilai data dalam bagan batang (bar chart) digambarkan dalam bentuk batang-batang vertikal ataupun batang-batang horisontal.

Kebaikan dari bagan batang adalah sebagai berikut:

- Baik untuk perbandingan.
- Dapat menunjukkan nilai dengan tepat.
- Mudah dimengerti.

Kelemahannya:

- Terbatas hanya pada satu titik saja.
- Spasi dapat menyesatkan.

✓ **Bagan Pastel.**

Bagan pastel (pie chart) merupakan bagan yang berbentuk lingkaran menyerupai kue pastel (pie). Tiap-tiap potong dari pie dapat menunjukkan bagian dari data.

Kebaikan dari bagan pastel adalah sebagai berikut ini.

- Baik untuk perbandingan sebagian dengan keseluruhannya.
- Mudah dimengerti.

Kelemahannya :

- Penggunaannya terbatas
- Ketepatannya kurang
- Tidak dapat menunjukkan hubungan beberapa titik

1. Pedoman Desain Laporan

- ✓ Untuk laporan formal, sedapat mungkin dibagi menjadi tiga bagian utama, yaitu :
  - Judul laporan.
  - Tubuh laporan.
  - Catatan kaki laporan yang dapat berisi ringkasan, subtotal atau grandtotal.
- ✓ Untuk laporan-laporan yang penting, gunakanlah kertas yang berkualitas baik, tidak mudah sobek serta tidak mudah kotor.
- ✓ Untuk tiap-tiap batas tepi laporan (margin), sebaiknya diberi jarak 2 1/2 cm, sehingga bila pinggir laporan tersobek tidak akan mengenai isi laporannya.
- ✓ Gunakanlah spasi baris yang cukup, sehingga laporan mudah dibaca.
- ✓ Untuk hal-hal yang ingin ditonjolkan, dapat ditulis dengan huruf besar, tebal, atau digaris-bawahi.
- ✓ Gunakanlah bentuk huruf cetak yang jelas dan tidak membingungkan serta hindari pengguanaan font yang sulit untuk dimengerti.
- ✓ Jika isi laporan menjelaskan suatu daftar urutan, gunakanlah tanda "." atau "-". Bila urutannya penting dapat dipergunakan tanda 1,2,3 dan seterusnya dan sajikan dalam urutan yang terpenting.
- ✓ Letakkanlah informasi yang mendetail di lampiran dan gunakanlah penunjuk yang mudah dipahami untuk

menjelaskan kepada pemakai laporan Ietak dari informasi detail tersebut.

- ✓ Usahakan di dalam laporan berisi keterangan-keterangan yang diperlukan yang mungkin akan ditanyakan oleh pemakai laporan bila keterangan-keter-angan tersebut tidak ada.
- ✓ Laporan untuk tingkat manajemen yang lebih tinggi, sebaiknya lebih tersaring dan untuk tingkat manajemen yang lebih rendah lebih terinci.
- ✓ Laporan harus dibuat dan didistribusikan tepat pada waktunya.
- ✓ Laporan harus sederhana tetapi jelas.
- ✓ Laporan harus diungkapkan dalam bentuk dan Bahasa yang mudah dimen-gerti dan dipahami oleh pemakainya.
- ✓ Isi laporan harus akurat.
- ✓ Laporan bilamana mungkin harus distandardisasi. Bentuk-bentuk laporan yang selalu berubah akan menyebabkan kebingungan bagi mereka yang menggunakannya.
- ✓ Laporan harus berguna
- ✓ Biaya pembuatan laporan harus dipertimbangkan

2. Alat-alat Desain Output Terinci

Dua buah alat desain sistem dapat digunakan untuk desain output terinci, yaitu sebagai berikut ini.

- ✓ Printer layout form atau printer spacing chart atau printer layout chart merupakan suatu bagan yang digunakan untuk menggambarkan sketsa bentuk dari output di printer.
- ✓ Kamus data output yang merupakan pengembangan dari kamus dari arus data. Kamus data output digunakan untuk menjelaskan secara terinci tentang data yang akan disajikan di laporan.

3. Mengatur Tata Letak Isi Output

Pengaturan isi dari output akan secara langsung menentukan kemudahan dari output untuk dipahami dan dimengerti. Pengaturan tata letak output merupakan pekerjaan desain yang penting dan sangat diperlukan baik bagi pemakai sistem maupun bagi programmer. Bagi pemakai sistem digunakan untuk menilai isi dan bentuk dari output apakah sudah sesuai dengan yang dinginkan atau belum. Bagi programmer akan digunakan sebagai dasar pembuatan program untuk menghasilkan output yang diinginkan. Programmer membutuhan desain output ini untuk menentukan posisi kolom, baris dan informasi yang harus disajikatt~i suatu output. Pengaturan tata letak isi output yang akan dicetak di printer dapat digunakan alat bagan tata letak printer (printer layout chart) dan kamus data output.

Supaya tidak dihasilkan sampah, maka input yang masuk dalam sistem informasi harus tidak boleh berupa sampah. Oleh karena itu desain input harus berusaha membuat suatu sistem yang dapat menerima input yang bukan sampah. Desain input terinci dimulai dari desain dokumen dasar sebagai penangkap input yang pertamakali. Jika dokumen dasar

tidak didesain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang. Pada bab ini akan dibahas desain input terinci untuk perancangan dokumen dasar dan kode-kode yang digunakan untuk input.

Dokumen Dasar

Dokumen dasar (source document) merupakan formulir yang digunakan untuk menangkap (capture) data yang terjadi. Data yang sudah dicatat di dokumen dasar kemudian dimasukkan sebagai input ke sistem informasi untuk diolah. Dokumen dasar sangat penting di dalam arus data di sistem informasi.

Dokumen dasar ini dapat membantu di dalam penanganan arus data sebagai berikut ini.

- ✓ Dapat menunjukkan macam dari data yang harus dikumpulkan dan ditangkap.
- ✓ Data dapat dicatat dengan jelas, konsisten dan akurat.
- ✓ Dapat mendorong lengkapnyg data, disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya.
- ✓ Bertindak sebagai pendistribusi data, karena sejumlah tembusan dari formulirformulir tersebut dapat diberikan kepada individu-individu atau departemendepartemen yang membutuhkannya.
- ✓ Dokumen dasar dapat membantu di dalam pembuktian terjadinya suatu transaksi yang sah, sehingga sangat berguna untuk audit trail (pelacakan pemeriksaan).

- ✓ Dokumen dasar dapat digunakan sebagai cadangan atau pelindung back up) dari file-file data di komputer.

Untuk mencapai maksud tersebut, dokumen dasar harus dirancang dengan baik. Berikut ini merupakan petunjuk-petunjuk di dalam perancangan dokumen dasar yang baik, sebagai berikut ini.

- ✓ Kertas yang dipergunakan.

  Beberapa faktor harus dipertimbangkan di dalam pemilihan kertas yang akan digunakan, yaitu sebagai berikut ini.

  - Lamanya dokumen dasar tersebut akan disimpan.
  - Penampilan dari dokumen dasar.
  - Banyaknya dokumen dasar tersebut ditangani.
  - Bagaimana menanganinya (secara halus, kasar, dilipat atau dibawa-bawa oleh pemakainya).
  - Lingkungan-lingkungannya (berlemak, kotor, panas, dingin, lembab atau mengandung asam).
  - Metode pengisian data di dokumen dasar tesebut, ditulis tangan atau dice-tak dengan mesin.
  - Kemananan terhadap pudarnya data yang dicatat di dokumen dasar.

- ✓ Ukuran dari dokumen dasar.

  Usahakan ukuran dari dokumen dasar berupa ukuran dari kertas yang standar dan banyak dijual. Ukuran kertas yang umum adalah ukuran kuarto (81/2" x 11") dan ukuran folio (81/2" x 14"). Jika dokumen dasar lebih kecil dari ukuran

kertas standar, sebaiknya dibuat ukuran yang merupakan kelipatan yang tidak membuang kertas, misalnya ukuran kertas standar dibagi 2, dibagi 3, dibagi 4 dan sebagainya.

- ✓ Warna yang digunakan.

  Penggunaan warna akan membantu di dalam mengidentifikasikan dengan cepat dokumen dasar yang dipergunakan. Warna yang baik adalah warna yang datanya mudah dibaca, terutama bila menggunakan karbon. Warna yang baik ini adalah warna yang cerah. Warna-warna gelap, seperti misalnya biru tua, hijau tua, merah tua, coklat, ungu, hitam dan lain sebagainya sebaiknya di-hindari untuk digunakan.

- ✓ Judul dokumen dasar.

  Dokumen dasar harus diberi judul yang dapat menunjukkan jenis dan kegu-naan dari dokumen dasar tersebut. Judul harus sesingkat mungkin tetapi jelas. Bila dokumen dasar akan digunakan oleh pihak-pihak luar perusahaan, selain judul yang ada, maka nama perusahaan sebaiknya juga dicantumkan.

- ✓ Nomor dokumen dasar.

  Nomor dokumen dasar dapat digunakan untuk menunjukkan keunikannya. Nomor dokumen dasar dapat diletakkan di pojok bawah kiri atau di pojok bawah kanan (jangan di atas kiri, karena tertutup bila distaples dan jangan di atas kanan, karena dapat membingungkan dengan nomor

urut dokumen dasar). Nomor dokumen dasar ini dapat juga digunakan untuk menunjukkan sumber dan jenisnya. Misalnya nomor dokumen dasar PJ-FO1 dapat menunjukkan bahwa sumbernya dari departemen penjualan (ditunjukan oleh kode PJ) dan jenisnya adalah faktur penjualan model ke 1 (ditunjukkan oleh kode FOl).

- ✓ Nomor urut dokumen dasar.

  Disamping nomor dokumen dasar, nomor unit dari masing-masing dokumen dasar biasanya dicantumkan di pojok kanan atas. Nomor urut ini sangat perlu untuk tujuan pengendalian (dapat diketahui bila ada dokumen dasar yang hilang bila nomornya meloncat), untuk pelacakan pemeriksaan dan untuk pengarsipan.

- ✓ Nomor dan jumlah halaman.

  Bila dokumen dasar terdiri lebih dari satu halaman, maka tiap-tiap halaman harus diberi nomor dan jumlah halamannya, supaya bila ada halaman yang hilang dapat diketahui. Misainya halaman pertama dapat diberi nomor halaman

  1-3 (menunjukkan halaman pertama dari sejumlah 3 halaman), halaman ke duadiberi nomor 2-3 dan seterusnya. Nomor dan jumlah halaman ini biasanya diletakkan pada sebelah kanan atas.

- ✓ Spasi.

Spasi antar baris dan spasi antar karakter pada dokumen dasar harus diperhat-ikan, terutama bila dokumen dasar akan diisi dengan data yang dicetak dengan mesin. Untuk spasi di dokumen dasar harus disesuaikan dengan spasi yang dibuat oleh mesin.

- ✓ Pembagian area.

  Dokumen dasar harus dibentuk dengan pembagian area sedemikian rupa, sehingga memudahkan untuk mencarinya guna pengisian atau pencarian data. Pembagian area ini meliputi area judul, area halaman, area kontrol, area organisasi, area obyek, area tubuh, area berita, area otorisasi, area jumlah dan area nomer.

BAB-XIII

UNIFED MODELING LANGUAGE (UML)

Overview

Pada bab ini membahas konsep normalisasi database berikut konsep-konsep lain yang mendasarinya. Dalam bab ini juga akan ditampilkan contoh penerapan normalisasi untuk tabel-tabel sederhana dalam kasus database.

Tujuan

1. Mahasiswa memahami definisi dan tujuan normalisasi
2. Mahasiswa mampu mengidentifikasi super key, candidate key dan primary ey dari sebuah table
3. Mahasiswa mampu mengidentifikasi functional dependency yang ada pada sebuah tabel, termasuk partial dan transitive FD
4. Mahasiswa mengenal bentuk normal pertama, ke dua, ke tiga dan BCNF serta mampu melakukan normalisasi dengan menerapkan bentuk-bentuk normal tersebut
5. Mahasiswa mengenal sekilas tentang bentuk-bentuk normal lain dan memahami konsep denormalisasi

Materi

1. Pendahuluan

Saat ini piranti lunak semakin luas dan besar lingkupnya, sehingga tidak bisa lagi dibuat asal-asalan. Piranti lunak saat ini seharusnya dirancang dengan memperhatikan hal-hal seperti scalability , security , dan eksekusi yang robust walaupun dalam kondisi yang sulit. Selain itu arsitekturnya harus didefinisikan dengan jelas, agar bug mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya. Keuntungan lain dari perencanaan arsitektur yang matang adalah dimungkinkannya penggunaan kembali modul atau komponen untuk aplikasi piranti lunak lain yang membutuhkan fungsionalitas yang sama.

Pemodelan (modeling) adalah proses merancang piranti lunak sebelum melakukan pengkodean (coding). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Dengan menggunakan model, diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti scalability, robustness, security , dan sebagainya. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebuan segitiga sukses (the triangle for success). Ketiga unsur tersebut adalah metode pemodelan (notation), proses (process) dan tool yang digunakan.

Memahami notasi pemodelan tanpa mengetahui cara pemakaian yang sebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadap metode pemodelan dan proses disempurnakan dengan penggunaan tool yang tepat.

2. Apa itu UML

Unified Modelling Language (UML) adalah sebuah “bahasa” yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti

lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax /semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch [1], metodologi coad [2], metodologi OOSE [3], metodologi OMT [4], metodologi shlaer-mellor [5], metodologi wirfs-brock [6], dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – http://www.omg.org). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson

menyusun tiga buku serial tentang UML pada tahun 1999 [7] [8] [9]. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

3. Konsep Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Major Area	View	Diagrams	Main Concepts
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
dynamic	state machine view	statechart diagram	state, event, transition, action
	actifity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		colaborating diagram	collaborating, interaction, collaboration rule, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Abstraksi konsep dasar UML yang terdiri dari structural classification , dynamic behavior , dan model management , bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari Diagrams . Main concepts bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut.

Lalu darimana kita mulai ? Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan:

- ✓ Menguasai pembuatan diagram UML
- ✓ Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

- use case diagram
- class diagram
- statechart diagram
- activity diagram
- sequence diagram
- collaboration diagram
- component diagram
- deployment diagram

4. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

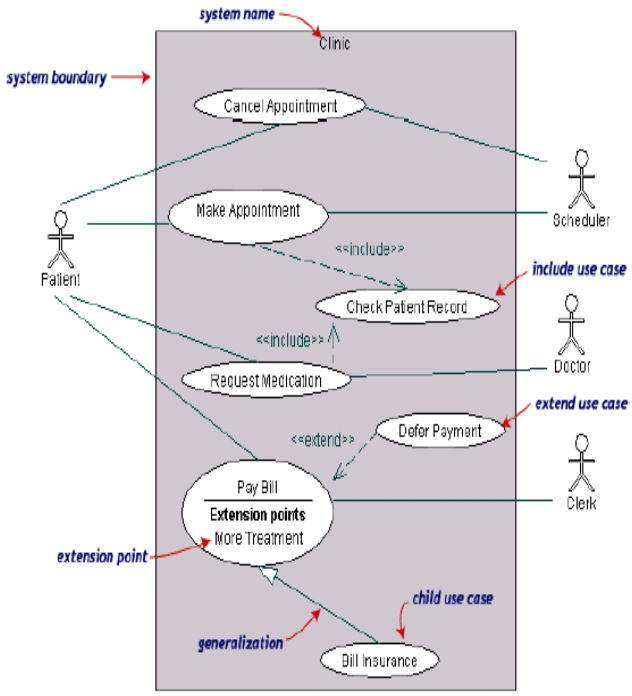
Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng- create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah

sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

Sebuah use case dapat meng- include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di- include akan dipanggil setiap kali use case yang meng- include dieksekusi secara normal. Sebuah use case dapat di- include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common .

Sebuah use case juga dapat meng- extend use case lain dengan behaviour -nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

Contoh Use Case



5. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment , pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

a. Nama (dan stereotype)

b. Atribut

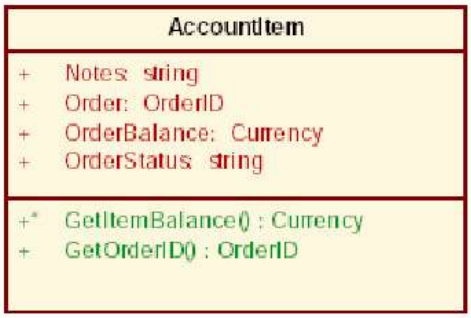
c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. Private , tidak dapat dipanggil dari luar class yang bersangkutan

b. Protected , hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya

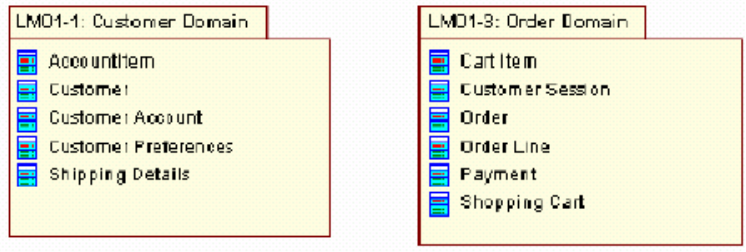
c. Public , dapat dipanggil oleh siapa saja



Class dapat merupakan implementasi dari sebuah interface , yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time .

Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi

package . Kita juga dapat membuat diagram yang terdiri atas package .



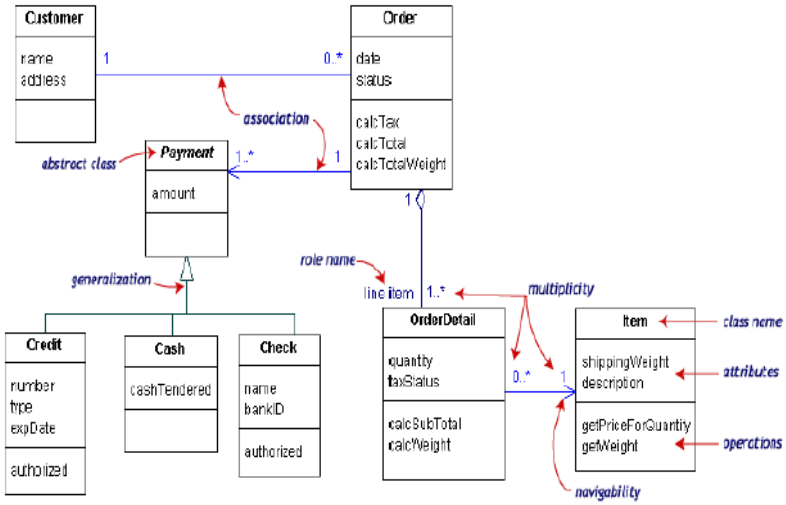
6. Hubungan Antar Class

✓ Asosiasi, yaitu hubungan statis antar class . Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus

mengetahui eksistensi class lain. Panah navigability m enunjukkan arah query antar class .

- ✓ Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
- ✓ Pewarisan, yaitu hubungan hirarkis antar class . Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
- ✓ Hubungan dinamis, yaitu rangkaian pesan (message) yang di- passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

Contoh Class Diagram



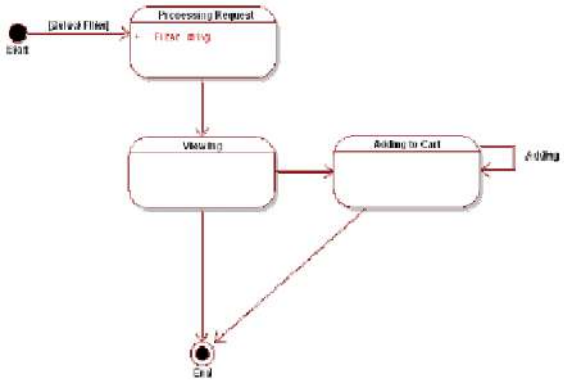
7. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang

diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.



8. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di- trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas

menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state , standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

9. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display , dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

Diawali dari apa yang men- trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class . Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity .Collaboration Diagram Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram , tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message .

Setiap message memiliki sequence number , di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

10. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code , baik berisi source code maupun binary code , baik library maupun executable , baik yang muncul pada compile time, link time , maupun run time . Umumnya komponen terbentuk dari beberapa class dan/atau package , tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface , yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

11. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di- deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah node adalah server, workstation , atau piranti keras lain yang digunakan untuk men- deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

12. Langkah-Langkah Penggunaan UML

Berikut ini adalah tips pengembangan piranti lunak dengan menggunakan UML:

- Buatlah daftar business process dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.

- Petakan use case untuk tiap business process untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian perhalus use case diagram dan lengkapi dengan requirement, constraints dan catatan-catatan lain.
- Buatlah deployment diagram secara kasar untuk mendefinisikan arsitektur fisik sistem. Definisikan requirement lain (non-fungsional, security dan sebagainya) yang juga harus disediakan oleh sistem.

Berdasarkan use case diagram , mulailah membuat activity diagram .

- Definisikan objek-objek level atas (package atau domain) dan buatlah sequence dan/atau collaboration diagram untuk tiap alir pekerjaan. Jika sebuah use case memiliki kemungkinan alir normal dan error, buatlah satu diagram untuk masing-masing alir.
- Buarlah rancangan user interface model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario use case .
- Berdasarkan model-model yang sudah ada, buatlah class diagram . Setiap package atau domain d ipecah menjadi hirarki class lengkap dengan atribut dan metodanya..
- Akan lebih baik jika untuk setiap class dibuat unit test untuk menguji fungsionalitas class dan interaksi dengan class lain.

Setelah class diagram dibuat, kita dapat melihat kemungkinan pengelompokan class menjadi komponen-komponen. Karena itu buatlah component diagram pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia berinteraksi dengan baik.

- Perhalus deployment diagram yang sudah dibuat. Detilkan kemampuan dan requirement piranti lunak, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam node.

Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan :

- Pendekatan use case , dengan meng- assign setiap use case kepada tim pengembang tertentu untuk mengembangkan unit code yang lengkap dengan tes.
- Pendekatan komponen, yaitu meng- assign setiap komponen kepada tim pengembang tertentu.
- Lakukan uji modul dan uji integrasi serta perbaiki model berserta code nya.
- Model harus selalu sesuai dengan code yang aktual.
- Piranti lunak siap dirilis.

13. Tool Yang Mendukung UML

Saat ini banyak sekali tool pendesainan yang mendukung UML, baik itu tool komersial maupun opensource. Beberapa diantaranya adalah:

Rational Rose (www.rational.com)

Together (www.togethersoft.com)

Object Domain (www.objectdomain.com)

Jvision (www.object-insight.com)

Objecteering (www.objecteering.com)

MagicDraw (www.nomagic.com/magicdrawuml)

Visual Object Modeller (www.visualobject.com)

BAB-XIV

EASY CASE

Overview

Pada bab ini membahas tools perancangan sistem secara tersetruktur, dengan menggunakan tools easy case.

Tujuan

1. Mahasiswa mampu memahami dan menggunakan tools easycase dalam aplikasi perancangan programnya.
2. Mahasiswa mampu merancang dengan tools untuk pembuatan contex diagram, dekomposisi hingga diagram levelnya.
3. Mahasiswa mampu membedakan model notasi dalam tools ini dengan baik, karena banyak sekali model notasi yang ada.
4. Mahasiswa mampu mengaplikasikan peracangan dengan model tersetruktur untuk skripsinya nanti.

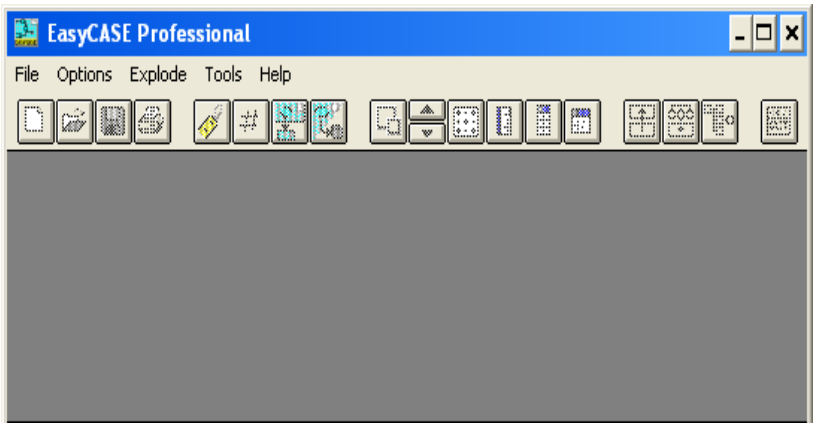
Materi

1. Easycase

EasyCase Propesional for Windows merupakan salah satu alat bantu perancangan sistem yang mempunyai kelebihan jika dibandingkan dengan alat bantu yang telah ada sebelumnya, diantaranya : Flowchart *(for Dos atau*

for Windows). Cara mejalankan easycase dapat digunakan langkah-langkah sebagai berikut :

- ✓ Double klik pada Shorcut EasyCase
- ✓ Start – Program – Development - Easycase

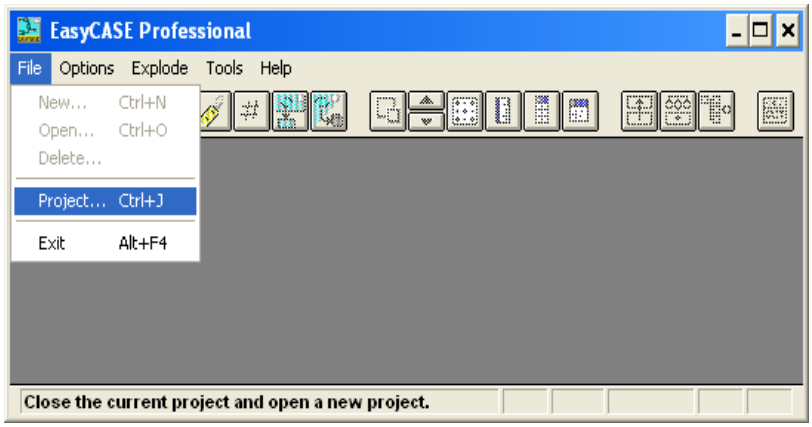


Gambar 1.1 Tampilan Awal Easycase

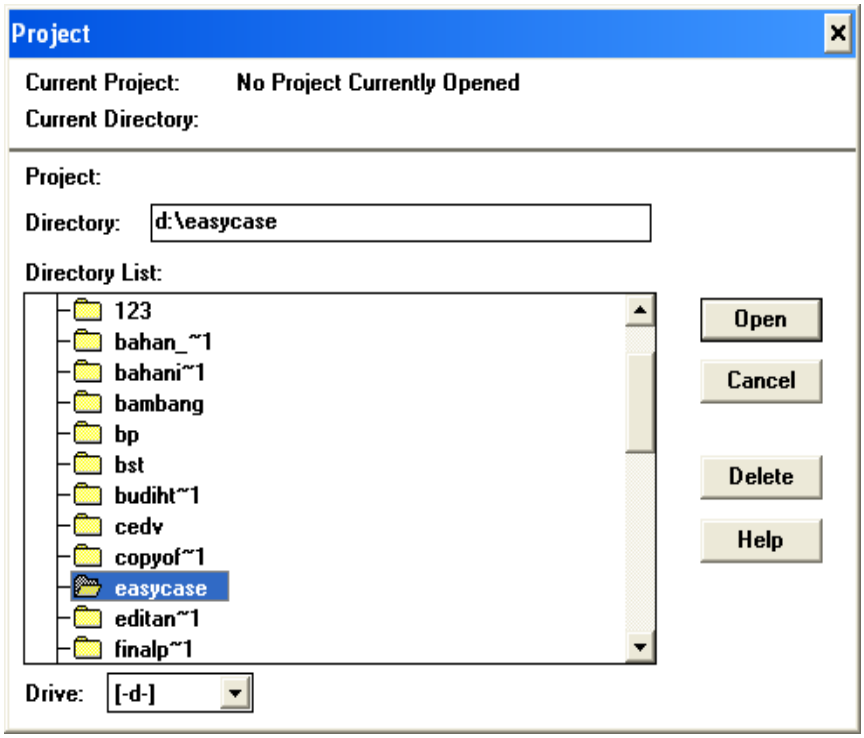
Langkah selanjutnya untuk dapat bekerja dengan EasyCase adalah membuat Project. Project tersebut akan digunakan untuk membuat dan menyimpan semua pekerjaan yang berhubungan dengan proyek tersebut.

2. Project dan Context Diagram

Untuk membuat suatu proyek dapat dilakukan dengan memilih menu file, kemudian pilih Project atau tekan tombol CTRL + J secara bersamaan, selanjutnya akan muncul tampilan seperti dibawah ini

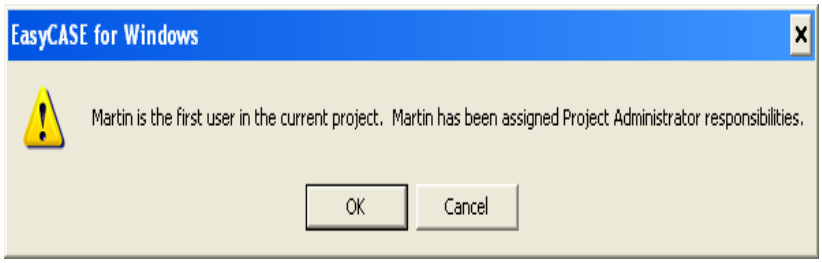


Gambar 1.2 Menu File

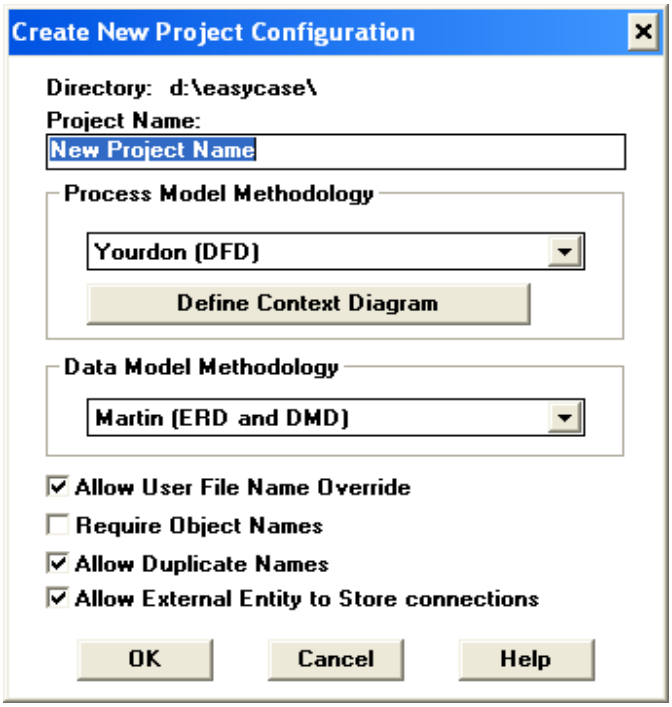


Gamabr 1.3 Simpan Project

Sebelum anda membuat sebuah project , maka harus disiapkan sebuah folder/directory yang akan Anda gunakan untuk project yang bersangkutan. Apabila Anda belum menyiapkan folder/directory, maka akan muncul komentar seperti terlihat pada gambar dibawah ini, tetapi apabila anda telah menyiapkan sebuah folder/directory untuk project tersebut , maka akan muncul seperti gambar 1.5 dibawah ini



Gambar 1.4 Pesan EasyCase

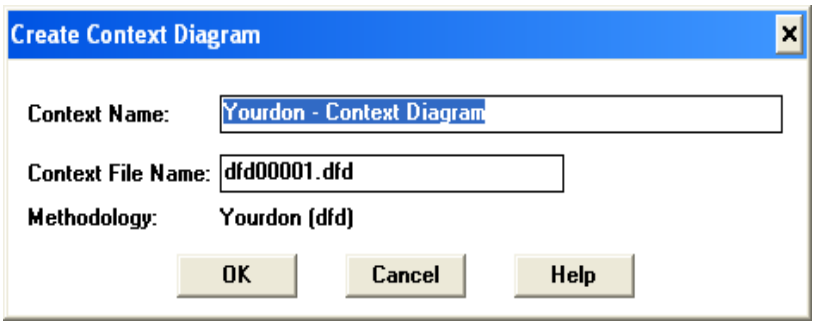


Gambar 1.5 Memberi Nama Project

Selanjutnya Anda dapat memasukkan Project Name, Prosess Model Methodology, Data Model Methodology, Define Contect Diagram serta berbagai options yang telah disediakan dan diakhiri dengan menekan tombol OK

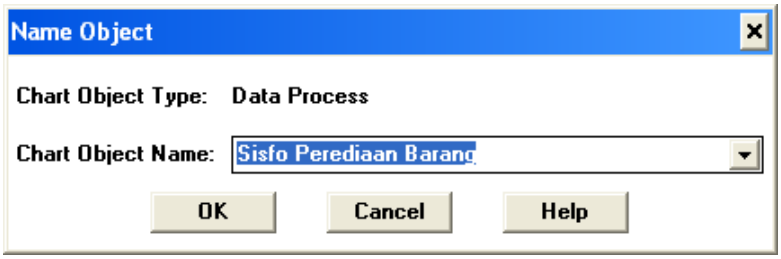
Untuk memilih atau menentukan jenis-jenis prosess model methodology atau data model methodology dapat dilakukan dengan cara menekan / meng-klik tanda panah kecil yang terletak disebelh kanan pada option yang bersangkutan. Misalnya nama contect diagram diisi dengan Sistem Informasi Stock Barang, Prosess model methodology (Yourdon DFD) dan Data model methodology (Martin DFD dan DMD).

Apabila semua pilihan telah selesai dengn yang Anda inginkan , maka selanjutnya pilih tombol Define Contect Diagram dan akan muncul tampilan seperti pada gambar 1.6 diubawah ini. Pada bagian ini anda diminta memberikan nama Contect , nama file contect dan diakhiri dengan menekan tombol OK

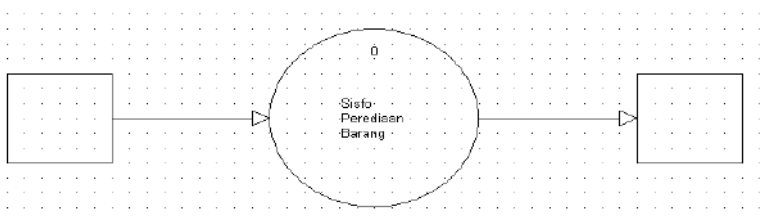


Gambar 1.6 Membuat Context Diagram

Masukan nama contect diagram dengn nama Sisfo Stock Barang – Contect Diagram , Context file name dengan nama Contect dan akhiri dengan OK. Setelah itu akan muncul tampilan seperti pada gambar 1.7 berikut ini

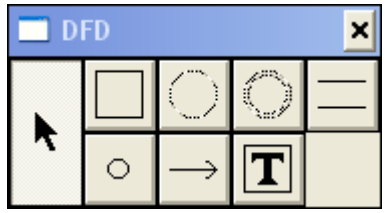


Gambar 1.7



Gambar 1.8 Memberi Nama Object

Pada bagian ini , anda bisa membuat contect diagram untuk sebuah sistem sesuai dengan yang anda butuhkan. Easycase menyediakan beberapa simbol yang dapat anda gunakan untuk membuat contect diagram tersebut



Gambar 1.9 Komponen DFD

- ✓ **Menyimpan File**

  Untuk menyimpan pekerjaan yang telah anda buat , dapat dilakukan dengan cara memilih menu file, kemudian Save, Save As atau Save All. Selain itu dapat dilakukan dengan cara menekan /meng-klik tool bergambar disket

- ✓ **Meletakan Simbol pada Worksheet**

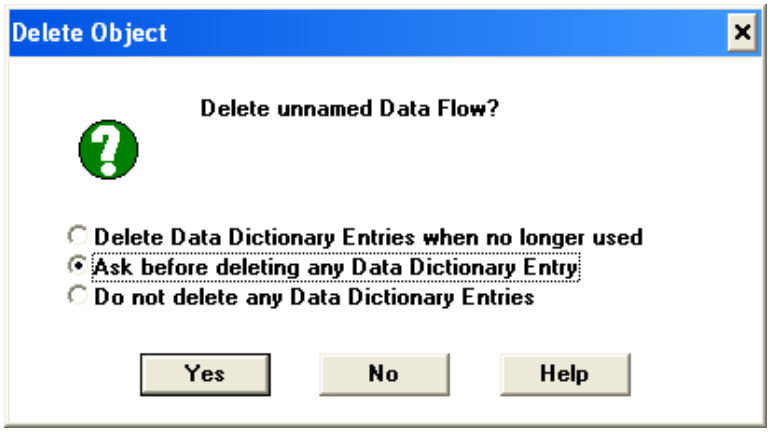
  Untuk meletakan simbol kedalam worksheet dapat anda lakukan dengan cara memilih simbol yang anda inginkan kemudian anda lakukan klik pada tempat yang anda inginkan pada worksheet anda

- ✓ **Memindah Simbol**

Simbol-simbol yang telah anda letakan pada worksheet dapat anda pindahkan ke tempat lain dengan cara meng-klik simbol tersebut , kemudian lakukan drag ke tempat lain sesuai dengan yang anda inginkan

✓ **Menghapus Simbol**

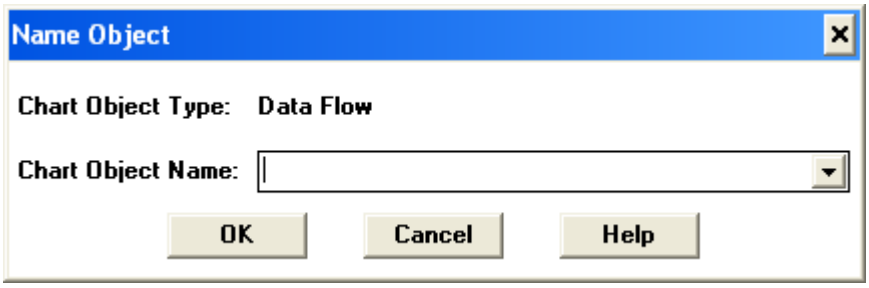
Apabila anda salah meletakan simbol dalam worksheet anda, maka anda dapat menghapusnya dengan cara meng-klik simbol yang akan anda hapus (sehingga muncul blok hitam disekeliling simbol yang anda pilih) , kemudian anda pilih menu edit, delete atau langsung menekan tombol Del pada keyboard. Selain menggunakan kedua cara tersebut diatas, anda juga dapat memanfaatkan pop-up menu dengan cara klik simbol yang akan anda hapus kemudian klik kanan pada mouse anda, kemudian pilih menu delete, maka akan muncul konfirmasi penghapusan sperti pada gambar 1.10



Gambar 1.10 Menghapus Object

✓ **Memberi Nama Simbol**

Untuk memberi nama simbol dapat dilakukan dengan cara memilih simbol yang akan diberi nama, kemudia klik kanan dan pilih menu Name. Selain itu juga dapat dilakukan dengan cara menggunakan menu Edit, kemudian pilih Name, seteleh itu akan muncul tampilan seperti pada gamabr dibawah ini

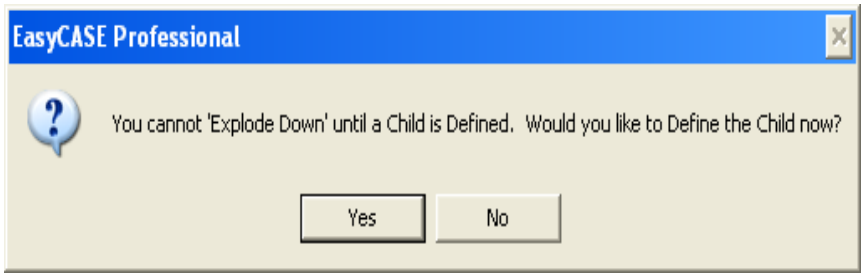


Gambar 1.11 Memberi Nama Object DFD

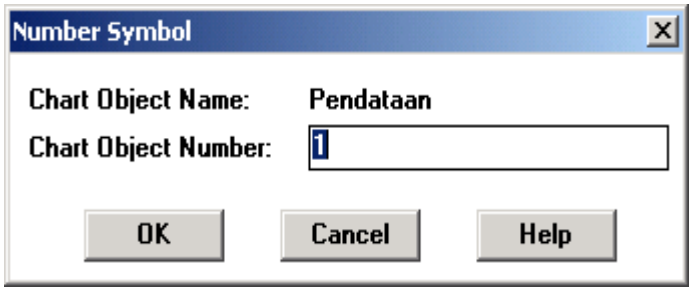
✓ **Memberi Nomor Simbol**

Tidak semua simbol dapat diberi nomor , tetapi hanya simbol-simbol tertentu yang dapat diberi nomor. Sebelumnya anda memberi nama simbol pada sebuah simbol, maka anda harus terlebih dahulu sudah memberi nama simbol. Apabila anda memberi nomor simbol dan simbol tersebut belum diberi nama maka akan muncl komentar dibawah ini

Selain itu juga ada simbol yang nomornya tidak dapat diganti , misalnya simbol proses pada contect diagram (0) yang harus diberi nilai atau angka 0



Gambar 1.2 Menanyakan Arah Turunan di Tampilkan



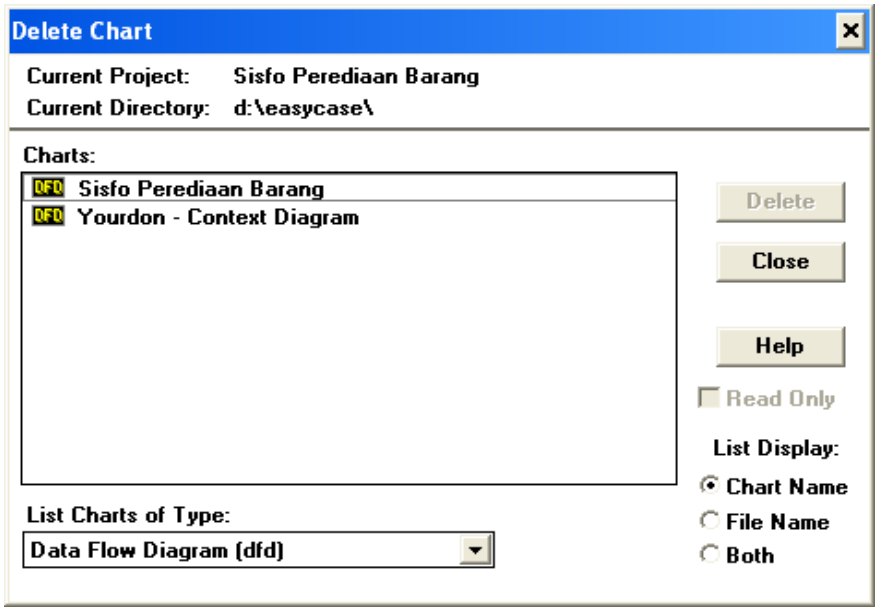
Gambar 1.13 Memberi Simbol dengan Nomor

✓ **Menghubungkan simbol satu dengan simbol lainya**

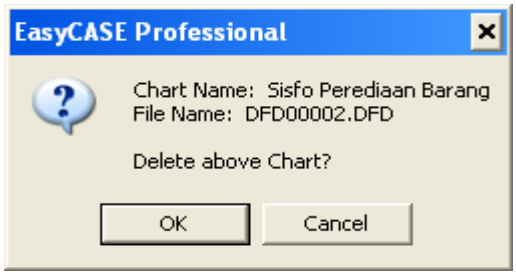
Pilih atau klik tanda simbol panah, kemudian klik simbol yang akan dihubungkan (sehingga muncul titik-titik putih disekeliling simbol tersebut), kemudian klik pada slah satu titik putih tersebut, setelah itu klik simbol yang lainnya (sehingga muncul titik-titik putih disekeliling simbol tersebut) kemudian klik pada salah satu titik putih tersebut, maka kedua simbol tersebut sudah terhubung

✓ **Menghapus Chart**

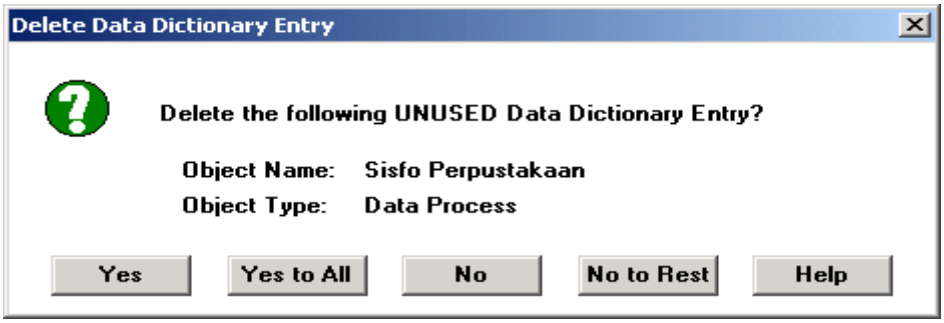
Untuk menghapus chart file (contect / DFD) dapat dilakukan dengan cara menggunkan menu file, delete, kemudian pilih Context/DFD yang akan dihapus. Perhatikan gambar 1. 14 berikut ini



Gambar 1.14 Tampilan Object DFD yang Sdh Dikerjakan



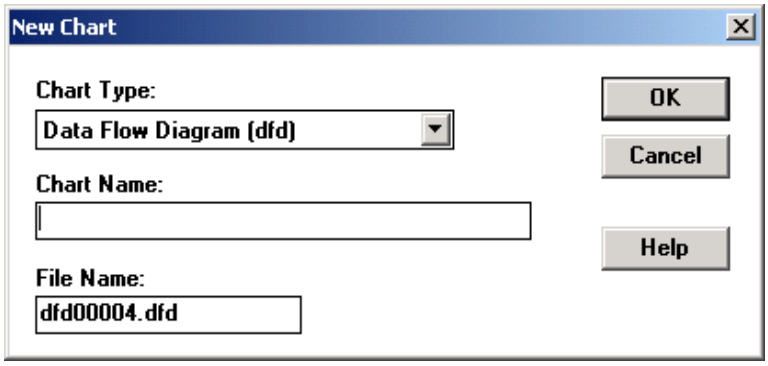
Gambar 1.15 Menghapus Chart



Gambar 1.16 Mengahapus Chart dan Subnya

3. Data Flow Diagram (DFD) Levelled 0

Setelah membuat contect diagram, langkah selanjutnya adalah membuat DFD levelled dengan menggunakan cara : Pilih menu File , kemudian pilih New Chart atau klik toolbar New Chart



Gambar 1.17 Membuat Object Baru

Dari tampilan diatas , anda diminta memasukan Type Chart (DFD, DMD atau DSD), nama chart dan nama file. Selanjutnya anda pilih OK sehingga anda sudah dapat membuat DFD yang anda harapkan.

Adapun cara atau teknik menggambarkan / membuat DFD sama dengan apabila anda membuat context diagram yang telah lalu

4. Child (DFD Levelled 1 dan seterusnya)

Untuk dapat membuat sebuah child, maka anda perlu melakukan dua langkah yaitu:

- Mendifinisikan Child
- Membuat/Menuju Child

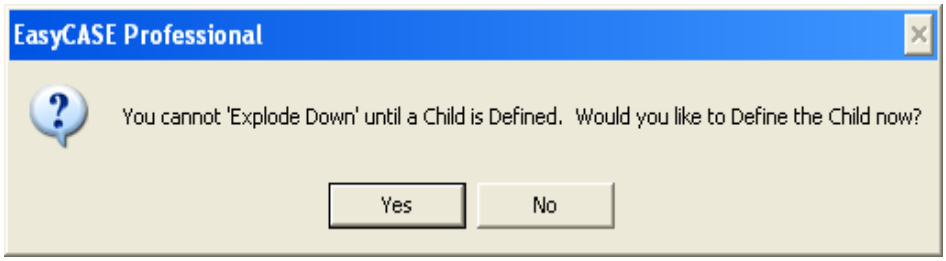
Untuk mendefinisikan child , anda pilih simbol proses yang akan dibuat sebuah child sehingga simbol proses tersebut dikelilingi dengan titik-titik hitam, kemudia pilih klik kanan sehingga muncul pop-menu dan pilih define child sehingga akan menampilkan gambar seperti berikut :

Gamba 1.18 Membuat Turunan Chart

Kemudian lakukan langkah-langkah sebagai berikut :

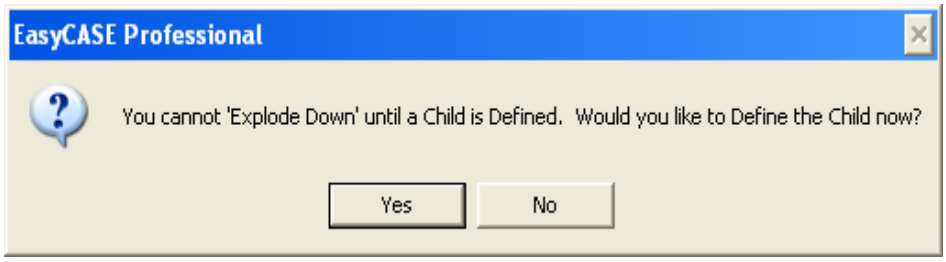
- Masukkan type child
- Masukkan nama child
- Masukkan nama file child
- Setelh itu pilih OK

Apabila anda akan membuat child tetapi belum pernah mendifinisikan child yang akan dibuat maka tampil pesan seperti berikut ini :



Gamba 1.19 Tanyaka Buat Chart dan Subnya

Langkah selanjutnya menuju child yang akan dibuat dengan cara memilih simbol proses yang teleh didifinisikan childnya tersebut, kemudian lakukan klik kanan dan pilih Goto Child. Selanjutnya akan ditampilkan seperti pada gambar berikut ini



Gambar 1.20 Menampilkan Chart dan Subnya

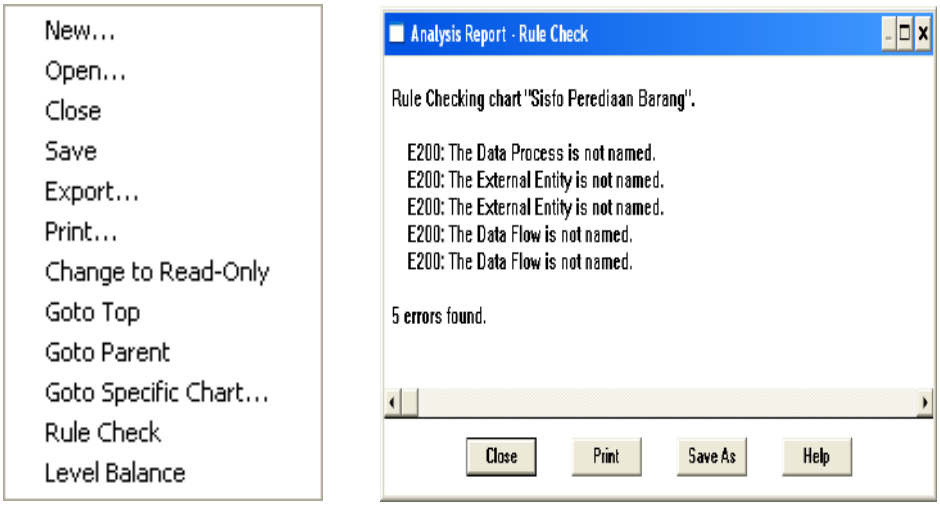
5. Rule Check & Level Balance

EasyCase mempunyai fasilitas untuk mengecek kesalah-kesalahn terhadap apa yang kita buat diantaranya :

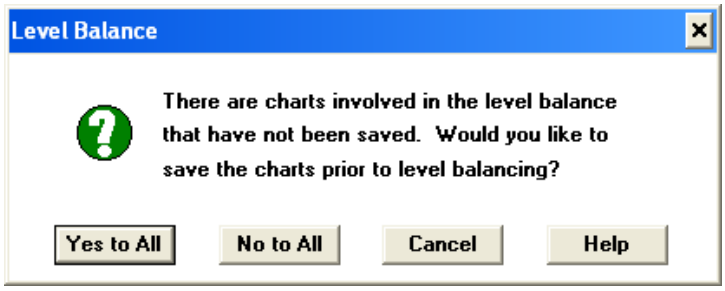
Rule Check digunakan untuk mengontrol aliran diagram yang anda buat

Level Balance digunakan untuk mengontrol kesimbangan diagram yang anda buat

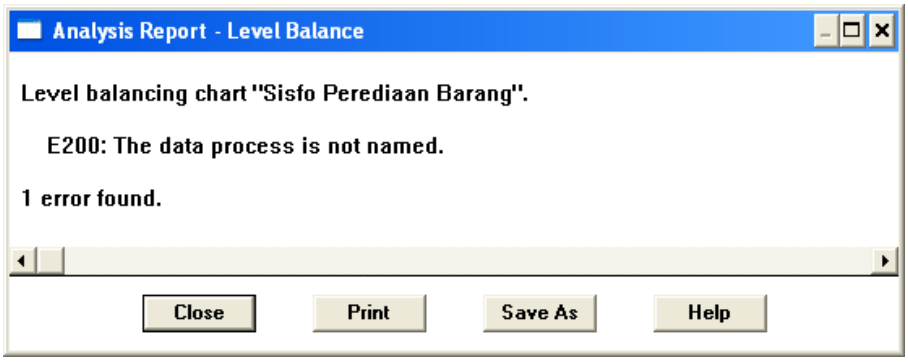
Untuk melakukan pengontrolan tersebut dapat dilakukan dengan cara memilih menu toolbars, kemudian pilih Rule Check atau Level Balance, perhatikan gambar berikut ini



Gambar 1.21 Rule Chek Kesalahan



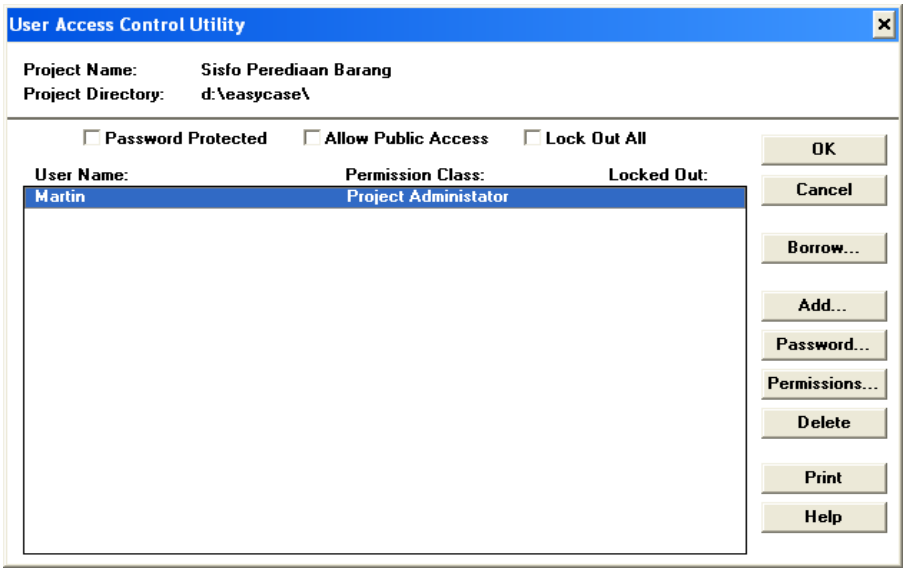
Gambar 1.22 Level Balance



Gambar 1.23 Hasil Level Balance

6. User Access Control Utility

Anda dapat menambah, menghapus user yang nantinya berhak menggunakan, merubah dan menghapus isi suatu project dengan cara pilih menu tools, kemudia Access Control maka komputer akan menampilkan gambar seperti dibawah ini

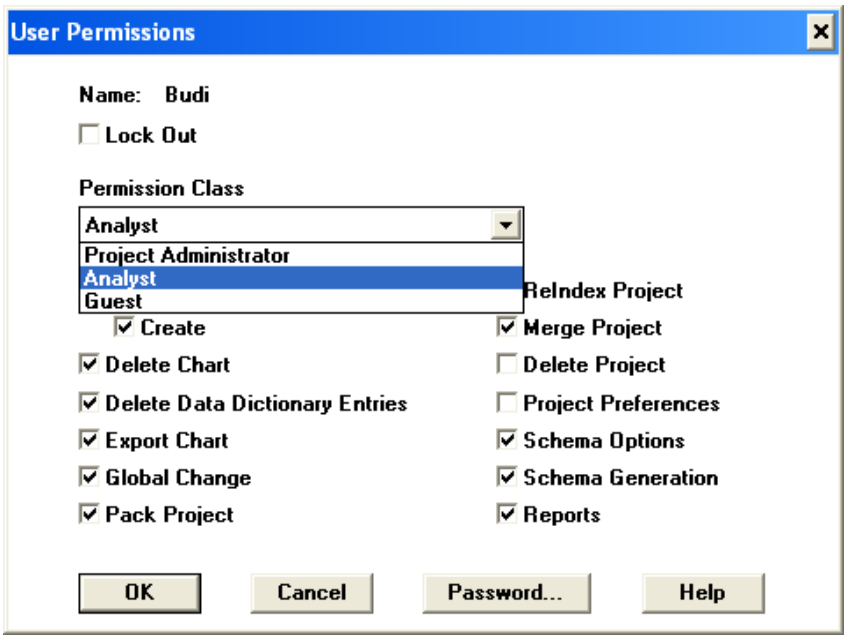


Gambar 1. 24 Mengganti Account User

Untuk menambah user pilih Add dan isikan user yang baru kemudian pilih OK

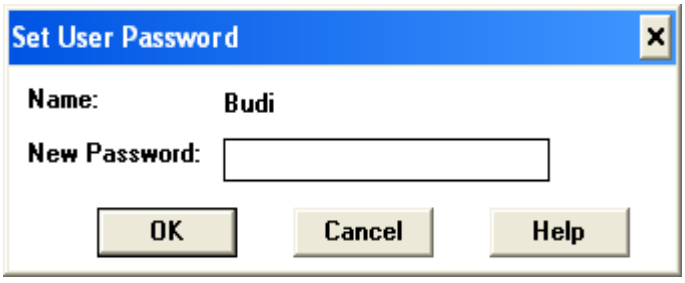
Gambar 1.25 Menambah User Baru

Sedangkan untuk memasukan hak aksesnya dapat dilaukan dengan cara memeilih user yang akan diatur haknya, kemudian pilih tombol permisson (perhatikan gambar berikut ini)



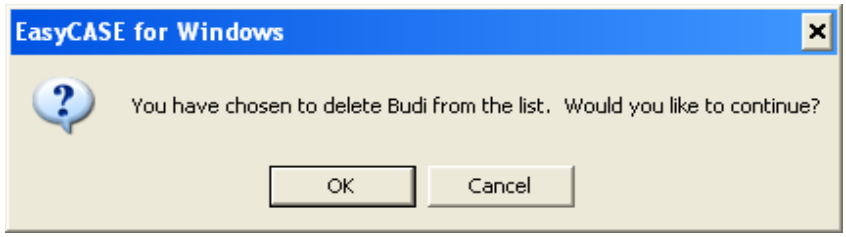
Gambar 1.26 Level User

Apabiula anda akan memberikan password untuk user , maka pilihlah tombol password sehingga akan menghasilkan tampilan seperti pada gamabr berikut ini



Gambar 1.27 Set Pass Ulang

7. Hapus User



Gambar 1.28 Delete User

BAB-XV

RATIONAL ROSE

Overview

Pada bab ini membahas konsep normalisasi database berikut konsep-konsep lain yang mendasarinya. Dalam bab ini juga akan ditampilkan contoh penerapan normalisasi untuk tabel-tabel sederhana dalam kasus database.

Tujuan

1. Mahasiswa memahami definisi dan tujuan normalisasi
2. Mahasiswa mampu mengidentifikasi super key, candidate key dan primary ey dari sebuah table
3. Mahasiswa mampu mengidentifikasi functional dependency yang ada pada sebuah tabel, termasuk partial dan transitive FD
4. Mahasiswa mengenal bentuk normal pertama, ke dua, ke tiga dan BCNF serta mampu melakukan normalisasi dengan menerapkan bentuk-bentuk normal tersebut
5. Mahasiswa mengenal sekilas tentang bentuk-bentuk normal lain dan memahami konsep denormalisasi

Materi

Salah satu pertanyaan yang paling sering saya temukan dari para mahasiswa adalah apa saja step-step yang harus ditempuh dalam merancang sistem berbasis UML? Saya biasanya akan memperbaiki pertanyaan ini karena UML pada dasarnya hanya kumpulan diagram atau teknik visualisasi; UML tidak mendikte langkah-langkah pengembangan sistem! Lalu apa yang harus dilakukan dalam merancang sebuah sistem?

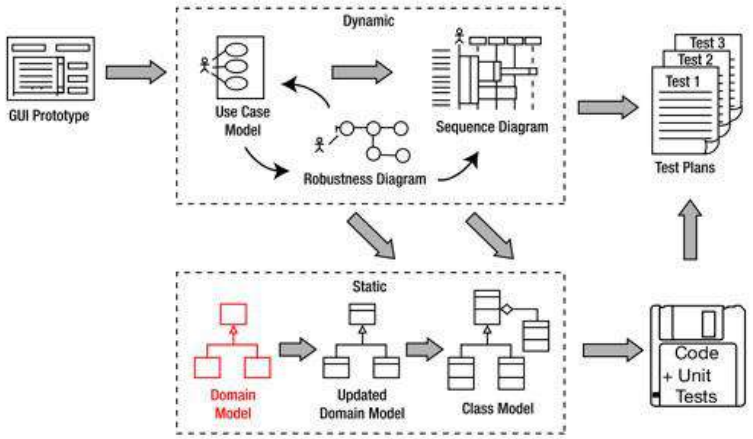
Saya yakin para mahasiswa telah memperoleh mata kuliah pengembangan sistem informasi sebelumnya, tapi kenapa mereka bertanya demikian disaat mereka harus merancang sistem yang nyata (misalnya untuk skripsi)? Salah satu penyebabnya adalah analisa & perancangan terlalu teoritis sehingga tidak mendukung implementasi/pembuatan kode program.

Dari beberapa buku analisa & perancangan yang saya baca, buku **Use Case Driven Object Modeling with UML: Theory and Practice** adalah salah satu buku yang spesial dan sangat membantu. Bukan hanya dilengkapi dengan humor, tetapi buku ini memberikan metode analisa & perancangan yang sangant berguna saat diterapkan dalam pembuatan kode program! Buku yang memakai metode **ICONIX Process** ini ditulis oleh analyst yang memiliki latar belakang programmer.

Saya tahu bahwa sangat sulit merangkum sebuah buku 438 halaman dalam sebuah artikel blog, tapi saya ingin menunjukkan bahwa diagram UML hasil analisa & perancangan bukanlah sekedar basa basi yang dihasilkan analyst. Dengan metode yang salah, analyst kerap terlihat tidak berguna di mata developer. Metode yang salah juga menyebabkan mahasiswa lebih senang membuat kode program terlebih dahulu, baru melakukan reverse engineering untuk menghasilkan diagram UML. Dengan kata lain, sistem dibuat tanpa analisas & perancangan, sementara diagram UML hanya produk

sampingan yang menambah ketebalan skripsi tanpa fungsi yang sangat berarti.

Gambar berikut ini memperlihat proses analisa & perancangan sistem informasi dengan **ICONIX process**:



ICONIX Process

Proses analisa & perancangan sistem yang saya rangkum dari buku **Use Case Driven Object Modeling With UML: Theory and Practice** adalah sebagai berikut ini:

1. Membuat Functional Requirement

Gunakan Microsoft Word untuk menuliskan **functional requirement** (apa yang dapat dilakukan oleh sistem?). Tahap ini melibatkan business analyst, pelanggan, end user, dan project stakeholders lainnya. **Functional requirement** bersifat tidak terstruktur dan tidak dapat dipakai dalam perancangan secara langsung.

Berikut ini adalah contoh potongan dari **function requirement** untuk sebuah sistem bengkel motor yang sederhana:

Sistem harus dapat memproses work order untuk motor mulai dari antri, sedang dikerjakan

oleh seorang mekanik, selesai di-servis, dan proses pembayaran.

Sebuah work order memiliki informasi jenis pekerjaan dan sparepart yang dijual.

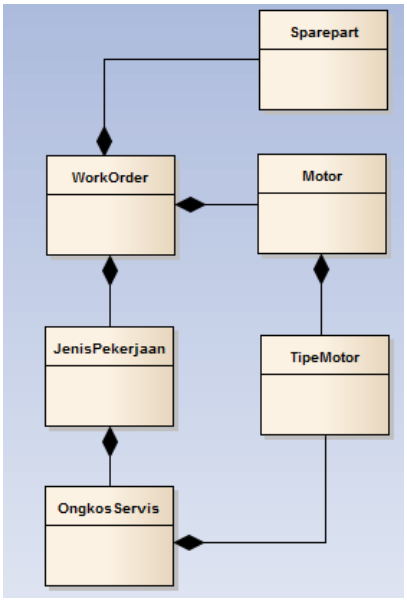
Harga ongkos servis berdasarkan jenis pekerjaan dan tipe motor.

2. Membuat Domain Model (sederhana)

Salah satu fungsi **domain model** adalah menyamakan istilah yang akan pakai diproses selanjutnya. Misalnya, apakah saya akan memakai istilah '*work order'* atau '*pekerjaan servis'*? Apa saya akan memakai sebutan '*sparepart*' atau '*suku cadang*'? Walau terlihat sepele, perbedaan istilah seringkali menimbulkan salah paham dalam komunikasi tim.

Pada tahap ini, **domain model** adalah *class diagram* yang hanya memakai relasi pewarisan (**is-a**/adalah sebuah) dan agregasi (**has-a**/memiliki sebuah). *Class diagram* ini belum memiliki atribut dan operasi. Nantinya, di proses selanjutnya, **domain model** akan diperbaiki dan dikembangkan menjadi lebih detail.

Gambar berikut ini memperlihatkan contoh **domain model** untuk **functional requirement** di langkah 1:



Domain Model Versi Awal

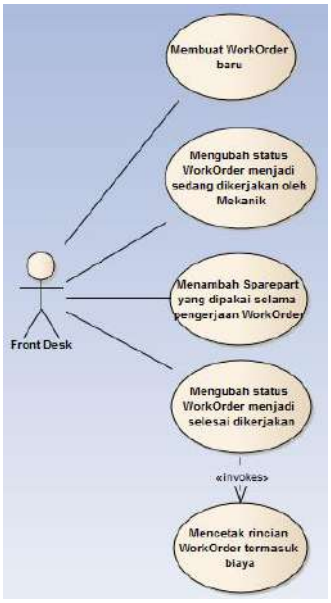
3. Membuat Use Case

Use case mendefinisikan **behavioral requirements** berdasarkan **functional requirement** (dan sumber lainnya). Berbeda dengan anjuran dari buku analisis sisfo lain, buku ini menyarankan untuk membuat use case dengan maksimal 2 paragraf! Tidak perlu mengikuti template yang detail! Sebuah use case yang panjang & detail malah akan memperlambat kita. Tim yang membuat use case bisa jadi akhirnya hanya mengisi form yang kosong tanpa banyak berpikir panjang (misalnya sekedar *copy paste)* sehingga proses membuat use case hanya sekedar ritual tanpa analisa mendalam.

Kalimat yang dipakai dalam use case harus berupa **kalimat aktif**, misalnya "pengguna men-klik tombol Login". **Kalimat pasif** seperti "Sistem menyediakan tombol Login" adalah ciri-ciri **functional requirement** dan bukan bagian dari use case.

Use case harus mengandung nama di **domain model**. Dengan demikian, saya bisa menghubungkan class-class yang akan dirancang dengan use case. Setiap halaman/layar yang direferensikan di dalam use case sebaiknya diberi nama yang konsisten, misalnya *Halaman Tambah Sparepart* atau *Screen TambahSparepart*.

Sebuah use case hampir mirip seperti dokumentasi sistem. Kita perlu menspesifikasikan seperti apa cara pakai sistem (termasuk respon sistem) sebelum sebuah sistem dibuat. Berikut ini adalah contoh use case diagram berdasarkan **functional requirement** di langkah 1 dan memakai **domain model** dari langkah 2:



Use Case Diagram

Sebuah use case selain memiliki **sunny-day scenario**, sebaiknya juga memiliki **rainy-day scenario** (apa yang akan terjadi bila sesuatu salah?) atau alternatif. Sebagai contoh, berikut ini contoh use case scenario untuk use case diagram di atas:

Membuat WorkOrder baru

Basic Scenario

Front Desk memilih Motor di Screen ListMotor dan men-klik tombol untuk membuat WorkOrder. Sistem akan membuat sebuah WorkOrder baru dengan status sedang antri.

Alternate Scenario

Motor belum terdaftar - Front Desk terlebih dahulu menambah Motor baru dengan men-klik tombol untuk menambah Motor baru sebelum mengerjakan langkah yang ada di Basic Scenario.

Motor sudah memiliki WorkOrder yang sedang antri - Sistem akan menampilkan pesan kesalahan.

Mengubah Status WorkOrder menjadi sedang dikerjakan oleh Mekanik

Basic Scenario

Front Desk memilih sebuah WorkOrder di Screen ListWorkOrder dan memilih menu untuk menandakan bahwa Workorder tersebut sedang dikerjakan. Sistem akan menampilkan Screen PengerjaanWorkOrder. Front Desk memilih nama Mekanik yang mengerjakan WorkOrder dan men-klik tombol untuk menyimpan perubahan. Sistem akan mengubah status WorkOrder menjadi sedang dikerjakan serta mencatat tanggal & jam mulai dikerjakan.

Alternate Scenario

Status WorkOrder yang dipilih bukan sedang antri - Sistem akan menampilkan pesan kesalahan.

Menambah Sparepart yang dipakai selama pengerjaan WorkOrder

Basic Scenario

Front Desk memilih sebuah WorkOrder di Screen ListWorkOrder dan memilih menu untuk menambah Sparepart di WorkOrder. Sistem akan menampilkan Screen TambahSparepart yang berisi daftar Sparepart untuk WorkOrder yang dipilih. Disini Front Desk akan mengisi data ItemSparepart dengan memilih Sparepart, memasukkan jumlah Sparepart, lalu men-klik tombol Tambah ItemSparepart. Sistem akan memperbaharui daftar ItemSparepart di layar.

Front Desk men-klik tombol Simpan untuk selesai. Sistem akan menyimpan perubahan sparepart pada WorkOrder terpilih.

Alternate Scenario

Terdapat lebih dari satu jenis Sparepart yang perlu ditambahkan - Front Desk kembali menambah data ItemSparepart. Setelah semua ItemSparepart selesai dimasukkan, Front Desk men-klik tombol Simpan di Screen TambahSparepart. Sistem akan menyimpan perubahan.

Status WorkOrder yang dipilih bukan sedang dikerjakan - Sistem akan menampilkan pesan kesalahan.

Mengubah status WorkOrder menjadi selesai dikerjakan

Basic Scenario

Front Desk memilih sebuah WorkOrder di Screen ListWorkOrder, lalu memilih menu untuk mengubah status WorkOrder menjadi selesai

dikerjakan. Sistem akan menampilkan dialog konfirmasi. Bila kasir menkonfirmasi, Sistem akan mengubah status WorkOrder tersebut menjadi selesai dikerjakan dan mencatat jam selesai dikerjakan. Front Desk kemudian mengerjakan use case "Mencetak rincian WorkOrder termasuk biaya".

Alternate Scenario

Status WorkOrder bukan sedang dikerjakan - Sistem akan menampilkan pesan kesalahan.

Mencetak rincian WorkOrder termasuk biaya

Basic Scenario

Front Desk memilih tombol untuk mencetak. Sistem kemudian mencetak detail WorkOrder ke printer. Detail WorkOrder yang dicetak meliputi tanggal, plat nomor motor, jam mulai dikerjakan, jam selesai dikerjakan, nama mekanik yang mengerjakan, rincian seluruh Sparepart yang dipakai (jumlah & harga eceran tertinggi Sparepart), ongkos servis dan total yang harus dibayar.

Alternate Scenario

Status WorkOrder bukan selesai dikerjakan - Sistem akan menampilkan pesan kesalahan.

4. Requirements Review

Pada saat melakukan analisa dalam membuat use case, saya menemukan hal yang masih kurang. Misalnya, saya perlu menambahkan class ItemSparepart pada **domain model**. Selain itu, pada beberapa situasi, saya bahkan bisa

menemukan ada use case yang masih kurang, misalnya use case "*Tambah Motor baru*".

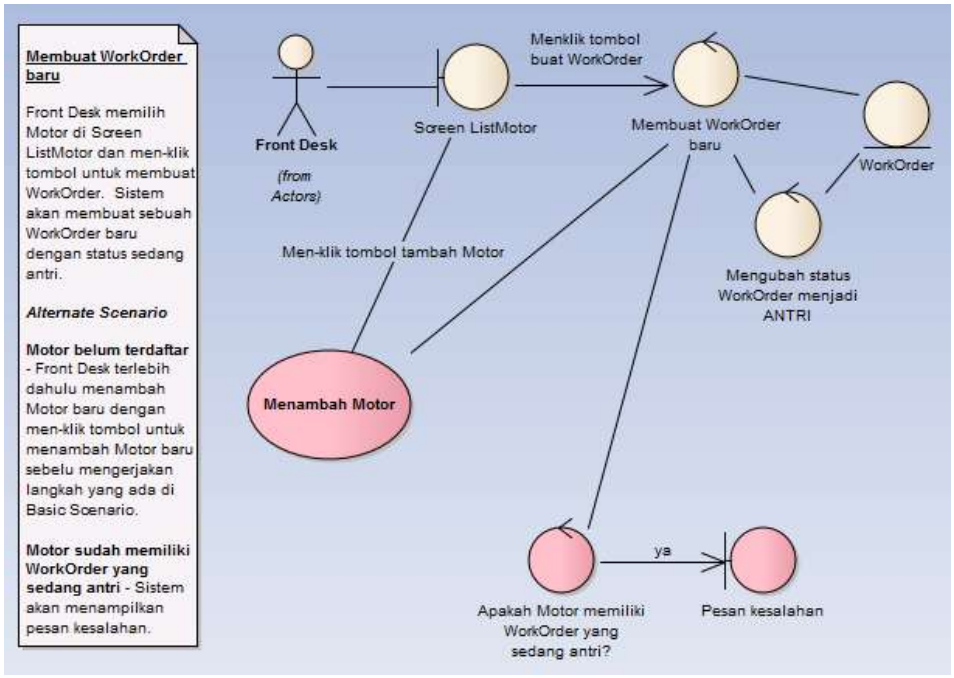
Pada langkah ini, saya kembali memastikan bahwa use case & **domain model** telah dibuat dengan baik. Pelanggan juga perlu dilibatkan untuk memastikan bahwa use case (**behavioral requirement**) & **functional requirement** sesuai dengan yang diharapkan. Ingatlah selalu bahwa bagian terpenting dari sebuah sistem bukanlah seberapa keren *design pattern* yang diterapkan di class diagram, tetapi sejauh mana sistem tersebut memberikan profit bagi penggunanya (memenuhi requirements).

5. Melakukan Robustness Analysis

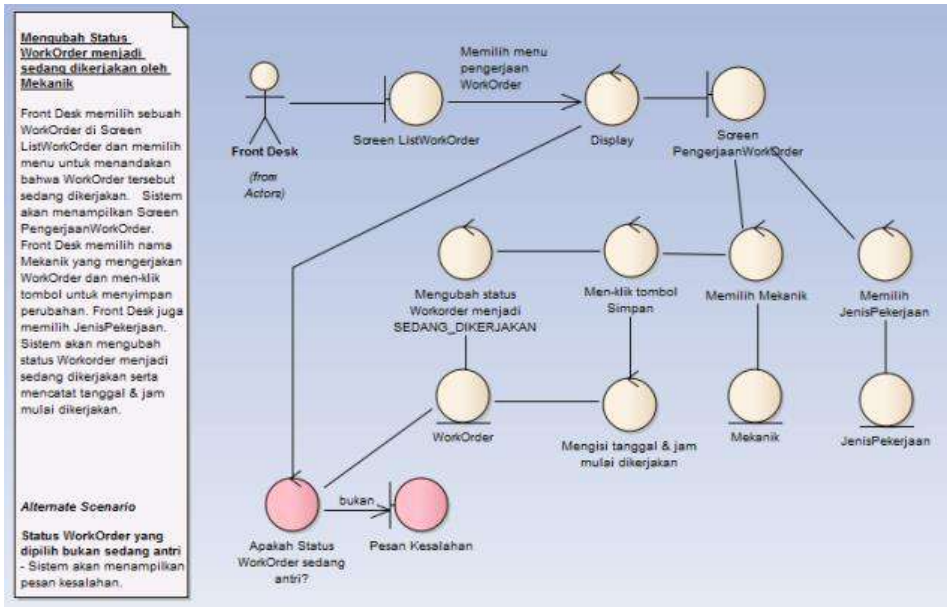
Analisis adalah memikirkan "apa" (what), sementara perancangan adalah memikirkan "bagaimana" (how). Salah satu alasan mahasiswa sering mengabaikan UML dan langsung terjun ke coding adalah celah yang cukup jauh antara analisis dan perancangan sehingga mereka memilih merancang secara eksperimental dengan langsung coding. Umumnya mereka berakhir dalam jebakan siklus perubahan "coding" terus menerus (guna memperbaiki rancangan). Padahal, perubahan "coding" adalah sesuatu yang sangat memakan waktu dan upaya bila dibandingkan dengan mengubah diagram UML.

Robustness analysis dipakai untuk menjembatani analisis dan perancangan. **Robustness analysis** harus diterapkan pada setiap use case yang ada. Pada Enterprise Architect, **robustness analysis** dapat digambarkan dengan menggunakan **Analysis Diagrams (**terdapat di kategori **Extended**).

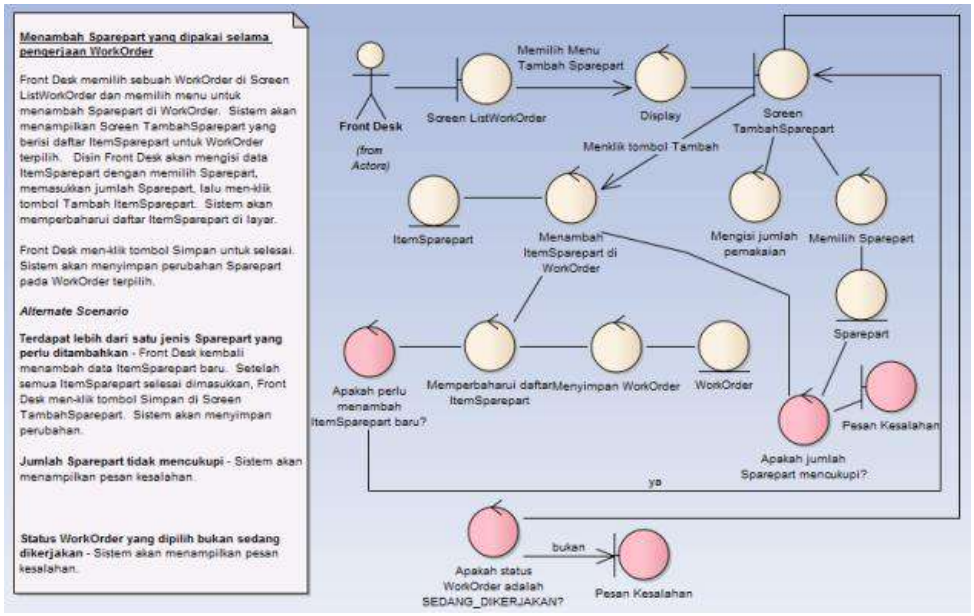
Berikut ini adalah contoh hasil **robustness analysis** untuk use case yang ada:



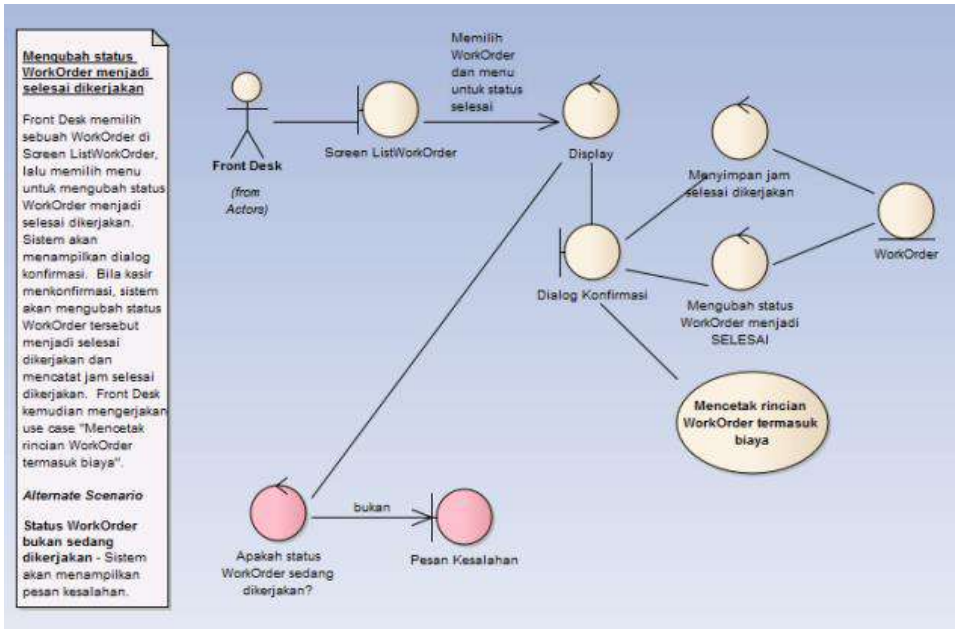
Analysis Diagram Untuk Use Case Membuat WorkOrder Baru



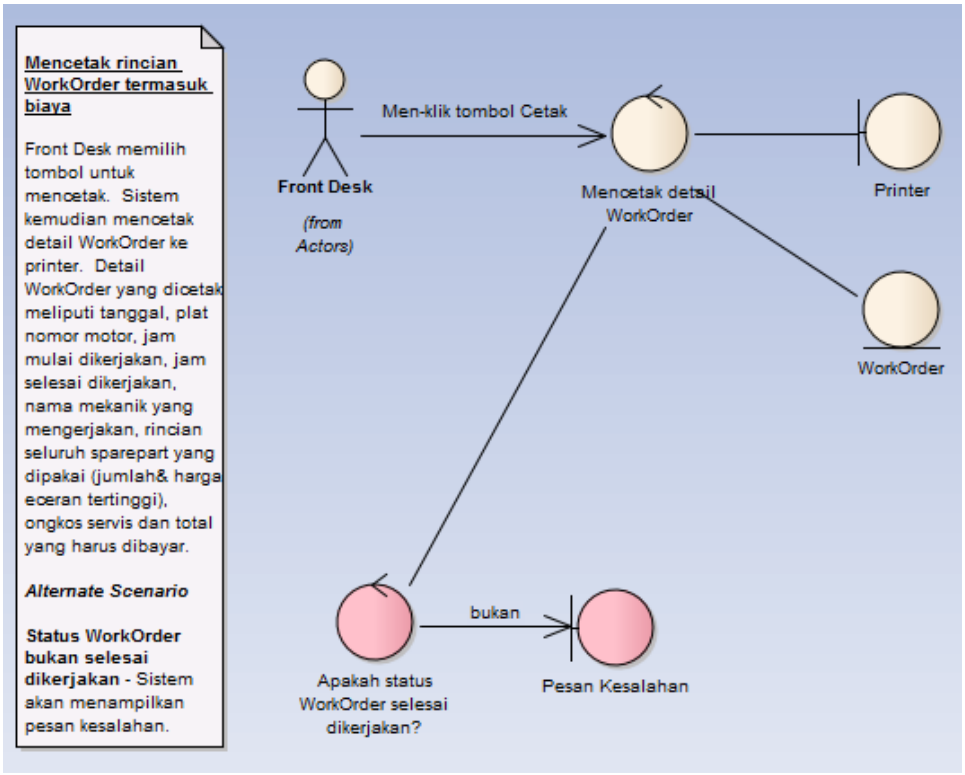
Analysis Diagram Untuk Use Case Mengubah Status WorkOrder Menjadi Sedang DIkerjakan Oleh Mekanik



Menambah Sparepart yang dipakai selama pengerjaan WorkOrder



Analysis Diagram Untuk Use Case Mengubah Status WorkOrder Menjadi Selesai Dikerjakan



Analysis Diagram Untuk Use Case Mencetak Rincian WorkOrder Termasuk Biaya

Semakin detail **robustness analysis**, maka semakin banyak hal yang kurang dari use case dan **domain model** yang akan ditemukan. Yup! Pada awalnya saya ragu kenapa saya harus menambah *control* "mengisi jumlah", "mengisi nama", dsb untuk setiap field yang ada. Tapi saya cukup terkejut saat menemukan dari hal sepele tersebut, saya menemukan beberapa alternate scenario yang kurang.

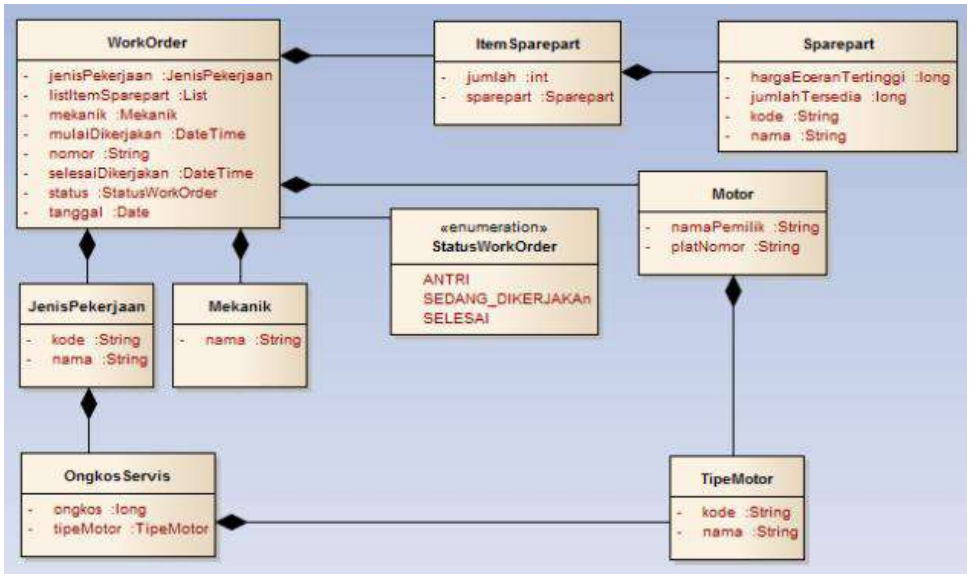
Sebagai contoh, saya menemukan bahwa saya lupa menambahkan alternate scenario "jumlah Sparepart tidak mencukupi" saat melakukan analisa

robustness pada use case “Menambah Sparepart yang dipakai selama pengerjaan WorkOrder”. Semakin cepat saya menyadari ada yang kurang, semakin baik! Idealnya adalah sebelum coding dilakukan. **Robustness analysis** adalah salah satu senjata yang ampuh untuk itu.

Saya juga menemukan bahwa pada use case “Mengubah Status WorkOrder menjadi sedang dikerjakan oleh Mekanik”, saya lupa menuliskan bahwa Front Desk officer juga perlu memilih JenisPekerjaan. Saya perlu segera mengubah teks use case tersebut.

Selain itu, terkadang saya juga dapat menemukan ada class yang kurang pada domain model. Misalnya, dari hasil **robutsness analysis**, terlihat bahwa saya perlu menambahkan class Mekanik di domain model.

Pada tahap ini, saya juga perlu mengisi domain model dengan atribut, seperti yang terlihat pada gambar berikut ini:



Domain Model Yang Telah Memiliki Atribut

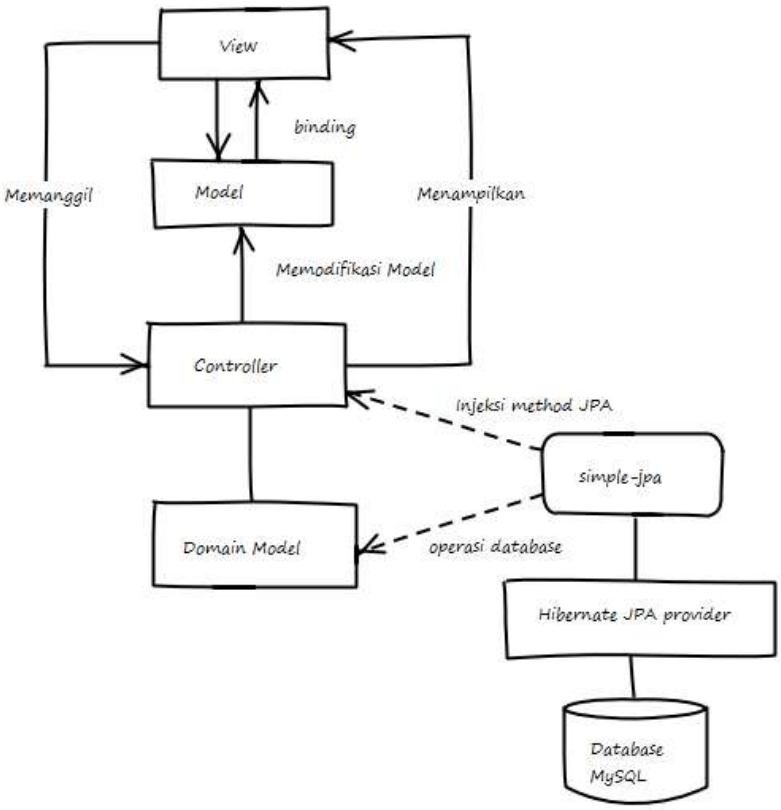
6. Preliminary Design Review

Kembali lagi seluruh tim melakukan review dan memastikan bahwa semua yang telah dibuat sesuai dengan requirement. Ini adalah langkah terakhir dimana pelanggan (stackholder) terlibat! Hal ini karena langkah berikutnya melibatkan proses techincal. Akan berbahaya bila membiarkan pelanggan yang non-technical atau *semi*-technical mengambil keputusan untuk hal-hal yang bersifat teknis (misalnya framework atau database yang dipakai). Walaupun demikian, pelanggan boleh memberikan komentar mengenai tampilan.

Setelah langkah ini, tidak ada lagi perubahan requirement. Lalu bagaimana bila pelanggan ingin menambah requirement? Buat sebuah rilis atau milestone baru dengan kembali lagi ke langkah pertama di atas.

7. Menentukan Technical Architecture

Tentukan framework apa yang akan dipakai. Sebagai contoh, saya akan membuat sebuah aplikasi desktop dengan Griffon. Pola arsitektur yang dipakai menyerupai Model View ViewModel (MVVM) dimana terdapat perbedaan antara **domain model** dan **view model**. Saya juga mengasumsikan penggunaan sebuah plugin fiktif yang dirujuk sebagai *simple-jpa*. Gambar berikut ini memperlihatkan contoh arsitektur yang dipakai:



Gambaran Technical Architecture

8. Membuat Sequence Diagram

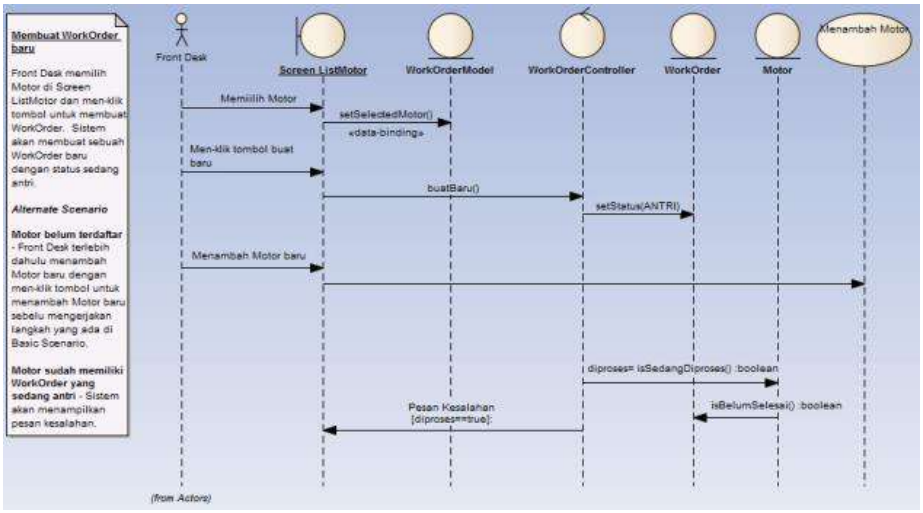
Object oriented pada dasarnya adalah menggabungkan antara data dan operasi ke dalam sebuah entitas. Saat ini, **domain model** baru berisi data. Oleh sebab itu, dibutuhkan sebuah upaya untuk menemukan operasi untuk **domain model**. Caranya adalah dengan memakai **sequence diagram**.

Saat membuat **sequence diagram**, sertakan juga elemen dalam arsitektur teknis/framework. Misalnya penggunaan MVC akan menyebabkan ada class baru seperti *controller*. Yup! Penggunaan MVC akan membuat operasi tersebar ke controller. Hal ini sering dikritik karena bukan pendekatan OOP melainkan kembali ke zaman prosedural. Baca buku **Object Design: Roles,**

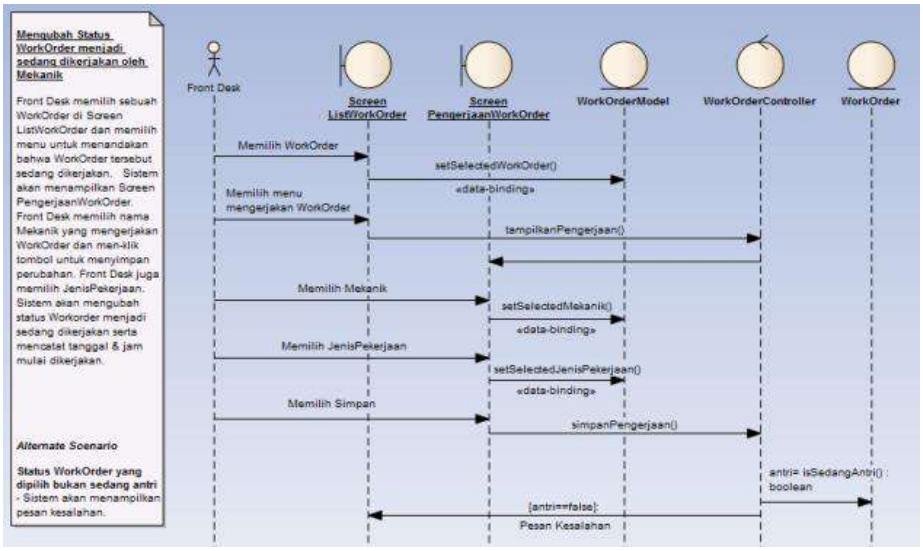
Responsibilities, and Collaborations untuk pendekatan yang OOP, akan tetapi jangan lupa kalau kita dibatasi oleh framework yang dipakai (ehem, seharusnya bukan framework yang membatasi kita, melainkan kita yang tegas dalam memilih framework).

Selama membuat **sequence diagram**, ingatlah selalu bahwa tujuannya adalah menemukan operasi (**behavior**) untuk setiap class yang ada, bukan menunjukkan step-by-step operasi secara detail! Untuk menjaga agar tidak tersesat menjadi membuat *flow-chart* yang detail, jangan memikirkan **focus of control** (matikan saja fitur tersebut!).

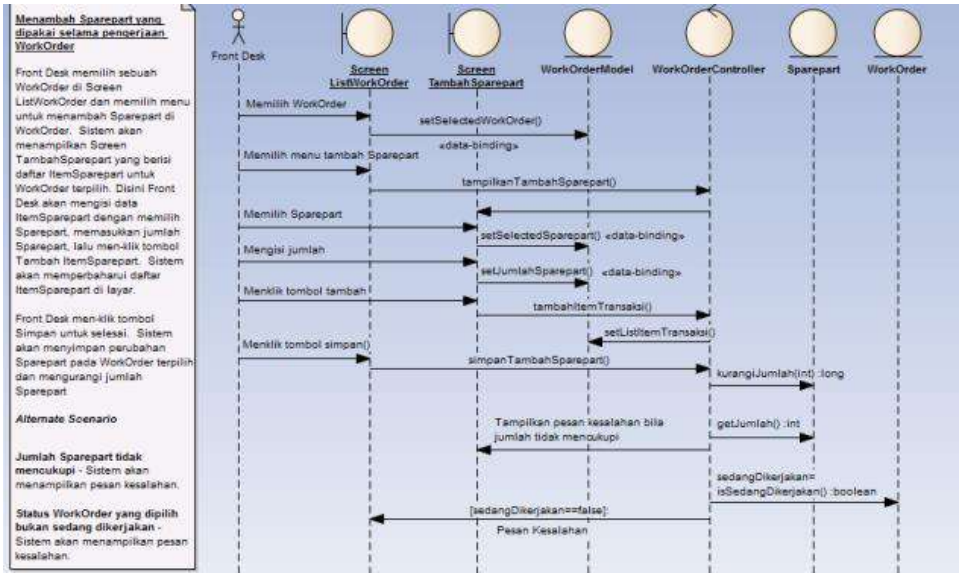
Sequence diagram dibuat untuk setiap use case yang ada, berdasarkan hasil **robustness anaylsis**. Gambar berikut ini memperlihat contoh **sequence diagram** yang dihasilkan (agar sederhana, operasi penyimpanan data oleh *simple-jpa* tidak ditampilkan):



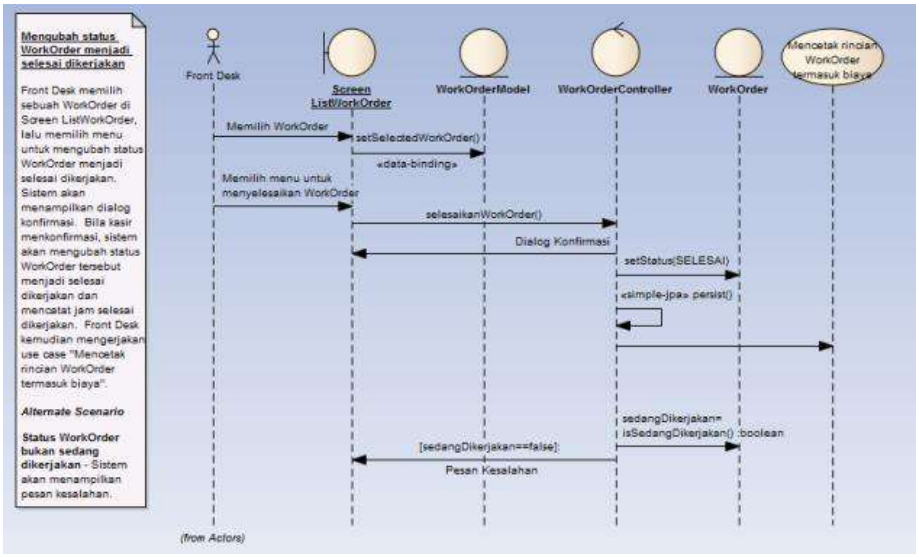
Sequence Diagram Untuk Membuat WorkOrder Baru



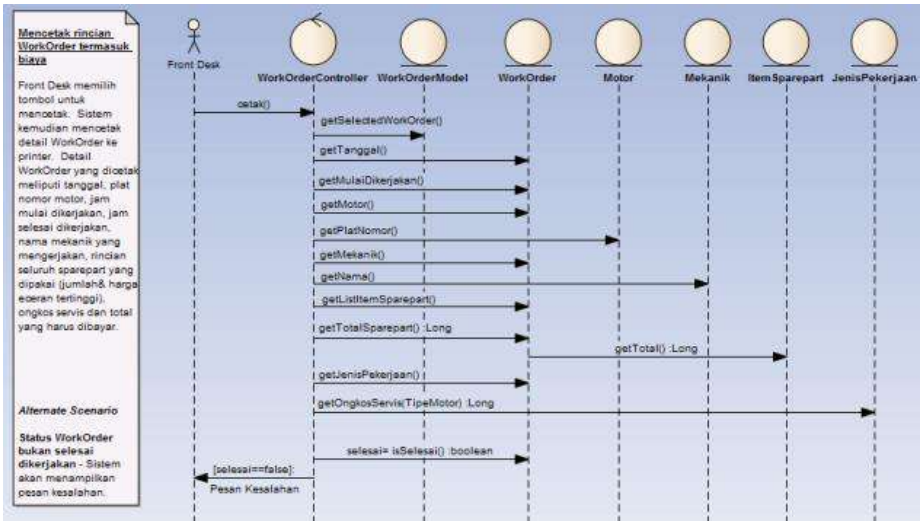
Sequence Diagram Untuk Mengubah Status WorkOrder Menjadi Sedang Dikerjakan Oleh Mekanik



Sequence Diagram Untuk Menambah Sparepart Yang Dipakai Selama Pengerjaan WorkOrder



Sequence Diagram Untuk Mengubah Status WorkOrderMenjadi Selesai Dikerjakan

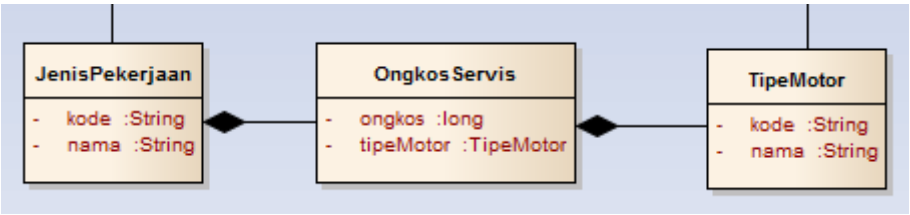


Sequence Diagram Untuk Mencetak Rincian WorkOrder Termasuk Biaya

Proses analisa yang berulang kali lagi-lagi membantu saya menemukan kekurangan. Saya selama ini ternyata lupa bahwa pada use case "Menambah Sparepart yang dipakai selaman pengerjaan WorkOrder", sistem harus

mengurangi jumlah stok Sparepart bila pengguna men-klik tombol simpan. Oleh sebab itu, saya segera memperbaharui teks use case.

Selain itu, saya juga menemukan sebuah kesalahan yang saya buat dari awal dan tidak terdeteksi hingga sekarang, yang berhubungan dengan class OngkosServis. Gambar berikut ini memperlihatkan perancangan awal class tersebut:



Kesalahan Rancangan Domain Model Akibat Berfokus Pada Penyimpanan Data

Bila membuat ERD atau design tabel, ini adalah sesuatu yang dapat diterima (ongkos disimpan dalam tabel yang mewakili hubungan one-to-many dari JenisPekerjaan ke TipeMotor). Tetapi, bila diterapkan ke dalam domain model, maka akan terjadi kejanggalan saat saya memakai domain model tersebut di sequence diagram. Untuk memperoleh ongkos servis, apa saya harus membuat instance objek OngkosServis baru? Bila diterapkan ke coding, maka ini berarti untuk memperoleh ongkos servis, saya harus selalu melakukan query JPQL yang kira-kira terlihat seperti berikut ini:

TipeMotor tipeMotor = model.selectedTipeMotor

JenisPekerjaan jenisPekerjaan = model.selectedJenisPekerjaan

Query query = em.createQuery ("SELECT o.harga FROM OngkosServis o WHERE " + "o.tipeMotor = :tipeMotor AND o.jenisPekerjaan = :jenisPekerjaan") query.setParameter ("tipeMotor", tipeMotor)

query.setParameter("jenisPekerjaan", jenisPekerjaan) Long ongkos = query.getSingleResult()

Terlihat ada yang tidak beres! Bukankah OOP harus intuitive & bisa dipakai dengan mudah? Kenapa tiba2 harus melakukan query secara eksplisit untuk memperoleh sebuah ongkos servis? Selain itu, class OngkosServis tidak pernah dipakai secara langsung di kode program, hanya dipakai di query saja!

Oleh sebab itu, saya memperbaikinya dengan membuang class OngkosServis, dan menambahkan atribut ongkosServis di class JenisPekerjaan dengan tipe data berupa Map<TipeMotor,Long>. Dengan demikian, saya bisa memakainya seperti berikut ini:

TipeMotor tipeMotor = model.selectedTipeMotor

JenisPekerjaan jenisPekerjaan = model.selectedJenisPekerjaan

Long ongkos = jenisPekerjaan.getOngkosServis(tipeMotor)

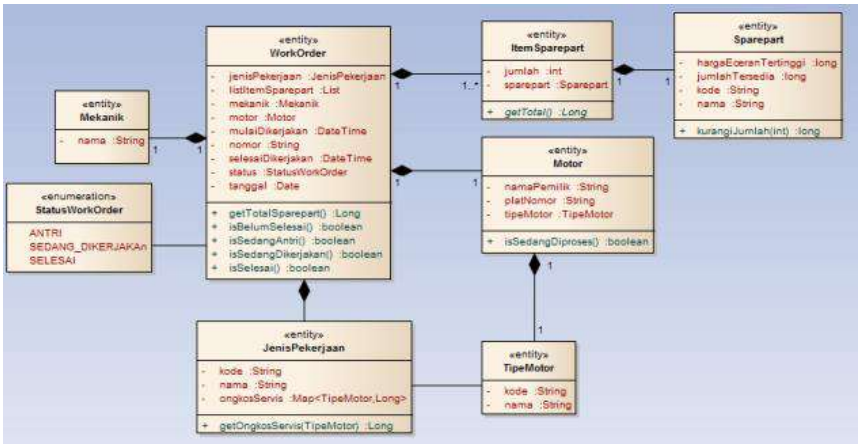
Cara di-atas jauh lebih intuitive dan mudah dipahami. Query database tetap terjadi, tetapi kali ini secara implisit (secara otomatis) oleh Hibernate tanpa campur tangan developer.

Ini adalah alasan kenapa saya selalu memberikan pesan pada mahasiswa agar tidak memikirkan proses penyimpanan data saat membuat **domain model**. Jangan menyamakan **domain model** dan ERD. Pada saat merancang **domain model**, pikirkan bagaimana class-class yang ada akan saling berinteraksi dan dipakai oleh developer. Bahkan bila tidak memakai ORM seperti Hibernate, tetap jangan memikirkan bagaimana penyimpanan data di **domain model**, melainkan buat ERD terpisah untuk dipakai oleh persistence layer (DAO).

9. Critical Design Review

Kembali melakukan review untuk memastikan bahwa tidak ada yang kurang pada **sequence diagram**. Pastikan bahwa setiap class yang ada telah memiliki atribut dan operasi yang didefinisikan secara lengkap (memiliki nama, tipe data, parameter, dsb).

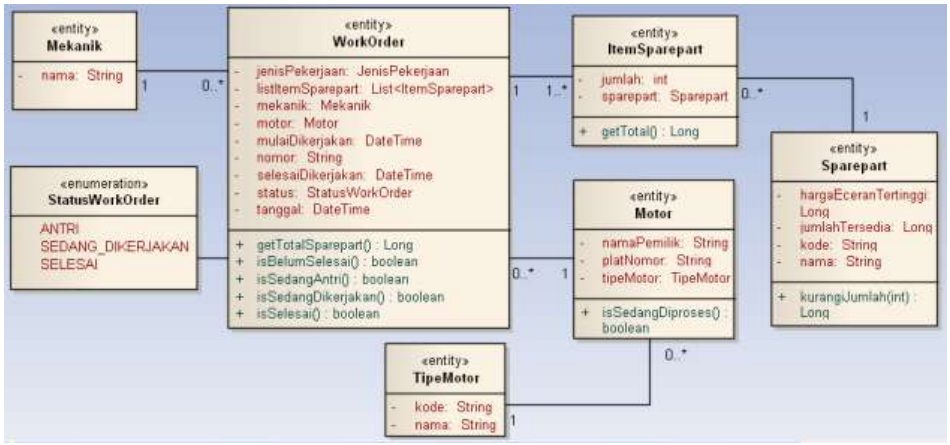
Berikut ini adalah contoh hasil domain model yang telah dilengkapi dengan operasi dan multiplicity:



Domain Model Yang Telah Lengkap

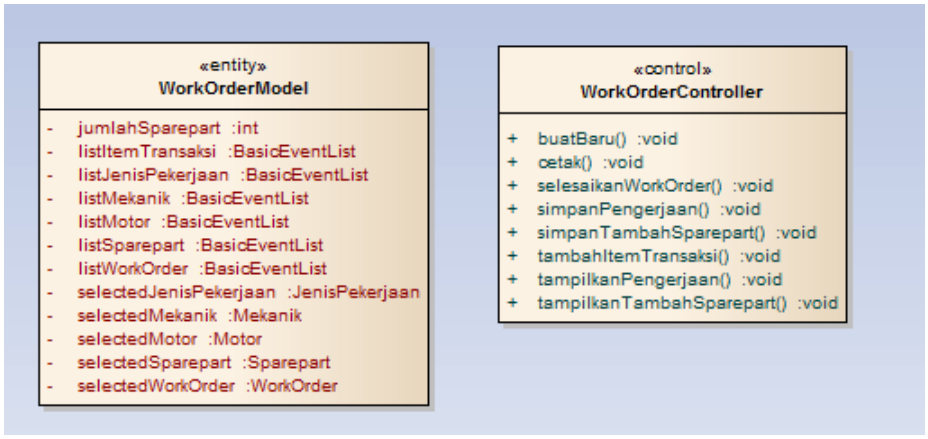
Getter dan setter tidak perlu ditampilkan karena hanya akan membuat class diagram terlihat ‘penuh’.

Versi *bidirectional*-nya akan terlihat seperti pada gambar berikut ini:



Domain Model Yang Telah Lengkap (Versi Bidirectional)

Selain **domain model**, saya juga menemukan terdapat class-class lain yang dihasilkan yang berkaitan dengan penggunaan framework, yang terlihat seperti pada gambar berikut ini:



Class pembantu untuk framework

10. Coding

Disini developer berperan mengubah rancangan (*design*) menjadi kode program. Karena semua telah direncanakan dan dipikirkan sebelumnya, maka proses coding dapat dianggap sebagai sebuah pembuktian (test) bahwa

rancangan yang dibuat sudah benar. Terkadang terdapat beberapa hal yang lolos dari perancangan dan baru terungkap saat coding; pada kasus tersebut, perubahan pada rancangan harus segera dilakukan sehingga kode program dan rancangan bisa tetap sinkron.

Bila pembuatan kode program tiba-tiba menjadi tidak terkendali (pada kasus skripsi, mahasiswa tiba-tiba merasa seolah otaknya hendak meledak dan jadi malas coding), maka ada beberapa kemungkinan:

- Hasil rancangan tidak bagus.
- Programmer tidak mengikuti hasil rancangan yang bagus dan mengerjakannya sesuka hatinya.
- Programmer tidak diikutsertakan dalam proses perancangan
- Mahasiswa tidak mau pikir panjang/hanya copy paste di bab analisa & perancangan, dalam pikirannya mau segera fokus ke bab implementasi dan kode program;
- Mahasiswa hanya memikirkan bab analisa & perancangan, tidak mau peduli dampaknya pada saat membuat kode program nanti.

DAFTAR PUSTAKA

HM., Yogiyanto, Analisis dan Disain Sistem Informasi : Pendekatan Terstruktur, Penerbit Andi Offset, Yogyakarta, 2010.

McLeod, Raymond, Management Information System, 9th ed., Prentice Hall, New Jersey, 2012.

Martin, Merle P., Analysis And Design of Business Information Sytems, Mac Millan Publishing Company, New York, 1991.

McNurlin, Barbara C,; Sparague, Ralph H Jr., Information Systems Management in Practice, 4th ed., Prentice Hall, New Jersey, 1998.

Booch, Grady. 1998. Object-oriented analysis and design with applications 2nd edition. Addison Wesley.

Grady Booch, James Rumbaugh and Ivar Jacobson, The UML User‟s Guide, 1st Edition, Addison and Wesley, 1998.

Bowman, Kevin. 2004. System Analysis: A Beginner‟s Guide. Palgrave

Macmillan. Knudson, Joan dan Ira Bitz. 1991. Project Management. Amacom.

Langer, Arthur M. 2008. Analysis and Design of Information Systems 3rd edition. Springer.

Rumbaugh, James dkk. 1999. The Unified Modeling Language Reference Manual. Addison-Wesley.

Simon Bennet, Steve McRobb, Ray Farmer, Object Oriented Systems Analysis and Design Using UML 2nd, McGraw Hill, 2002.

Cara Mudah dan Cepat Belajar Pengembangan Sistem Informasi

Budi Hartono. M.Kom

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-6141-07-6 (PDF)