

BAB I

Konsep Dasar Sistem

1.1 Pengertian Sistem

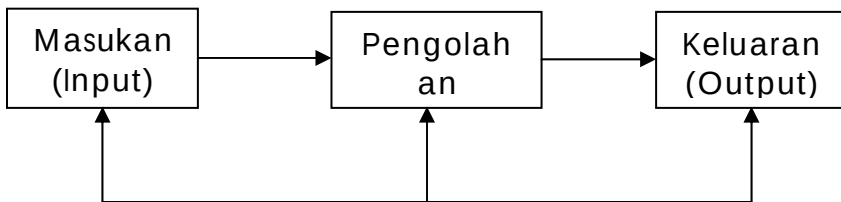
Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Disini akan diberikan beberapa definisi sistem secara umum:

- Kumpulan dari bagian-bagian yang bekerja bersama-sama untuk mencapai tujuan yang sama
 - Contoh
 - Sistem tatasurya
 - Sistem pencernaan
 - Sistem Transportasi umum
 - Sistem Otomotif
 - Sistem Komputer
 - Sistem Informasi
- Sekumpulan dari objek-objek yang saling berelasi dan berinteraksi dan hubungan antar objek bisa dilihat sbg 1 kesatuan yang dirancang untuk mencapai 1 tujuan

Dengan demikian secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling teroganisasi, saling berinteraksi dan saling bergantung sama lain. *Murdick dan Ross (1993)* mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Sedangkan definisi sistem dalam kamus *Webster's Unbrided*

adalah elemen-elemen yang saling berhubungan membentuk satu kesatuan atau organisasi.

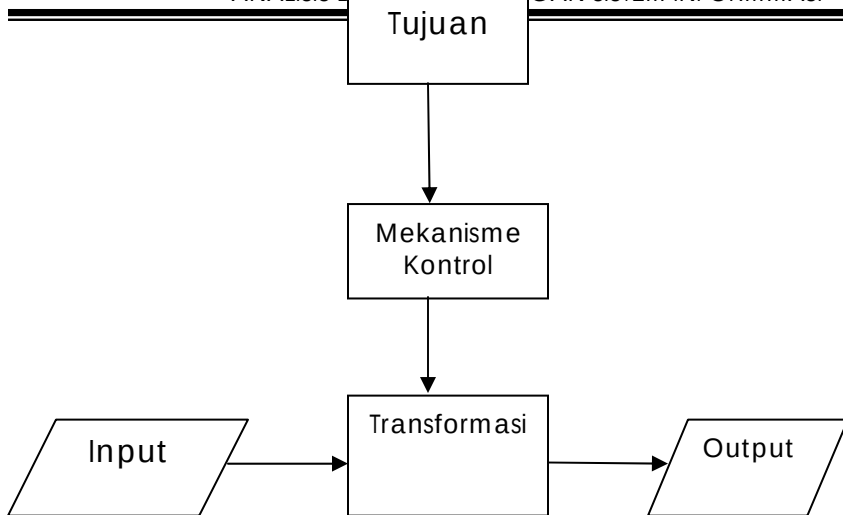
Scott (1996) mengatakan sistem terdiri dari unsur-unsur seperti masukan (*input*) , pengolahan (*processing*) , serta keluaran (*output*) . Ciri pokok sistem menurut Gapsert ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan, terdiri atas unsur-unsur, ditandai dengan saling berhubungan dan mempunyai satu fungsi atau tujuan utama.



Gambar. 1.1 Model sistem

Gambar diatas menunjukkan bahwa sistem atau pendekatan sistem minimal harus mempunyai empat komponen, yakni masukan, pengolahan, keluaran da, balikan atau control.

Sementara *Mc. Leod (1995)* mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumberdaya mengalir dari elemen output dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan mekanisme control. Untuk lebih jelasnya elemen sistem tersebut dapat digambarkan dengan model sebagai berikut :



Gambar. 1.2. *Model hubungan elemen-elemen sistem*

Banyak ahli mengajukan konsep sistem dengan deskripsi yang berbeda namun pada prinsipnya hampir sama dengan konsep dasar sistem umumnya. *Schronderberg (1971)* dalam *Suradinata (1996)* secara ringkas menjelaskan bahwa sistem adalah

1. Komponen-komponen sistem saling berhubungan satu sama lainya.
2. Suatu keseluruhan tanpa memisahkan komponen pembentukanya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki input dan output yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang merubah input menjadi output.
6. Menunjukan adanya entropi
7. terdapat aturan
8. Terdapat subsistem yang lebih kecil.
9. terdapat deferensiasi antar subsistem.
10. Terdapat tujuan yang sama meskipun mulainya berbeda.

1.2. Karakteristik Sistem.

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut ini karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya.

1. Batasan (*boundary*) : Penggambaran dari suatu elemen atau unsur mana yang termasuk didalam sistem dan mana yang diluar sistem.
2. Lingkungan (*environment*) : Segala sesuatu diluar sistem, lingkungan yang menyediakan asumsi, kendala dan input terhadap suatu sistem
3. Masukan (*input*) : Sumberdaya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*) : Sumber daya atau produk (informasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*) : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*) : Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) : Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

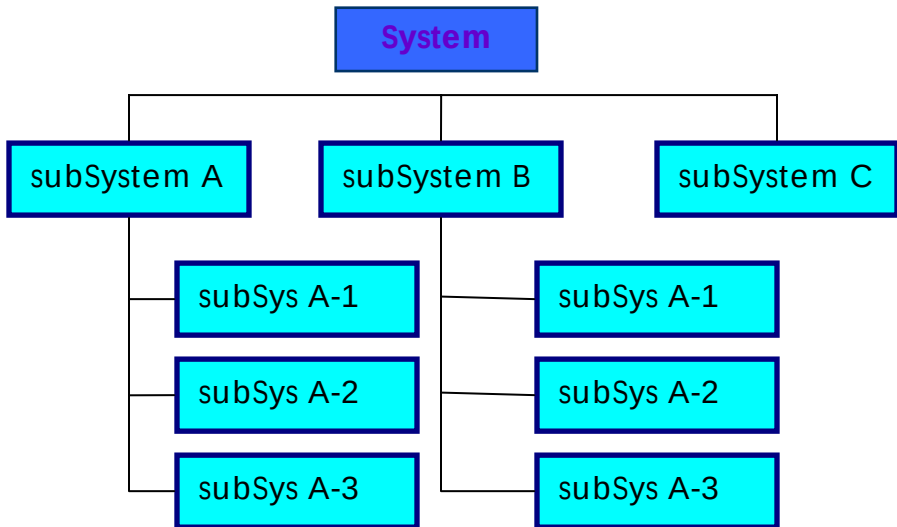
1.3 Pengertian Subsistem

Suatu sistem yang kompleks biasanya tersusun atas beberapa subsistem. Subsistem bisa dijelaskan sebagai sebuah sistem dalam sistem yang lebih besar. Sebagai contoh :

Automobile adalah sistem yang terdiri dari beberapa subsistem:

- Sistem mesin
- Sistem Body
- Sistem Roda
- Setiap sub sistem bisa terdiri dari beberapa sub-sub --systems.
 - Sistem mesin: sistem karburator, sistem generator, sistem bahan bakar dan lain-lain

Untuk lebih jelasnya bisa dilihat pada bagan berikut ini:



Gambar 1.3 Gambaran subsistem dalam sistem

1.4 Sistem Yang Buruk

Untuk menghindari pengembangan suatu sistem yang buruk maka perlu diketahui beberapa ciri-ciri dari sistem yang buruk:

- Tidak memenuhi kebutuhan user
- Performance buruk
- Reliabilitas rendah
- Kegunaan rendah
- Contoh-contoh kesulitan:
 - Tidak terjadwal
 - Tidak ada rencana anggaran
 - Bisa jalan = 100% over budget atau jadwal

1.5 Beberapa Konsep sistem yang penting

Untuk lebih mudah memahami pengertian sistem dan sistem informasi lebih jauh maka perlu diingat beberapa konsep yang penting dalam pengembangan sistem yaitu :

1. Decomposition

- Proses pembagian sistem ke dalam komponen-komponen yang lebih kecil
- Memungkinkan sistem analisis untuk:
 - Memecah sistem menjadi bagian-bagian (sub sistem) yang lebih kecil sehingga mudah di-manage
 - Fokus pada 1 area pada 1 waktu
- Bisa membangun komponen-komponen secara paralel

● Modularity

- Proses membagi sistem menjadi modul-modul yang relatif sama ukurannya
- Modul menyederhanakan desain sistem

● Coupling

- Subsystems yang saling bergantung 1 sama lain di-couple (dipasangkan)

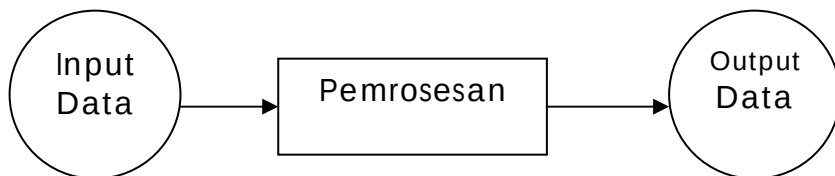
● Cohesion

- Diperluas ke sub-sub sistem yang berdiri sendiri

1.6 Pengertian Sistem Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang (*Davis, 1995*). *McLeod (1995)* mengatakan bahwa informasi adalah data yang telah diproses, atau data yang memiliki arti.

Akhirnya Sistem Informasi Manajemen (SIM) dapat didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya (*Kertahadi, 1995*). Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan dan menyajikan sinergi organisasi pada proses (*Murdick dan Ross, 1993*). Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output – IPO*) dapat dilihat pada gambar berikut.



Gambar. 1.4. *Konsep Sistem Informasi.*

1.6.1. Komponen Sistem Informasi

Stair (1992) menjelaskan bahwa sistem informasi berbasis komputer (CBIS) dalam suatu organisasi terdiri dari komponen-komponen berikut :

- **Hardware**, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukan data, memproses data dan keluaran data.
- **Software**, yaitu program dan instruksi yang diberikan kekomputer.
- **Database**, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
- **Telekomunikasi**, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama kedalam suatu jaringan kerja yang efektif.
- **Manusia**, yaitu personil dari sistem informasi, meliputi manajer, analis, programmer, operator dan bertanggungjawab terhadap perawatan sistem.

Prosedur, yakni tata cara yang meliputi strategi, kebijakan, metode dan peraturan-peraturan dalam menggunakan sistem informasi berbasis komputer.

Pendapat *Burch dan Grudnistki (1986)*, sistem informasi terdiri dari komponen-komponen diatas disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok mkeluaran (*output block*), blok teknologi (*technology block*) dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. **Blok Masukan.** Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan yang dapat berupa dokumen-dokumen dasar.
2. **Blok Model.** Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. **Blok Keluaran.** Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.
4. **Blok Teknologi.** Teknologi merupakan kotakalat (tool-box) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.
5. **Blok Basis Data.** Basis Data (Data Base) merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. **Blok Kendali.** 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.

Sedangkan menurut pendapat *Davis (1995)* sistem informasi manajemen terdiri dari elemen-elemen berikut :

1. Perangkat keras komputer (*hardware*).
2. Perangkat Lunak (*software*), yang terdiri dari perangkat lunak sistem umum, perangkat lunak terapan, program aplikasi.
3. Database.
4. Prosedur.
5. Petugas pengoperasian.
- 6.

1.6.2 Computer Based Information System

Istilah Computer Based Information Sistem (CBIS), sebenarnya mengacu kepada sistem informasi yang dikembangkan berbasis teknologi komputer.

**Computer-based Information System = Hardware +
Software + People + Procedures + Information**

Dalam modul ini, CBIS selanjutnya akan disebut sebagai sistem informasi saja.

1.7 Tipe-tipe sistem informasi

CBIS biasanya dibedakan menjadi beberapa tipe aplikasi, yaitu :

- **Transaction Processing Systems (TPS)**
- **Management Information Systems (MIS)**
- **Decision Support Systems (DSS)**
- **Expert System and Artificial Intelligence (ES &AI)**

1.7.1 Transaction Processing System

TPS adalah sistem informasi terkomputerisasi yang dikembangkan untuk memproses sejumlah besar data untuk transaksi bisnis rutin.

- 1 Mengotomasi penanganan data-data aktifitas bisnis dan transaksi. Yang bisa dianggap sebagai kejadian diskrit dalam kehidupan organisasi :
- 2 Data setiap transaksi ditangkap
- 3 Transaksi di verifikasi untuk diterima atau ditolak
- 4 Transaksi yang telah di validasi disimpan untuk pengumpulan data berikutnya.
- 5 Laporan bisa dihasilkan untuk menyediakan rangkuman dari setiap transaksi
- 6 Transaksi bisa dipindah dari 1 proses ke proses yang lainnya untuk menangani seluruh aspek bisnis

1.7.2 Management Information System

Management Information System (MIS) atau Sistem informasi Manajemen adalah sebuah sistem informasi pada level manajemen yang berfungsi untuk membantu perencanaan, pengendalian dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu. SIM mengambil data mentah dari TPS dan mengubahnya menjadi kumpulan data yang lebih berarti yang dibutuhkan manager untuk menjalankan tanggung jawabnya. Untuk mengembangkan suatu SIM diperlukan pemahaman yang baik tentang informasi apa saja yang dibutuhkan manajer dan bagaimana mereka menggunakan informasi tersebut.

1.7.3 Decision Support System

Sistem informasi pada level manajemen dari suatu organisasi yang mengkombinasikan data dan model analisa canggih atau peralatan data analisis untuk mendukung pengambilan yang semi terstruktur dan tidak terstruktur. DSS dirancang untuk membantu pengambilan keputusan organisasional. DSS biasanya tersusun atas :

- Basis Data (bisa diekstraksi dari TPS/MIS)
- Model grafis atau Matematis yang digunakan untuk proses bisnis
- Use interface yang digunakan oleh user untuk berkomunikasi dengan DSS

1.7.4 Expert System

Representasi pengetahuan yang menggambarkan cara seorang ahli dalam mendekati suatu masalah. ES lebih berpusat pada bagaimana mengkodekan dan memanipulasi pengetahuan daripada informasi (misalnya aturan if...then). Biasanya ES bekerja sebagai berikut:

- User berkomunikasi dengan sistem menggunakan interaktif dialog
- ES menanyakan pertanyaan (yang akan ditanyakan seorang pakar), dan pengguna memberikan jawaban.
- Jawaban digunakan untuk menentukan aturan mana yang dipakai, dan ES sistem menyediakan rekomendasi berdasarkan aturan yang telah disimpan.
- Seorang knowledge engineer bertanggung jawab pada bagaimana melakukan akuisisi pengetahuan, sama seperti seorang analis tetapi dilatih untuk menggunakan teknik yang berbeda.

BAB 2

Stake Holder dalam Sistem Informasi

2.1 Stake Holder

Stake Holder adalah orang yang memiliki kepentingan tertentu pada suatu kegiatan bisnis. Di dalam pengembangan sebuah sistem informasi stake holder dapat dibedakan menjadi:

- Manager SI
- System analyst pada pengembangan sistem
- Programmer dalam pengembangan sistem
- End user dalam pengembangan sistem
- Supporting end user
- Business manager
- Teknisi SI lainnya

2.1.1 Manager SI

Manager dalam departemen Sistem informasi memiliki peranan secara langsung dalam proses pengembangan sistem jika organisasi yang ditanganinya berskala kecil. Manager SI berperan dalam mengalokasikan dan mengawasi proyek pengembangan sistem daripada terlibat langsung dalam proses pengembangan sistem. Ada beberapa manager SI pada departemen SI yang berskala besar:

- Manager untuk keseluruhan departemen SI biasa disebut sebagai Chief Information Officer dan berada dibawah president atau direktur perusahaan.

- Setiap divisi dalam departemen SI juga memiliki seorang manager misalnya manager pengembangan SI, Manager operasi ,manager programmer SI dan lain-lain

2.1.2 Systems Analysts

Sistem analis merupakan individu kunci dalam proses pengembangan sistem. Sistem analis mempelajari masalah dan kebutuhan dari organisasi untuk menentukan bagaimana orang, data, proses, komunikasi dan teknologi informasi dapat meningkatkan pencapaian bisnis. Seorang sistem analis juga merupakan orang yang paling bertanggung jawab pada proses analisa dan perancangan sistem informasi.

Seorang sistem analis yang sukses harus memiliki beberapa skill.

Keahlian analisa

- Memahami organisasi
- Keahlian memecahkan masalah
- Pemahaman sistem, untuk melihat organisasi dan sistem informasi sebagai sebuah sistem.

Keahlian teknis

- Memahami potensi dan limitasi dari suatu teknologi

Keahlian Managerial

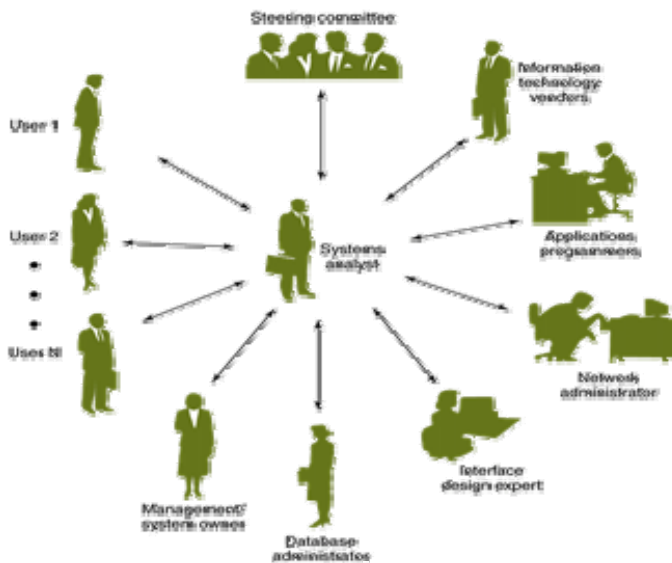
- Kemampuan untuk mengatur proyek, sumber daya resiko dan perubahan.

Interpersonal skills

- Kemampuan untuk berkomunikasi secara aktif baik tertulis maupun lisan
- Sangat membantu untuk komunikasi dengan end user, sistem analis maupun programmer

Adapun tanggung jawab dari seorang sistem analis meliputi :

1. Pengambilan data yang efektif dari sumber bisnis
2. Aliran data menuju ke komputer
3. Pemrosesan dan penyimpanan data dengan komputer
4. Aliran dari informasi yang berguna kembali ke proses bisnis dan penggunaanya



Gambar 2.1 Posisi sistem analis di antara stack holder yang lain

2.1.3 Programmer

Programmer mengubah Spesifikasi yang diberikan oleh sistem analis ke dalam instruksi yang bisa dijalankan oleh komputer. Langkah mengubah ke dalam kode yang bisa dijalankan komputer ini disebut coding. Code generator telah dikembangkan untuk menghasilkan kode dari spesifikasi yang telah dibuat, menghemat waktu dan biaya. Tujuan dari penggunaan CASE (Computer Aided Software Engineering) adalah untuk menyediakan beberapa code generator yang secara otomatis menghasilkan 90% atau lebih dari spesifikasi sistem normal yang diberikan oleh programmer secara normal.

2.1.4 Business manager

Kelompok lain dalam pengembangan sistem adalah manajer bisnis misalnya kepala bagian atau kepala departemen atau eksekutif perusahaan. Manajer-manajer ini penting karena mereka memiliki kekuatan pendanaan pengembangan sistem dan mengalokasikan sumber daya yang diperlukan untuk keberhasilan proyek.

2.1.5 Teknisi lainnya

Masih banyak lagi teknisi lain yang terlibat dalam pengembangan sistem diantaranya:

- database administrator
- Ahli network dan telekomunikasi

BAB 3

Analisa dan Perancangan Sistem

3.1 Pengertian Analisa dan Perancangan Sistem

Analisa sistem didefinisikan sebagai bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sedangkan sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Sehingga Analisa dan desain sistem informasi (ANSI) bisa didefinisikan sebagai: Proses organisasional kompleks dimana sistem informasi berbasis komputer diimplementasikan. Sehingga bisa diringkas sebagai berikut:

Analysis: mendefinisikan masalah

— **From requirements to specification**

Design: memecahkan masalah

— **From specification to implementation**

Kenapa tahapan-tahapan ini penting?

- Kesuksesan suatu sistem informasi tergantung pada analisa dan perancangan yang baik
- ANSI telah digunakan secara luas di berbagai industri (teknologi yang telah teruji)
- Bagian dari karir dalam dunia IT, menawarkan banyak kesenangan dan tantangan serta gaji yang tidak rendah
- Kenaikan permintaan akan keahlian sistem analis

Secara singkat berdasarkan pendekatan ANSI, seorang sistem analis adalah orang yang bertugas:

- Bagaimana membangun sistem informasi
- Bagaimana menganalisa kebutuhan dari sistem informasi
- Bagaimana merancang sebuah Sistem Informasi berbasis komputer
- Bagaimana memecahkan masalah dalam organisasi melalui sistem informasi

3.2 Metodologi pengembangan Sistem

Proses-proses standard yang digunakan untuk membangun suatu sistem informasi meliputi langkah-langkah berikut ini:

- o Analisa
- o Desain
- o Implementasi
- o Maintenance

Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang dikenal dengan nama Systems Development Life Cycle (SDLC) yang merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan dari usaha analisa dan desain. SDLC meliputi fase-fase sebagai berikut:

1. Identifikasi dan seleksi proyek
2. Inisiasi dan perencanaan proyek
3. Analisa
4. Desain
 - o Desain logikal
 - o Desain Fisikal
5. Implementasi
6. Maintenance

3.2.1 Identifikasi dan seleksi proyek

Langkah pertama dalam SDLC keseluruhan informasi yang dibutuhkan oleh sistem diidentifikasi, dianalisa, diprioritaskan dan disusun ulang. Dalam langkah ini dilakukan beberapa hal diantaranya:

- o Mengidentifikasi proyek-proyek yang potensial
- o Melakukan klasifikasi dan me-rangking proyek
- o Memilih proyek untuk dikembangkan.

Adapun sumber daya yang terlibat adalah *user, sistem analis, manager yang mengkoordinasi proyek*

Aktivitas yang dilakukan meliputi: *mewawancarai manajemen user, merangkum pengetahuan yang didapatkan, dan mengestimasi cakupan proyek dan mendokumentasikan hasilnya*

Output: Laporan kelayakan berisi definisi masalah dan rangkuman tujuan yang ingin dicapai

3.2.2 Inisiasi dan perencanaan proyek

Dalam tahapan ini Proyek SI yang potensial dijelaskan dan argumentasi untuk melanjutkan proyek dikemukakan. Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Hasil dari tahapan ini adalah :

Langkah-langkah detail-rencana kerja-high level system requirement-penugasan untuk anggota tim.

3.2.3 Tahapan Analisa

Fase ketiga dalam SDLC dimana sistem yang sedang berjalan dipelajari dan sistem pengganti diusulkan. Dalam tahapan ini dideskripsikan sistem yang sedang berjalan, masalah dan kesempatan didefinisikan, dan rekomendasi umum untuk bagaimana memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan diusulkan. Tujuan utama dari fase analisis adalah untuk memahami dan mendokumentasikan kebutuhan

bisnis (business need) dan persyaratan proses dari sistem baru. Ada 6 aktifitas utama dalam fase ini:

- o Pengumpulan informasi
- o Mendefinisikan sistem requirement
- o Membangun prototype untuk menemukan requirement
- o Memprioritaskan requirement
- o Menyusun dan mengevaluasi alternatif
- o Mereview requirement dengan pihak manajemen

3.2.4 Tahapan Desain

Pada tahapan ini deskripsi dari requirement yang telah direkomendasikan diubah ke dalam spesifikasi sistem physical dan logical.

Logical Design

Bagian dari fase desain dalam SDLC dimana semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis dideskripsikan terpisah dari platform komputer yang nanti digunakan. Hasil dari tahapan ini adalah :

- o Deskripsi fungsional mengenai data dan proses yang ada dalam sistem baru
- o Deskripsi yang detail dari spesifikasi sistem meliputi:
 - Input
 - Output
 - Process

Physical design

Pada bagian ini spesifikasi logical diubah ke dalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan. Adapun output dari sistem ini adalah :

- o Deskripsi teknis
- o Deskripsi yang detail dari spesifikasi sistem meliputi:
 - programs,
 - files,
 - network,
 - system software
 -

Pada tahapan desain ada beberapa aktifitas utama yang dilakukan yaitu:

- o Merancang dan mengintegrasikan network
- o Merancang Arsitektur aplikasi
- o Mendesain user interface
- o Mendesain sistem interface
- o Mendesain dan mengintegrasikan database
- o Memnuat prototype untuk detail dari desain
- o Mendesain dan mengintegrasikan kendali sistem

3.2.5 Implementasi

Tahapan kelima pada SDLC, dimana pada tahapan ini dilakukan beberapa hal yaitu:

- o Coding
- o Testing
- o Instalasi

Output dari tahapan ini adalah : source code, prosedur pelatihan.

3.2.6 Maintances

Langkah terakhir dari SDLC dimana pada tahapan ini sistem secara sistematis diperbaiki dan ditingkatkan. Hasil dari tahapan ini adalah Versi baru dari software yang telah dibuat.

3.3 Kelemahan dari SDLC tradisional

- o Terlalu mahal (biaya dan waktu) ketika terjadi perubahan ketika sistem sudah dikembangkan
- o SDLC merupakan metode dengan pendekatan terstruktur yang mensyaratkan mengikuti semua langkah yang ada
- o Biaya maintenance cukup besar

BAB 4

Pendekatan-Pendekatan Pengembangan Sistem

Disamping metode tradisional SDLC, ada beberapa metode yang dikembangkan untuk melengkapi kelemahan-kelemahan yang ada dalam metode SDLC. Metode-metode itu antara lain:

- o Structured analysis and structured design
- o Object oriented analysis and design
- o Prototyping
- o Joint Application Design (JAD)
- o Participatory design

4.1 Structured analysis dan structured design

Pendekatan ini lebih berfokus pada bagaimana mereduksi waktu dan maintenance dalam pengembangan sistem. Pendekatan ini juga langsung mengintegrasikan perubahan jika diperlukan.

4.2 Object Oriented Analysis and Design (OOAD)

Pendekatan baru untuk pengembangan sistem, sering disebut sebagai pendekatan ketiga setelah pendekatan yang berorientasi data dan berorientasi proses. OOAD adalah metode pengembangan sistem yang lebih menekankan pada objek

dibandingkan dengan data atau proses. Ada beberapa ciri khas dari pendekatan ini yaitu object, Inheritance dan object class

Object adalah struktur yang mengenkapsulasi atribut dan metode yang beroperasi berdasarkan atribut-atribut tadi. Objek adalah abstraksi dari benda nyata dimana data dan proses diletakkan bersama untuk memodelkan struktur dan perilaku dari objek dunia nyata.

Object class adalah sekumpulan objek yang berbagi struktur yang sama dan perilaku yang sama.

Inheritance. Properti yang muncul ketika tipe entitas atau object class disusun secara hirarki dan setiap tipe entitas atau object class menerima atau mewarisi atribut dan metode dari pendahulunya.

4.3. 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. Prototyping merupakan bentuk dari Rapid Application Development (RAD). Beberapa kerugian RAD:

- o RAD mungkin mengesampingkan prinsip-prinsip rekayasa perangkat lunak
- o Menghasilkan inkonsistensi pada modul-modul sistem
- o Tidak cocok dengan standar
- o Kekurangan prinsip reusability komponen

Prototype methodology

- o Analis bekerja dengan tim untuk mengidentifikasi requirement awal untuk sistem
- o Analis kemudian membangun prototype. Ketika sebuah prototype telah selesai. User bekerja dengan prototype itu dan menyampaikan pada analis apa yang mereka sukai dan yang tidak mereka sukai.
- o Analis kemudian menggunakan feedback ini untuk memperbaiki prototype
- o Versi baru diberikan kembali ke user
- o Ulangi langkah-langkah tersebut sampai user merasa puas

Keuntungan prototype

- o Prototype melibatkan user dalam analisa dan desain
- o Punya kemampuan menangkap requirement secara konkret daripada secara abstrak
- o Untuk digunakan secara standalone
- o Digunakan untuk memperluas SDLC

4.4 Joint Application Design (JAD)

Pada akhir 1970 an personil pengembangan sistem di IBM mengembangkan proses baru untuk mengumpulkan requiremen SI dan mereview desain dengan nama JAD. JAD adalah proses terstruktur dimana user, manager dan analis bekerja bersama-sama selama beberapa hari dalam 1 pertemuan bersama untuk mengumpulkan requiremen sistem yang akan dibangun.

4.5 Participatory design

End user dilibatkan dalam pengembangan sistem dalam satu meja untuk persetujuan tentang sistem requirement dan sistem desain.

Pada perkembangannya desain sistem banyak disupport oleh penggunaan software dan teknologi baru. Analisis mengandalkan tool dengan tujuan :

- o Meningkatkan produktifitas
- o Berkomunikasi lebih efektif dengan user
- o Mengintegrasikan pekerjaan yang telah dilaksanakan dari awal pengembangan sampai akhir.

Contoh-contoh tool yang digunakan adalah :

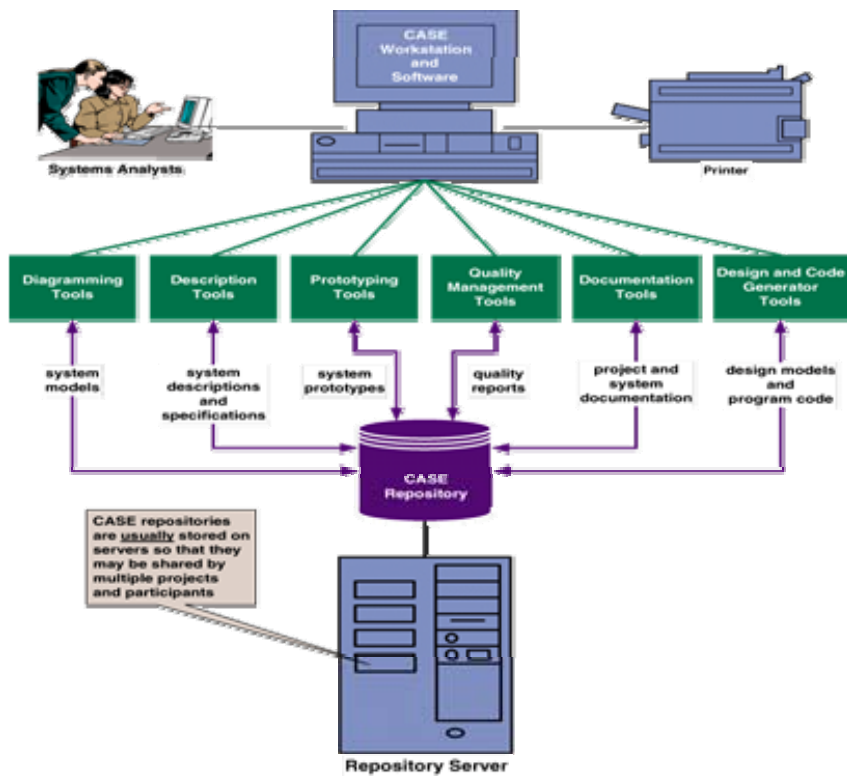
- Computer-Aided Systems Engineering (CASE -tools)
- Application Development Environments (ADE -tools)
- Process and Project Managers

4.6 meningkatkan produktifitas pengembangan Sistem Informasi

Teknologi komputer dapat digunakan untuk meningkatkan produktifitas. CASE tool sebagai contoh, menyediakan lebih banyak produktifitas untuk meningkatkan kemampuan seperti pengembangan kode, tool diagramming dan pendesainan layar dan laporan.

Computer-Aided Systems Engineering: CASE tools

Case adalah software yang digunakan untuk mengotomasi atau mendukung penggambaran dan analisa dari model sistem dan menyediakan translasi dari model sisystem ke sistem aplikasi. Berikut ini gambaran arsitektur CASE



Gambar 4.1 arsitektur CASE Tool

BAB 5

Analisa Sistem

5.1 Definisi Analisis Sistem

Analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka.

5.2 Definisi Desain Sistem

Sebuah teknik pemecahan masalah yang saling melengkapi (dengan analisa sistem) yang merangkai kembali bagian-bagian komponen menjadi sistem yang lengkap-harapannya, sebuah sistem yang diperbaiki. Hal ini melibatkan penambahan, penghapusan, dan perubahan-perubahan bagian relatif pada sistem awal (aslinya).

5.3 Analisa sistem informasi

Fase-fase pengembangan dalam pembangunan sistem informasi yang utamanya difokuskan pada masalah dan persyaratan-persyaratan bisnis, terpisah dari teknologi apapun yang dapat atau akan digunakan untuk mengimplementasikan solusi pada masalah tersebut.

5.4 Pendekatan-pendekatan analisis model Driven

Model Driven adalah analisis yang menggunakan gambar untuk mengkombinasikan masalah-masalah, persyaratan-persyaratan, dan solusi-solusi bisnis. Contoh model yang sering dipakai adalah flowchart dan DFD. Contoh dari analisis dengan pendekatan model driven adalah :

- o Analisis terstruktur
- o Rekayasa informasi
- o Analisa berorientasi objek

5.4.1 Analisis Terstruktur

Dulu adalah salah satu pendekatan formal pertama untuk analisis sistem informasi. Analisis ini terfokus pada aliran data dan proses bisnis dan perangkat lunak. Analisis ini disebut proses oriented. Analisis terstruktur sederhana dalam konsep. Para analis menggambarkan serangkaian proses dalam bentuk diagram alir data (Data flow diagram) yang menggambarkan proses yang ada atau yang diusulkan bersama-sama dengan input, output dan file mereka.

5.4.2 Rekayasa informasi dan pemodelan data

Rekayasa informasi dulu terfokus pada struktur data yang tersimpan pada sebuah sistem. Jadi dikatakan data centered. Model-model data dalam rekayasa informasi disebut entity relationship. Untuk lebih jelasnya akan dibahas pada modul berikutnya. Rekayasa informasi dikatakan berpusat pada data karena menekankan pada pembelajaran dan analisa persyaratan data sebelum persyaratan-persyaratan proses. Hal ini didasarkan pada tingkat kepercayaan bahwa data dan pengetahuan adalah sumber daya perusahaan yang harus direncanakan dan dipelihara. Akibatnya analis menggambarkan model data mentah ke dalam hubungan entitas, baru kemudian membuat diagram alir data yang menjelaskan proses-proses

yang terjadi. Rekayasa informasi sebenarnya berusaha mensinkronkan pemodelan data dengan pemodelan proses. Perbedaannya hanya pada analisis terstruktur digambarkan pemodelan proses terlebih dahulu, baru pemodelan data.

5.4.3 Analisa berorientasi objek

Selama 30 tahun, kebanyakan pendekatan pengembangan sistem telah memisahkan pengetahuan(data) dari proses. Teknik objek muncul untuk memisahkan menghilangkan pemisahan data dan proses ini. Sebaliknya data dan proses spesifik yang membuat, membaca, memperbarui atau menghapus data itu diintegrasikan ke dalam konstruksi yang disebut data atau objek. Satu satunya cara untuk membuat, membaca, memperbarui atau menghapus data adalah dengan cara proses perlekatan (embedded) yang disebut metode. OOA adalah teknik yang model driven yang mengintegrasikan data dan proses yang disebut objek. Model OOA adalah gambar-gambar yang mengilustrasikan objek-objek sistem dari berbagai perspektif, seperti struktur, perilaku dan interaksi antar objek. Contoh yang paling terkenal adalah UML (Unified Modelling Language).

5.5 Arti penting tahapan Analisis

Banyak sistem informasi bagus yang akhirnya ditinggalkan user karena sistem analisis tidak punya pengertian yang benar tentang organisasi. Tujuan dari analisa sistem adalah menghindari kondisi ini, dengan menjawab pertanyaan-pertanyaan berikut:

- o Siapa yang akan menggunakan sistem?
- o Sistem yang dikembangkan akan seperti apa?
- o Apa yang bisa dikerjakan sistem?
- o Kapan sistem akan digunakan?

BAB 6

ANALISA SISTEM

(Bagian 2)

6.1 Pendefinisian masalah

Pada tahapan analisis sistem , analis mempunyai tugas mendefinisikan masalah sistem, melakukan studi kelayakan dan menganalisis kebutuhan sistem yang akan dikembangkan. Terdapat 3 pertanyaan kunci yang harus dijawab, yaitu :

- o Apa masalah yang harus diselesaikan dengan sistem informasi?
- o Apa penyebab masalah tersebut?
- o Siapa pemakai akhir dari sistem?

Masalah yang dipelajari analis sistem adalah masalah yang dihadapi pemakai. Dengan mempelajari masalah ini, maka analis bekerjasama dengan pemakai untuk mendapatkan permasalahan secara kasar. Langkah-langkah yang harus dijalankan adalah:

- o Mendefinisikan batasan dan sasaran
- o Mendefinisikan masalah yang dihadapi pemakai
- o Mengidentifikasi penyebab masalah dan titik keputusan
- o Mengidentifikasi pemakai akhir
- o Memilih prioritas penanganan masalah
- o Memperkirakan biaya dan manfaat secara kasar
- o Membuat laporan hasil pendefinisian masalah

6.2 Sasaran dan batasan sistem informasi

Sasaran sistem informasi adalah peningkatan kinerja, peningkatan efektifitas informasi, penurunan biaya, peningkatan keamanan aplikasi, peningkatan efisiensi dan peningkatan pelayanan pada pelanggan. Penyimpangan dari keenam sasaran inilah yang menimbulkan masalah pada sistem informasi. Batasan sistem adalah lingkungan yang membatasi aplikasi, misalnya peraturan-peraturan siapa yang boleh menggunakan sistem, dan siapa yang tidak boleh.

Masalah dalam sistem informasi

Masalah dalam sistem informasi adalah kondisi atau situasi yang menyimpang dari sasaran sistem informasi, bahkan menyimpang dari sasaran organisasi atau perusahaan, misalnya kinerja mengalami penurunan, informasi tidak efektif, sistem informasi tidak aman. Biasanya masalah dinyatakan dalam pertanyaan misalnya:

- o Apakah sistem informasi ini dapat meningkatkan kinerja
- o Apakah sistem informasi dapat menurunkan biaya
- o Apakah sistem informasi bisa meningkatkan keamanan
- o Apakah sistem informasi bisa menurunkan pemborosan
- o Apakah sistem informasi bisa meningkatkan penjualan
- o Apakah sistem informasi bisa meningkatkan pelayanan

6.3 ANALISIS PIECES

Untuk mengidentifikasi masalah, maka harus dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Panduan ini dikenal dengan PIECES analysis (performance, Information, economy, Control,

eficiency dan Services). Dari analisis ini biasanya didapatkan beberapa masalah utama. Hal ini penting karena biasanya yang muncul dipermukaan bukan masalah utama, tetapi hanya gejala dari masalah utama saja.

a. Analisis Kinerja

Masalah kinerja terjadi ketika tugas-tugas bisnis dijalankan dan tidak mencapai sasaran. Kinerja diukur dengan jumlah produksi dan waktu tanggap. Jumlah produksi adalah jumlah pekerjaan yang bisa diselesaikan selama jangka waktu tertentu. Bagian pemasaran kinerjanya diukur berdasarkan volume pekerjaan atau pangsa pasar yang diraih atau citra perusahaan.

Waktu tanggap adalah keterlambatan rata-rata antara suatu transaksi dengan tanggapan yang diberikan kepada transaksi tersebut.

b. Analisis Informasi

Informasi merupakan komoditas krusial bagi pemakai akhir. Kemampuan sistem informasi dalam menghasilkan informasi yang bermanfaat dapat dievaluasi untuk menangani masalah dan peluang untuk mengatasi masalah tersebut. Dalam hal ini meningkatkan kualitas informasi tidak dengan menambah jumlah informasi, karena terlalu banyak informasi juga menghasilkan masalah baru. Situasi yang membutuhkan peningkatan informasi meliputi:

- o Kurangnya informasi mengenai keputusan atau situasi yang sekarang
- o Kurangnya informasi yang relevan mengenai keputusan ataupun situasi sekarang
- o Kurangnya informasi yang tepat waktu
- o Terlalu banyak informasi
- o Informasi tidak akurat

Informasi juga dapat merupakan fokus dari suatu batasan atau kebijakan. Sementara analisis informasi memeriksa output sistem, analisis data meneliti data yang tersimpan dalam sebuah sistem. Permasalahan yang dihadapi meliputi:

Data yang berlebihan. Data yang sama ditangkap dan/atau disimpan di banyak tempat.

Kekakuan data. Data ditangkap dan disimpan, tetapi diorganisasikan sedemikian rupa sehingga laporan dan pengujian tidak dapat atau sulit dilakukan.

c. analisis ekonomi

Ekonomis barangkali merupakan motivasi paling umum bagi suatu proyek. Pijakan dasar bagi kebanyakan manajer adalah biaya atau rupiah. Persoalan ekonomis dan peluang berkait dengan masalah biaya. Adapun hal-hal yang perlu diperhatikan dapat disimak berikut ini:

Biaya

- o Biaya tidak diketahui
- o Biaya tidak dapat dilacak ke sumber
- o Biaya terlalu tinggi

Keuntungan

- o Pasar-pasar baru dapat dieksplorasi
- o Pemasaran saat ini dapat diperbaiki
- o Pesanan-pesanan dapat ditingkatkan.

d. Analisis keamanan

Tugas-tugas bisnis perlu dimonitor dan dibetulkan jika ditemukan kinerja yang dibawah standar. Kontrol dipasang untuk meningkatkan kinerja sistem, mencegah atau mendeteksi kesalahan sistem, dan menjamin keamanan data, informasi dan persyaratan. Adapun hal-hal yang perlu diperhatikan adalah:

Keamanan atau kontrol yang lemah

- o Input data tidak diedit dengan cukup
- o Kejahatan (misalnya penggelapan atau pencurian) terhadap data
- o Etika dilanggar pada data atau informasi-mengacu pada data atau informasi yang diakses orang yang tidak berwenang.
- o Data tersimpan secara berlebihan, tidak konsisten pada file-file atau database-database yang berbeda.
- o Peraturan atau panduan privasi data dilanggar (atau dapat dilanggar)
- o Error pemrosesan terjadi (oleh manusia, mesin atau perangkat lunak)
- o Error pembuatan keputusan terjadi.

Kontrol atau keamanan berlebihan

- o Prosedur birokratis memperlamban sistem
- o Pengendalian mengganggu para pelanggan atau karyawan
- o Pengendalian berlebihan menyebabkan penundaan pemrosesan.

e. analisis Efisiensi

Efisiensi menyangkut bagaimana menghasilkan output sebanyak-banyaknya dengan dengan input yang sekecil mungkin. Untuk melihat apakah efisiensi dari suatu sistem baik atau tidak dapat dengan melihat indikator-indikator berikut ini:

- o Orang, mesin atau komputer membuang-buang waktu
- o Data secara berlebihan di input atau disalin
- o Data secara berlebihan di proses
- o Informasi secara berlebihan dihasilkan
- o Orang, mesin atau komputer membuang
- o Usaha yang dibutuhkan untuk tugas-tugas terlalu berlebihan
- o Material yang dibutuhkan untuk tugas-tugas terlalu berlebihan.

f. Services

Analisa untuk menilai kualitas dari suatu sistem dapat dilihat dari kriteria-kriteria berikut ini :

- o Sistem menghasilkan produk yang tidak akurat
- o Sistem menghasilkan produk yang tidak konsisten
- o Sistem menghasilkan produk yang tidak dipercaya
- o Sistem tidak mudah dipelajari
- o Sistem tidak mudah digunakan
- o Sistem canggung untuk digunakan
- o Sistem tidak fleksibel

Berikut ini contoh hasil analisa PIECES yang dilakukan pada sistem informasi penggajian yang ada pada BBTCLPPM (Balai Besar Teknik Kesehatan Lingkungan dan Pemberantasan Penyakit Menular) Yogyakarta

A. ANALISIS KINERJA (PERFORMANCE)

Adalah kemampuan didalam menyelesaikan tugas bisnis dengan cepat sehingga sasaran dapat segera tercapai. Kinerja diukur dengan jumlah produksi (*throughput*) dan waktu jawaban (*respon time*) dari suatu system. System yang dikembangkan ini akan menyediakan *throughput* dan *respon time* yang memadai untuk kebutuhan manajemen pada BBTCLPPM.

Kelemahan:

- Keterlambatan memberikan informasi oleh pegawai mengenai jumlah anak yang bertambah, sehingga Bendahara Gaji harus melakukan perhitungan ulang gaji dengan menambahkan jumlah tunjangan anak. Dan hal tersebut akan memperlambat kinerja proses.
- Adanya kenaikan pangkat pegawai yang menyebabkan Bendahara Gaji harus memintakan kekurangan gaji atau rapel pada bulan berikutnya, sehingga akan semakin memperpanjang *response time* (waktu tanggap).

B. ANALISIS INFORMASI (INFORMATION)

Laporan-laporan yang sudah selesai diproses digunakan untuk menghasilkan informasi yang dibutuhkan oleh manajemen di dalam pengambilan keputusan.

Informasi merupakan hal yang tidak kalah penting karena dengan informasi tersebut pihak manajemen akan merencanakan langkah-langkah selanjutnya.

Kelemahan:

- Tidak adanya slip gaji pegawai sehingga pegawai tidak mengetahui informasi perincian gaji yang diterima secara keseluruhan.

C. ANALISIS EKONOMI (ECONOMI)

Adalah penilaian sistem didalam pengurangan dan keuntungan yang akan didapatkan dari system yang dikembangkan. System ini akan memberikan penghematan operasional dan meningkatkan keuntungan perusahaan. Penghematan didapat melalui pengurangan bahan baku dan perawatan. Sedangkan

keuntungan didapat dari peningkatan nilai informasi dan keputusan yang dihasilkan.

Kelemahan:

Apabila Bendahara Gaji ingin mendapatkan aplikasi system penggajian yang lebih baik dan mudah penggunaannya maka harus membeli software untuk aplikasi tersebut dengan harga yang mahal.

D. ANALISIS KEAMANAN (SECURITY)

Sistem keamanan yang digunakan harus dapat mengamankan data

dari kerusakan, misalnya dengan membuat BackUp data. Selain itu

system keamanan juga harus dapat mengamankan data dari akses

yang tidak diijinkan, biasanya dilakukan dengan password terutama

pada form aplikasi dan databasenya.

Kelemahan:

Aplikasi sistem penggajian pada BBTKLPPM keamanannya belum begitu diperhatikan sehingga bisa dilihat oleh pihak lain.

E. ANALISIS EFISIENSI

Berhubungan dengan sumber daya yang ada guna meminimalkan

pemborosan. Efisiensi dari system yang dikembangkan adalah pemakaian secara maksimal atas sumber daya yang tersedia yang

meliputi manusia, informasi, waktu, uang, peralatan, ruang dan keterlambatan pengolahan data.

Kelemahan:

Apabila Bendahara Gaji BBTKLPPM yang mengurus system penggajian berhalangan hadir pada saat penggajian

dilaksanakan, maka akan menghambat proses penggajian dan tidak adanya pengganti yang mengurus system penggajian tersebut

F. ANALISIS PELAYANAN (SERVICE)

Perkembangan organisasi dipicu dengan peningkatan pelayanan yang

lebih baik. Dari peningkatan pelayanan terhadap system yang dikembangkan akan memberikan suatu:

1. Akurasi dalam pengolahan data
2. Kehandalan terhadap konsistensi dalam pengolahan input dan outputnya serta kehandalan dalam menangani pengecualian
3. Kemampuan dalam menangani masalah yang diluar kondisi normal
4. Sistem mudah pakai
5. Mampu mengkoordinasi aktifitas untuk mencapai tujuan dan sasaran.

Kelemahan:

Proses penyerahan gaji masih menggunakan system manual (tidak melalui rekening bank, tetapi diserahkan langsung kepada pegawai) sehingga membingkinkan antrian dalam penerimaan gaji.

6.4 STUDI KELAYAKAN

- Setiap proposal proyek harus dievaluasi kelayakannya dari berbagai segi :
 - ⇒ Kelayakan Teknis
 - ⇒ Kelayakan Operasional
 - ⇒ Kelayakan Ekonomi

♦ Kelayakan Teknis

- ⇒ Apakah teknologi yang dibutuhkan sudah tersedia?
- ⇒ Apakah teknologi yang akan digunakan ini dapat berintegrasi dengan teknologi yang sudah ada?
- ⇒ Apakah sistem yang sudah ada dapat dikonversikan ke sistem dengan teknologi baru?
- ⇒ Apakah organisasi memiliki orang yang menguasai teknologi baru ini?

♦ Kelayakan Operasional

- Aspek Teknis
 - ⇒ Apakah sistem dapat memenuhi tujuan organisasi untuk mendapatkan informasi yang dibutuhkan?
 - ⇒ Apakah sistem dapat diorganisasikan untuk menghasilkan informasi pada saat yang tepat untuk setiap orang yang membutuhkannya?
- Aspek (psikologis) penerimaan oleh orang-orang yang ada di dalam organisasi
 - ⇒ Apakah sistem baru memerlukan restrukturisasi organisasi dan bagaimana akibat strukturisasi ini terhadap orang-orang yang ada di organisasi?
 - ⇒ Apakah diperlukan pelatihan atau pelatihan ulang?
 - ⇒ Apakah personil di dalam organisasi dapat memenuhi kriteria untuk sistem baru?

♦ Kelayakan Ekonomi

- Kelayakan ekonomi berhubungan dengan *return on investment* atau berapa lama biaya investasi dapat kembali.
- Apakah bermanfaat melakukan investasi ke proyek ini atau kita harus melakukan sesuatu yang lain?
- Pada suatu proyek yang besar biasanya lebih ditekankan kepada kelayakan ekonomi karena umumnya berhubungan dengan biaya yang jumlahnya besar.

Untuk menganalisis kelayakan ekonomi digunakan kalkulasi yang dinamakan **Cost Benefit Analysis** atau **Analisa Biaya dan Manfaat**.

Tujuan dari analisa biaya dan manfaat ini adalah :

Memberikan gambaran kepada user apakah manfaat yang diperoleh dari sistem baru “lebih besar” dibandingkan dengan biaya yang dikeluarkan.

Metode kuantitatif yang dapat digunakan adalah :

1. Analisa Payback (Payback Period).
2. Analisa Net Present Value.
3. Return Of investment (ROI)

Payback Period

Jangka waktu yang diperlukan untuk membayar kembali biaya investasi yang telah dikeluarkan.

(dalam ribuan Rp)

Deskripsi	Tahun 1	Tahun 2	Tahun 3
Biaya Investasi	20.000		
Biaya Operasional	5.000	5.000	5.000
Total Biaya	25.000	5.000	5.000
Pendapatan	10.000	17.000	11.500
Keuntungan Bersih	(15.000)	12.000	6.500
Keuntungan Bersih (Kumulatif)	(15.000)	(3.000)	3.500

Proyek mampu membayar kembali investasi karena keuntungan bersih (kumulatif) pada tahun ke-3 telah mencapai nilai (positif) 3.500.

Dengan demikian waktu pelunasan investasi tercapai pada tahun ke-3.

Tepatnya, jangka waktu pelunasan adalah :

$$2 + \{6.500 - 3.500\} / \{6.500\} \\ = 2,46 \text{ tahun} \approx (2 \text{ tahun} + 5,5 \text{ bulan}).$$

Analisa Net Present Value (NPV)**Present Value :**

Nilai sekarang dari penerimaan (uang) yang akan didapat pada tahun mendatang.

Net Present Value :

Selisih antara penerimaan dan pengeluaran per tahun.

Discount Rate :

Bilangan yang digunakan untuk men-discount penerimaan yang akan didapat pada tahun mendatang menjadi nilai sekarang.

Discount rate dapat dilihat dari tabel discount rate yang ditentukan oleh tingkat suku bunga (i) dan tahun (t).

Discount rate pada tahun ke-5 dengan interest rate 10% adalah :
0,621

Untuk menghitung discount rate ini dapat digunakan rumus :

$$d = 1/(1+i)^t$$

d = discount rate

i = Interest rate

t = tahun

NPV dapat dihitung dengan menggunakan rumus :

$$NPV = \sum_{n=1}^n \frac{(B_t - C_t)}{(1+i)^t} - K_0$$

$$t = 1$$

B_t = Benefit tahun ke- t

C_t = Cost tahun ke- t

i = Interest rate yang ditentukan

t = tahun

K_0 = Investasi awal tahun ke-0 (sebelum proyek dimulai)

Kriteria :

$NPV > 0$

Feasible

$NPV = 0$

Indifferent

$NPV < 0$

Unfeasible

Contoh :

(dalam juta Rp.)

No.	Deskripsi	Tahun ke - 0	Tahun ke - 1	Tahun ke - 2	Tahun ke - 3	Tahun ke - 4	Tahun ke - 5
A	Benefit						
1	Penghematan biaya perawatan		50	50	20	30	40
2	Penjualan Informasi			25	25	30	30
	Total Benefit		50	75	45	60	70
B	Cost						
1	Investasi Awal (membeli komputer baru)	95					
2	Biaya operasional		30	30	30	30	30
	Total Cost	95	30	30	30	30	30
	Net Benefit	(95)	20	45	15	30	40
	Discount Rate 15%	1,00	0,870	0,756	0,658	0,572	0,497
	NPV pada Disc. Rate 15%	(95)	17,4	34,02	9,87	17,16	19,88

$$\text{NPV} = 17,4 + 34,02 + 9,87 + 17,16 + 19,88 - 95 = 8,33 > 0$$

Karena NPV > 0 maka proyek feasible.

Return On Invesment

Return on invesment adalah besarnya keuntungan yang bisa diperoleh (dalam %) selama periode waktu yang telah ditentukan untuk menjalankan proyek, untuk menghitungnya digunakan rumus:

$$ROI = \frac{TotalManfaat - TotalBiaya}{TotalBiaya}$$

Jika dinyatakan dalam persen (%)

$$ROI = \frac{Totalmanfaat - TotalBiaya}{TotalBiaya} \times 100\%$$

Berikut ini contoh perhitungan analisa biaya dan manfaat untuk pengembangan sebuah sistem pembelajaran berbasis komputer di sebuah SMP untuk melengkapi sistem pembelajaran yang sudah ada:

C.7. Analisis Biaya dan Manfaat

Rincian biaya dan manfaat	Tahun ke-0	Tahun ke-1	Tahun-2	Tahun-3
Rincian Biaya				
Biaya pengembangan Sistem				
Pengadaan Sistem (Fisika)	Rp 1.500.000	-	-	-
Hardware (2 unit)	Rp 9.000.000	-	-	-
LCD Projector dan Screen	Rp 7.500.000	-	-	-
Software Education Pack	Rp 5.000.000	-	-	-
Biaya pelatihan (8 orang)	Rp 2.000.000	-	Rp 500.000	Rp. 625.000
Biaya Pemeliharaan	-			
Total Biaya (TB)	Rp25.000.000	-	Rp 500.000	Rp. 625.000
Rincian Manfaat				
1. Manfaat Wujud				
• Peningkatan biaya pendidikan	-	Rp. 3.400.000	Rp 4.420.000	Rp. 5.746.000
• Peningkatan Informasi	-	Rp. 4.050.000	Rp. 5.467.500	Rp. 7.381.125
Total Wujud (TW)	-	Rp. 7.450.000	Rp. 9.887.500	Rp.13.127.125
2. Manfaat Tak Wujud				
• Perbaikan citra/nama baik	-	Rp. 1.300.000	Rp. 1.950.000	Rp. 2.925.000
Total Tak Wujud (TTW)	-	Rp. 1.300.000	Rp. 1.950.000	Rp. 2.925.000
Total Manfaat (TM)	-	Rp. 8.750.000	Rp.11.837.500	Rp.16.052.125
Proceed (Selisih TM & TB)	-	Rp. 8.750.000	Rp.11.337.500	Rp.15.427.125

Tabel 4.1 Rincian biaya dan manfaat

Adapun metode-metode untuk melakukan analisis biaya dan manfaat diantaranya sebagai berikut:

1. Analisis Periode Pengembalian (*Payback Period*)
2. Analisis Pengembalian Investasi (*Return On Investment*)
3. Analisis Nilai Sekarang Bersih (*Net Present Value*)

Berdasarkan para rincian biaya dan manfaat pada tabel di atas, maka dapat dilakukan analisis biaya dan manfaat sebagai berikut:

1. Analisis *Payback Period*

Perhitungan analisis *Payback Period* (waktu pengembalian investasi):

	Total Biaya Pengadaan Sistem pada tahun 0	: Rp. 25.000.000
	<i>Proceed</i> pada tahun I	: <u>Rp. 8.750.000</u>
(-)		
	Sisa Biaya Sistem pada tahun I	: Rp. 16.250.000
	<i>Proceed</i> pada tahun II	: <u>Rp. 11.337.500</u>
(-)		
	Sisa Biaya Sistem pada tahun II	: Rp. 4.912.500
	Sisa	$= \frac{4.912.500}{15.427.125} \times 1 \text{ tahun} = 0,31 \text{ tahun}$

Payback Periode sudah dapat dicapai pada tahun ke-3, secara detailnya adalah 2,31 tahun. Dari fakta tersebut, dapat disimpulkan bahwa yang ditanamkan pada rancangan sistem ini akan mencapai titik impas pada waktu 2,31 tahun, yang berarti bahwa pada tahun ke-3 mulai dapat mengambil keuntungan dari sistem tersebut.

2. Analisis Return On Investment (ROI)

Perhitungan prosentase manfaat yang dihasilkan oleh sistem dibandingkan dengan biaya pengadaan sistem.

Biaya perhitungan sistem :	
- Biaya Pengadaan sistem bulan 0	: Rp 25.000.000
- Biaya Perawatan sistem bulan I	: Rp 0
- Biaya Perawatan sistem bulan II	: Rp 500.000
- Biaya Perawatan sistem bulan III	: <u>Rp 625.000</u>
(+)	
Total Biaya	: Rp 26.125.000

Total Keuntungan yang didapat :	
- Total Manfaat pada bulan I	: Rp 8.750.000
- Total Manfaat pada bulan II	: Rp 11.827.500
- Total Manfaat pada bulan III	: <u>Rp 16.052.125</u>
(+)	
Total Manfaat	: Rp 36.639.625

Rumus (ROI) adalah :

$$\text{ROI} = \frac{\text{Total Manfaat} - \text{Total Biaya}}{\text{Total Biaya}} \times 100 \%$$

$$\text{ROI} = \frac{36.639.625 - 26.125.000}{26.125.000} \times 100 \% = 40,24 \%$$

Sistem ini akan memberikan keuntungan pada tahun ke-3 sebesar 40,24 % dari biaya pengadaan, sehingga sistem ini layak di gunakan.

3. Analisis *Net Present Value* (NPV)

Perhitungan Analisis NPV:

$$\text{NPV} = - \text{nilai proyek} + \frac{\text{Proceed 1}}{(1+i\%)^1} + \frac{\text{Proceed 2}}{(1+i\%)^2} + \dots + \frac{\text{Proceed n}}{(1+i)^n}$$

Apabila diaplikasikan pada penerapan sistem yang baru, maka perhitungan nilainya dengan tingkat bunga diskonto 15% per tahun adalah:

$$\text{NPV} = - 25.000.000 + \frac{8.750.000}{(1+0,15)^1} + \frac{11.337.500}{(1+0,15)^2} + \frac{15.427.625}{(1+0,15)^n}$$

$$\text{NPV} = 1.325.059,587$$

Dari hasil perhitungan diatas, dapat disimpulkan bahwa jumlah keuntungan yang diterima sekarang pada tahun ke-3 apabila sistem ini diterapkan adalah : Rp. 1.325.059,587 atau NPV lebih besar dari 0, maka proyek tersebut layak dilaksanakan.

BAB 7

PENENTUAN KEBUTUHAN SISTEM

7.1 System requirement (Kebutuhan Sistem)

Tujuan dari fase analisis adalah memahami dengan sebenarnya kebutuhan dari sistem baru dan mengembangkan sebuah sistem yang memadai requirement tersebut-atau memutuskan bahwa sebenarnya pengembangan sistem baru tidak dibutuhkan. Penentuan kebutuhan sistem merupakan langkah yang paling crucial dalam tahapan SDLC. Kebutuhan Sistem bisa diartikan sebagai berikut:

Pernyataan tentang apa yang harus dikerjakan oleh sistem

Pernyataan tentang karakteristik yang harus dimiliki sistem

7.2 Tipe-tipe Kebutuhan Sistem

Kebutuhan Fungsional (Functional requirement)

Functional requirement adalah jenis requirement yang berisi:

Proses-proses yang harus dilakukan oleh sistem

Informasi-informasi yang harus ada di dalam sistem

Nonfunctional Requirements

Adalah tipe requirement yang berisi properti perilaku yang dimiliki oleh sistem, meliputi:

- ❖ Operasional
- ❖ Performance
- ❖ Keamanan
- ❖ Politik dan budaya

Berikut ini contoh sebuah dokumen kebutuhan sistem untuk pengembangan sistem informasi perpustakaan berbasis komputer :

FUNCTIONAL REQUIREMENT

❖ **Sistem harus dapat melakukan entri buku yang berhubungan dengan pendataan buku:**

- a. User bisa memasukkan berbagai jenis buku beserta dengan kode buku, kategori buku, judul buku, penerbit, pengarang, jumlah halaman, ISBN buku dan lain-lain
- b. User bisa menambahkan koleksi buku baru yang akan dikeluarkan
- c. User bisa menghitung berapa jumlah koleksi buku yang ada secara keseluruhan
- d. User dapat menampilkan berapa jenis atau kategori koleksi buku yang ada beserta dengan jumlahnya
- e. User dapat menampilkan jenis buku berdasarkan kategori tertentu beserta jumlah bukunya
- f. User dapat mencari koleksi buku berdasarkan pengarang, penerbit, judul, dan lain-lain
- g. User dapat menampilkan nama buku yang paling sering di pinjam
- h. User dapat mengganti beberapa kode buku dan lain-lain yang berhubungan dengan buku
- i. User dapat menghapus beberapa jenis buku, judul buku dan lain-lain yang sudah rusak atau hilang
- j. User dapat mengurutkan buku berdasarkan kodenya, judul buku, jumlah halaman, dan lain-lain
- k. User dapat menampilkan buku yang belum diberi kode
- l. User dapat mengelompokkan buku berdasarkan pengarang dan judul buku apa saja yang telah ditulis yang ada di perpustakaan

❖System harus dapat melakukan pendataan anggota

- a. User dapat memasukkan data anggota baru dengan memasukkan kode anggota, nama, alamat, nomor telepon
- b. User dapat menampilkan biaya pendaftaran anggota yang harus di bayarkan
- c. User dapat menampilkan print kartu anggota baru
- d. User dapat menampilkan menghitung berapa jumlah anggota yang ada secara keseluruhan
- e. User dapat menampilkan data jumlah anggota yang berada pada daerah kecamatan yang sama
- f. User bisa menambahkan data anggota baru

- g. User bisa merubah data anggota apabila alamat atau nomor telepon anggota berubah
- h. User dapat menampilkan anggota yang paling sering meminjam serta kategori buku yang di pinjam
- i. User dapat menampilkan data peningkatan atau penurunan anggota baru setiap tahunnya

❖Sistem harus dapat melakukan transaksi peminjaman:

- a. User dapat mencatat semua transaksi peminjaman
- b. User dapat memasukan data-data dari anggota baik nama, alamat, nomor telepon dan lain-lain
- c. User dapat memasukkan jumlah buku yang di pinjam
- d. User dapat memasukkan judul buku, nama pengarang, dan nama penerbit
- e. User dapat menampilkan tanggal kembali buku yang di pinjam
- f. User dapat mengetahui apakah anggota sudah mengembalikan buku yang di pinjam

- g. User dapat menampilkan anggota yang paling sering meminjam

❖ **Sistem harus dapat melakukan transaksi pengembalian:**

- a. User dapat mencatat semua transaksi pengembalian
- b. User dapat menampilkan judul buku, nama pengarang, nama penerbit dari buku yang di kembalikan
- c. User dapat menampilkan denda yang harus dibayar oleh peminjam buku (denda 1 buku Rp.100/perhari)
- d. User dapat mengetahui total keseluruhan buku yang di pinjam
- e. User dapat menampilkan tanggal pengembalian buku
- f. User dapat menampilkan judul buku, nama pengarang, dan nama penerbit dari buku apabila peminjam menghilangkan buku yang di pinjam
- g. User dapat menampilkan harga buku yang harus diganti oleh anggota apabila buku tersebut hilang
- h. User dapat mengetahui nama buku yang belum di kembalikan oleh peminjam/anggota
- i. User dapat mengetahui nama anggota beserta alamat anggota yang belum mengembalikan buku beserta tanggal buku tersebut harus di kembalikan

❖ **Sistem harus dapat melakukan laporan keuangan secara otomatis:**

- a. User dapat menampilkan laporan pendapatan yang diterima baik berdasarkan tanggal tertentu, bulan tertentu, atau tahun tertentu
- b. User dapat menampilkan laporan pendapatan yang di terima baik berdasarkan biaya pendaftaran anggota baru atau anggota yang memperbarui kartu anggota dan denda yang di bayarkan oleh anggota

- c. User dapat mengetahui jumlah biaya-biaya yang dikeluarkan dan untuk apa biaya-biaya tersebut
- d. User dapat mengetahui jumlah dana yang di gunakan untuk pembelian buku baru serta jumlah buku yang harus di beli
- e. User dapat mengetahui atau mengecek jumlah laba kotor maupun laba bersihnya dari pendapatan yang di terima
- f. User dapat menampilkan jumlah kas yang dapat digunakan

NON-FUNCTIONAL REQUIREMENT

Operasional

- a. Digunakan pada system operasi Microsoft Windows XP®, Microsoft Windows® NT, Microsoft Windows®2000
- b. Pentium III-class or higher processor
- c. 128 MB – 256 MB of RAM
- d. Bisa digunakan untuk barcode render
- e. Printer untuk mencetak kartu anggota dan laporan keuangan maupun yang lain-lain

Security

- a. Dilengkapi password untuk sistem aplikasinya maupun databasenya
- b. Dilengkapi dengan kamera untuk mengawasi anggota yang membaca di ruang baca dan ruang penyimpanan tas yang tersambung kekomputer

❖Informasi

- a. Digunakan untuk menginformasikan apabila password yang di masukkan oleh user salah
- b. Digunakan untuk menampilkan prosedur pendaftaran anggota baru

❖Performance

- a. Waktu untuk transaksi peminjaman buku dibatasi 2 menit
- b. Waktu untuk transaksi pengembalian buku di batasi 1 menit
- c. Jumlah transaksi peminjaman dan pengembalian

7.3 Teknik Pengumpulan Requirement

Dalam menyusun requirement, ada beberapa teknik yang biasa digunakan. Pada uraian berikut kita akan membahas beberapa teknik pengumpulan requiremen yang biasa digunakan.

Interview

Interview atau wawancara adalah teknik pengumpulan requirement yang paling umum digunakan. Langkah-langkah dasar dalam teknik interview adalah:

- a. Memilih target interview
- b. Mendisain pertanyaan-pertanyaan interview
- c. Persiapan interview
- d. Melakukan interview
- e. Follow up hasil interview

Joint Application Development

Proses kelompok terstruktur yang terfokus untuk menentukan requiremen, melibatkan tim proyek, pengguna, dan manajemen bekerja bersama-sama, teknik ini sangat berguna untuk mereduksi waktu pengumpulan informasi sampai 50%.

Pelaku JAD

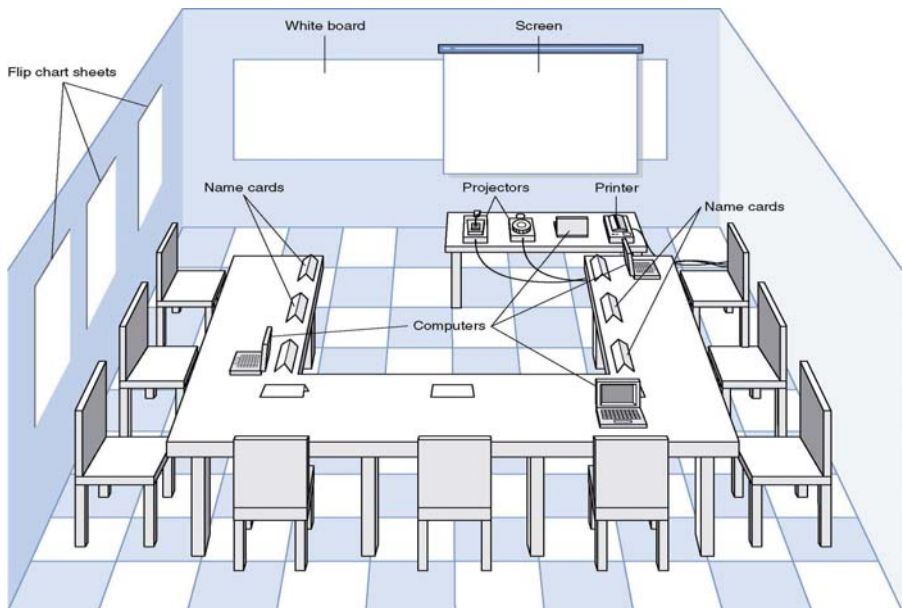
a. Facilitator

1. Terlatih untuk teknik JAD
2. Menyiapkan agenda dan memandu proses kelompok

b. Scribe

Mencatat isi dari sesi JAD

c. User dan **manager** dari perusahaan yang akan dibuat Sistemnya



Gambar 7.1 JAD Meeting Room

Questionnaires (Kuisisioner)

Kuisisioner adalah sekumpulan pertanyaan tertulis, dan sering melibatkan banyak orang. Kuisisioner bisa dilakukan paper based atau secara elektronik. Biasanya sampel dipilih untuk mewakili populasi tertentu. Setelah hasil kuisisioner diperoleh diperlukan analisa untuk mengambil data yang sesuai dengan keperluan pengumpulan requirement.

Analisa Dokumen

Teknik ini dilakukan dengan mempelajari material yang menggambarkan sistem yang sedang berjalan. Biasanya dokumen yang diamati berupa form, laporan, manual kebijakan, grafik organisasi

Observasi

Teknik ini dilakukan dengan melakukan pengamatan secara langsung pada proses-proses yang sedang berjalan. Hal ini penting karena kadang-kadang user atau manager tidak dapat mengingat secara keseluruhan apa yang mereka lakukan dan menceritakan kembali ke analis.

BAB 8

ANALISA USE CASE

8.1 Analisa Use Case

Use case adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks. Use case menambahkan detail untuk requirement yang telah dituliskan pada definisi sistem requirement. Use case dikembangkan oleh sistem analis bersama-sama dengan user. Pada tahapan selanjutnya berdasarkan use case ini analis menyusun model data dan model proses.

8.2 Peranan Use case

Semua kemungkinan tanggapan terhadap suatu kejadian didokumentasikan. Use case sangat berguna ketika situasi yang dianalisa sangat kompleks. Sebuah use terdiri dari elemen-elemen sebagai berikut:

1. Informasi Dasar
 - a. Nama, jumlah dan deskripsi singkat
 - b. Trigger-kejadian yang menyebabkab adanya usecase
 - c. Eksternal trigger: trigger yang berasal dari luar sistem
 - d. Temporal Trigger: kejadian yang berbasis waktu
 - e. Sudut pandang use case harus konsisten
2. Input-output utama
 - a. Asal dan tujuan
 - b. Tujuan harus lengkap dan komprehensif

3. Detail

- a. Harus ada detail dari langkah-langkah yang harus dilakukan berikut data masukan dan keluarannya.

Contoh sebuah use case:

Scenario Name: Patient makes, cancels, or changes an appointment		ID number: <u>1</u>
Short description: This describes how we make a new appointment as well as changing or canceling an appointment		
Trigger: Patient calls and asks for an appointment or asks to cancel an existing appointment		
Type: <u>External</u> Temporal		
Major Inputs		Major Outputs
Description	Source	Description
Patient name	Patient	Appointment
Desired appointment	Patient	Patient
Appointment to change/cancel	Patient	Possible appointments
Patient information	Patient's DB	
Available appointments	Appointment's DB	
Major Steps Performed		Information for Steps
1. If this is a change or cancellation, then find current appointment in the Appointment File and cancel it. 2. Check to make sure the patient is a current patient and has no unpaid bills. If this is a new patient, perform the "Add New Patient" use case before continuing. If the patient has unpaid bills, then transfer the call to the business office. 3. Find the available times for an appointment and select ones to propose to the patient (some appointment times will be too short or too long for the patient's problem). This may be repeated several times until a good time is found. 4. The patient picks an appointment time to be scheduled.		Patient name Appointment to change/cancel Revised appointment Patient name Available appointments Possible appointments Desired appointment Appointment

Proses pengembangan Use Case

1. Identifikasi use case utama
2. Identifikasi setiap langkah dalam setiap use case
3. Identifikasi elemen-elemen dalam setiap langkah
4. Konfirmasikan use case
5. Ulangi langkah-langkah diatas secara iteratif

Langkah 1: Identifikasi use case utama

Aktifitas	Pertanyaan tipikal yang diajukan
Mulailah sebuah form use case dari setiap use case Jika lebih dari 9 kelompokkan dalam paket-paket	Tanyakan who, what dan where tentang tugas dan input/output mereka: • Tugas utama apa yang dilakukan? • Trigger apa yang memicu tugas ini? • Informasi/form/laporan apa yang diperlukan untuk melakukan tugas ini? • Siapa yang memberi informasi/form/laporan ini? • Informasi apa yang dihasilkan dan pergi ke mana informasi tersebut?

Langkah 2: Identifikasi setiap langkah dalam setiap use case

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap use case, masukkan langkah-langkah utama untuk memproses input dan menghasilkan output	Menanyakan "how" tentang setiap use case: • Bagaimana anda memproduksi laporan? • Bagaimana informasi berubah pada laporan? o you process forms? • Tool apa yang dilakukan pada langkah ini?

Langkah 3: Identifikasi elemen-elemen dalam setiap langkah

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap langkah, identifikasi setiap langkah, input dan outputnya	Tanyakan How tentang setiap langkah ini: • Bagaimana orang mengetahui kapan melakukan langkah ini? • Laporan/form apa yang dihasilkan dari langkah ini? • Laporan/form apa yang dibutuhkan langkah ini? • Apa yang terjadi jika laporan/form ini tidak ada?

Langkah 4: Konfirmasikan use case

Aktifitas	Pertanyaan tipikal yang diajukan
Untuk setiap langkah, validasi bahwa semuanya lengkap dan benar	Minta user untuk mengeksekusi proses-proses yang tertulis dalam use case.

Berikut ini contoh use case yang diterapkan pada proses pemilihan CD pada sebuah rental CD:

Hasil langkah 1

Use case name: Take requests for CDs		ID number: 1	
Short description: This describes how customers can search the Web site and place requests to hold CDs in stock or place special orders			
Trigger: Customer searches Web and places request to hold a CD or to special order it			
Type: External Temporal			
Major Inputs: Description		Major Outputs: Description	
Search request	Customer	Special order	Special order DBs
CDs selected for request	Customer	Hold for in-stock CD	In-store hold DB
Customer information	Customer		
Marketing materials	Marketing DB		

Use case name: Maintain marketing materials		ID number: 2	
Short description: This adds, deletes, and modifies the additional marketing material from vendors (e.g. reviews, music clips)			
Trigger: Materials from vendors, distributors, wholesalers, record companies, and articles in trade magazines			
Type: External Temporal			
Major Inputs: Description		Major Outputs: Description	
Marketing materials	Vendor	Marketing materials	Marketing DB

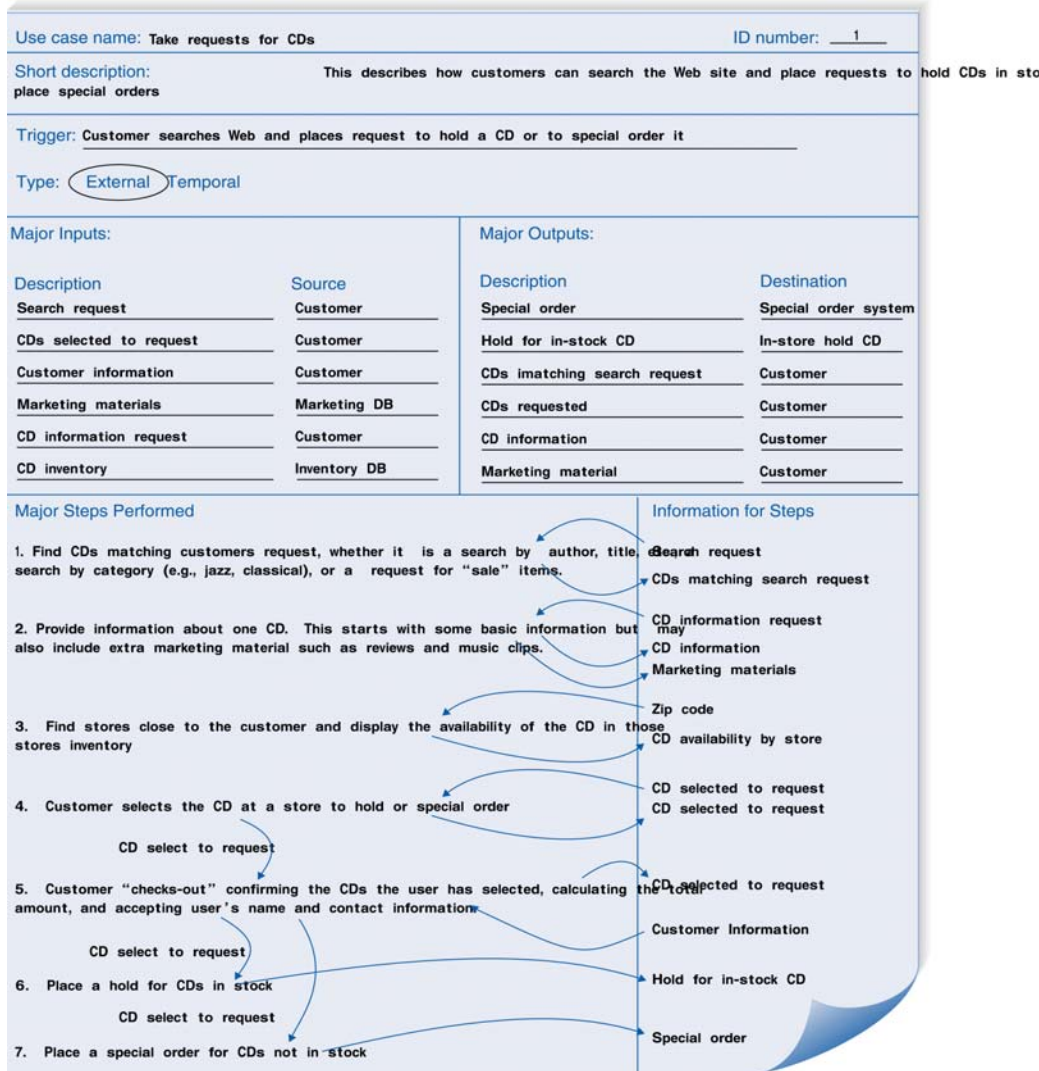
Use case name: Process in-store holds		ID number: 3	
Short description: This alerts the store staff to pull a requested CD from the shelves the special order section			
Trigger: Hold request from take request use case			
Type: External Temporal			
Major Inputs: Description		Major Outputs: Description	
Hold request	Take request use case	Hold label	Store staff

Major Steps Performed		Information for Steps
-----------------------	--	-----------------------

Hasil Langkah 2:

Use case name: <u>Take requests for CDs</u>		ID number: <u>1</u>																																			
Short description: <u>place special orders.</u>		This describes how customers can search the Web site and place requests to hold CDs in stock																																			
Trigger: <u>Customer searches Web and places requests to hold a CD or to special order it</u>																																					
Type: <u>External</u> Temporal																																					
Major Inputs <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Description</th> <th style="text-align: left; border-bottom: 1px solid black;">Source</th> </tr> </thead> <tbody> <tr> <td><u>Search request</u></td> <td><u>Customer</u></td> </tr> <tr> <td><u>CDs selected to request</u></td> <td><u>Customer</u></td> </tr> <tr> <td><u>Customer information</u></td> <td><u>Customer</u></td> </tr> <tr> <td><u>Marketing materials</u></td> <td><u>Marketing DB</u></td> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>		Description	Source	<u>Search request</u>	<u>Customer</u>	<u>CDs selected to request</u>	<u>Customer</u>	<u>Customer information</u>	<u>Customer</u>	<u>Marketing materials</u>	<u>Marketing DB</u>									Major Outputs <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Description</th> <th style="text-align: left; border-bottom: 1px solid black;">Destination</th> </tr> </thead> <tbody> <tr> <td><u>Special order</u></td> <td><u>Special order DB</u></td> </tr> <tr> <td><u>Hold for CDs in stock</u></td> <td><u>Hold DB</u></td> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>		Description	Destination	<u>Special order</u>	<u>Special order DB</u>	<u>Hold for CDs in stock</u>	<u>Hold DB</u>										
Description	Source																																				
<u>Search request</u>	<u>Customer</u>																																				
<u>CDs selected to request</u>	<u>Customer</u>																																				
<u>Customer information</u>	<u>Customer</u>																																				
<u>Marketing materials</u>	<u>Marketing DB</u>																																				
Description	Destination																																				
<u>Special order</u>	<u>Special order DB</u>																																				
<u>Hold for CDs in stock</u>	<u>Hold DB</u>																																				
Major Steps Performed 1. Find CDs matching customer's request, whether it is a search by author, title, etc., a search by category (e.g., jazz, classical), or a request for "sale" items. 2. Provide information about one CD. This starts with some basic information but may also include extra marketing material such as reviews and music clips. 3. Find stores close to the customer and display the availability of the CD in those stores' inventory. 4. Customer selects the CD at a store to hold or special order. 5. Customer "checks out " confirming the CDs the user has selected, calculating the total amount, and accepting user's name and contact information. 6. Place hold(s) for CD(s) in stock. 7 Place special order for CDs not in stock.		Information for Steps 																																			

Hasil langkah 3:



Use case name: <u>Take requests for CDs</u>		ID number: <u>1</u>																												
Short description: <u>This describes how customers can search the Web site and place requests to hold CDs in stock or place special orders</u>																														
Trigger: <u>Customer searches Web and places request to hold a CD or to special order it</u>																														
Type: <u>External</u> Temporal																														
Major Inputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>Search request</td> <td>Customer</td> </tr> <tr> <td>CDs selected for request</td> <td>Customer</td> </tr> <tr> <td>Customer information</td> <td>Customer</td> </tr> <tr> <td>Marketing materials</td> <td>Marketing DB</td> </tr> <tr> <td>CD information request</td> <td>Customer</td> </tr> <tr> <td>CD inventory</td> <td>Inventory DB</td> </tr> </tbody> </table>		Description	Source	Search request	Customer	CDs selected for request	Customer	Customer information	Customer	Marketing materials	Marketing DB	CD information request	Customer	CD inventory	Inventory DB	Major Outputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>Special Order</td> <td>Special order DBs</td> </tr> <tr> <td>Hold for in-stock CD</td> <td>In-store hold DB</td> </tr> <tr> <td>CDs matching search request</td> <td>Customer</td> </tr> <tr> <td>CDs requested</td> <td>Customer</td> </tr> <tr> <td>CD information</td> <td>Customer</td> </tr> <tr> <td>Marketing materials</td> <td>Customer</td> </tr> </tbody> </table>	Description	Destination	Special Order	Special order DBs	Hold for in-stock CD	In-store hold DB	CDs matching search request	Customer	CDs requested	Customer	CD information	Customer	Marketing materials	Customer
Description	Source																													
Search request	Customer																													
CDs selected for request	Customer																													
Customer information	Customer																													
Marketing materials	Marketing DB																													
CD information request	Customer																													
CD inventory	Inventory DB																													
Description	Destination																													
Special Order	Special order DBs																													
Hold for in-stock CD	In-store hold DB																													
CDs matching search request	Customer																													
CDs requested	Customer																													
CD information	Customer																													
Marketing materials	Customer																													

Major Step: Use case name: <u>Maintain marketing materials</u>		ID number: <u>2</u>																				
Short description: <u>This adds, deletes, and modifies the additional marketing material from vendors (e.g., reviews, music clips)</u>																						
Trigger: <u>Materials from vendors, distributors, wholesalers, record companies, and articles in trade magazines</u>																						
Type: <u>External</u> Temporal																						
Major Inputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>Marketing materials</td> <td>Vendor</td> </tr> <tr> <td>Marketing materials</td> <td>Marketing manager</td> </tr> <tr> <td>CD information</td> <td>CD DB</td> </tr> <tr> <td>Vendor information</td> <td>Vendor</td> </tr> </tbody> </table>		Description	Source	Marketing materials	Vendor	Marketing materials	Marketing manager	CD information	CD DB	Vendor information	Vendor	Major Outputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>Marketing materials</td> <td>Marketing DB</td> </tr> <tr> <td>Marketing material report</td> <td>Marketing manager</td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Description	Destination	Marketing materials	Marketing DB	Marketing material report	Marketing manager				
Description	Source																					
Marketing materials	Vendor																					
Marketing materials	Marketing manager																					
CD information	CD DB																					
Vendor information	Vendor																					
Description	Destination																					
Marketing materials	Marketing DB																					
Marketing material report	Marketing manager																					

Use case name: <u>Process in-store holds</u>		ID number: <u>3</u>																										
Short description: <u>This alerts the store staff to pull a request CD from the shelves and place it in the special order section</u>																												
Trigger: <u>Hold request from take request use case</u>																												
Type: <u>External</u> Temporal																												
Major Inputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>Hold request</td> <td>In-store hold DB</td> </tr> <tr> <td>Hold confirmation</td> <td>In-store staff</td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>		Description	Source	Hold request	In-store hold DB	Hold confirmation	In-store staff							Major Outputs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Description</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>Hold Label</td> <td>In-store staff</td> </tr> <tr> <td>Hold request alert</td> <td>In-store staff</td> </tr> <tr> <td>Hold confirmation</td> <td>In-store hold DB</td> </tr> <tr> <td>Inventory adjustment</td> <td>Inventory DB</td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Description	Destination	Hold Label	In-store staff	Hold request alert	In-store staff	Hold confirmation	In-store hold DB	Inventory adjustment	Inventory DB				
Description	Source																											
Hold request	In-store hold DB																											
Hold confirmation	In-store staff																											
Description	Destination																											
Hold Label	In-store staff																											
Hold request alert	In-store staff																											
Hold confirmation	In-store hold DB																											
Inventory adjustment	Inventory DB																											

Major Steps Performed	Information for Steps
-----------------------	-----------------------

BAB 9

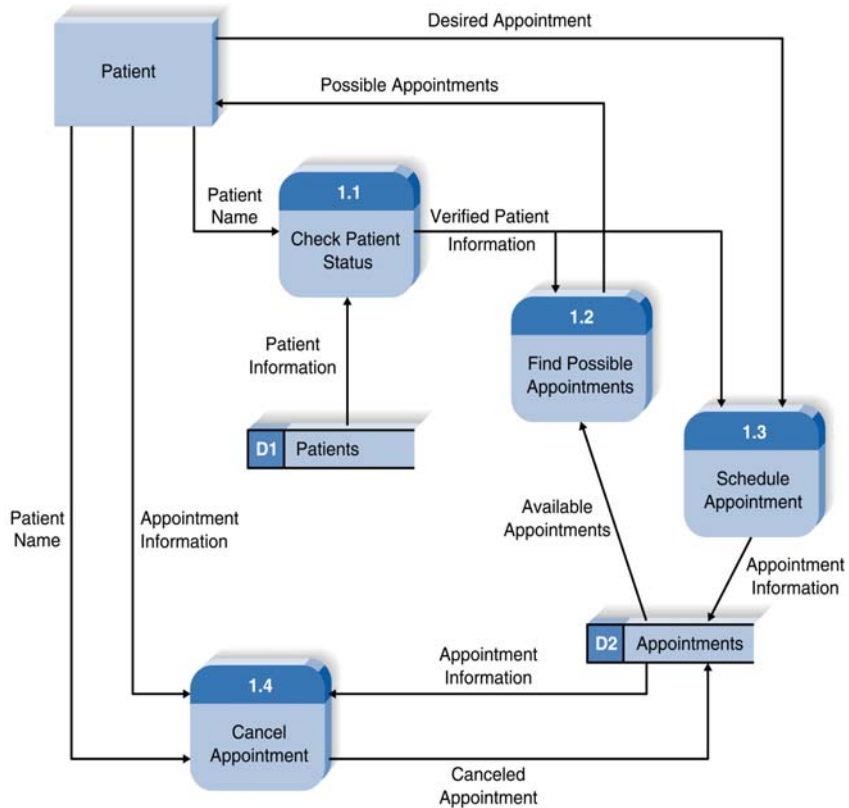
PROCESS MODELLING

9.1 Process model

Proses modelling adalah cara formal untuk menggambarkan bagaimana bisnis beroperasi. Mengilustrasikan aktivitas-aktivitas yang dilakukan dan bagaimana data berpindah diantara aktifitas-aktifitas itu. Ada banyak cara untuk merepresentasikan proses model, cara yang populer adalah dengan menggunakan data flow diagram (DFD). Ada 2 jenis DFD, Logical DFD yang menggambarkan proses tanpa menyarankan bagaimana mereka akan dilakukan. Kedua adalah Fisikal DFD yang menggambarkan proses model berikut implementasi pemrosesan informasinya.

9.2 DATA FLOW DIAGRAMS

Berikut ini contoh sebuah DFD yang menggambarkan proses pendaftaran pasien pada rumah sakit:



Gambar 9.1 DFD pendaftaran pasien pada rumah sakit

Untuk membaca suatu DFD kita harus memahami dulu , elemen-elemen yang menyusun suatu DFD. Ada 4 elemen yang menyusun suatu DFD yaitu:

Proses

Aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

Data flow

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

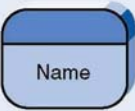
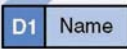
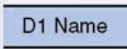
Data Store

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam data store. Aliran data diupdate atau ditambahkan ke data store.

External entity

Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.

Masing-masing elemen akan diberi lambang tertentu untuk membedakan satu dengan yang lain. Ada beberapa metode untuk menggambarkan elemen-elemen tersebut. Untuk lebih jelasnya dapat dilihat pada diagram berikut ini :

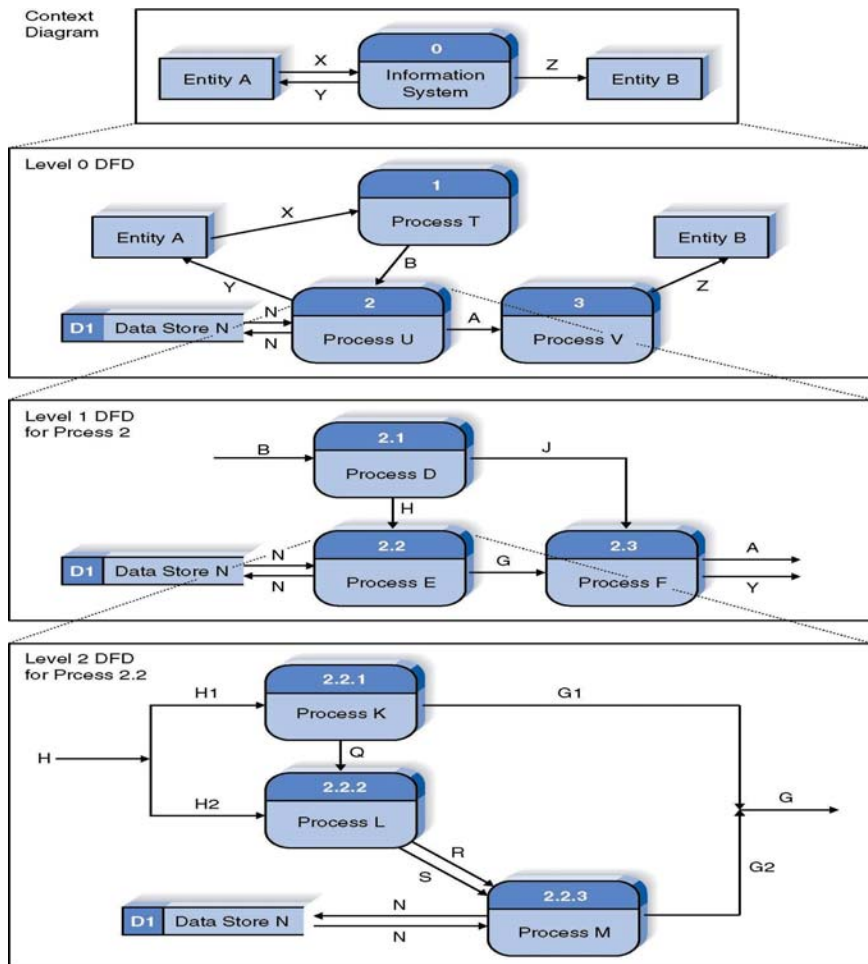
Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
Every <i>process</i> has A number A name (verb phrase) A description One or more output data flows Usually one or more input data flows	Label (name) Type (process) Description (what is it) Process number Process description (Structured English) Notes		
Every <i>data flow</i> has A name (a noun) A description One or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>data store</i> has A number A name (a noun) A description One or more input data flows Usually one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>external entity</i> has A name (a noun) A description	Label (name) Type (entity) Description Alias (another name) Entity description Notes		

Gambar 9.2 Elemen-elemen dari DFD dan lambangnya

9.3 Menggambarkan proses bisnis dengan menggunakan DFD

Proses bisnis biasanya terlalu kompleks untuk ditunjukkan dalam 1 DFD. Dekomposisi adalah proses untuk

menggambarkan sistem dalam hirarkie dari diagram DFD. Diagram anak menggambarkan proses yang lebih detail dibandingkan dengan diagram induk. Harus ada proses Balancing untuk menjamin informasi yang disajikan dalam satu level dari suatu DFD secara akurat direpresentasikan pada DFD level berikutnya. Adapun hirarki dari suatu DFD dapat dilihat pada diagram berikut ini:



Gambar 9.3 Hirarki penyusunan DFD

Keterangan :

Context Diagram: DFD pertama dalam proses bisnis. Menunjukkan konteks dimana proses bisnis berada. Menunjukkan semua proses bisnis dalam 1 proses tunggal (proses 0). Context Diagram juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem.

Level 0 Diagram: Menunjukkan semua proses utama yang menyusun keseluruhan sistem. Level ini juga menunjukkan komponen internal dari proses 0 dan menunjukkan bagaimana proses-proses utama direlasikan menggunakan data flow. Pada level ini juga ditunjukkan bagaimana proses-proses utama terhubung dengan entitas eksternal. Pada level ini juga dilakukan penambahan data store.

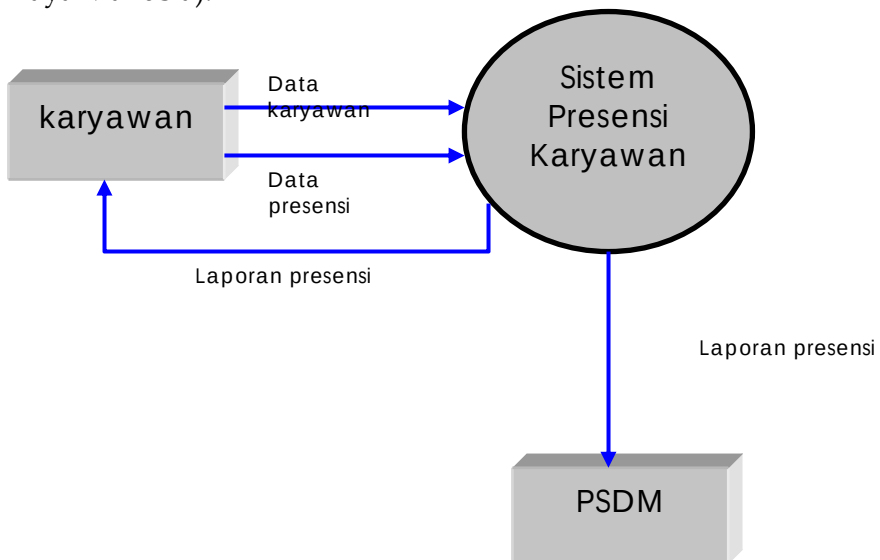
Level 1 Diagrams: Umumnya diagram level 1 diciptakan dari setiap proses utama dari level 0. Level ini menunjukkan proses-proses internal yang menyusun setiap proses-proses utama dalam level 0, sekaligus menunjukkan bagaimana informasi berpindah dari satu proses ke proses yang lainnya. Jika misalnya proses induk di pecah, katakanlah menjadi 3 proses anak, maka 3 proses anak ini secara utuh menyusun proses induk.

Level 2 Diagrams: Menunjukkan semua proses yang menyusun sebuah proses pada level 1. Bisa saja penyusunan DFD tidak mencapai level 2 ini. Atau mungkin harus dilanjutkan ke level berikutnya (level 3, level 4...dst)

Berikut ini contoh penyusunan DFD untuk sistem presensi karyawan yang menggunakan webcam untuk meng-capture wajah karyawan yang melakukan presensi. Sistem ini bekerja dengan prinsip pengenalan wajah. Perhatikan bahwa diagram yang dipakai adalah diagram yang mengacu pada metode De marco-Jordan.

Context Diagram

Pada context diagram, sistem digambarkan dengan sebuah proses saja, kemudian diidentifikasi entitas luar yang berinteraksi dengan proses tunggal tadi. Didapatkan 2 entitas luar yaitu karyawan dan bagian PSDM (Personalia dan Sumber Daya Manusia).



Gambar 9.4. context diagram untuk sistem presensi

Keterangan Gambar :

Karyawan : entitas luar karyawan STMIK AMIKOM Yogyakarta yang

diwajibkan melakukan presensi

PSDM : entitas luar Personalia dan Sumber Daya Manusia

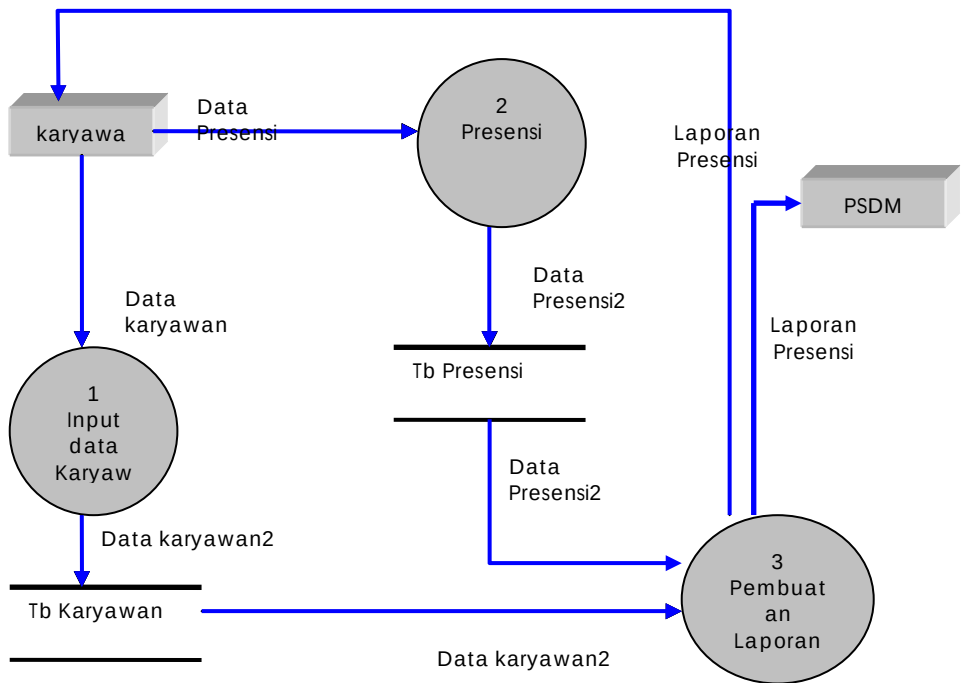
Data Karyawan: Meliputi No induk, Nama, Bagian, Jabatan, Gaji

Data Presensi: Meliputi wajah karyawan yang melakukan presensi

Laporan presensi: Laporan presensi karyawan perbulan berisi no induk, nama, bulan, tahun dan total potongan.

DFD Level 1

Pada level ini, proses tunggal dari context diagram dipecah menjadi 3 proses yang lebih terperinci yaitu proses input data karyawan, presensi dan pembuatan laporan presensi.



Gambar 9.5 DFD level 1 untuk sistem presensi

Keterangan Gambar:

Karyawan : entitas luar karyawan STMIK AMIKOM Yogyakarta yang

diwajibkan melakukan presensi

PSDM : entitas luar Personalia dan Sumber Daya Manusia

Data Karyawan: Meliputi No induk, Nama, Bagian, Jabatan, Gaji

Data Karyawan2: Meliputi Id karyawan, No induk, Nama, Bagian, Jabatan, Gaji

Data Presensi: Meliputi wajah karyawan yang melakukan presensi

Data Presensi2: Meliputi id presensi, citra wajah karyawan, tanggal dan waktu presensi dilakukan.

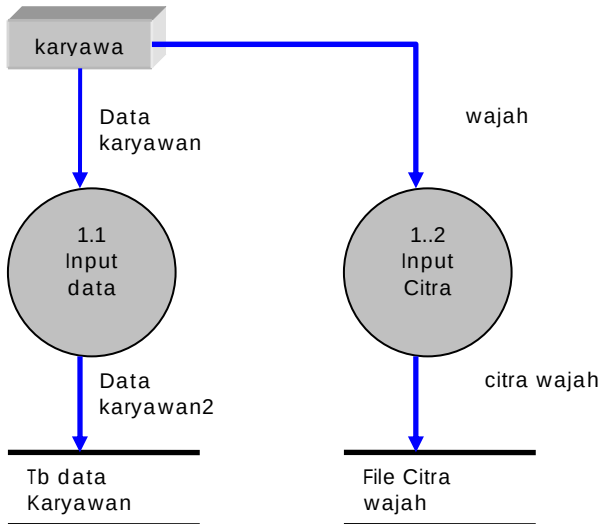
Tb Karyawan : Tabel Karyawan, untuk menyimpan data karyawan

Tb Presensi : Tabel Presensi untuk menyimpan data presensi

Laporan presensi: Laporan presensi karyawan perbulan berisi no induk, nama, bulan, tahun dan total potongan.

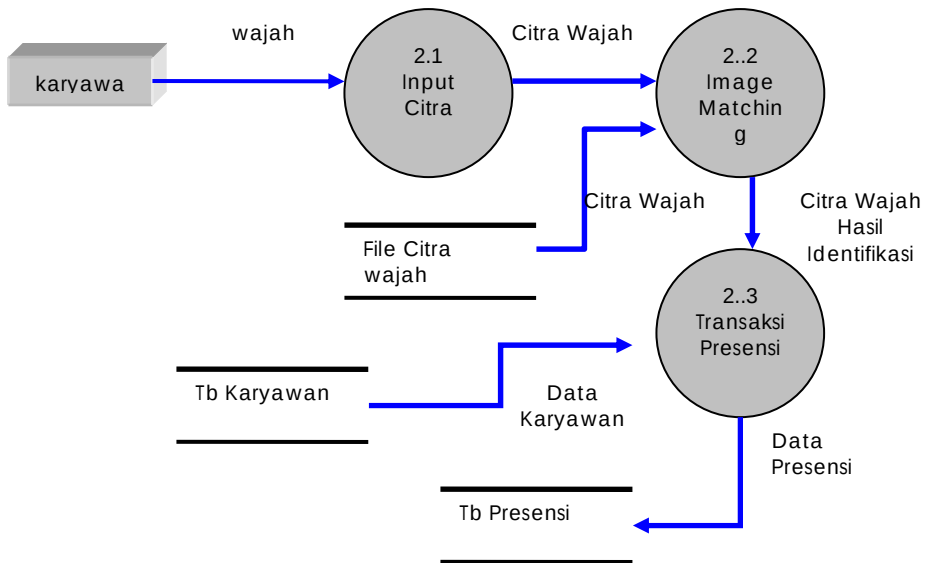
DFD Level 2

Pada DFD Level 2, proses Input data Karyawan diperinci menjadi 2 proses yaitu proses input data, digunakan untuk menginputkan data karyawan dan proses input citra, proses meng-capture wajah karyawan sehingga didapatkan citra wajah. Pada Level ini didapatkan juga beberapa penyimpanan data yaitu Tb Data Karyawan untuk menyimpan data-data karyawan dan File Citra Wajah Karyawan untuk menyimpan citra wajah karyawan.



Gambar 9.6 DFD Level 2 untuk proses Input data Karyawan

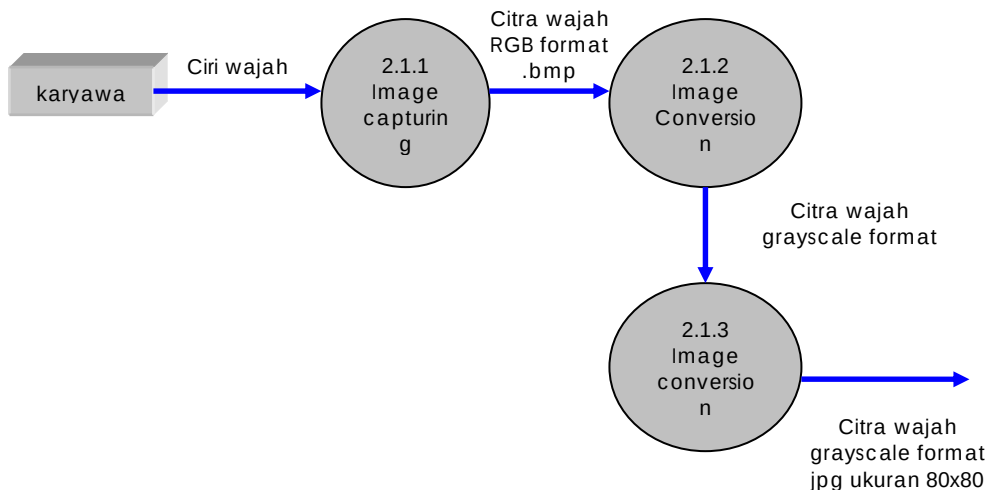
Untuk proses Presensi, dapat diperinci menjadi proses input citra yang menerima input wajah dari karyawan yang dicapture dengan webcam, Image matching yang bertugas mencari citra wajah yang sesuai pada database dan Transaksi presensi yang memproses data-data saat transaksi karyawan dilakukan.



Gambar 9.7 DFD Level 2 untuk proses Presensi

DFD Level 3

Untuk proses input citra baik pada proses input data karyawan (proses 1.2) maupun proses presensi (proses 2.1) mempunyai rincian proses yang sama, sehingga untuk penyusunan DFD level 3 akan diambil proses 2.1 saja sebagai berikut:



Gambar 9.8 DFD Level 3 untuk proses input citra

BAB 10

DATA MODELLING

10.1 DATA MODEL

Data model adalah cara formal untuk menggambarkan data yang digunakan dan diciptakan dalam suatu sistem bisnis. Model ini menunjukkan orang, tempat atau benda dimana data diambil dan hubungan antar data tersebut. Data modelling juga dibedakan menjadi 2: *Logical data model*: Menunjukkan pengaturan data tanpa mengindikasikan bagaimana data tersebut disimpan, dibuat dan dimanipulasi. *Physical data model*: Menunjukkan bagaimana data akan disimpan sebenarnya dalam database atau file. Penyusunan model data harus seimbang dengan model proses. Salah satu cara pemodelan data adalah dengan ERD (Entity Relationship Diagram).

10.2 THE ENTITY-RELATIONSHIP DIAGRAM (ERD)

Apakah ERD itu?

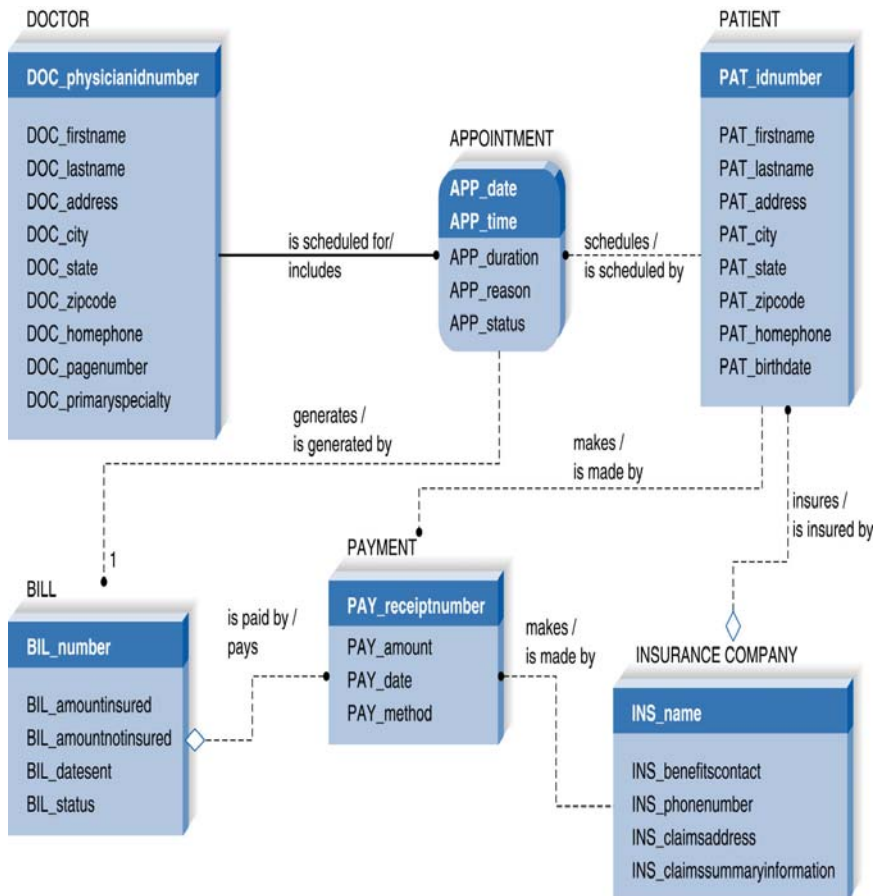
- Gambar yang menunjukkan informasi dibuat disimpan dan digunakan dalam sistem bisnis.
- Entitas biasanya menggambarkan jenis informasi yang sama
- Garis yang menghubungkan antar entitas menunjukkan hubungan antar data

- ERD juga bisa digunakan untuk menunjukkan aturan-aturan bisnis

Menggunakan ERD untuk menunjukkan aturan bisnis

- Aturan bisnis adalah batasan yang harus diikuti ketika sistem beroperasi.
- Simbol ERD hanya menunjukkan satu instance dari entitas harus ada sebelum instance lain dari suatu entitas. Sebagai contoh: Seorang dokter harus ada sebelum perjanjian ketemu dengan dokter dibuat.
- Simbol ERD dapat menunjukkan ketika salah satu instance dari suatu entitas dapat direlasikan dengan satu atau lebih instance dari entitas lainnya. Contohnya: satu dokter bisa memiliki banyak pasien, satu pasien bisa jadi hanya memiliki satu dokter utama.
- Simbol ERD juga menunjukkan ketika eksistensi dari instance suatu entity adalah opsional untuk sebuah relasi dengan instance dari suatu entitas. Contohnya: Pasien mungkin memiliki atau mungkin tidak memiliki biaya asuransi.

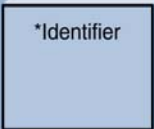
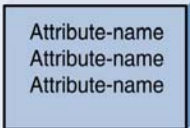
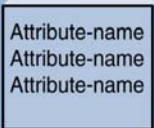
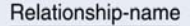
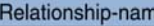
Berikut ini contoh sebuah ERD



Gambar 10.1 Contoh Diagram ERD

Elemen-elemen ERD

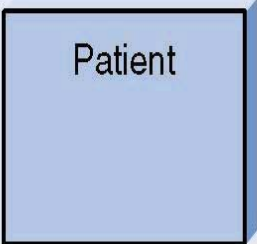
Seperti data flow diagram, ERD juga menggunakan simbol-simbol khusus untuk menggambarkan elemen-elemen ERD. Berikut ini simbol-simbol yang digunakan dalam ERD:

	IDEFIX	Chen	Information Engineering
An ENTITY: ✓ Is a person, place, or thing ✓ Has a singular name spelled in all capital letters ✓ Has an identifier ✓ Should contain more than one instance of data	ENTITY-NAME 	ENTITY-NAME 	ENTITY-NAME 
An ATTRIBUTE: ✓ Is a property of an entity ✓ Should be used by at least one business process ✓ Is broken down to its most useful level of detail	ENTITY-NAME 		ENTITY-NAME 
A RELATIONSHIP: ✓ Shows the association between two entities ✓ Has a parent entity and a child entity ✓ Is described with a verb phrase ✓ Has cardinality (1 : 1, 1 : N, or M : N) ✓ Has modality (null, not null) ✓ Is dependent or independent			

Gambar 10.2 Elemen-elemen dari ERD

Keterangan :**Entity**

Entitas bisa berupa orang, kejadian, atau benda dimana data akan dikumpulkan. Untuk menjadi sebuah entity, suatu objek harus menampilkan beberapa kali event. Sebagai contoh : Jika sebuah firma hanya memiliki 1 gudang, maka gudang tersebut bukan entitas. Tetapi jika perusahaan memiliki banyak gudang, maka gudang bisa menjadi entitas suatu entitas jika perusahaan ingin menyimpan data untuk setiap instance dari gudang.

Entity	Example Instances
 Patient	John Smith Susan Jones Peter Todd Dale Turner Pat Turner

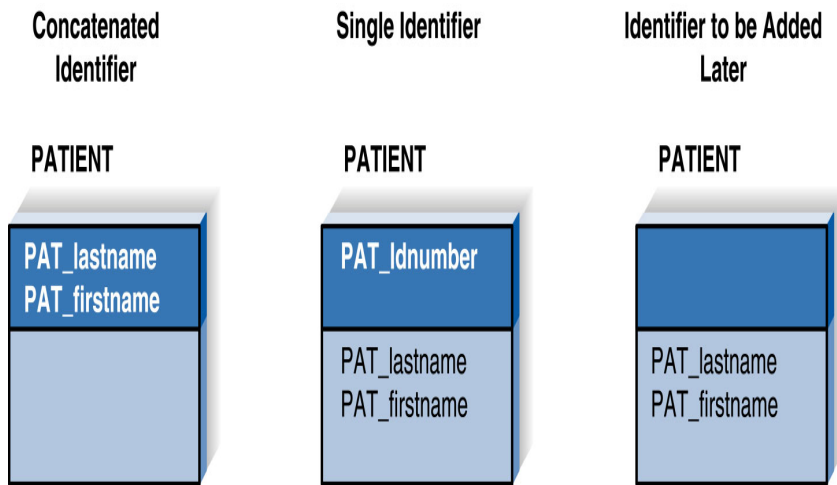
Gambar 10.3 Contoh Entitas dan Instance

Atribut

- Informasi yang diambil tentang sebuah entitas
- Hanya yang digunakan oleh organisasi yang dimasukkan dalam model
- Nama atribut harus merupakan kata benda
- Kadang nama entitas diletakkan di depan nama atribut untuk ketelitian.

Identifier

- Satu atau lebih atribut dapat menjadi identifier entitas, yang secara unik mengidentifikasi setiap instance dari entitas.
- Concatenated identifier (identifier gabungan) terdiri dari beberapa atribut.
- Identifier bisa saja artifisial, seperti dengan membuat ID number
- Identifier tidak akan dikembangkan sampai fase desain.



Gambar 10.4 tipe-tipe identifier

Berikut ini contoh penggunaan identifier dalam sebuah case repository untuk suatu atribut

The image shows a software window titled "Case repository" with a light blue border. Inside, there is a form with several fields, each with a label on the left and a text input area on the right. The fields are: "Attribute Name:" with the value "PAT_idnumber"; "Aliases:" with the value "Patient ID Number, Social Security Number, SSN"; "Entity:" with the value "PATIENT"; "Description:" with the text "The ID number assigned by the doctor's office that uniquely identifies a patient."; "Data Type:" with the value "Alpha-Numeric"; "Identifier:" with the value "Yes"; and two "Notes:" sections. The first note says: "Usually, the patient's social security number is used. When a patient does not have a social security number, the number assigned by the patient's insurance provider is used. If the patient does not have an insurance provider, a driver's license number is used." The second note says: "If the patient's insurance provider number is used and it is greater than 9 characters, the first 9 characters are used to create this attribute. Zeros are added to the number if it is less than 9 characters." A vertical scrollbar is on the right side of the form.

Attribute Name:	PAT_idnumber
Aliases:	Patient ID Number, Social Security Number, SSN
Entity:	PATIENT
Description:	The ID number assigned by the doctor's office that uniquely identifies a patient.
Data Type:	Alpha-Numeric
Identifier:	Yes
Notes:	Usually, the patient's social security number is used. When a patient does not have a social security number, the number assigned by the patient's insurance provider is used. If the patient does not have an insurance provider, a driver's license number is used.
Notes:	If the patient's insurance provider number is used and it is greater than 9 characters, the first 9 characters are used to create this attribute. Zeros are added to the number if it is less than 9 characters.

Gambar 10.5 Case repository untuk atribut patient_SSN

Relationships

- Hubungan antar entitas
- Entitas pertama dalam relationship disebut entitas induk, entitas kedua disebut sebagai entitas anak.
- Relationship harus mempunyai nama yang berupa kata kerja
- Relationship berjalan 2 arah

Sebagai contoh, jika dimiliki dua entitas yaitu buku dan toko buku maka bisa dibuat beberapa relationship, diantaranya:

- Toko buku memesan buku
- Toko buku menampilkan buku
- Toko buku menstock buku
- Toko buku menjual buku
- Toko Buku mengembalikan buku.

Relationship *memesan*, *menampilkan*, *menstock*, *menjual* dan *mengembalikan* mendefinisikan hubungan yang relevan antara buku dan toko buku.

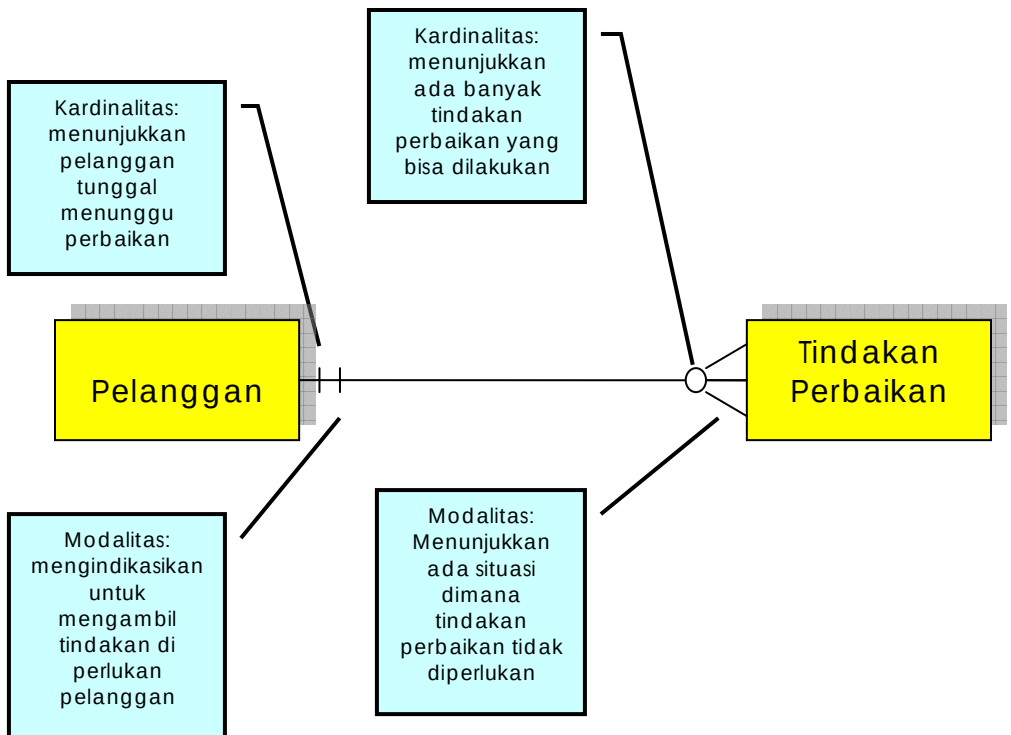
Cardinalitas

- Kardinalitas mengacu pada berapa kali instance dari satu entitas dapat berelasi dengan instance lain di entitas yang berbeda.
- Satu instance dalam 1 entitas mengacu pada satu dan hanya satu instance pada entitas lainnya (1:1)
- Satu instance dalam suatu entitas mengacu ke satu atau lebih instance yang berelasi (1:N)
- Satu atau lebih instance dalam suatu entitas mengacu pada satu atau lebih instance pada entitas yang berelasi (M:N).

Modalitas

- Mengacu pada apakah suatu instance dari entitas anak dapat ada tanpa suatu relasi dengan instance dari entitas induk atau tidak.
- **Not Null** berarti bahwa suatu instance pada entitas yang berelasi harus ada untuk suatu instance dari entitas lain untuk disebut valid
- **Null** berarti bahwa tidak ada instance dalam entitas yang berelasi yang diperlukan untuk instance pada relasi lain untuk dikatakan valid.

Berikut ini contoh penggunaan kardinalitas dan modalitas dalam suatu Relationship:



Data Dictionary dan Metadata

- Metadata adalah informasi yang tersimpan yang berisi komponen dari model data
- Metadata disimpan dalam data dictionary sehingga bisa dibagi dengan developer dan user melalui SDLC
- Data dictionary yang lengkap dan bisa dibagi membantu meningkatkan kualitas dari sistem yang sedang dikembangkan.

berikut ini frame umum dari sebuah metadata:

Sebuah entitas yang bisa dideskripsikan sebagai:	Sebuah Atribut yang bisa dideskripsikan sebagai:	Sebuah Relationship yang bisa dideskripsikan sebagai:
Nama Definisi Catatan Khusus Kontak User Kontak Analis	Nama Deskripsi Alias Contoh nilai Nilai yang bisa diterima Format Tipe Catatan khusus	Frase kata kerja Entitas induk Entitas anak Definisi Kardinalitas Modalitas

Gambar 10.6 Frame Metadata

MEMVALIDASI ERD

Untuk membuat ERD, diperlukan latihan dan jam terbang, Ada beberapa pedoman yang perlu diperhatikan untuk membuat ERD diantaranya:

- Entitas harus memiliki banyak kejadian/realitas
- Hindari penggunaan atribut yang tidak perlu
- Berilah label yang jelas untuk semua komponen
- Pasangkan kardinalitas dan modalitas yang jelas dan benar
- Pecah atribut menjadi level serendah mungkin yang diperlukan

- Label harus merefleksikan istilah-istilah bisnis yang umum
- Asumsi harus disebutkan dengan jelas.

Normalisasi

Normalisasi adalah teknik yang digunakan untuk memvalidasi model data. Serangkaian aturan diberlakukan pada data model logik untuk meningkatkan pengaturannya. Biasanya digunakan 3 aturan

Langkah-langkah normalisasi

Berikut ini adalah langkah-langkah yang digunakan untuk melakukan normalisasi terhadap data model yang telah kita peroleh :

0 normal form	
Apakah ada atribut yang memiliki nilai ganda untuk satu instance dari suatu entitas?	Ya: Hilangkan atribut yang berulang dan grup yang berulang. Buat entitas yang menggambarkan atribut-atributnya. Biasanya diperlukan penambahan relasi untuk menghubungkan entitas baru dan lama. Tidak : Model data ada dalam bentuk 1NF (1 Normal Form)
1 Normal Form	
Apakah identifier terdiri dari lebih dari satu atribut? Jika ya. Apakah nilai atribut tergantung hanya pada satu bagian dari identifier?	Ya: Hilangkan ketergantungan parsial. Hilangkan atribut suatu entitas dimana nilai-nilai mereka tergantung pada ke semua identifier. Biasanya

	diperlukan penambahan relasi untuk menghubungkan entitas baru dan lama. Tidak: Model data dalam bentuk 2NF (2 Normal Form)
2 Normal Form	
Apakah ada nilai-nilai atribut yang tergantung pada entitas yang bukan identifier?	Ya: Hilangkan ketergantungan transitif atau entitas turunan. Pindahkan atribut ke entitas dimana atribut tersebut bergantung pada identifier. Biasanya diperlukan penambahan relasi untuk menghubungkan entitas baru dan lama. Tidak : Model data ada dalam bentuk 3NF (3 Normal Form)
3 Normal Form	

Unnormalized Entity

Mulai dengan entitas dari model data logik

SPECIAL ORDER

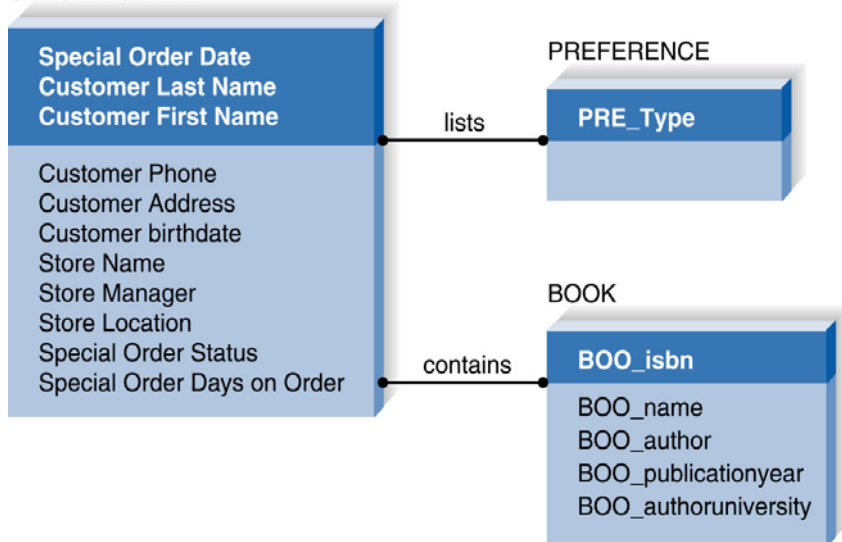
Special Order Date
Customer Last Name
Customer First Name

Customer phone
Customer Address
Customer birthdate
Customer Book Preferences
Book ISBN1
Book Name1
Book Author1
Book Publication Year1
Book Author University1
Book ISBN2
Book Name2
Book Author2
Book Publication Year2
Book Author University2
Book ISBN3
Book Name3
Book Author3
Book Publication Year3
Book Author University3
Store Name
Store Manager
Store Location
Special Order Status
Special Order Days on Order

First Normal Form (1NF)

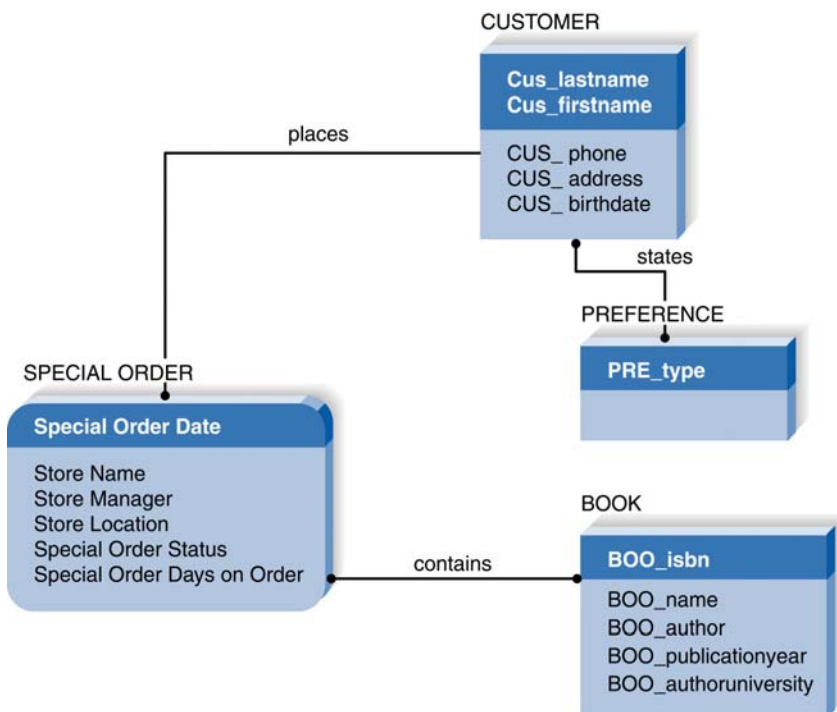
Cari kelompok-kelompok entitas yang berulang dan pisahkan ke dalam entitas yang berbeda.

SPECIAL ORDER



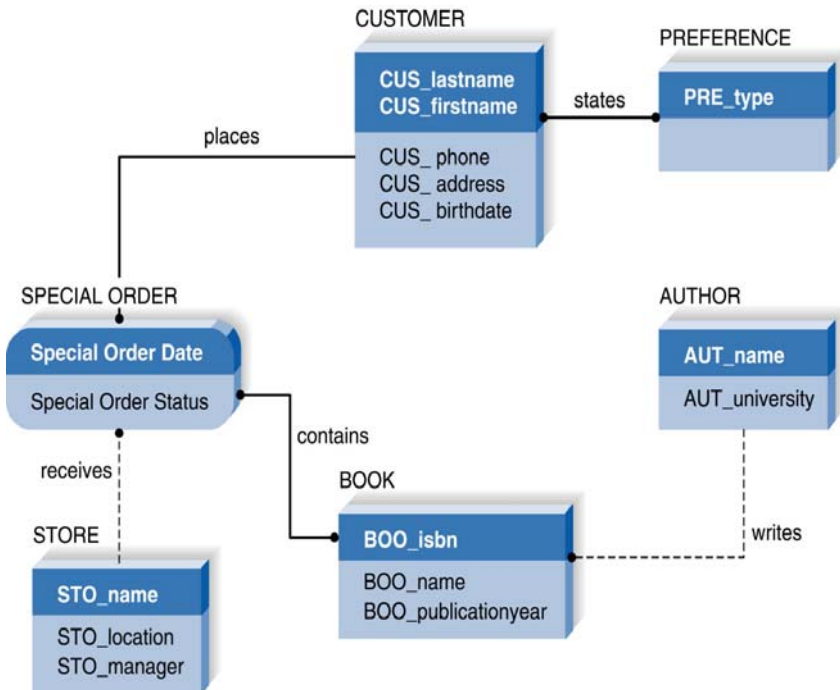
Second Normal Form (2NF)

Jika ada entitas yang memiliki identifier gabungan, cari atribut yang hanya bergantung pada identifier. Jika ditemukan pindahkan ke entitas baru.



Third Normal Form (3NF)

Cari atribut yang bergantung hanya pada atribut lain yang bukan merupakan identifier. Jika ditemukan pindahkan menjadi entitas baru, juga pindahkan atribut-atribut yang dirasa perlu dipindahkan.



Menyeimbangkan ERD dengan DFD

Semua aktifitas analisa merupakan aktifitas-aktifitas yang saling berkaitan, termasuk proses modelling dan data modelling. Proses model akan berisi dua hal data flow dan data store. Komponen data dalam DFD ini harus diseimbangkan dengan ERD dimana Data store diseimbangkan dengan entitas dan elemen data diseimbangkan dengan atribut. Untuk mempermudah, banyak tool CASE yang menyediakan fitur untuk mengecek ketidakseimbangan.

BAB 11

DESAIN SISTEM

11.1 FASE DESAIN

Fase desain adalah fase dimana didefinisikan bagaimana akan membangun sistem. Pada tahapan ini detail dari deskripsi teknis tentang bagaimana membangun sistem didefinisikan. Biasanya orang mengatakan hasil dari fase desain adalah spesifikasi sistem. Spesifikasi sistem adalah produk akhir dari fase desain, yang menjelaskan dengan tepat perancangan sistem seperti apa yang akan diimplementasikan oleh tim desain selama fase implementasi.

Langkah-langkah Fase Desain:

Adapun langkah-langkah fase desain dapat dijelaskan sebagai berikut:

- Menyajikan alternatif desain (membuat sendiri, membeli atau outsourcing)
- Mengubah proses-proses logik dan model data ke dalam model fisik
- Merancang arsitektur sistem
- Membuat pemilihan hardware dan software
- Merancang bagaimana data akan disimpan
- Mendesain program untuk proses yang bersesuaian
- Membuat spesifikasi sistem

Dalam fase desain sebaiknya dihindari kesalahan-kesalahan klasik seperti :

- Pengurangan waktu perancangan
- Penambahan fitur dari apa yang sudah ditentukan dalam fase analisis
- Mengganti tool dan teknologi pada pertengahan proyek.

11.2 STRATEGI DESAIN

Ada 3 strategi umum yang biasa dipakai dalam pengembangan sistem:

- *Custom development* : membangun sistem dari nol dalam rumah produksi sendiri
- *Membeli paket software* dan melakukan beberapa kostumisasi
- Melakukan *outsourcing* ke pihak ketiga.

Costum development

Strategi desain ini punya satu keunggulan utama, yaitu biaya pengembangannya dapat kita kendalikan sendiri dan relatif mudah. Rincian kelebihan dan keunggulannya adalah :

Kelebihan:

- Tingkat fleksibilitas dan kreatifitas lebih tinggi

- Cocok dan konsisten dengan teknologi dan standar yang ada
- Bisa membangun kemampuan dan pengetahuan fungsional dari tenaga kerja yang kita miliki.

Kekurangan :

- Membutuhkan waktu dan usaha yang signifikan
- Mungkin membutuhkan keahlian atau pengetahuan fungsional yang belum dikuasai
- Kadang-kadang biaya bisa membengkak
- Sering membutuhkan waktu lebih banyak dari jadwal yang telah tersedia.
- Ada resiko kegagalan proyek.

Pembelian software Paket

Saat ini telah tersedia berbagai software paket untuk banyak kebutuhan bisnis. Dalam paket software, program yang digunakan telah dites dan diujikan serta bisa menghemat waktu dan biaya pengembangan. Tetapi software paket sangat jarang sesuai 100% dengan kebutuhan bisnis. Untuk mengatasi ini beberapa software paket memungkinkan user untuk melakukan kostumisasi. Kelemahan utamanya adalah paket software bisa jadi tidak bisa diintegrasikan dengan sistem yang sudah ada jika dikembangkan oleh perusahaan yang berbeda.

Outsourcing

Outsourcing adalah menyewa vendor luar, developer atau service provider untuk menyelesaikan tahapan desain . Keuntungan utama adalah mengurangi waktu produksi dan ada nilai tambah diantaranya bisa menerima lebih banyak proyek dalam periode yang sama dan melakukan outsourcing pada pihak-pihak ketiga. Tetapi strategi ini ada beberapa resikonya, diantaranya:

- Kehilangan informasi penting
- Kehilangan kendali akan pengembangan di masa depan
- Kehilangan kesempatan untuk belajar

Beberapa pedoman yang perlu diperhatikan untuk melakukan outsourcing adalah:

- Pelihara jalur komunikasi tetap terbuka dengan pihak ketiga.
- Definisikan dan stabilisasi requirement sebelum menandatangani kontrak
- Pandang kerjasama sebagai hubungan partnership
- Pilihlah vendor, developer dan service provider dengan hati-hati
- Tugaskan seseorang untuk memelihara kerjasama
- Jangan melakukan outsourcing apa yang tidak kita mengerti
- Tekankan pada requirement yang fleksibel, hubungan jangka panjang dan kontrak jangka pendek.

Memilih strategi desain

Dalam memilih strategi desain, pertimbangkan hal-hal berikut ini :

- Kebutuhan bisnis
- Pengalaman sendiri
- Kemampuan proyek
- Manajemen proyek
- Frame waktu yang tersedia

Untuk lebih jelasnya, perhatikan tabel berikut ini :

	Custom Development	Software Paket	Outsourcing
Business need	kebutuhan bisnis unik	kebutuhan bisnis umum	Kebutuhan bisnis bukan core/inti dari bisnis
In-house experience	Tenaga terlatih dan berkemampuan baik banyak dimiliki	Ikemampuan standar untuk kostumisasi dimiliki	SDM dengan Kemampuan teknis yang terlatih tidak dimiliki
Project skills	Punya keinginan ingin meningkatkan Skill SDM yang dimiliki	Skill bukan hal atau isu yang strategis	Outsourcing adalah keputusan strategis
Project management	Memiliki manajer proyek yang handal dan teknologi yang telah terbukti	Manajer proyek bisa mengkoordinasikan usaha-usaha dari vendor	Proyek manajer berkemampuan tinggi pada level manajemen tertentu level
Time frame	Waktu yang tersedia fleksibel	Waktu yang tersedia pendek	Waktu yang tersedia bisa pendek bisa panjang

BAB 12

DISAIN ARSITEKTUR

12.1 Desain Arsitektur

Desain arsitektur adalah perencanaan bagaimana sistem akan di distribusikan di antara komputer-komputer yang ada dan software dan hardware apa yang akan digunakan untuk masing-masing komputer. Dalam desain arsitektur ada 2 hal yang perlu diperhatikan:

- Spesifikasi dari software dan hardware
- Deskripsi detail dari komponen software/hardware untuk mempermudah pihak yang akan membeli software/hardware tersebut.

12.2 Elemen dari desain arsitektur

Dari sudut pandang software komponen arsitektural dapat dibagi menjadi:

Data Storage : komponen yang digunakan untuk menyimpan data

Data access logic: prosedur yang digunakan untuk mengakses data yang disimpan dalam data storage.

Application Logic: bagian logika pemrosesan dalam lapis aplikasi

Presentation Logic : komponen pemrosesan tampilan dan perintah dari user.

12.3 Tujuan desain arsitektur

Tujuan dari desain arsitektur adalah untuk menentukan komponen software mana yang akan di-assign ke hardware yang tersedia. Adapun pilihan hardware yang tersedia adalah:

Klien :

Input/output device yang dioperasikan oleh user
PC, Laptop, peralatan mobile, telepon selular

Pilihan untuk Klien:

- Terminal
- Microcomputer
- Minicomputer

Server:

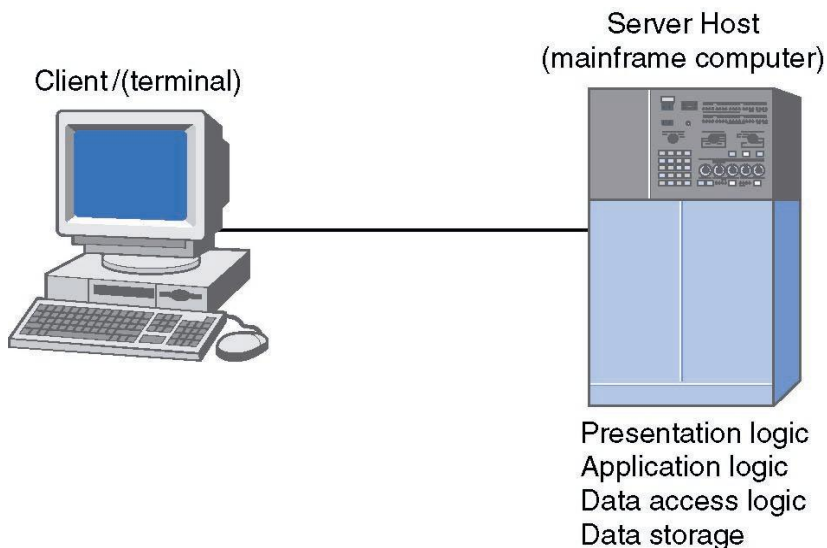
Komputer lebih canggih untuk menyimpan software
Bisa diakses oleh banyak user.

Pilihan untuk server :

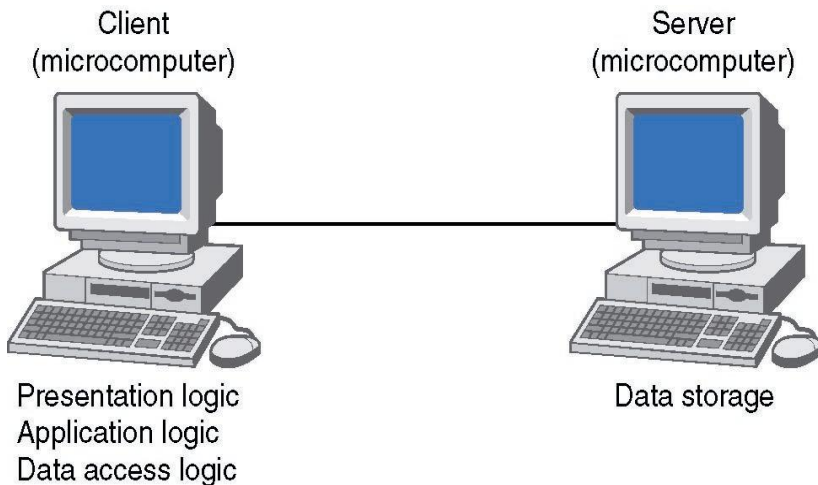
- Mainframe
- Microcomputer (PC)
- Terminal khusus seperti ATM, kiosk dll

12.4 Pilihan arsitektur

a. Server-Based Architecture

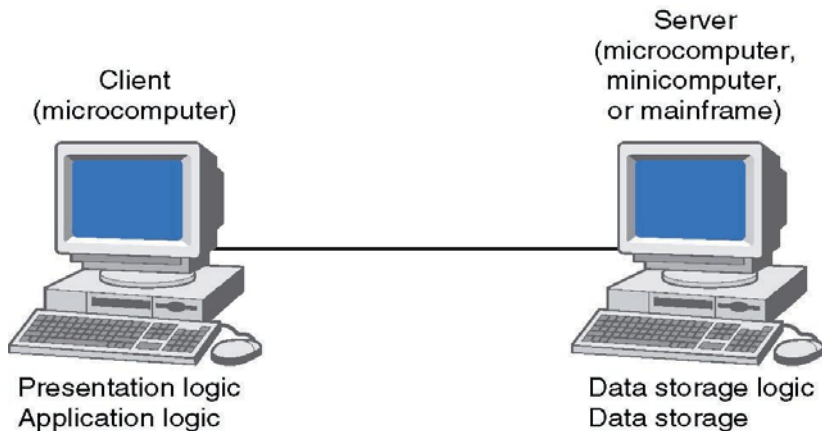


Pada arsitektur jenis ini, 4 komponen software diletakkan pada bagian yang sama yaitu server. Terminal yang tersedia hanya digunakan sebagai dumb terminal, perpanjangan layanan dari server saja. Pada arsitektur ini data relatif aman tetapi beban kerja server sangat berat, karena semua komponen dijalankan di server.

b. Client-Based Architecture

Pada arsitektur ini, server hanya berperan sebagai data storage. Komputer klien yang digunakan untuk mengolah data yang dikirim oleh server. Beban kerja server sangat ringan tetapi keamanan data sangat tergantung kinerja dari klien.

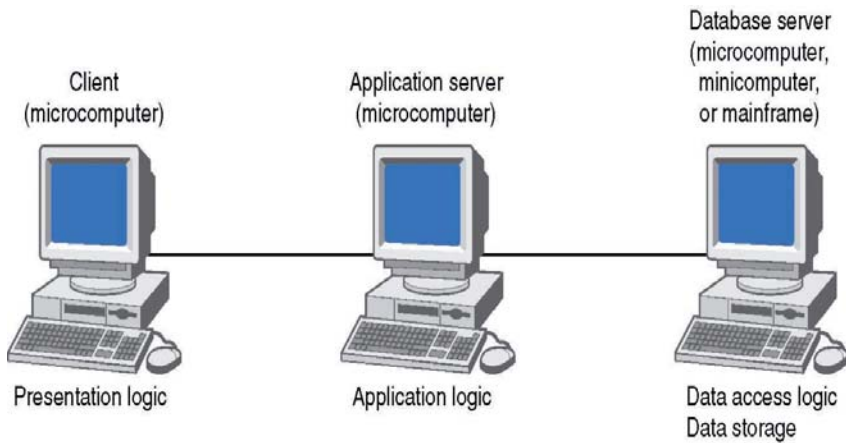
c. Client-Server Architecture (Two-Tiered)



Pada arsitektur ini data dan prosedur pengaksesan data dilayani oleh server dan masalah tampilan dan logika aplikasi dilayani oleh user. Pada arsitektur ini keamanan data cukup terjamin karena prosedur bagaimana mengakses data oleh klien ditangani oleh server. Arsitektur ini biasa disebut arsitektur 2-tier.

- ❑ Keunggulan
 - Scalable
 - Meningkatkan modularitas dari sistem berbasis web
 - Tidak ada titik pusat kegagalan
- ❑ Kelemahan
 - Kompleksitas
 - Perlu bahasa dan teknik pemrograman baru
 - Lebih kompleks untuk di update

d. Three-Tiered Client-Server Architecture



Pada arsitektur ini ditambahkan hardware untuk menjalankan lapis application logic, sehingga keamanan pengaksesan data menjadi lebih terjamin.

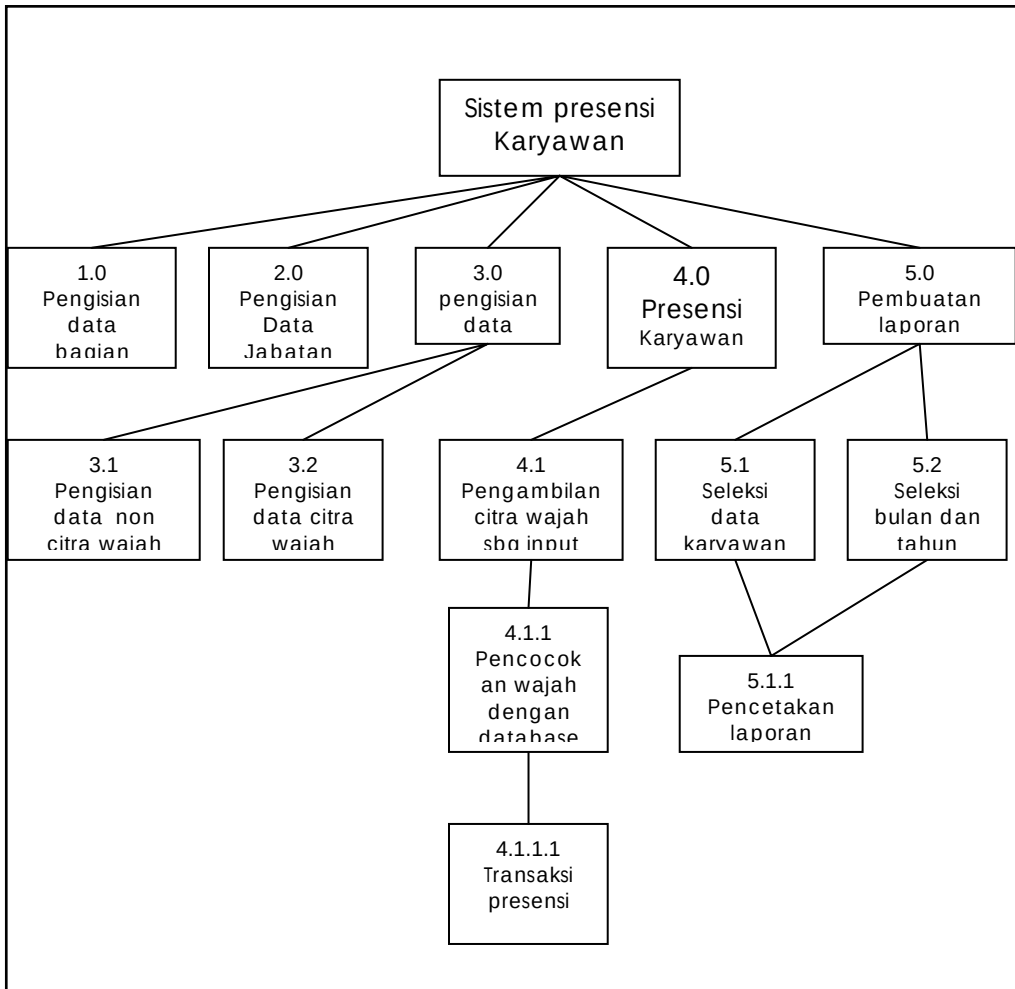
BAB 13

HIRARCHY INPUT OUTPUT CHART (HIPO)

HIPO merupakan teknik untuk mendokumentasikan pengembangan suatu sistem yang dikembangkan oleh IBM. HIPO dapat digunakan untuk memenuhi kebutuhan dari beberapa pemakai yang menggunakannya untuk kepentingan yang berbeda-beda. Antara lain sebagai berikut:

- Seorang manajer dapat menggunakan dokumentasi HIPO untuk memperoleh gambaran umum sistem
- Seorang programmer menggunakannya untuk menentukan fungsi-fungsi dalam program yang dibuatnya.
- Programmer juga dapat menggunakan dokumen ini untuk mencari fungsi-fungsi yang dimodifikasi dengan cepat

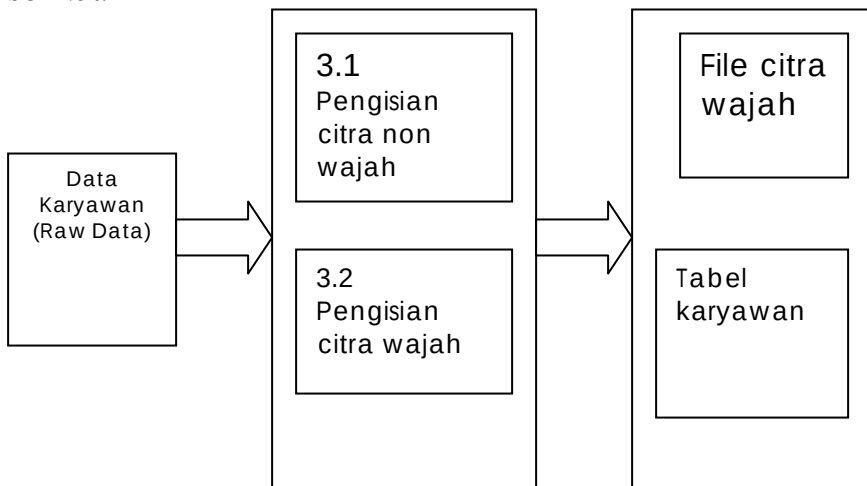
Teknik ini mempunyai beberapa tujuan utama. Pertama dapat dibuat sebuah struktur yang menggambarkan hubungan antar fungsi dalam program secara hirarkis. Perhatikan gambar berikut ini:



Gambar 13.1 Diagram DIV dari HIPO

Sasaran kedua adalah untuk menentukan fungsi-fungsi apa saja yang harus ada dalam sistem yang dikembangkan. Dari gambar diatas terlihat jelas fungsi apa saja yang harus dibuat. Sasaran ketiga adalah untuk mendapatkan gambaran input dari fungsi

dan output apa yang dihasilkan. Sebagai contoh perhatikan modul fungsi no 3. Diagram HIPO yang bisa dibuat sebagai berikut:



Gambar 13.2 Diagram Ringkas

13.2 Jenis Diagram HIPO

Paket HIPO terdiri dari 3 jenis diagram yaitu diagram daftar isi visual (Visual Table of content). Diagram Ringkas (Overview diagram) dan diagram rinci (detail diagram)

Daftar Isi Visual (DIV)

Diagram ini memuat semua modul yang ada dalam sistem berikut nama dan nomornya, yang nantinya akan diperinci dalam diagram ringkas dan diagram rinci. Dalam DIV juga bisa

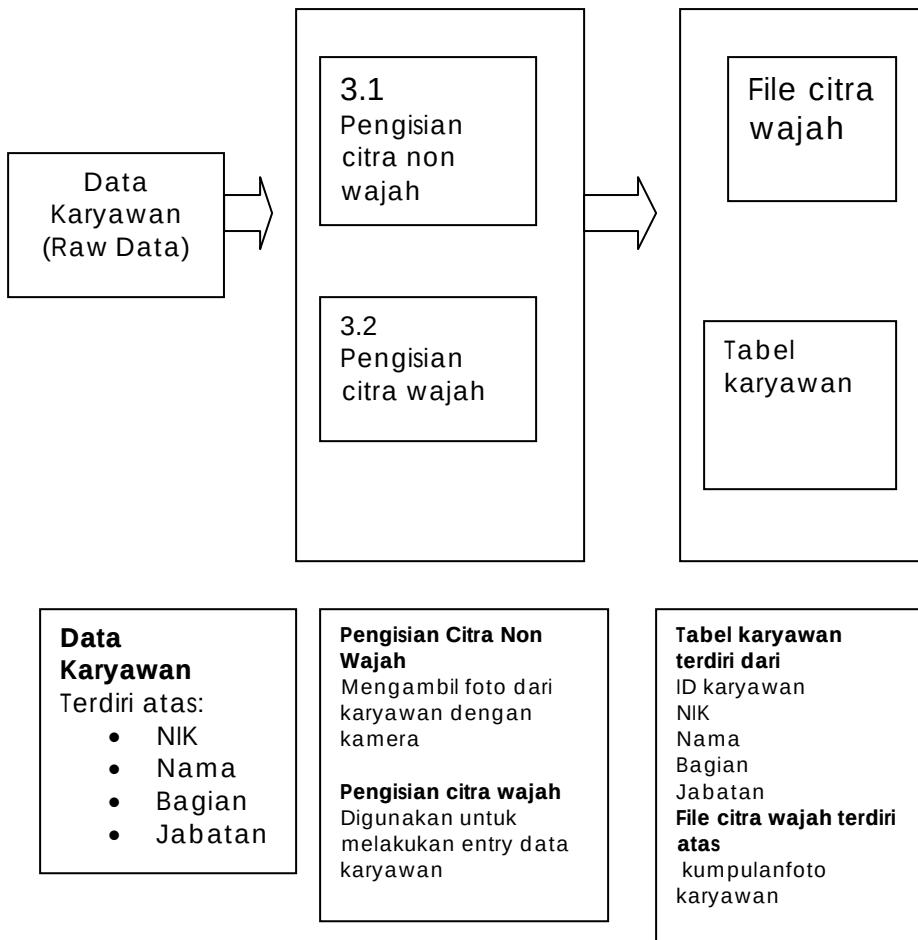
dilihat fungsi-fungsi utama yang menyusun sebuah sistem dan hubungan antar fungsi tersebut.

Diagram Ringkas

Diagram Ringkas menerangkan input, proses dan output dari sistem. Diagram ringkas menggambarkan input dan output dari fungsi-fungsi yang telah didefinisikan dalam Daftar Isi Visual. Perhatikan gambar 13.2 untuk contoh sebuah diagram ringkas.

Diagram Rinci

Diagram Rinci HIPO digunakan untuk memperinci input, proses dan output yang telah digambarkan dalam diagram ringkas. Input data dijelaskan field-field datanya secara detail. Untuk Fungsi, juga dideskripsikan proses apa yang dilakukan oleh fungsi-fungsi tersebut. Rincian field-field data output juga dijelaskan dengan lebih detail.



BAB 15

PEDOMAN DESAIN INTERFACE

User interface desain

- Beberapa aplikasi akan memiliki user interface yang sederhana, yang lain akan memiliki user interface yang kompleks.
- Akan ada 1 user interface untuk setiap pengguna
- Tujuan dari UI adalah untuk memungkinkan user menjalankan setiap tugas dalam user requirement
- Jadi dalam membangun sebuah user interface harus berdasar pada User Requirement

Prinsip-prinsip user interface yang baik:

- UI yang baik tidak mengharuskan pengguna untuk mengingat tampilan UI
- UI menampilkan apa yang dimengerti oleh user atau visualisasi dari keadaan dari sistem sekarang.
- hal yang harus dihindari :
- menampilkan terlalu banyak informasi dan terlalu banyak pilihan

- Menampilkan terlalu sedikit informasi, terlalu sedikit pilihan dan tanpa konteks
- Eksploitasi struktur menu standar yang sudah familiar dengan software yang sering digunakan user.

BAB 16

IMPLEMENTASI SISTEM

16.1 Implementasi sistem

Tahapan implementasi sistem dapat terdiri dari langkah-langkah berikut ini:

- Menerapkan rencana implementasi
- Melakukan kegiatan implementasi
- Tindak lanjut implementasi

16.1.1 Menerapkan rencana implementasi

Supaya kegiatan implementasi nantinya dapat beroperasi sesuai dengan yang diharapkan, maka suatu rencana implementasi . Sehingga rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan untuk mengatur biaya dan waktu yang dibutuhkan. Dalam tahapan ini biaya yang dibutuhkan dikalkulasi dalam bentuk anggaran biaya. Anggaran biaya inilah yang digunakan untuk kendali terhadap biaya-biaya yang dikeluarkan. Waktu yang diperlukan untuk melaksanakan tahapan-tahapan pekerjaan juga perlu ditentukan disini. Lebih jauh tentang penjadwalan akan di bahas pada modul tersendiri.

16.1.2 Kegiatan Implementasi

Kegiatan implementasi dilakukan dengan dasar kegiatan yang sudah direncanakan dalam rencana implementasi. Kegiatan yang dilakukan dalam tahapan implementasi adalah :

- Pemilihan dan pelatihan personil

- Pemilihan tempat dan instalasi perangkat lunak dan perangkat keras.
- Pemrograman dan pengetesan program
- Pengetesan sistem
- Konversi sistem.

16.1.3 Pemilihan dan pelatihan personil

Personil yang dipilih dapat berasal dari 2 sumber yaitu karyawan yang telah ada dari perusahaan atau calon karyawan yang berasal dari luar.

Pelatihan karyawan

Ada beberapa pendekatan pelatihan yaitu:

- Ceramah/seminar
- Pelatihan prosedural
- Pelatihan tutorial
- Simulasi
- Latihan langsung di pekerjaan (on-the-job-training)

16.1.4 Pemilihan tempat dan instalasi perangkat lunak dan perangkat keras.

Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk hardware baru harus juga dipersiapkan. Sistem yang besar membutuhkan tempat dan lingkungan yang harus diperhitungkan, misalnya dengan pemasangan AC, penerangan, pendeteksi kebakaran, UPS, telepon dan lain sebagainya. Langkah selanjutnya adalah menginstall hardware dan softwarenya.

16.2 Pengetesan Sistem

16.2.1 Filosofi testing

Sangat berbahaya hanya mengetes modul-modul awal tanpa perencanaan testing secara keseluruhan.

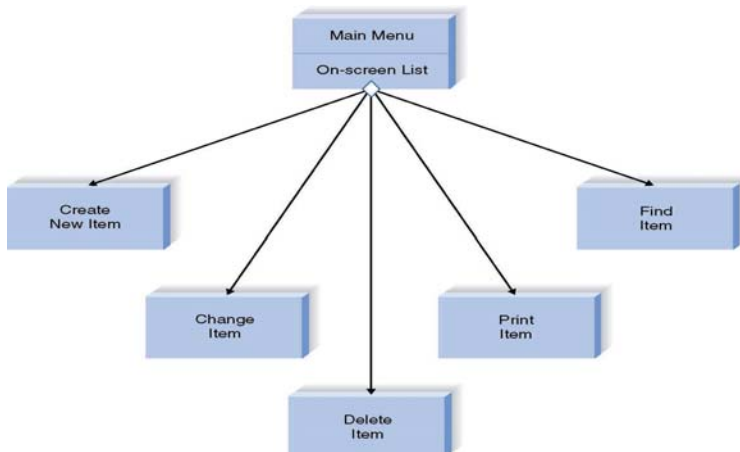
Akan cukup sulit untuk untuk menghasilkan serangkaian urutan kejadian yang menyebabkan error

Testing harus dilakukan secara sistematis dan hasilnya didokumentasikan dengan cermat.

16.2.2 Kategori testing

a. Stub testing

Tes struktur kendali sebelum semua modul dituliskan

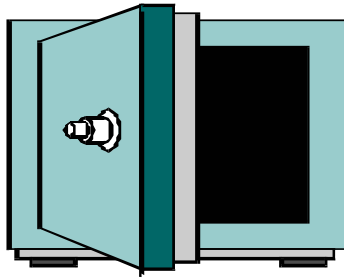


b. Unit testing

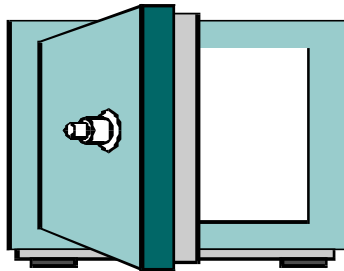
Tes setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing yaitu :

Black Box Testing

Terfokus pada apakah unit program memenuhi requirement yang disebutkan dalam spesifikasi.

**White-Box Testing**

Melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisa apakah ada kesalahan atau tidak.



c.Integration testing

Tes interaksi dari modul-modul untuk menjamin bahwa mereka bekerja dengan baik. Integration test terdiri dari serangkaian tes sebagai berikut:

Ujicoba Antarmuka

Ujicoba setiap fungsi dari interface

Ujicoba skenario pengguna

Pastikan setiap skenario berjalan dengan baik

Ujicoba aliran data

Uji setiap proses dalam langkah per-langkah

Ujicoba interface sistem

Pastikan data mengalir antar proses

d.System testing

Tes untuk menjamin software bekerja dengan baik sebagai bagian dari keseluruhan sistem.

e.Requirements Testing

Pastikan bahwa integrasi sistem tidak menyebabkan kesalahan. Tes ini terdiri dari serangkaian testing yaitu:

Usability Testing

Uji bagaimana sistem sudah bebas dari kesalahan dan mudah digunakan

Security Testing

Pastikan fungsi-fungsi keamanan berjalan dengan baik

Performance Testing

Pastikan bahwa sistem bekerja dengan baik pada aktivitas pekerjaan dengan volume besar

Documentation Testing

Uji apakah dokumen yang dihasilkan sudah benar dan sesuai dengan yang dibutuhkan

e.Acceptance testing

Tes untuk menjamin bahwa sistem telah melayani kebutuhan organisasi

•Alpha Testing

Tes ini dilakukan oleh user untuk menjamin bahwa mereka menerima sistem, tes dilakukan dengan menggunakan data tes .

•Beta Testing

Pengujian dengan menggunakan data riil, bukan data tes. User sesungguhnya memonitor kesalahan yang terjadi atau perbaikan yang dibutuhkan.