



SILABUS MATA KULIAH

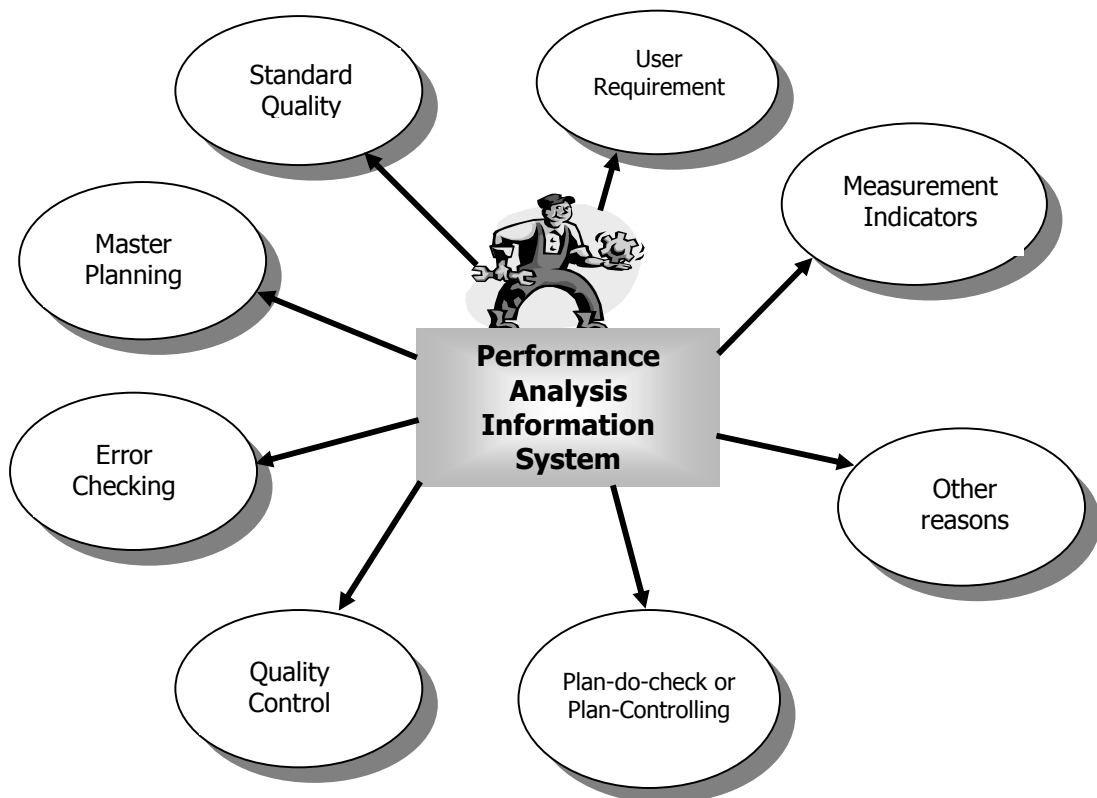
ANALISIS KINERJA SISTEM INFORMASI

Kode Kuliah	KK328
Mata Kuliah	Analisis Kinerja Sistem Informasi
Beban Kredit	3 SKS
Tujuan	Menjelaskan bagaimana suatu sistem informasi yang sudah dibangun dinilai kinerjanya dilihat dari sudut pandang proses dan produknya.
Prasyarat	Analisis dan Perancangan Sistem Informasi
Materi	<ol style="list-style-type: none">1. Teknik dan pendekatan analisis kinerja sistem informasi2. Mengukur kinerja sistem informasi dari proses pembuatan, produk yang dihasilkan dan dokumentasinya.3. Quality management
Kepustakaan	<ol style="list-style-type: none">1. Ron Weber, "<i>Information Systems Control and Audit</i>", Prentice-Hall, USA., 1999.2. Martin, Merle P, "<i>Analysis and Design of Business Information Systems-2 nd ed</i>", Prentice Hall, New Jersey, USA., 1995.

BAB I

Gambaran Pemeriksaan SI *(Overview of Information System Auditing)*

⊕ Performance Information System Reasons



Gbr. 1.1. Performance Information System Reason



⊕ Information Systems Auditing Defined

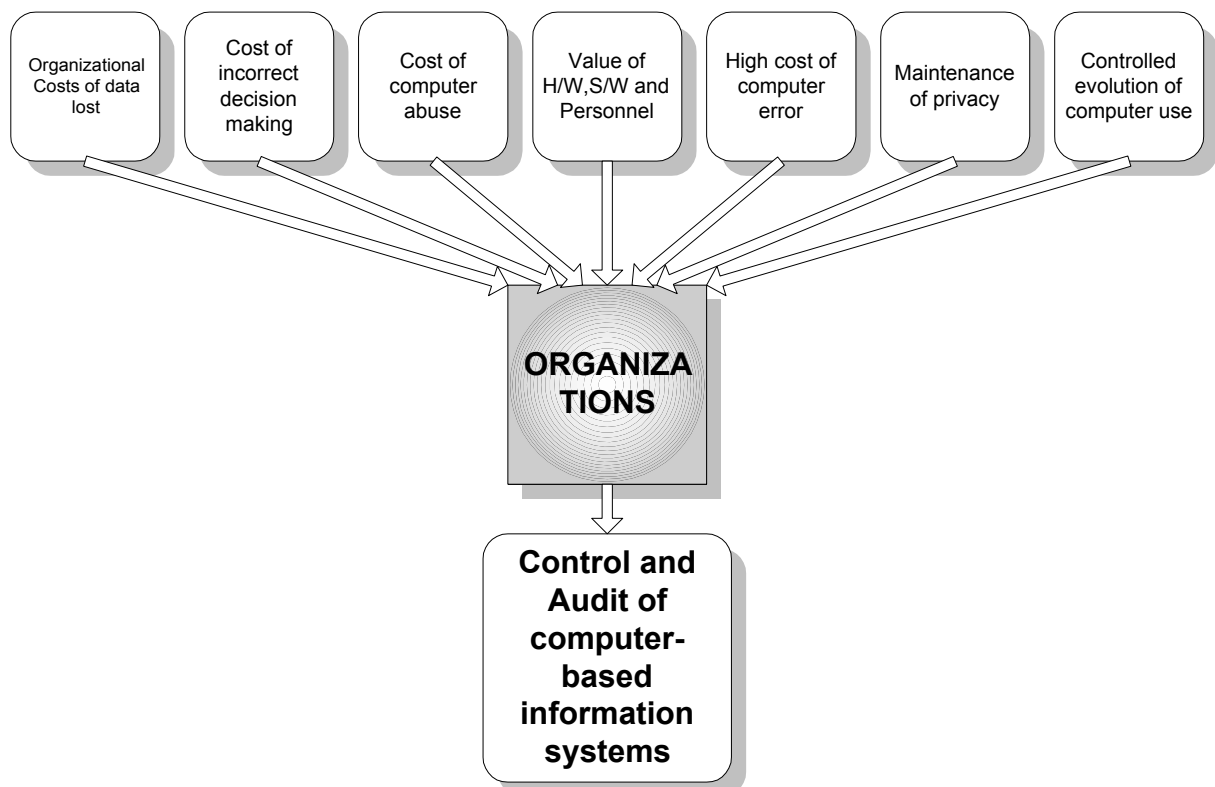
(ref.1 pp.5-21)

Information systems auditing is the process of collecting and evaluating evidence to determine whether a computer system safeguards assets, maintains data integrity, allows organizational goals to be achieved effectively, and uses resources efficiently.

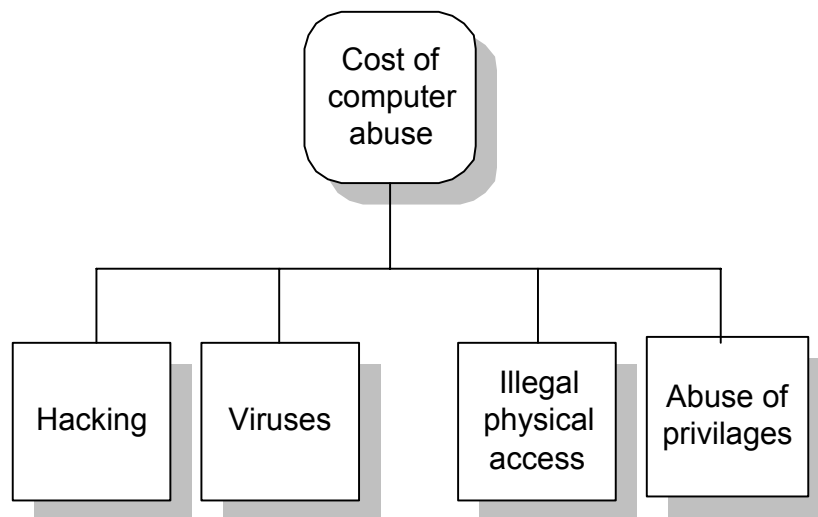
(Audit SI adalah proses mengumpulkan dan mengevaluasi fakta untuk memutuskan apakah sistem komputer yang merupakan aset bagi perusahaan terlindungi, integritas data terpelihara, sesuai dengan tujuan organisasi untuk mencapai efektifitas, dan efisiensi dalam penggunaan sumber daya).

⊕ Need For Control and Audit of Computers

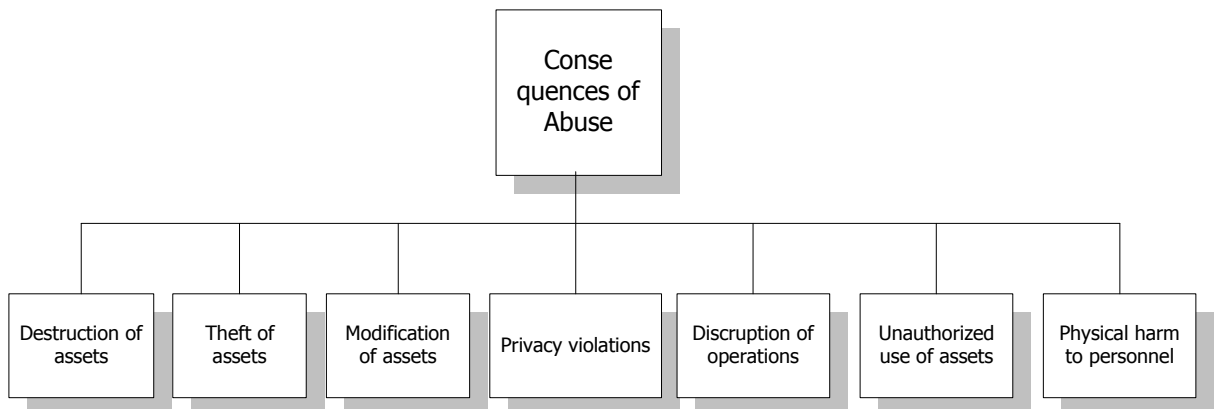
(ref.1 pp.5-21)



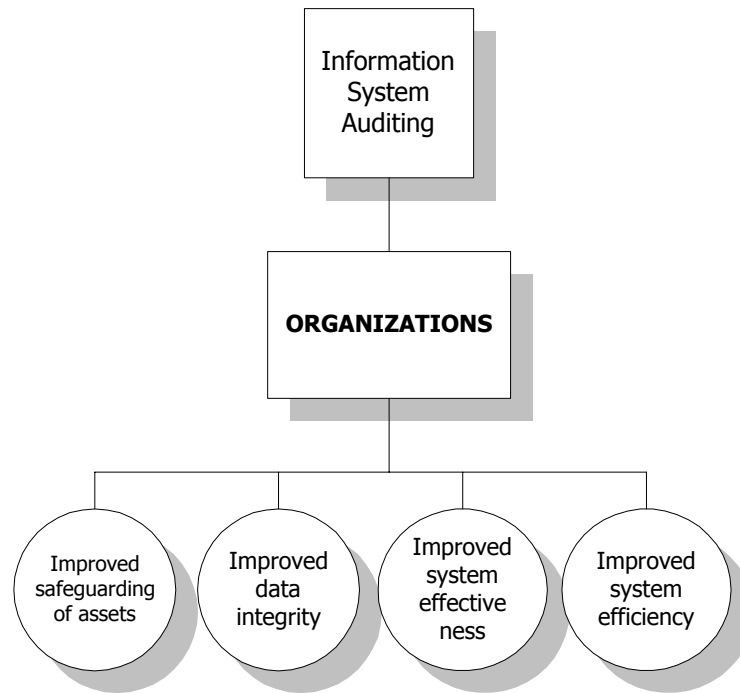
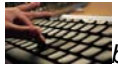
Gbr. 1.2 Factors influencing an organization toward control and audit of computers



Gbr. 1.3 Cost of computer abuse



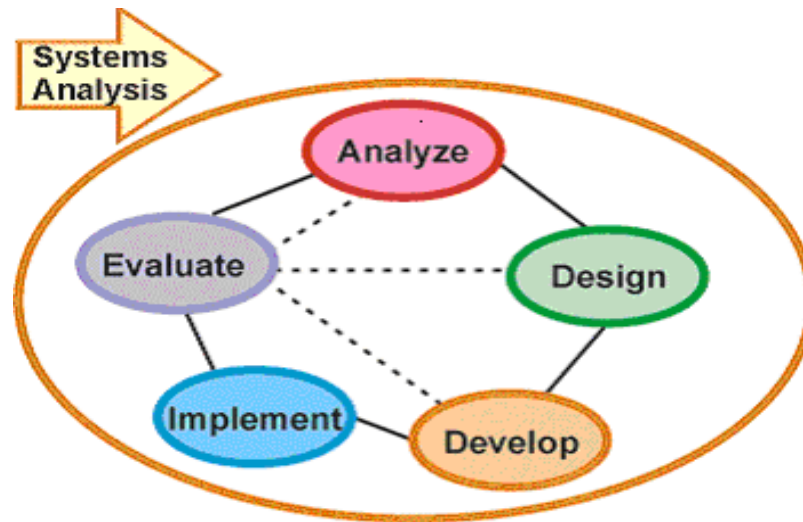
Gbr. 1.4 Consequences of Abuse



Gbr. 1.5 Impact of the IS audit function on organization



<http://www.concordassoc.com/main.aspx>



Gbr. 1.6 Evaluate of the System Analysis



⊕ Organizational Costs of Data Loss

- ▶ Data dapat menyebabkan kebutuhan sumber daya menjadi kritis untuk keberlangsungan operasional organisasi (baik untuk memberikan gambaran masa lalu, masa kini dan masa yang akan datang).
- ▶ Jika data akurat, maka organisasi akan mempunyai kemampuan untuk beradaptasi dan bertahan dalam lingkungan yang berubah. Jika tidak (data hilang), maka organisasi akan mengalami kehilangan data yang cukup penting.
- ▶ Contoh jika data master barang di suatu toko swalayan rusak, maka kasir tidak dapat melakukan transaksi pembelian yang dilakukan oleh konsumen.

⊕ Cost of Incorrect Decision Making

- ▶ Untuk membuat keputusan yang berkualitas dan dapat dipercaya, maka perlu di dukung oleh data yang akurat melalui sistem informasi berbasis komputer.
- ▶ Termasuk : deteksi, investigasi, dan koreksi proses yang diluar kontrol (*connection of out-of-control process*)
- ▶ Akibat data yang salah akan mempunyai dampak terhadap minat investor terhadap perusahaan. Contoh : jika penyediaan laporan keuangan salah (*inaccurate financial information*), maka investor akan membatalkan atas keputusan investasinya.
- ▶ Penting juga diperhatikan tentang `aturan-aturan keputusan yang akurat (*accurate decision rules*). Contoh jika aturan pengambilan keputusan (*decision rule*) dalam sistem pakar untuk mendukung diagnosis, salah, mengakibatkan dokter akan salah dalam memberikan keputusan / pemberian resep kepada pasiennya, ini akan berakibat fatal.

⊕ Cost of Computer Abuse

- ▶ Sebagian besar sebab yang mendorong pengembangan fungsi audit SI di perusahaan adalah akibat seringnya terjadi penyalahgunaan komputer.
- ▶ Penyalahgunaan komputer : "segala kejadian yang berhubungan dengan teknologi komputer yang mengakibatkan kerugian pada korban atau mengakibatkan kehilangan yang diakibatkan oleh pelaku kejahatan untuk mencari keuntungan"
- ▶ Sebagian besar tipe penyalahgunaan komputer adalah :
 - 1) **Hacking** : seseorang yang tidak mempunyai akses otoritas terhadap sistem komputer untuk membaca, memodifikasi atau menghapus program atau data untuk mengacaukan proses.
 - 2) **Virus** : adalah program yang menyerang file executable, area sistem atau disk, atau file data yang berisi macro yang mengakibatkan kekacauan operasi komputer atau kerusakan data / program.
 - 3) **Illegal Physical Access** : seseorang yang mengambil keuntungan melalui akses fisik secara ilegal terhadap fasilitas komputer. Contoh memasuki ruang komputer atau ruang terminal secara ilegal, merusak H/W, atau copy program dan data yang bukan merupakan wewenangnya.
 - 4) **Abuse of Privileges** : seseorang yang menggunakan hak-hak istimewanya untuk maksud dan tujuan yang bukan merupakan otoritasnya. Contoh : membuat copy data yang rahasia (sensitif) akan tetapi tidak meminta ijin atau persetujuan kepada yang berwenangnya.



Menurut survey Benbow (1990) : 80% penyalahgunaan komputer diakibatkan oleh 'pegawai intern'.

⊕ **Konsekuensi Penyalahgunaan Komputer**

1. **Destruction of asset** (perusakan aset) : Hardware, software, data, fasilitas, dokumentasi atau persediaan barang dapat dirusak.
2. **Theft of asset** (pencurian asset): Hardware, software, data, dokumentasi, atau persediaan barang dapat dipindahkan secara ilegal.
3. **Modification of asset**: Hardware, software, data atau dokumentasi dimodifikasi dengan cara yang tidak sah
4. **Privacy violaction** (pelanggaran privasi) : privasi mengenai data seseorang atau organisasi di gunakan untuk kepentingan yang tidak sah.
5. **Disruption of Operations** (pengacauan operasi) : operasi fungsi sehari-hari ('day-to-day') SI dapat terhenti sementara yang diakibatkan oleh operasi yang dikacaukan.
6. **Unauthorized use of asset** (penyalahgunaan otorisasi aset) : Hardware, software, data, fasilitas, dokumentasi atau persediaan barang digunakan untuk maksud yang tidak sah. contoh penggunaan komputer dinas di kantor untuk maksud private atau konsultasi.
7. **Physical harm to personnel** (kejahatan fisik terhadap personal) : personal / pegawai dapat menderita akibat kejahatan fisik.

⊕ **Value of computer H/W,S/W, Personnel**

- Data, H/W, S/W dan personal adalah merupakan sumber daya kritis organisasi
- Beberapa organisasi telah menginvestasikan ratusan miliar dollar untuk itu.

⊕ **High Cost of Computer Error**

- Komputer saat ini mempunyai peranan / fungsi penting dengan lingkungan sosial. Contoh monitor digunakan untuk memantau pasien, memonitor missile, pengendali reaktor nuklir, dll. Akibatnya jika komputer 'error', maka akan mengakibatkan kerugian yang sangat besar (mahal).
- Contoh : 257 orang meninggal di pegunungan antartika, akibat error pada sistem yang diakibatkan oleh pekerjaan 'iseng' seseorang yang mengganti isi / data sistem komputer yang terkait dengan penerbangan.

⊕ **Maintenance of Privacy**

- Sebagian besar data dikumpulkan, merupakan data individu seperti: data pembayar pajak, credit, medical, educational, employment, dan yang lainnya. Data ini dikumpulkan sebelum proses komputerisasi, dan data privasi ini harus dilindungi. Agar hak-hak privasinya terjaga.

⊕ **Controlled of Evolution of Computer Use**

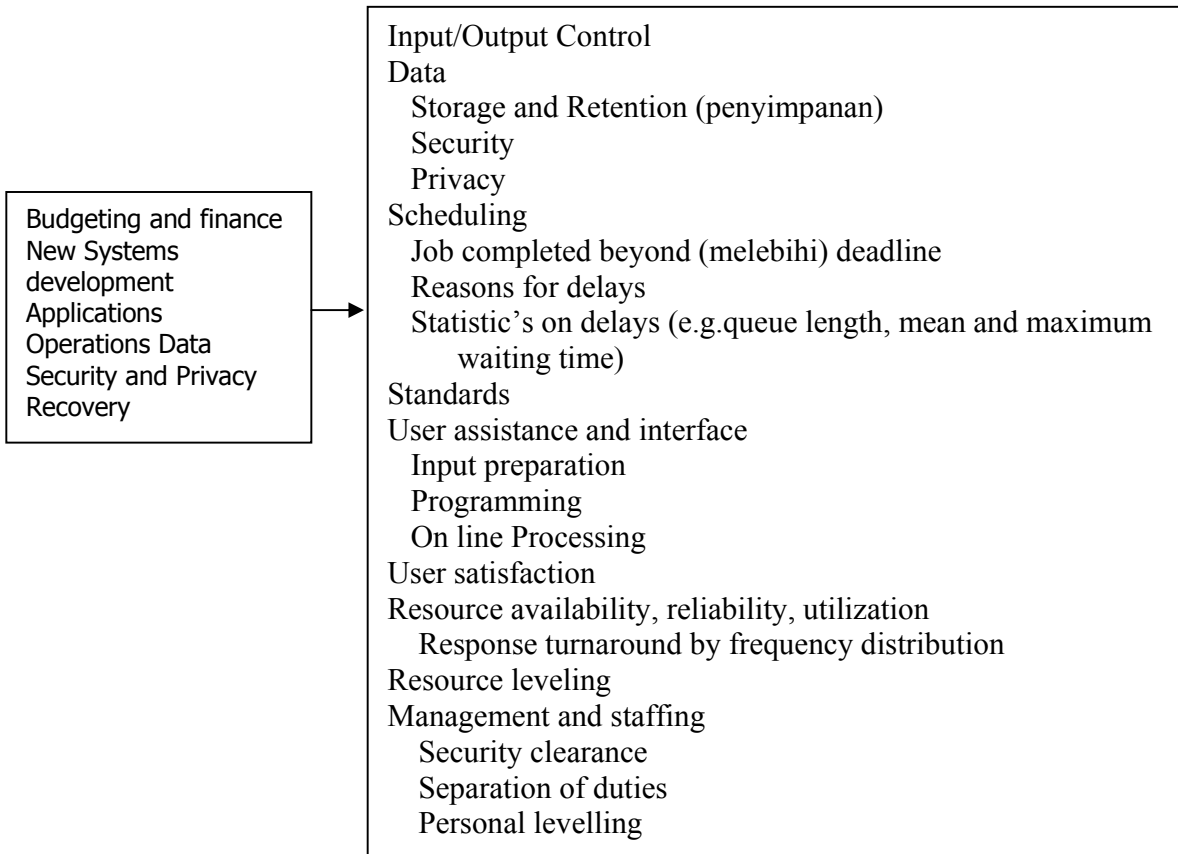
- Konflik, satu sisi komputer digunakan untuk hal-hal yang berguna, tapi di sisi lain komputer digunakan untuk pengendalian nuklir yang mungkin saja digunakan untuk hal-hal yang tidak berguna.



Tugas-2 :

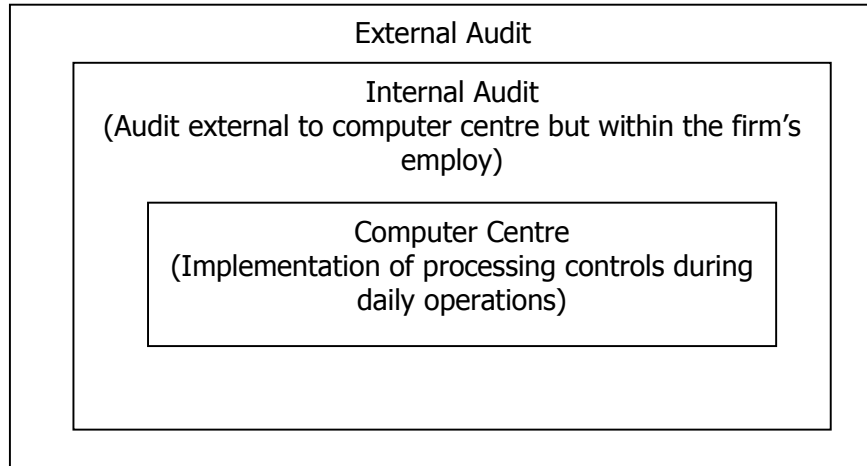
1. Cari artikel di majalah / internet tentang kejahatan / kasus2 penyalahgunaan komputer
2. Analisis kinerja S/W yg dibuat oleh salah seorang alumni

Auditing Concerns : < dari copy-an mk MIS, pak marie h)





Layer of control :



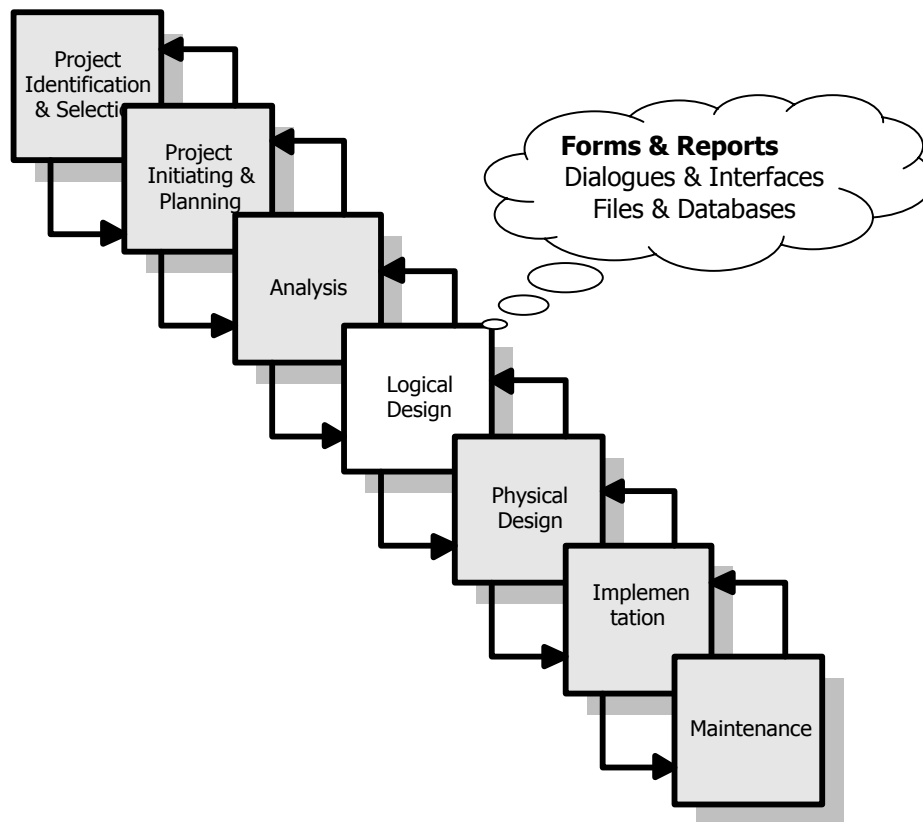


Design Forms & Reports

(rb.1,pp.513-727)

✚ Designing Forms & Reports

Pada pembahasan ini kita akan memfokuskan pada desain logika (*logical design*) dalam tahapan SDLC seperti pada gbr.2.1 di bawah ini.



Gbr.2.1 Tahap SDLC, Desain Logika



Pada bab ini kita akan membahas materi yang berkaitan dengan sistem input dan output - **forms & reports**.

Pada bab berikutnya kita juga akan membahas logical design dari **dialog dan interface**, yaitu bagaimana interaksi antara user dan sistem.

Pada bab selanjutnya juga kita akan membahas mengenai **logical databases design**.

Selama tahapan analisis, kita tidak dapat kosen dengan desain yang tepat tentang form dan report. Untuk itu kita dapat membuat prototipe (contoh) form dan report lalu dikonfirmasi sesuai dengan kebutuhan user.

Sebagai contoh setiap form input akan berhubungan dengan sebuah data flow masukkan pada sebuah DFD, dan setiap output form atau report akan menghasilkan sebuah data flow oleh sebuah proses dalam sebuah DFD. Ini artinya setiap form atau report berhubungan dengan isi elemen data yang berhubungan dengan data flow.

Selanjutnya data pada semua form dan report harus berisi data elemen di dalam data store dan pada ER data model untuk sebuah aplikasi atau harus berisi hasil perhitungan dari elemen data.

Formating Forms and Reports

(br 1, pp 525)

➤ Pedoman Umum dalam pembuatan Form dan Report (*Generating Formatting Guidelines*)

General guidelines for Design Forms and Reports

1. Judul harus mewakili obyeknya dan penuh arti (*Meaningful Titles*) :

- Berisi penjelasan judul yang spesifik dan jelas, baik untuk form maupun report (*Clear and spesific titles describing content and use of form or report*)
- Cantumkan tanggal revisi dan kode untuk membedakan form atau report dari versi sebelumnya (*Revision date or code to distinguish a form or report from prior versions*)
- Cantumkan tanggal, pada saat form dan report dibuat (*Current date which identifies when the form or report was generated*)
- Validasi data sebagai identifikasi apakah tanggal / waktu data di dalam form atau report akurat (*Valid date which identifies on what date (or time) the data in the form or report were accurate*)



-
2. Informasi harus penuh arti (*Meaningful Information*):
 - Hanya informasi yang dibutuhkan yang ditampilkan (*Only needed information should be displayed*)
 - Informasi harus mengandung arti, dapat digunakan dengan modifikasi (*Information should be provided in a manner that is usable without modification*).
 3. Keseimbangan Tata Letak (*Balanced Layout*) :
 - Informasi harus seimbang di tampilkan di layar maupun di halaman (*Information should be balanced on the screen or page*)
 - Gunakan spasi dan margin yang baik (*Adequate spacing and margins should be use*)
 - Semua data dan entri field harus dengan label yang jelas (*All data and entry fields should be clearly labeled*).
 4. Kemudahan arah penunjuk (*easy navigation*) :
 - Mudah memindahkan arah penunjuk ke depan atau kebelakang (*clearly show how to move forward and backward*).
 - Mudah melihat halaman (*clearly show how where you are*) (e.g. page 1 of 3).
 - Mudah melihat halaman berikutnya pada urutan halaman yang banyak (*notify user when on the last page of a multipaged sequence*)

Highlighting Information

Metode pen-sorotan (highlighting)

1. Teknik blink atau suara (dengan suara jelas) (*blinking and audible tones*)
2. Pembedaan warna (*color difference*)
3. Pembedaan intensitas (*intensity difference*)
4. Pembedaan ukuran (*size difference*)
5. Pembedaan huruf (*font difference*)
6. Kebalikan warna layar (reverse video)
7. Gunakan kotak (*boxing*)
8. Gunakan garis bawah (*underlining*)
9. Gunakan huruf besar semua (*all capital letters*)
10. Gunakan posisi atau informasi yang tidak standar (*offsetting the position of nonstandard information*).

Gunakan warna atau tidak

Keuntungan dan kerugian penggunaan warna

● Keuntungan :

1. Enak dipandang mata (menyejukan) (*soothes or strikes the eye*)
2. Penekanan pada tampilan yang tidak diminati (*accents an uninteresting display*)



-
3. Sebagai penajaman pembeda tampilan yang komplek (*facilitates subtle discriminations in complex displays*)
 4. Memperluas organisasi logika informasi (*emphasizes the logical organization of information*)
 5. Pemberian warna untuk focus penekanan (peringatan) (*draws attention to warnings*)
 6. Menggambarkan reaksi emosi (*evokes more emotional reactions*)
- Kerugian :
 1. Penggunaan warna yang tidak tepat dapat menimbulkan masalah, contoh warna yang tidak terlihat dengan jelas (*color pairings may wash out or cause problems for some users*)
 2. Perbedaan resolusi dapat menyebabkan degradasi perbedaan tampilan warna (*resolution may degrade with different displays*)
 3. Ketepatan warna dapat menimbulkan degradasi tampilan (*color fidelity may degrade on different displays*)
 4. Pencetakan atau konversi untuk media yang lain tidak mudah untuk dilakukan (*printing or conversion to other media may not easily translate*)

Tampilan Teks

Pedoman untuk tampilan teks

1. Huruf besar (case), tampilan teks dalam huruf besar atau campuran huruf besar dan kecil.
2. Spasi (spacing), gunakan spasi double atau single, gunakan jarak spasi antar paragraph.
3. Justifikasi (justification), gunakan rata kiri-kanan atau rata kiri dan kanan.
4. Hyphen (hyphenation), tidak menggunakan hyphen untuk kata antar baris.
5. Singkatan (abbreviations), gunakan singkatan dan akronim yang dapat dipahami, jika teks cukup banyak (panjang).

Desain Tabel dan Daftar (List)

Pedoman umum untuk desain tabel dan daftar (list)

1. Gunakan label yang penuh arti :
 - a. Seluruh kolom dan baris harus mempunyai arti
 - b. Label harus dipisahkan dari informasi lain dengan menggunakan pensorotan (highlighting).



- c. Tampilkan kembali label sewaktu data ditampilkan dalam satu layar atau halaman.
2. Format kolom, baris dan teks:
 - a. Urutkan dalam urutan tertentu (ascending, descending atau alphabetic)
 - b. Tempatkan baris kosong setiap lima baris dalam kolom yang panjang.
 - c. Tampilan informasi dalam banyak kolom diurutkan secara vertical (di baca dari atas ke bawah, tidak dari kiri ke kanan)
 - d. Antar kolom paling sedikit terdapat jarak dua spasi.
 - e. Boleh menggunakan spasi 'putih' pada pencetakan laporan untuk user untuk catatan tulisan.
 - f. Gunakan bentuk tunggal, selain untuk 'dibesarkan'.
 - g. Gunakan bentuk yang sama untuk kros tampilan dan report
 - h. Hindari penggunaan huruf yang berlebihan
 3. Format data dalam Numerik, Textual, dan alphanumeric
 - a. Data numerik : Rata kanan (Right-justify numeric data)
 - b. Data teks : rata kiri
 - c. Alphanumeric data : alphanumeric yang panjang dipecah dalam grup-grup kecil antara 3-4 karakter.



Pedoman Desain Interface

(rb:rpl-roger s pressman, pp.471-474)

Pedoman Desain Interface

Banyak sumber didalam literature menyajikan serangkaian pedoman desain HCI (Human Computer Interaction) yang akan menghasilkan interface yang `ramah` dan efisien.

Terdapat 3 katagori pedoman desain HCI, yaitu :

1. Interaksi Umum
2. Tampilan dan
3. Entri Data

● Interaksi Umum

1. Konsisten, gunakan format yang konsisten untuk pemilihan menu, input, perintah, tampilan data, dll
2. Berikan umpan balik yang sangat berarti, berikan umpan balik yang berkaitan dengan interaksi antara user dan sistem (komputer), misal melalui suara, teks, dll.
3. Mintalah verifikasi terhadap sembarang aksi destruktif yang signifikan, jika user meminta penghapusan file, indikasikan dengan suatu interaksi, misal `apakah anda yakin?`
4. Ijinkan kemudahan pembatalan sebagian besar aksi, fasilitas pembatalan harus disediakan dalam sistem. Misal fasilitas sejenis UNDO atau REDO.
5. Kurangi jumlah informasi yang harus di ingat diantara aksi-aksi, user jangan di bebani dengan ingatan-ingatan yang banyak, misal mengingat perintah interaksi, dll.
6. Usahakan adanya efisiensi dalam dialog, gerakan, dan pemikiran, penekanan tombol harus diminimalkan, user harus tahu persis sedang berada pada aksi apa saat ini.
7. Memaafkan kesalahan, sistem harus melindungi dirinya sendiri dari kesalahan yang dapat menyebabkan kegagalan pada sistem.
8. Katagorikan aktivitas menurut fungsi dan atur letak layar yang sesuai, misal melalui pengaturan `menu pull-down`.
9. Sediakan fasilitas help
10. Gunakan instruksi yang sederhana atau pendek untuk memberi nama perintah.



- Tampilan

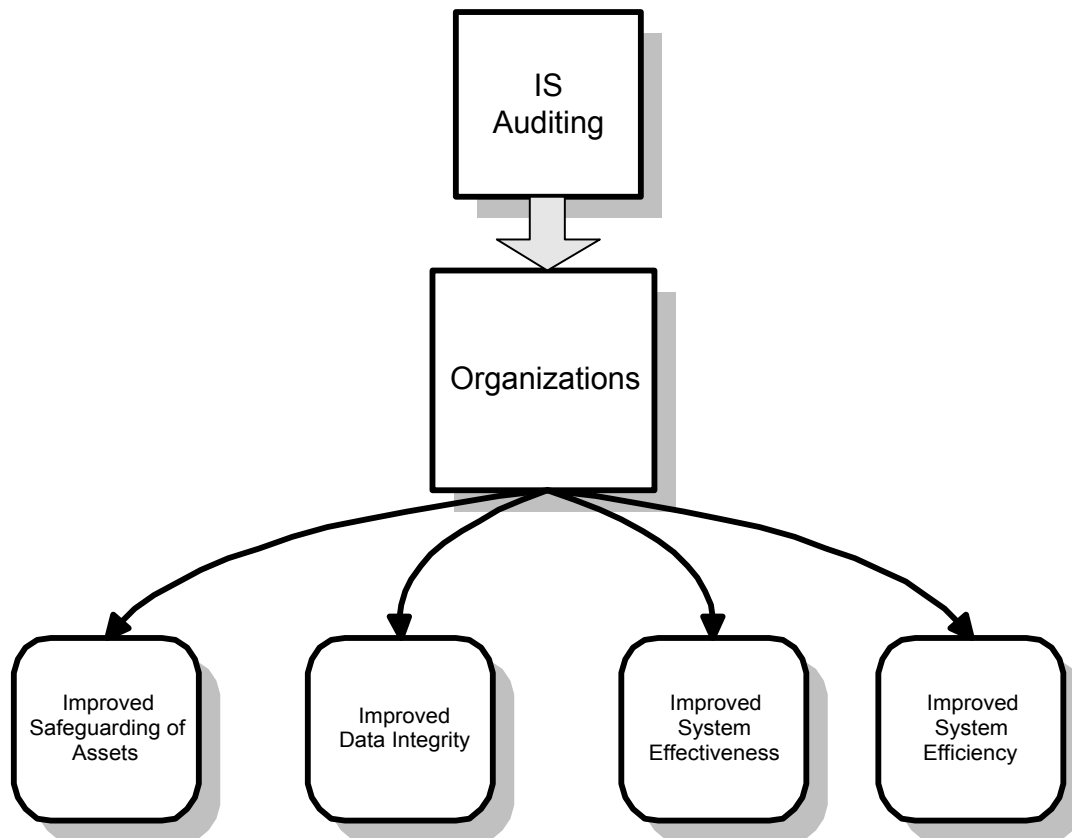
1. Menampilkan hanya informasi yang relevan dengan konteks yang ada
2. Jangan membanjiri pemakai dengan data, gunakan format representasi yang memungkinkan asimilasi informasi yang tepat
3. Gunakan label-label yang konsisten, penyingkatan standar, dan warna yang dapat diprediksi.
4. Ijinkan user untuk memelihara konteks visual, user memahami lokasi relatif dari pembagian citra yang sedang dipandang.
5. Hasilkan pesan kesalahan yang berarti
6. Gunakan huruf besar dan kecil, indentasi, dan pengelompokan teks untuk membantu pemahaman.
7. Gunakan jendela untuk menggolongkan tipe-tipe informasi yang berbeda
8. Gunakan tampilan 'analog' untuk merepresentasikan informasi yang lebih mudah diasimilasikan dengan bentuk representasi yang lain.
9. Pertimbangkan ketersediaan letak layar tampilan dan gunakan secara efisien.

- Entri Data

1. Minimalkan jumlah aksi input yang dibutuhkan dari pemakai
2. Jagalah konsistensi diantara tampilan informasi dan input data
3. Ijinkan user mengkustomasi input, user yang sudah mahir mungkin hanya memerlukan perintah tertentu saja, HCI harus memungkinkan hal tersebut.
4. Interaksi harus fleksibel tetapi juga diatur ke mode input yang disukai pemakai.
5. Non-aktifkan perintah yang tidak sesuai di dalam konteks aksi yang sedang berlangsung
6. Biarkan user mengendalikan aliran interaktif, misal user dapat melompati perintah yang tidak diperlukan.
7. Sediakan help untuk membantu semua aksi input
8. Hilangkan input 'mickey-mouse', misal jangan meminta user untuk mengetik '.00' untuk seluruh bilangan rupiah, berikan bilangan default jika mungkin. Dan jangan pernah meminta user untuk memasukkan data yang dapat diperoleh secara otomatis atau hasil yang sebenarnya sudah di hitung oleh program (sistem).



✚ Pengaruh Fungsi Audit SI untuk Organisasi



Gbr. 2.2 Impact of the IS audit function on organization

✚ **Obyek Perlindungan Aset** (*Asset Safeguarding Objectives*)

- Aset SI di dalam organisasi adalah H/W, S/W, Fasilitas, user (Knowledge), file data, dokumentasi sistem, dan ??????persediaan barang.
- Sebaiknya semua aset harus dilindungi oleh sistem pengendalian internal.

✚ **Obyek Integritas Data** (*Data Integrity Objectives*)

- Integritas data adalah merupakan konsep dasar di dalam audit SI. Data terdiri dari atribut-atribut yang harus berisi : lengkap (completeness), dapat dipercaya (soundness), bersih (purity), and benar (veracity).
- Jika integritas data tidak dipelihara, maka organisasi tidak akan mendapatkan representasi data yang benar untuk suatu aktifitas, akibatnya organisasi tidak dapat berkompetisi.



⊕ **Obyek Efektivitas Sistem** (*System Effectiveness Objectives*)

- Audit efektivitas sering dilakukan setelah sistem berjalan untuk beberapa waktu. Manajemen membutuhkan hasil audit efektivitas untuk mengambil keputusan apakah sistem terus dijalankan atau dihentikan sementara untuk proses modifikasi.

⊕ **Obyek Efisiensi Sistem** (*System Efficiency Objectives*)

- Efisiensi SI dilakukan dengan cara menggunakan sumber daya yang minimum untuk menyelesaikan suatu tujuan obyek (pekerjaan). Variasi sumber daya terdiri dari mesin, waktu, peripheral, S/W sistem, dan pekerja.

✚ **Pengaruh Komputer dalam Pengendalian Internal**

Tujuan dari perlindungan aset, integritas data, efektivitas sistem, dan efisiensi sistem dapat dicapai dengan baik jika manajemen organisasi meningkatkan sistem pengendalian internalnya, yaitu dengan cara :

1. Pemisahan Tanggung Jawab

(Separation of Duties)

2. Pendelegasian Wewenang dan Tanggung Jawab

(Delegation of Authority and Responsibility)

3. Personal yang Kompeten dan Dapat dipercaya

(Competent and Trustworthy Personnel)

4. Otorisasi Sistem

(System of Authorizations)

5. Kecukupan Catatan dan Dokumen

(Adequate Documents and Records)

6. Pengendalian Fisik atas banyaknya Rekord dan Aset

(Physical Control over Assets and Records)

7. Kecukupan Supervisi dari pihak Manajemen

(Adequate Management Supervision)

8. Bentuk Pengecekan yang Independen

(Independent Checks on Performance)

9. Perbandingan Akuntabilitas Rekord dengan Aset

(Comparing Recorded Accountability with Assets)

✚ **Pengaruh Komputer dalam Pengendalian**

⊕ **Perubahan dalam Pengumpulan fakta** (*Changes to Evidence Collection*)

⊕ **Perubahan dalam Evaluasi Fakta** (*Changes to Evidence Evaluation*)



BAB II

Memeriksa Sistem Informasi *(Conducting an Information System Audit)*

Introduction

Kaitan dengan Kompleksitas *(Dealing with Complexity)*

Resiko Pemeriksaan *(Audit Risks)*

Tipe Prosedur Pemeriksaan *(Types of Audit Procedures)*

Gambaran Langkah-langkah dalam Pemeriksaan *(Overview of Steps in an Audit)*

Pemeriksaan Lingkungan Komputer Menyeluruh *(Auditing Arround or Through the Computer)*

2.1. Introduction

- Hal yang menenangkan dalam tugas audit SI di organisasi yaitu karena terdapat banyak programmer, analis, banyak komputer dan ribuan file. Yang jelas, tidak semua organisasi mempunyai kapasitas yang sama. Untuk organisasi kecil, bagaimanapun, peran auditor tidak untuk menyelesaikan pengecekan secara detil seluruh persoalan pengolahan data dalam fungsi SI. Sebagai gantinya mereka mengandalkan contoh data untuk memutuskan apakah tujuan audit SI telah tercapai.
- Bagaimana, selanjutnya kita dapat menyelesaikan audit SI, juga untuk mendapatkan alasan jaminan dan perlindungan aset pengolahan data, memelihara integritas data, dan mencapai sistem yang efektif dan efisien? akan dibahas d bawah ini.

2.2. Pemeriksaan Tradisional *(The Nature of Controls)*

Audit SI harus konsen dengan pengendalian evaluasi kehandalan atau efektivitas operasi. Yang penting lagi kita juga harus memahami apa pengertian dari pengendalian (control).

"Pengendalian (control) adalah sebuah sistem yang digunakan untuk mencegah (prevents), mendeteksi (detects), atau mengkoreksi kebenaran (keabsahan) suatu peristiwa / kegiatan sesuai dengan aturan / hukum".

Tiga aspek kata kunci definisi kontrol, yaitu :

1. Pengendalian adalah sebuah sistem *(a control is a system)*

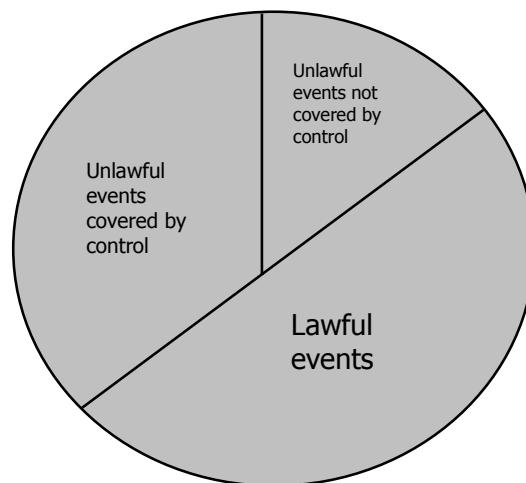
Dengan kata lain, terdiri dari sekumpulan komponen yang saling berelasi yang berfungsi secara bersama-sama untuk menyelesaikan suatu maksud atau tujuan.



2. Keabsahan / kebenaran dari suatu kegiatan (*unlawful events*)
Keabsahan kegiatan dapat muncul jika tidak ada otorisasi (*unauthorized*), tidak akurat (*inaccurate*), tidak lengkap (*incomplete*), redundansi (*redundant*), tidak efektif (*ineffective*) atau tidak efisien (*inefficient*) pemasukan data kedalam sistem.
3. Ketiga, pemeriksaan digunakan untuk mencegah (*prevent*), mendeteksi (*detect*), atau mengoreksi (*correct*) kejadian / peristiwa yang tidak sesuai dengan aturan / hukum (*unlawful events*).

Beberapa contoh dapat dipertimbangkan :

- a. Pemeriksaan Pencegahan (Preventive control)
Instruksi yang ditempatkan pada dokumen dasar (sumber) untuk mencegah kemungkinan petugas salah DALAM mengisi dokumen (out incorrectly).
- b. Pemeriksaan Detektif / pengintaian (Detective control)
Program dapat mengidentifikasi kesalahan pemasukan data ke dalam sistem melalui terminal (alat masukan).
- c. Pemeriksaan Koreksi (Corrective control)
Program menggunakan kode khusus yang memungkinkan sistem dapat mengoreksi kesalahan data akibat gangguan (noise) komunikasi.



Gbr. 2.1a Kejadian yang tidak dibenarkan atau tidak sesuai dengan aturan / hukum dalam SI.

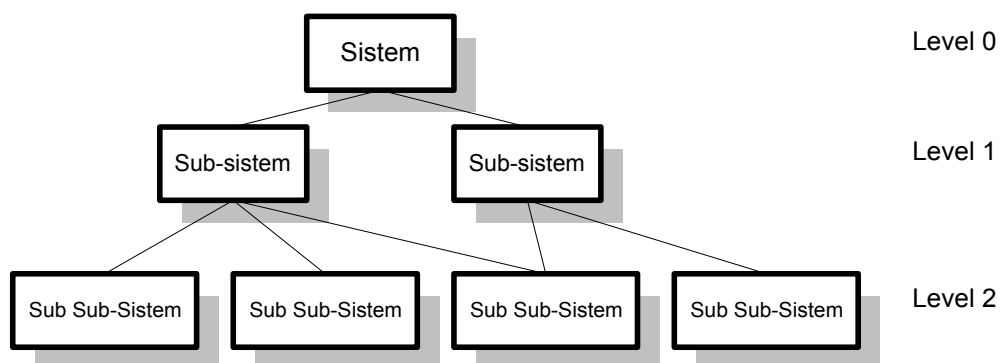


2.3. Kaitan dengan Kompleksitas (*Dealing with Complexity*)

Dua pedoman pendekatan sewaktu pemeriksaan (audit) SI :

1. Penentuan rencana audit SI, faktor sistem di evaluasi kedalam sub sistem.
2. Menentukan keandalan dari tiap sub-sistem dan implikasi kehandalan dari tiap level sub-sistem untuk kehandalan sistem secara menyeluruh.

Faktor sub-sistem



Gbr. 2.2a Struktur dari level sistem dan sub-sistem

Beberapa tipe sub-sistem manajemen dapat diidentifikasi berdasarkan hubungan antara hirarki organisasi dan sebagian besar tugas fungsi system informasinya :

1.



3.1.1.1. Pengendalian Manajemen Puncak

- (i) Evaluasi Fungsi Perencanaan
- (ii) Evaluasi Fungsi Organisasi
- (iii) Evaluasi Fungsi Kepemimpinan
- (iv) Evaluasi Fungsi Pengendalian

3.1.1.2. Pengendalian Manajemen Pengembangan Sistem

- (i) Evaluasi Sebagian Besar Tahap Proses Pengembangan Sistem

3.1.1.3. Pengendalian Manajemen Pemrograman

- (i) Siklus Hidup Pengembangan Program
- (ii) Pengorganisasian Tim Pemrograman
- (iii) Pengelolaan Kelompok Pemrograman

3.1.1.4. Pengendalian Manajemen Sumber Data

- (i) Fungsi DA dan DBA
- (ii) Beberapa Isu Pengorganisasian
- (iii) Sistem Penyimpanan Data
- (iv) Pengendalian atas DA dan DBA

3.1.1.5. Pengendalian Manajemen Keamanan

- (i) Pelaksanaan Program Keamanan
- (ii) Sebagian Besar Ancaman dan Pengukuran Perbaikan
- (iii) Pengendalian Muara Akhir

3.1.1.6. Pengendalian Manajemen Operasi

- (i) Operasi Komputer



-
- (ii) Operasi Jaringan
 - (iii) Penyiapan dan Pengentrian Data
 - (iv) Pengendalian Produksi
 - (v) Pustaka File
 - (vi) Dokumentasi dan Pustaka Program
 - (vii) Help Desk / Dukungan Teknik
 - (viii) Perencanaan Kapasitas dan Pengawasan Kinerja
 - (ix) Pengelolaan Operasi Outsource

3.1.1.7. Pengendalian Manajemen Jaminan Kualitas

- (i) Fungsi QA
- (ii) Pertimbangan Pengorganisasian
- (iii) Hubungan Antara QA dan Auditing

3.1.1.8. Pengendalian Pembatasan (*Boundary*)

- (i) Pengendalian Cryptographic
- (ii) Pengendalian Akses
- (iii) Nomor Identifikasi Personal
- (iv) Tandatangan Digital
- (v) Kartu Kredit
- (vi) Pengendalian Audit Trail

3.1.1.9. Pengendalian Masukan

- (i) Metode Input Data
- (ii) Rancangan Dokumen Sumber / Dasar
- (iii) Rancangan Layar Entri Data
- (iv) Pengendalian Kode Data
- (v) Pengecekan Digit



-
- (vi) Pengendalian Batch
 - (vii) Validasi Input Data
 - (viii) Instruksi Masukan
 - (ix) Validasi Instruksi Masukan
 - (x) Pengendalian Audit Trail

3.1.1.10. Pengendalian Komunikasi

- (i) Ekspos Sub sistem Komunikasi
- (ii) Pengendalian Komponen Fisik
- (iii) Pengendalian Baris Kesalahan
- (iv) Pengendalian Flow
- (v) Pengendalian Hubungan
- (vi) Pengendalian Topologi
- (vii) Pengendalian Akses Chanel
- (viii) Pengendalian Atas Ancaman Subversif
- (ix) Pengendalian Antar Jaringan
- (x) Pengendalian dan Arsitektur Komunikasi
- (xi) Pengendalian Audit Trail
- (xii) Pengendalian Eksistensi

3.1.1.11. Pengendalian Pemrosesan

- (i) Pengendalian Pemroses
- (ii) Pengendalian Memori Nyata
- (iii) Pengendalian Memori Virtual
- (iv) Integrasi Sistem Operasi
- (v) Pengendalian Integritas
- (vi) Pengendalian Software Aplikasi
- (vii) Pengendalian Audit Trail



(viii) Pengendalian Eksistensi

3.1.1.12. Pengendalian Database

- (i) Pengendalian Akses
- (ii) Pengendalian Integritas
- (iii) Pengendalian Software Aplikasi
- (iv) Pengendalian Konkuren
- (v) Pengendalian Cryptographic
- (vi) Pengendalian Penanganan File
- (vii) Pengendalian Audit Trail
- (viii) Pengendalian Eksistensi

3.1.1.13. Pengendalian Output

- (i) Pengendalian Inferensi
- (ii) Pengendalian Distribusi dan Produksi Output Batch
- (iii) Pengendalian Rancangan Laporan Batch
- (iv) Pengendalian Distribusi dan Produksi Output On-line
- (v) Pengendalian Audit Trail

Pengendalian Eksistensi

Sistem Manajemen

Sub-sistem Manajemen	Penjelasan
Top Manajemen	Top manajemen harus memastikan fungsi sistem informasi di kelola dengan baik. Bertanggung jawab terutama untuk pengambilan keputusan dalam jangka panjang dan bagaimana sistem informasi akan di gunakan di dalam organisasi
Manajemen Sistem Informasi	Manajemen sistem informasi, secara keseluruhan bertanggung jawab dalam perencanaan dan pengendalian aktivitas sistem informasi. Juga menyampaikan masukan (saran) bagi top manajemen dalam kaitannya dengan kebijakan pengambilan keputusan jangka panjang dan menerjemahkan kebijakan jangka panjang tersebut ke dalam sasaran (<i>goal</i>) dan tujuan (<i>objectives</i>).

**diktat – A K S I**

Manajemen Pengembangan Sistem	Manajemen pengembangan sistem, bertanggung jawab atas desain, implementasi, dan pemeliharaan dari sistem aplikasi.
Manajemen Pemrograman	Manajemen Pemrograman, bertanggung jawab untuk pembuatan program sistem yang baru, pemeliharaan sistem yang lama, dan menyediakan dukungan sistem softwarenya.
Administrasi Data	Administrasi data, bertanggung jawab untuk mengarahkan perencanaan dan pengendalian dalam hubungannya dengan akses dan pengorganisasian data.
Manajemen Jaminan Kualitas	Manajemen Jaminan Kualitas, bertanggung jawab untuk penjaminan pengembangan sistem informasi, implementasi, operasional, dan pemeliharaan untuk meningkatkan standar kualitas.
Administrasi Keamanan	Administrasi keamanan, bertanggung jawab untuk pengendalian akses dan keamanan fisik dari fungsi sistem informasi.
Manajemen Operasional	Manajemen Operasional, bertanggung jawab untuk perencanaan dan pengendalian operasional sehari-hari (<i>day-to-day operations</i>) dari sistem informasi.

2. Sistem Aplikasi

Sub-sistem Aplikasi	Penjelasan
Pembatasan (<i>Boundary</i>)	Berupa komponen yang menetapkan hubungan (pembatasan) antara user dan sistem.
Input	Berupa komponen yang menangkap (<i>capture</i>), menyiapkan, dan memasukan perintah dan data kedalam sistem.
Communication	Berupa komponen yang mengirimkan data antar sub-sistem dan sistem.
Processing	Berupa komponen yang melaksanakan (memproses) pengambilan keputusan, perhitungan, klasifikasi, pemesanan, peringkasan data dalam sistem.
Database	Berupa komponen yang mendefinisikan, menambah, mengakses, memodifikasi, dan menghapus data dalam sistem.
Output	Berupa komponen yang mengambil dan mempresentasikan data untuk user dari sistem.

2.4. Resiko Pemeriksaan (*Audit Risks*)

Jika kita tinjau kembali, perhatian (konsen) dari audit SI mempunyai 4 tujuan yaitu :

1. Perlindungan asset (asset safeguarding),
2. Integritas data (data integrity),
3. Efektifitas Sistem (system effectiveness), dan
4. Efisiensi sistem (system efficiency).



Auditor eksternal dan internal, mereka kosen dengan apakah kesalahan atau ketidakberesan di sebabkan oleh hilangnya material (material losess) atau pernyataan yang salah tentang material dalam penyiapan informasi keuangan oleh organisasi.

Jika anda sebagai auditor internal, mungkin anda juga akan kosen dengan masalah kehilangan material dan akan mencoba untuk melakukan efisiensi dan efektifitas operasi. Auditor eksternal, juga sama, kosen sewaktu kemungkinan operasi tidak efisien dan efektif yang akan menimbulkan menurunnya kinerja organisasi. Lagi pula, banyak laporan auditor eksternal seperti permasalahan sebagian dari pelayanan professional lainnya untuk mengatur (memenej) organisasi.

Untuk mengukur apakah tujuan perlindungan asset, integritas data, efektivitas sistem, dan efisiensi sistem tercapai, auditor mengumpulkan fakta-fakta. Dikarenakan biasanya di dalam pengetesan audit, seoranag auditor harus mampu mendeteksi realitas atau potensi kemungkinan hilangnya data atau kesalahan pencatatan.

Resiko kelemahan auditor untuk mendeteksi secara dini atau menyebabkan hilangnya materi yang potensial atau kesalahan dalam pencatatan sebagai jalan keluar (konklusi) dari audit disebut resiko audit (audit risk).

Auditor harus memilih pendekatan dan merancang prosedur audit di dalam mencoba untuk mengurangi resiko untuk tingkat pertimbangan yang dapat diterima.

Faktor Resiko	Penjelasan
Sistem Keuangan (<i>financial systems</i>)	sistem digunakan untuk menyediakan pengendalian keuangan yang merupakan bagian terbesar aset organisasi. Contoh penerimaan dan pengeluaran kas, penggajian, penerimaan dan pembayaran accounts. Pada umumnya mempunyai resiko yang tinggi. Sering menjadi sasaran penipuan dan penggelapan.
Sistem Strategis (<i>strategic system</i>)	Sistem yang menyediakan organisasi untuk keunggulan kompetitif, contoh rahasia (kunci) customer,supplier atau bagian dari paten atau merk atau rahasia dagang, mempunyai resiko tinggi. Sering dijadikan sasaran / target spionase dan aksi / tindakan pembalasan oleh pesaing.
Sistem Operasional Kritis (<i>Critical Operational Systems</i>)	Sistem yang dapat melemahkan organisasi jika terjadi kesalahan. Contoh sistem reservasi pelanggan, sistem pengendalian produksi. Sering mempunyai resiko tinggi.
Sistem Keunggulan Teknologi (<i>Technologically Advanced Systems</i>)	Sistem yang menggunakan teknologi tinggi, sering mempunyai resiko tinggi, karena biasanya sangat kompleks dan banyak memberikan celah yang dapat memperlemah organisasi.

2.5. Tipe Prosedur Pemeriksaan (*Types of Audit Procedures*)

Tipe Prosedur Pemeriksaan :

1. Procedure to obtain an understanding of controls
2. Test of controls



3. Substantive test of details of transactions
4. Substantive test of details of accounts balances
5. Analytical review procedures

Auditor dapat menggunakan tipe prosedur sejenis untuk menilai efisiensi dan efektivitas operasi organisasi :

1. Procedures to obtain an understanding of controls
2. Test of controls
3. Substantive test of details of transactions
4. Substantive test of details of accounts balances
5. Analytical review procedures

2.6. Gambaran Langkah-langkah dalam Pemeriksaan (*Overview of Steps in an Audit*)

Perencanaan Audit

Hubungan lima komponen kontrol internal :

1. Lingkungan Pemeriksaan (*Control Environment*)
2. Perkiraan Resiko (*Risk Assessment*)
3. Aktivitas Pemeriksaan (*Control Activities*)
4. Komunikasi dan Informasi (*Information and Communication*)
5. Pengawasan (*Monitoring*)

Tes Pemeriksaan

1. Tes Transaksi (*Test of Transactions*)
2. Tes Keseimbangan atau Hasil Keseluruhan (*Test of Balances or Overall Results*)
3. Penyelesaian Audit (*Completion of the Audit*)
 - a. Disclaimer of opinion
 - b. Adverse opinion
 - c. Qualified opinion
 - d. Unqualified opinion

2.7. Pemeriksaan Lingkungan Komputer Menyeluruh (*Auditing Arround or Through the Computer*)



Pemeriksaan Sekitar Komputer (*Auditing Around the Computer*)

1. Their inherent risk is low
2. Their logic is straightforward
3. Input transaction are batched
4. Processing primarily consist
5. A clear audit trail exist
6. The Task environment

Seluruh Pemeriksaan Komputer (*Auditing Throught the Computer*)

1. Their inherent risk association with application system is high
2. The application system processes large volumes of input and produces large volumes of output.
3. Significant parts of the internal control system are embodied in the computer system.
4. The processing logic mbedded within the application system is complex
5. Because of cost-benefit considerations, substantial gaps in the visible audit trail are common in the system.

Design Interface & Dialog

(rb.1,pp.553-598)

Learning Objectives :

1. menjelaskan proses desain interface dan dialog dan kreasinya
 2. membandingkan dan menerapkan beberapa metoda untuk interaksi dengan sistem
 3. mendaftarkan dan menjelaskan variasi input device dan diskusi tentang isu kegunaan kaitannya dengan pelaksanaan tugas yang berbeda.
 4. menjelaskan dan menerapkan pedoman umum desain interface dan pedoman spesifik untuk desain layout, penstrukturan field entri data, penyediaan fasilitas feedback 'providing feedback', dan sistem help.
 5. memilih mekanisme variasi untuk penyediaan fasilitas sekuriti interface sistem.
 6. desain dialog manusia-komputer (human-computer dialogues), termasuk penggunaan diagram dialog
 7. desain GUI (graphical user interface)
-

Introduction

Pada bab ini akan di pelajari mengenai desain dialog dan interface sistem.

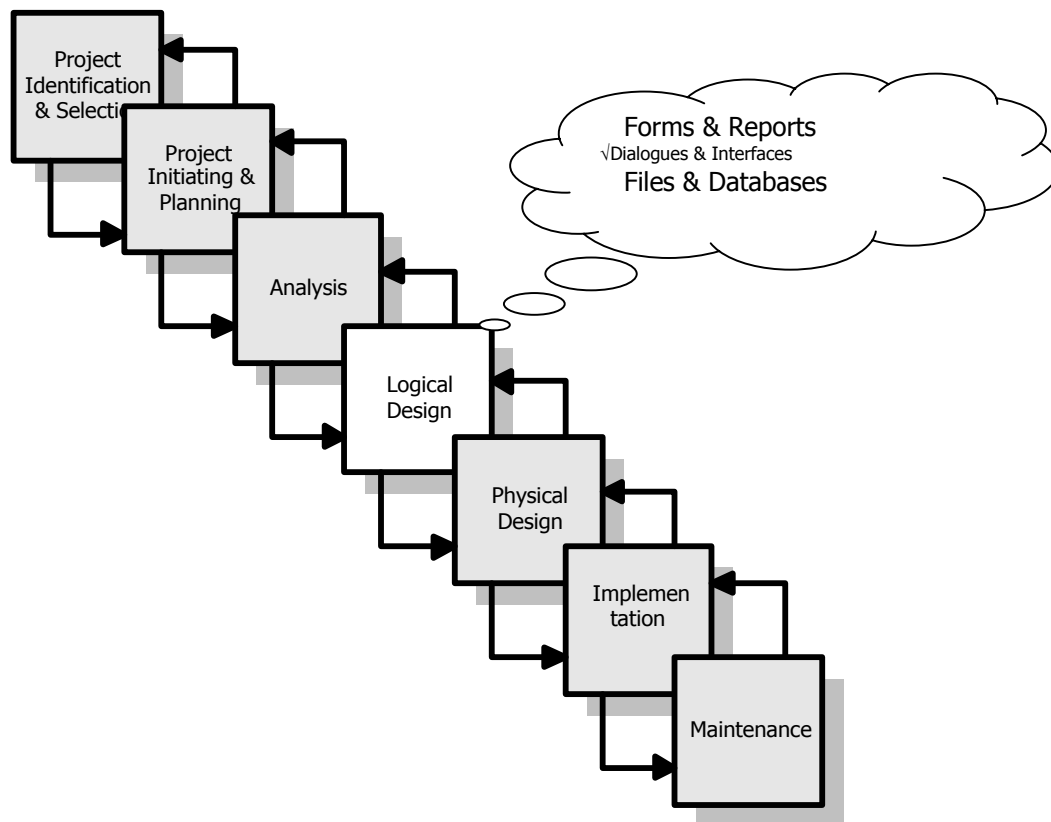
- Fokus desain interface sistem adalah bagaimana informasi di sediakan untuk dan di tangkap (capture) dari user.
- Fokus desain dialog adalah pada rangkaian display (tampilan) interface.



Desain interface dan dialog adalah proses mendefinisikan pengertian pertukaran informasi antara user dan komputer.

✚ Designing Interfaces & Dialogues

Pada pembahasan ini kita akan masih memfokuskan pembahasan pada fase desain logika (*logical design*) dalam tahapan SDLC seperti pada gbr.2.1 di bawah ini.



Gbr.2.1 Tahap SDLC, Desain Logika, dengan pensorotan dialog dan interface

Pada bab ini kita akan membahas materi yang berkaitan dengan sistem **dialog dan interface** pada bab sebelumnya telah kita bahas desain input dan output - **forms & reports**.

Pada bab selanjutnya juga kita akan membahas mengenai **logical databases design**.

✚ Proses Desain Interface & Dialog



Seperti pada desain form dan report, proses desain interface dan dialog adalah suatu kegiatan (aktivitas) yang fokusnya pada user (user-focused activity). Untuk mendesain interface dan dialog, kita dapat menggunakan rumusan 4W+H yaitu who, what, when, where, dan how-bagaimana pertanyaan digunakan sebagai pedoman desain form dan report, lihat table berikut :

Tabel 2.1 Pertanyaan mendasar sewaktu mendesain form dan report

1. Who - siapa yang akan menggunakan ?
2. What - apakah maksud form dan report ?
3. When – kapan form dan report di butuhkan dan digunakan?
4. Where – dimana form dan report di butuhkan untuk di sampaikan dan digunakan?
5. How – bagaimana kebutuhan orang banyak untuk menggunakan atau memanfaatkan form dan report?

Hasil dan Keluaran

Hasil dan keluaran dari interface dan desain dialog sistem adalah kreasi spesifikasi desain. Spesifikasi ini juga seperti untuk menghasilkan spesifikasi untuk form dan desain report- dengan satu pengecualian. Spesifikasi desain di bagi dalam tiga bagian, yaitu :

1. Narrative Overview (ikstisar narasi)
2. Sample Design (contoh desain)
3. Testing and Usability assessment (pengetesan dan perkiraan kebergunaan).

Untuk interface dan desain dialog, satu tambahan sub seksi adalah termasuk seksi iktisar urutan dialog (cara user berpindah dari satu dialog ke dialog lainnya). Pada pembahasan selanjutnya akan kita pelajari urutan desain dialog dengan menggunakan diagram dialog (dialogue diagramming) dan diagram transisi status (state-transition diagramming). Garis besar untuk spesifikasi desain untuk interface dan dialog digambarkan seperti gambar berikut :

Design Specification

1. Narrative Overview
 - a. Interface/Dialogue Name
 - b. User Characteristics
 - c. Task Characteristics
 - d. System Characteristics
 - e. Environmental Characteristics



2. Interface/Dialogue Design
 - a. Form/Report Designs
 - b. Dialogue Sequence Diagram(s) and Narrative Description.
3. Testing and Usability Assesment
 - a. Testing Objectives
 - b. Testing Procedures
 - c. Testing Results
 - a. Time to learn
 - b. Speed of Performance
 - c. Rate of errors
 - d. Retention Over Time (daya tahan terhadap waktu)
 - e. User Satisfaction and other perceptions

Gbr.2.2 Garis besar spesifikasi desain interface dan dialog

Metoda Interaksi dan Alat (device)

Metoda Interaksi

Metoda interaksi dapat dilakukan dengan teknik :

1. Command Line
2. Menu
3. Form
4. Object
5. Natural Language

Command Language Interaction

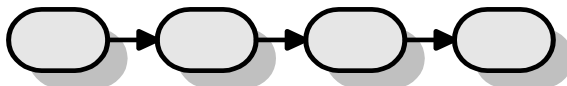
Contoh : COPY C:PAPER.DOC A:PAPER.DOC

Menu Interaction

- a. Single Menu

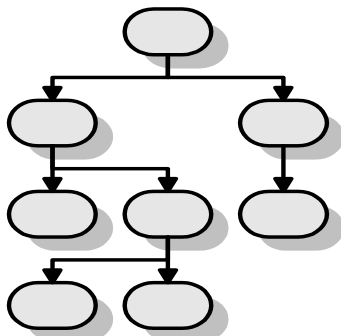


- b. Linier Sequence Menu

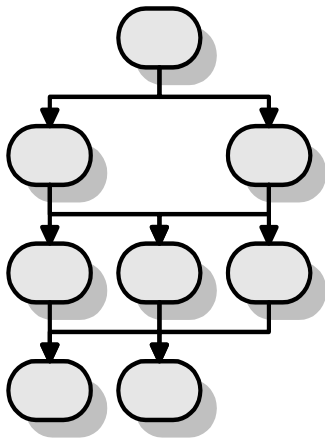




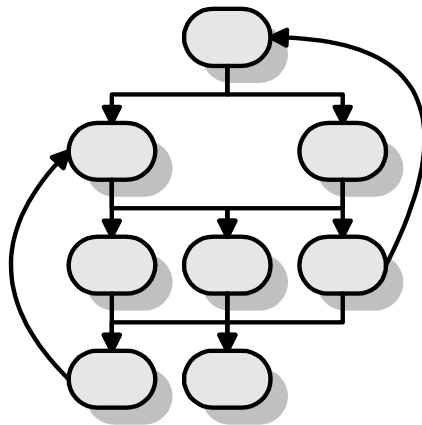
c. Multi-level Tree Menu



d. Multi-level Tree Menu with Multiple Parents



e. Multi-level Tree Menu with Multiple Parents and Multi-level Traversal



Single-Level Menu

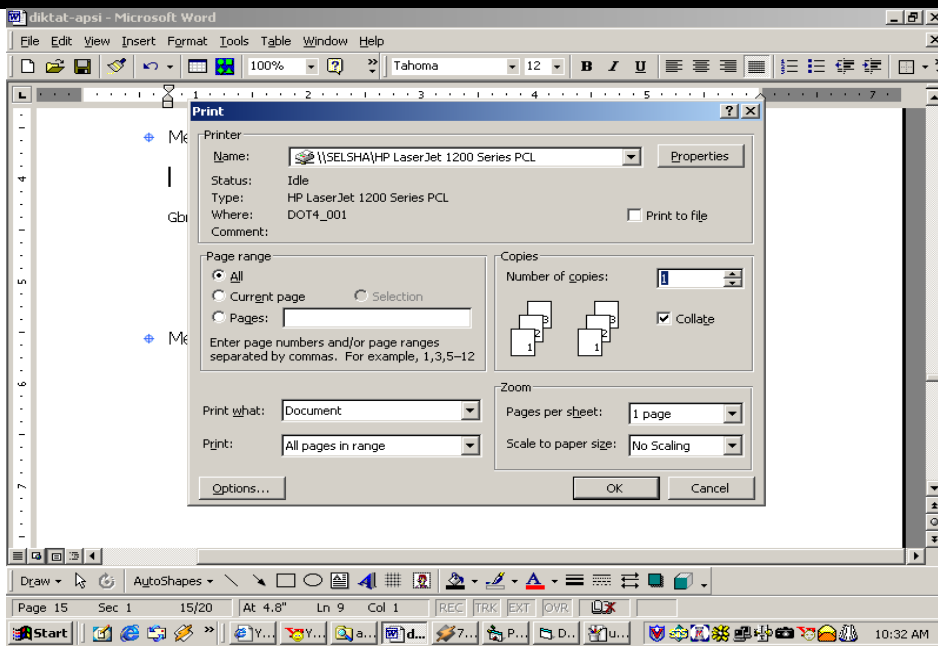


Gbr.2.3 Menu level-tunggal (level-single menu) untuk aplikasi DOS

Pop-Up Menu

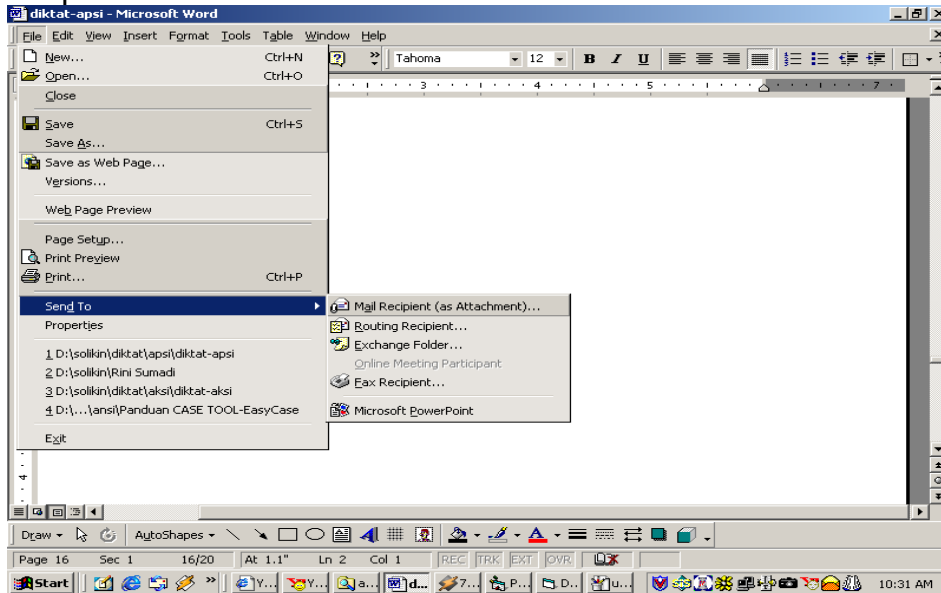


diktat – AKSI



Gbr.2.4 Menu Pop-Up

Drop Down Menu





Gbr.2.5 Drop-Down Menu

Definisi-definisi :

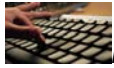
- ✦ **Interface (antar muka)** adalah sebuah metoda interaksi antara user dan system.
- ✦ **Command Language Interaction** adalah metoda interaksi user dan komputer, dimana user memasukkan perintah (statement) kedalam sistem untuk suatu operasi tertentu.
- ✦ **Menu Interaction** adalah metoda interaksi user dan komputer, dimana sejumlah pilihan sistem didaftarkan / disediakan, user tinggal menentukan / memilih pilihan yang tersedia.
- ✦ **Pop-Up Menu** adalah metoda posisi menu yang di lakukan melalui pendekatan posisi cursor.
- ✦ **Drop-Down Menu** adalah metoda posisi menu yang dapat dipilih mulai dari pilihan paling atas untuk ditampilkan, sewaktu diakses, menu di buka oleh 'dropping down' untuk ditampilkan.
- ✦ **Form Interaction** adalah metoda interaksi user dan komputer melalui form (spreadsheet).
- ✦ **Object Base Interaction** adalah metoda interaksi user dan komputer, dimana digunakan simbol untuk merepresentasikan perintah atau fungsi.
- ✦ **Icon** adalah gambar garfish yang merepresentasikan fungsi spesifik dalam sistem.
- ✦ **Natural Language Interaction** adalah interaksi user dan komputer, dimana masukan untuk dan keluaran dari aplikasi berbasis komputer dilakukan melalui percakapan konvensional bahasa seperti bahasa Inggris.

Tabel 2.2 Pedoman Desain Menu

Wording	<ul style="list-style-type: none"> ▪ Each menu should have a meaningful title ▪ Command verbs should clearly and specifically describe operations ▪ Menu items should be displayed in mixed upper and lower case letters and have a clear, unambiguous interpretation
Organization	<ul style="list-style-type: none"> ▪ A consistent organizing principle should be used that relates to the tasks the intended user perform; for example, related options should be grouped together and the same option should have the same wording and codes each time it appears Each menu should have a meaningful title.

**diktat – AKSI**

Length	<ul style="list-style-type: none">▪ The number of menu choices should not exceed the length of the screen▪ Submenus should be used to break up exceedingly long menus.
Selection	<ul style="list-style-type: none">▪ Selection and entry methods should be consistent and reflect the size of the application and sophistication of the users▪ How the user is to select each option and the consequences of each option should be clear (e.g. whether another menu will appear).
Hightlighting	<ul style="list-style-type: none">▪ Hightlighting should be minimized and used only to convey selected options (e.g. a check mark) or unavailable options (e.g. dimmed text)



BAB III

Manajemen Kontrol Pengembangan Sistem

*Kontrol Manajemen Pengembangan Sistem?
(Systems Development Management Controls)*

- Pendahuluan
- Pendekatan Audit (Kontrol) Pengembangan Sistem
 1. Model Normatif Proses Pengembangan Sistem
 - Pendekatan Daur Hidup Pengembangan Sistem (SDLC)
 - Pendekatan Desain Siosioteknik
 - Pendekatan Politik
 - Pendekatan Soft-System
 - Pendekatan Prototipe
 - Pendekatan Ketidaktentuan (Contingency)
 2. Tahap Evaluasi Proses Pengembangan Sistem
 - Definisi Masalah atau Peluang
 - Proses Perubahan Manajemen
 - Penilaian Kelayakan dan Masukan
 - Analisis Sistem yang Sedang Berjalan (Existing System)
 - Strategi Kebutuhan Formulasi
 - Organisasi dan Rancangan Kerja (Job)
 - Pengolahan Informasi Desain Sistem
 - Pengembangan dan Pengadaan Software Aplikasi
 - Pengadaan Hardware / Software Sistem
 - Pengembangan Prosedur
 - Pengetesan Penerimaan
 - Konversi
 - Operasional dan Pemeliharaan
- Pendahuluan

Manajemen pengembangan sistem mempunyai tanggung jawab terhadap proses penganalisaan, perancangan, pembangunan, penerapan, dan pemeliharaan sistem informasi. Dalam banyak cara, bagaimana kita dapat melakukan fungsi itu semua adalah merupakan seni. Walaupun kita sudah membuat kemajuan substansil dalam kaitan dengan pendekatan heuristik dan teori yang ada untuk memandu praktek, pekerjaan pengembangan sistem yang baik masih mengacu pada pemahaman yang mendalam, intuisi, dan pengalaman perancang dan analis sistem.

Bab ini menguraikan bagaimana kita dapat melakukan suatu audit subsistem manajemen pengembangan sistem. Kita mulai dengan mempertimbangkan tiga cara pendekatan berbeda untuk audit: sebagai pelaku di dalam proses pengembangan sistem (Concurrent Audit), sebagai reviewer pasca implementasi suatu sistem aplikasi spesifik (Postimplementation Audit), atau reviewer proses pengembangan sistem secara umum (General Audit). Berikutnya kita menguji beberapa pendekatan utama yang diharapkan untuk menyediakan petunjuk berdasarkan norma untuk pengembangan sistem. Akhirnya, kita mempertimbangkan tugas utama yang dilakukan selama pengembangan sistem, percobaan pengendalian atas tugas ini, dan tatacara kita mengevaluasi keandalan dari pengendalian ini.



Concurrent Audit : auditor sebagai anggota tim pengembangan sistem. Mereka membantu tim dalam meningkatkan kualitas pengembangan sistem untuk pembangunan dan pengimplementasian sistem spesifik

Postimplementation Audit : auditor mencari untuk membantu organisasi belajar dari pengalaman dalam pengembangan sistem aplikasi yang spesifik. Tambahan, mereka boleh juga melakukan evaluasi kebutuhan sistem untuk di susun kembali, dilanjutkan atau di modifikasi dalam berbagai cara.

General Audit : auditor mengevaluasi pengendalian pengembangan sistem secara menyeluruh.

PENDEKATAN UNTUK AUDITING PENGEMBANGAN SISTEM

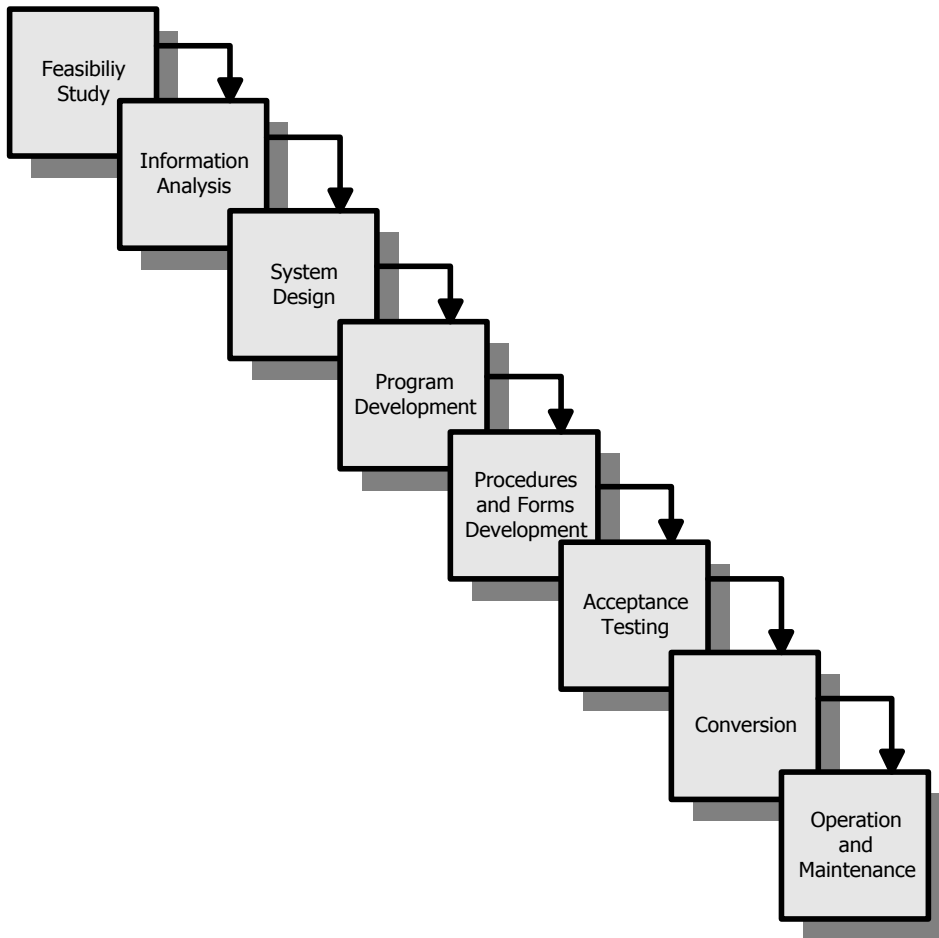
- Model Normatif Proses Pengembangan Sistem

- **Pendekatan Daur Hidup Pengembangan Sistem (SDLC)**

Biasanya, personal pengembangan sistem sudah memikirkan bagaimana tahapan utama proses pengembangan sistem dilakukan dalam siklus hidup.

Pendekatan siklus hidup muncul dari usaha dini untuk menerapkan teknik manajemen proyek untuk proses pengembangan sistem. Menurut sejarahnya, banyak sistem yang dibangun dengan biaya tinggi, evaluasi ekonomi yang tidak cukup, desain sistem yang tidak cukup, penyerahan kepada manajemen, komunikasi yang kurang baik, pengarahan yang kurang cukup, dan lain sebagainya. Pendekatan siklus hidup dikembangkan untuk membantu beberapa masalah. Yang dapat diselesaikan dengan baik oleh tugas dalam batas siklus hidup, manajemen proyek dan teknik kontrol dapat di aplikasikan. Untuk pengembangan sistem dengan kualitas yang baik, setiap tahap siklus hidup harus direncanakan dan di kontrol dan dikembangkan sesuai dengan standar, dokumentasi yang cukup, diorganisir oleh personel berkompeten, mempunyai poin pengecekan dan lain sebagainya.

Banyak bentuk SDLC, salah satunya adalah seperti terlihat pada gambar 3.1 dibawah ini, yang sering juga di sebut dengan waterfall model.



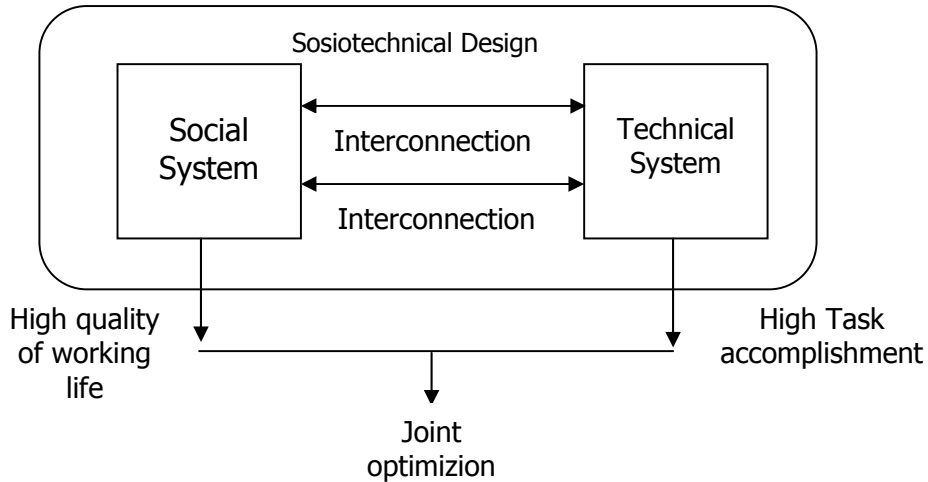
gbr.3.1 Siklus hidup tradisional atau model waterfall pengembangan sistem

Phase	Penjelasan
Feasibility Study	Penerapan kriteria untung rugi (cost-benefit) untuk aplikasi yang diusulkan
Information analysis	Menentukan kebutuhan informasi user
System Design	Menentukan antarmuka (interface) user, file yang digunakan, dan fungsi proses informasi untuk membentuk system
Program Development	Desain, coding, compiling, testing, dan dokumentasi program
Procedures and forms Development	Mendesain dan mendokumentasikan prosedur sistem dan form dari user tentang system
Acceptance Testing	Pengetesan akhir dari system dan persetujuan formal dan penerimaan oleh pihak manajemen dan user.
Conversion	Perubahan dari system lama ke system baru
Operation and Maintenance	Menjalankan (running) produk (program) system dan modifikasi, pemeliharaan terhadap problem yang muncul.



▪ **Pendekatan Desain Siosioteknik**

Pendekatan Desain Siosioteknik dapat digambarkan sbb :

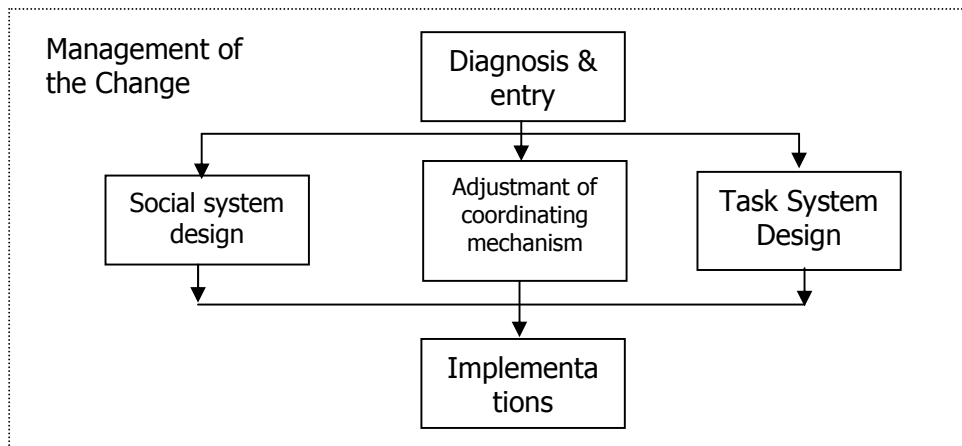


gbr.3.2 Sasaran desain system sosiotechnical

Pada pertengahan tahun 1970, sebuah pendekatan baru muncul yang fokusnya pada problem perilaku. Pendekatan ini disebut desain sosiotechnical, mencari solusi untuk mengoptimalkan dua sistem secara bersama-sama, yaitu :

- (a) sistem teknik (the technical system), sasarannya adalah untuk memaksimalkan pemenuhan tugas dan
- (b) sistem social (the social system), sasarannya adalah untuk memaksimalkan kualitas kerja pemakai sistem

Tahapan pendekatan sosiotechnical adalah sbb :

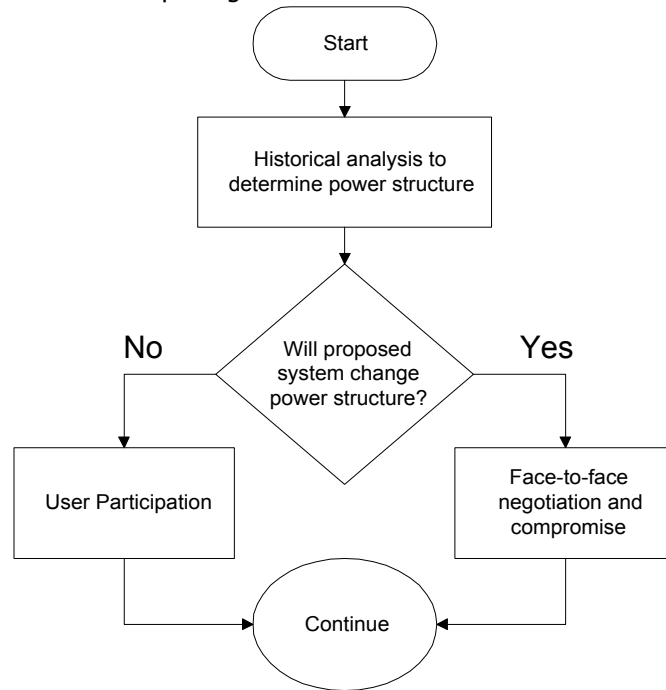


gbr.3.3 Tahap utama proses desain sistem sociotechnical



- **Pendekatan Politik**

Pendekatan Politik dapat digambarkan sbb :



gbr.3.4 Pendekatan politik dan keterlibatan user dalam proses pengembangan system

Sewaktu pendekatan politik untuk pengembangan sistem informasi diadopsi, sebuah tugas kritis adalah untuk mempelajari latar belakang (sejarah) organisasi. Dalam mempelajari latar belakang organisasi, perancang dapat mengevaluasi apakah sistem yang diinginkan akan tetap sama seperti sistem yang sedang berjalan ataukah mengharuskan untuk mengadakan perubahan struktur. Strategi pengembangan dan implementasi harus diganti atau tidak tergantung pada dampak dari sistem yang diajukan akan mempunyai kekuatan untuk mengubah struktur sistem yang sedang berjalan atau tidak. Lihat gbr. 3.4 tersebut diatas.



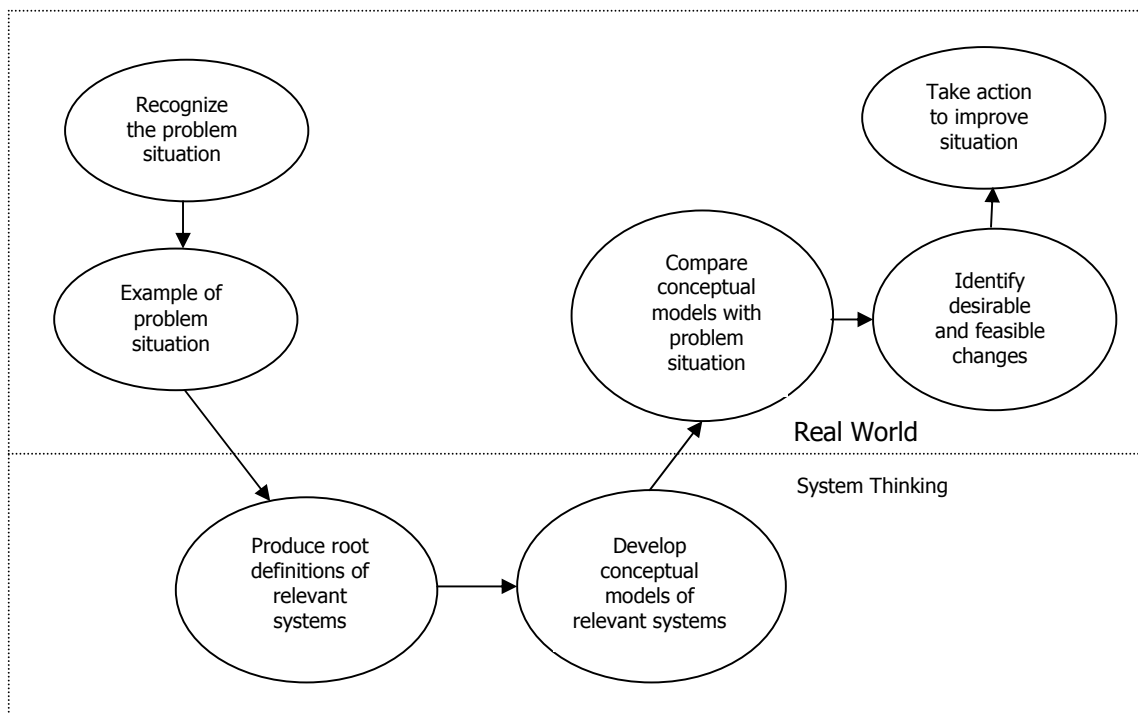
▪ Pendekatan Soft-System

Pada pertengahan tahun 1970, Checkland(1981) dan koleganya mengembangkan pendekatan yang didesain untuk membantu pengambil keputusan untuk mempelajari tentang dan pemahaman yang lebih dari problem struktur yang kurang baik. Mereka menyebutnya pendekatan "soft-system methodology" (SSM). Disebut SSM karena fokusnya pada learning (pembelajaran) dan innovation (inovasi) pada situasi masalah (problem).

Mereka membedakan pendekatan mereka dari pendekatan "hard system" dengan asumsi itu (terutama sekali pembuat keputusan) mempunyai tujuan spesifik dan memahami substansi solusi masalahnya.

SSM melibatkan tujuh langkah yaitu :

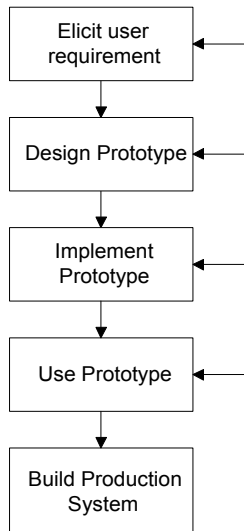
1. Recognize the problem situation (pengenalan terhadap situasi masalah)
2. Example of problem situation (contoh dari situasi masalah)
3. Produce root definitions of relevant systems (definisikan hasil utama sistem yang relevan)
4. Develop conceptual models of relevant systems (kembangkan model konseptual system yang relevan)
5. Compare conceptual models with problem situation (Bandingkan model konseptual dengan situasi masalah)
6. Identify desirable and feasible changes (identifikasi keinginan dan perubahan yang mungkin)
7. Take action to improve situation (lakukan aksi untuk perbaikan situasi)



gbr.3.5 Langkah utama dalam metodologi soft-systems



▪ **Pendekatan Prototipe**



gbr.3.6 Metodologi prototyping untuk pengembangan sistem

▪ **Pendekatan Ketidaktentuan (Contingency)**

1. Dampak Sistem sosial (Social Systems Impact)
2. Dampak Sistem Tugas (Task Systems Impact)
3. Ukuran Sistem (System Size)
4. Penggunaan komponen sama (Commonality)
5. Ketidakpastian Kebutuhan (Requirement Uncertainty)
6. Ketidakpastian Teknologi (Technological Uncertainty)



▪ Tahap Evaluasi Proses Pengembangan Sistem

Kualitas pengembangan sistem akan bergantung pada bagaimana keterlibatan (tanggapan) stakeholder terhadap proyek pengembangan sistem yang sedang ditangani. Jika Auditor mengambil bagian dalam proses pengembangan sistem, mereka akan merupakan bagian dari pertimbangan pengambilan keputusan kelompok stakeholder tentang pengarahannya terhadap isu-isu yang ada bagi pengembangan sistem. Jika, pada sisi lain, mereka melaksanakan juga tinjauan ulang (ex post view) terhadap sistem yang spesifik atau proses pengembangan sistem secara umum, mereka akan mengevaluasi bagaimana stakeholder mengarahkan isu-isu yang ada tersebut dan bagaimana pengaruhnya setelah sistem di kembangkan.

Pada uraian berikut akan dijelaskan tugas yang harus di kerjakan dan kontrol (pengendalian) yang mungkin cukup penting dalam 13 fase utama pengembangan sistem, yaitu :

1. Definisi Peluang dan Tantangan
2. Proses Perubahan Manajemen
3. Penilaian Kelayakan dan Masukan
4. Analisis Sistem yang Sedang Berjalan (Existing System)
5. Perumusan Kebutuhan Strategis
6. Organisasi dan Rancangan Kerja (Job)
7. Desain Sistem Pengolahan Informasi
8. Pengembangan dan Pengadaan Software Aplikasi
9. Pengadaan Hardware / Software Sistem
10. Pengembangan Prosedur
11. Pengetesan Penerimaan
12. Konversi
13. Operasional dan Pemeliharaan

Dalam tiap fase, kita harus mempertimbangkan bagaimana melakukan itu semua mungkin berbeda, tergantung pada tugas sistem dan dampak sosial, ukuran sistem, komponen sistem, dan kebutuhan dan lingkup ketidakpastian teknologi sistem. Kita juga harus mempertimbangkan bagaimana melakukan kontrol yang berbeda tergantung pada level factor ketidakpastian ini. Tujuan kita adalah untuk membangun fasilitas yang sama pada pemilihan model normatif yang baik dari proses pengembangan sistem itu, Auditor dapat melakukannya selama pengumpulan fakta dan evaluasi pekerjaan / tugas.

I. Definisi Peluang atau Kesempatan

Sistem informasi dapat dikembangkan untuk membantu memecahkan masalah atau untuk memperluas peluang (kesempatan). Peluang (kesempatan) dan tantangan itu boleh jadi dapat dipecahkan untuk dukungan sistem informasi yang dapat dikenali dengan dua cara yaitu :

Pertama, mereka dapat memikirkan seluruh hubungan proses formal dengan penyiapan rencana sistem informasi. Kedua, mereka dapat memikirkannya secara kebetulan.

Selama fase pendefinisian peluang / kesempatan, stakeholder harus mencoba terbuka untuk memahami bersama persoalan atau kesempatan yang ingin diraih. Adalah peluang atau kesempatan yang baik atau tidak?, mempunyai implikasi kecil atau besar terhadap personal?, Akankah solusi mempunyai dampak yang besar pada struktur organisasi dan job?, Akankah teknologi baru hampir dapat dipastikan dibutuhkan untuk mendukung solusi yang mungkin?.



II. Proses Perubahan Manajemen

Proses perubahan manajemen dilakukan secara paralel untuk semua fase. Proses perubahan dimulai dengan inialisasi konsep sistem dan berlanjut sampai dengan sistem yang baru berjalan dan organisasi dapat menyesuaikan dengan sistem yang baru tersebut.

Proses perubahan manajemen melibatkan dua tugas utama, yaitu : (1) Pengaturan proyek (project management) dan (2) Perubahan Fasilitas (change facilitation) lihat gbr. 3.7.

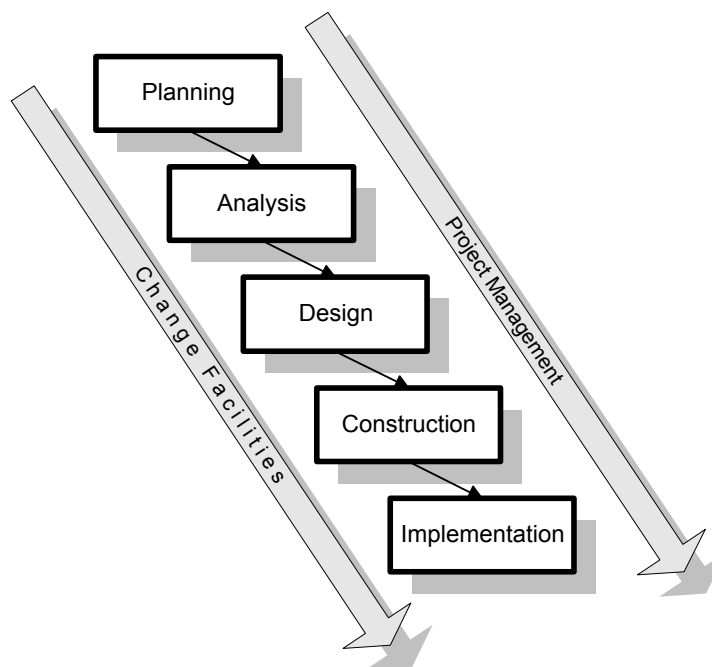
1. Pengaturan proyek : penganggaran (budgeting), laporan khusus / pengecualian (exception reporting), pemeriksaan (checkpoints), dan user signoffs.
2. Perubahan fasilitas adalah aspek pengembangan sistem kritis yang mungkin dapat mempengaruhi struktur organisasi dan job.

Model pandangan sekarang untuk perubahan fasilitas dalam pemeliharaan proses pengembangan sistem antara lain model yang dikemukakan oleh Lewin/Schein atau Kolb/Frolman :

Tiga aktivitas utama yang dibutuhkan dalam perubahan fasilitas adalah :

Kelas aktivitas	Penjelasan
<i>Unfreezing the Organization</i> (pembekuan organisasi)	Menyiapkan organisasi untuk perubahan.
<i>Moving the organization</i> (menjalankan organisasi)	Perubahan sistem kerja untuk sistem baru
<i>Refreezing the organization</i> (menjalankan kembali organisasi)	Membantu user untuk menyesuaikan dengan sistem yang baru

Selama fase ini, auditor harus mengevaluasi kualitas pembuatan keputusan tentang pengaturan proyek dan perubahan fasilitas (kemudahan) .



gbr.3.7 Pengaturan proses perubahan selama pengembangan system



III. Penilaian Kelayakan dan Masukan

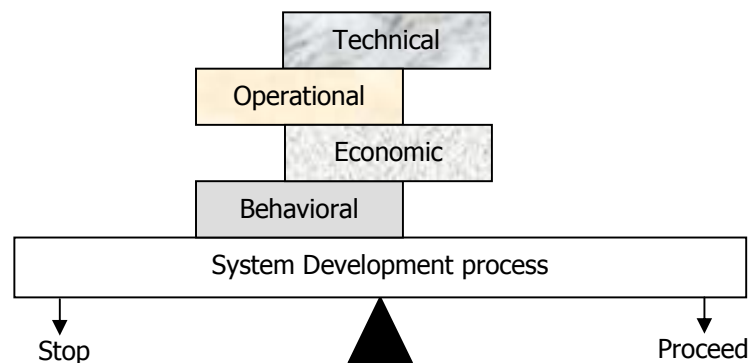
Tujuan dari fase penilaian kelayakan dan masukan adalah untuk memperoleh komitmen perubahan dan mengevaluasi apakah solusi penghematan biaya memungkinkan untuk mengarahkan peluang dan tantangan yang sudah berhasil diidentifikasi.

Pertimbangan, pertama, sebuah situasi dimana peluang atau masalah (tantangan) sudah dikenali oleh kelompok pemakai (user). Mereka yakin, mereka dapat merancang dan mengimplementasikan sebuah solusi yang digunakan dengan bahasa tingkat tinggi. Sistem yang diajukan akan mempunyai side efek / dampak kecil kepada organisasi, tidak akan mempengaruhi material dari sudut pandang keseluruhan organisasi. Pada situasi ini, user harus selalu diberi motivasi untuk melakukan perubahan. Selanjutnya, aktivitas untuk memenuhi kesuksesan masukan adalah tidak perlu atau kecil.

Pertimbangan lain, sebuah situasi dimana solusi potensi akan mempunyai dampak yang luas pada keseluruhan organisasi. Aktivitas untuk memenuhi kesuksesan masukan sekarang adalah kritis. Profesional system informasi harus mencari untuk menetapkan dirinya sebagai legitimasi agen perubahan antar stakeholder. Selanjutnya, mereka harus mencari untuk membantu antar stakeholder sebuah komitmen untuk perubahan. Jika solusinya potensial akan mempunyai dampak yang penting (significant) pada tugas dan sistem sosial, sebuah dorongan dari analisis kolaborasi dan evaluasi antar stakeholder harus dibangun.

Jika masukan sukses, perancang kemudian dapat menyelesaikan suatu studi persiapan untuk mengevaluasi kelayakan sistem baru menggunakan empat kriteria (lihat gbr.4.8), yaitu :

1. Kelayakan Teknik (Technical Feasibility)
2. Kelayakan Operasional (Operational Feasibility)
3. Kelayakan Ekonomi (Economic Feasibility)
4. Kelayakan Perilaku (behavioral feasibility)

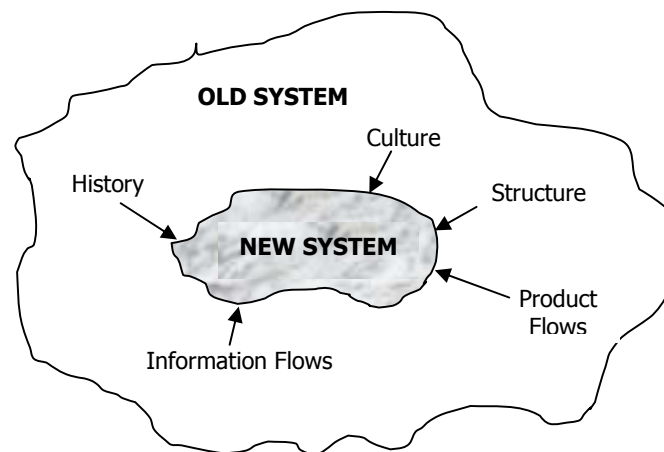


gbr.3.8 Kriteria kelayakan untuk pengembangan sistem



IV. Analisis Sistem yang Sedang Berjalan (Existing System)

Ketika sistem baru diajukan, dalam beberapa kasus ia akan menggantikan system yang sedang berjalan. Tenaga / fikiran Perancang dibutuhkan untuk memahami sistem yang sedang berjalan (existing system). Jika mereka menghendaki untuk melaksanakan pekerjaan dengan kualitas yang tinggi (baik) dalam pengembangan dan pengimplementasian sistem yang baru (lihat gbr.3.9).



Gbr.3.9 Keadaan existing system dapat mempengaruhi (memberi efek) terhadap sistem yang baru

Analisis penggunaan existing system melibatkan dua tugas utama, yaitu :

1. Studi terhadap sejarah perusahaan, struktur organisasi dan kultur (budaya)
2. Studi produk yang ada dan aliran informasi

Pada tahap studi terhadap sejarah perusahaan dan stuktur organisasi, auditor harus konsen dengan evaluasi keputusan perancang pada:

1. Apakah mereka membutuhkan untuk mempelajari sejarah organisasi sampai dengan sekarang, struktur dan kultur
2. Jika ya, apakah aspek yang dibutuhkan mereka untuk dipelajari
3. Berikan pilihan aspek yang dipelajari, tingkatan untuk dimana mereka mempunyai sesuatu untuk pemeriksaan.

Seperti perancang, auditor akan mempunyai pertimbangan keadaan dimana keputusan di buat dan refleksi pada pilihan yang mereka ambil. Mereka dapat melanjutkan perbandingan, pilihan perancang berlawanan dengan yang mereka miliki untuk mengevaluasi implikasi dari kemiripan dan perbedaan sikap audit.

V. Perumusan Kebutuhan Strategis

Kebutuhan strategis untuk sebuah sistem yang spesifik secara keseluruhan sasaran dan tujuan sistemnya harus terpenuhi. Biasanya mungkin masih kurang jelas, sebagai contoh "meningkatkan kekayaan pemegang saham" atau yang lebih spesifik, "mengurangi mutasi staff di bagian penjualan sebesar 30%". Kebutuhan strategis adalah dasar identifikasi untuk memahami sistem yang ada atau memahami peluang untuk meningkatkan kinerja tugas dan kualitas pekerjaan.

Auditor akan konsen untuk melihat rancangan system, itu penting agar kebutuhan strategis jelas untuk kualitas kerja rancangan berikutnya. Jika sistem yang diajukan akan mempunyai



dampak perilaku substansil, mereka akan menguji dan mengevaluasi prosedur yang digunakan oleh stakeholder untuk memperoleh persetujuan pada kebutuhan strategis. Jika sistem tidak mempunyai dampak substansil, mereka akan menguji dan mengevaluasi prosedur-prosedur yang digunakan untuk membantu klarifikasi kebutuhan strategis.

VI. Organisasi dan Rancangan Kerja (Job)

Dalam beberapa kasus, penyelesaian pilihan kebutuhan strategis untuk sistem akan memerlukan rancangan awal (initial design) atau rancangan ulang (redesign) struktur organisasi dan job.

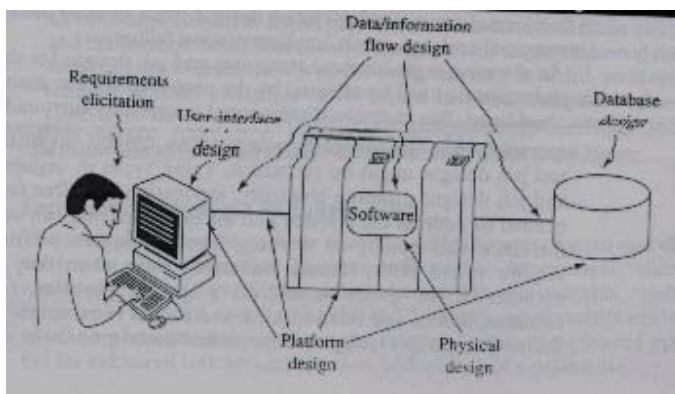
Dalam pemilihan struktur organisasi dan rancangan job suatu bagian organisasi akan efektif melalui usulan pengajuan sistem, banyak faktor yang dapat dipertimbangkan.

Dalam jangka pendek, desain stuktur organisasi dan job merupakan aktivitas yang kompleks. Jika auditor menilai sistem yang diajukan akan mempunyai dampak terhadap struktur organisasi dan job, maka mereka akan konsen untuk melihat masukan (advis) yang berkualitas yang diberikan oleh tim pengembangan sistem.

VII. Desain Sistem Pengolahan Informasi

Ketika mengevaluasi fase desain sistem pengolahan informasi, salah satu dari partisipan dalam proses desain atau dalam postimplementation atau meneliti kapasitas secara umum, maka auditor harus memeriksa enam aktivitas utama (lihat gbr. 3.10), yaitu :

1. Memunculkan kebutuhan secara detil (elicitation of detailed requirements)
2. Desain aliran data / informasi (design of the data / information flow)
3. Desain database (design of the database)
4. Desain hubungan dengan user (design of the user interface)
5. Desain fisik (physical design) dan
6. Desain dan akuisisi hardware dan platform system software (design and acquisition of the hardware / system software platform).

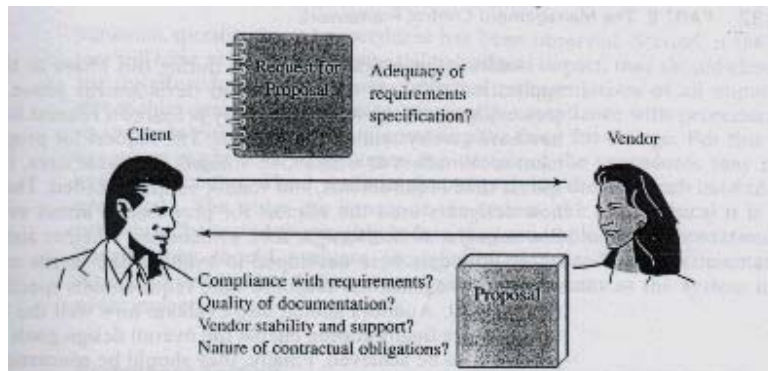


Gbr.3.10 Aktivitas utama dalam desain sistem pengolahan



VIII. Pengembangan dan Pengadaan Software Aplikasi

Auditor harus mempunyai konsentrasi selama akuisisi dan fase pengembangan. Jika aplikasi s/w terakuisisi, mereka akan konsen tentang pemenuhan spesifikasi kebutuhan yang disediakan untuk vendor, kualitas prosedur yang digunakan untuk mengevaluasi s/w, dll (lihat gbr.3.11).



Gbr.3.11 Konsen audit selama akuisisi h/w atau s/w

IX. Akuisisi (Pengadaan) Hardware / Software Sistem

Jika h/w baru atau s/w sistem mengharuskan membeli untuk mendukung aplikasi sistem yang baru, maka harus mengajukan permintaan agar disediakan / di lakukan pembelian. Auditor akan memeriksa pengadaan h/w dan s/w tersebut.

X. Pengembangan Prosedur

Pengembangan prosedur melibatkan empat tugas utama yaitu :

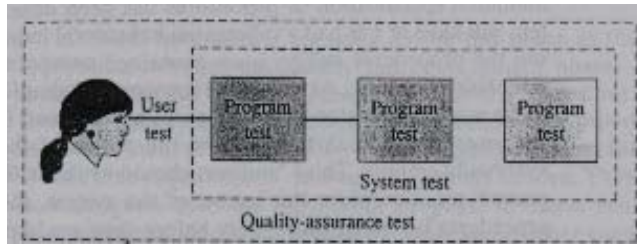
1. Desain of procedures
2. Testing of procedures
3. Implementation of procedures dan
4. Documentation of procedures



XI. Pengetesan Penerimaan

Terdapat empat tipe pengetesan masukan yaitu (lihat gbr.3.12) :

1. Program Testing
2. System Testing
3. User Testing
4. Quality assurance Testing



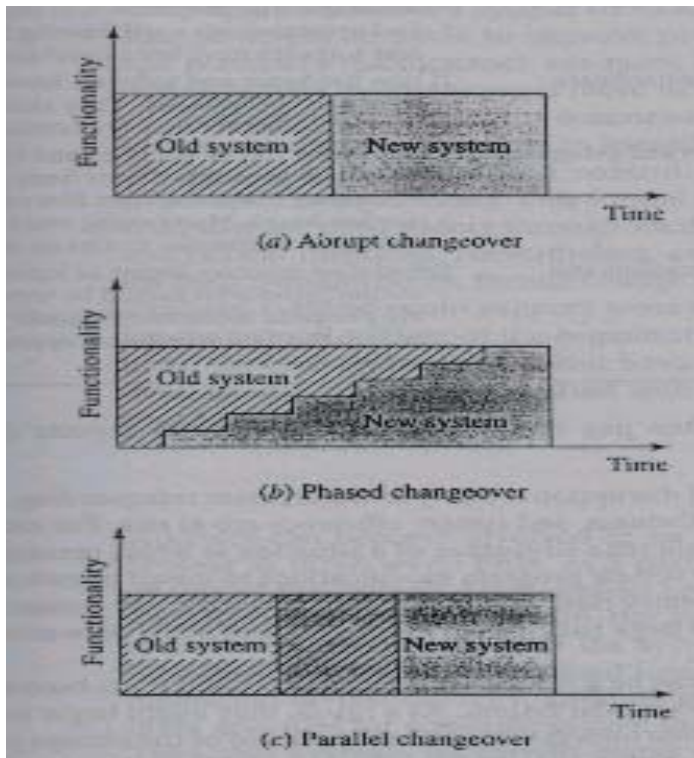
Gbr.3.12 Wilayah pengetesan



XII. Konversi

Terdapat 3 jenis konversi (lihat gbr.3.13) :

1. Abrupt changeover (tiba-tiba /sekaligus)
2. Phased changeover (bertahap)
3. Parallel changeover



Gbr.3.13 Strategi Konversi

XIII. Operasional dan Pemeliharaan

Selama fase operasional dan pemeliharaan, sistem baru dijalankan sebagai produk dari sistem.

Pemeliharaan mencakup :

1. Repair Maintenance (pemeliharaan perbaikan)
2. Adaptive Maintenance dan (pemeliharaan penyesuaian)
3. Perfective maintenance (pemeliharaan penyempurnaan)



BAB IV

Manajemen Kontrol Programming *(Programming Management Controls)*

1. Pendahuluan
2. Siklus hidup pengembangan Program :
 - 2.1. Perencanaan (Planning)
 - 2.2. Pengendalian (Control)
 - 2.3. Perancangan (Design)
 - 2.4. Pengkodean (Coding)
 - 2.5. Pengtesan (Testing)
 - 2.6. Pengoperasian dan Pemeliharaan (Operation and Maintenance)
3. Pengorganisasian Tim Programming
 - 3.1. Ketua Tim Programmer
 - 3.2. Penyesuaian Tim
 - 3.3. Desentraliasi Pengendalian Tim
4. Pengaturan Kelompok Programming Sistem
 - 4.1. Permasalahan Pengendalian
 - 4.2. Pengukuran Pengendalian

▪ **Pendahuluan**

Pada bab ini kita akan memeriksa secara praktis pengadaan atau produksi s/w dengan kualitas yang tinggi. Kita mulai dengan memeriksa fase utama pada siklus hidup pengembangan program. Diskusi di fokuskan pada tipe pengalaman yang baik dan konsen pengendalian yang harus dilakukan oleh Auditor pada tiap fase. Selanjutnya kita akan memeriksa cara lain untuk mengorganisasikan dan mengatur tim programmer, dari sudut pandang pengendalian kita akan memfokuskan pada keuntungan dan kelemahan perbedaan struktur tim yang akan digunakan. Pada akhirnya, kita akan memeriksa masalah pengendalian khusus yang timbul dalam kaitan untuk kegiatan dari sistem programmer. Kita mempertimbangkan beberapa pendekatan yang dapat digunakan untuk memudahkan persoalan pengendalian.

1. Siklus Hidup Pengembangan Program

Pembuatan dan pengembangan program adalah merupakan tahap penting dalam siklus hidup pengembangan sistem. Tujuan utama tahap ini adalah untuk menghasilkan atau memperoleh dan menerapkan program yang berkualitas.

Beberapa karakteristik program yang berkualitas adalah :

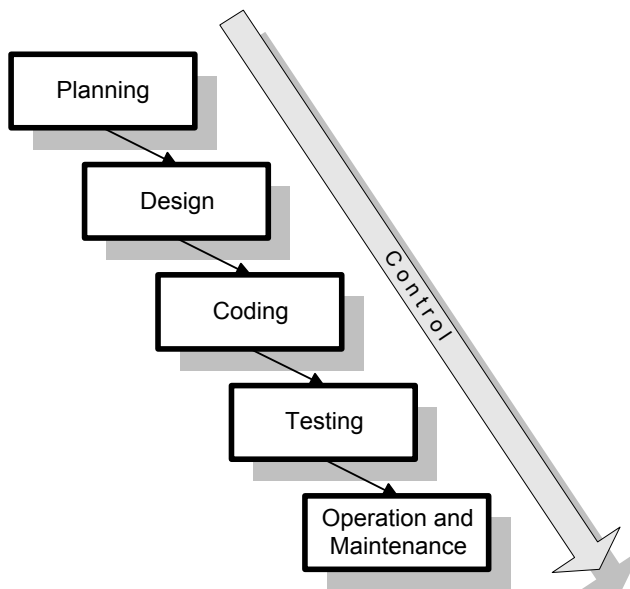
- a. Berfungsi dengan tepat dan lengkap
- b. Mempunyai user interface dengan kualitas tinggi (baik)
- c. Bekerja secara efisien
- d. Dirancang dan di dokumentasikan dengan baik
- e. Mudah untuk pemeliharaan
- f. Dapat menyesuaikan di bawah kondisi tidak normal



Jika program mempunyai karakteristik tersebut, aktivitas pengembangan, penerimaan dan implementasi program dapat diatur dengan baik. Dalam pengembangan sistem, auditor dapat menggunakan model siklus hidup untuk lebih memahami, merencanakan, dan menyelesaikan tugas dalam rangka untuk memperoleh s/w dengan kualitas baik. Selama proses audit, model ini juga dapat digunakan sebagai pedoman aktivitas pengumpulan dan evaluasi fakta.

Terdapat enam pedoman dalam pengembangan program, yaitu (lihat gbr.4.1) :

- 1.Planning
- 2.Control
- 3.Design
- 4.Coding
- 5.Testing dan
- 6.Operation and maintenance



Gbr.4.1 Siklus hidup pengembangan program



Perencanaan (Planning)

Tugas utama dari manajemen dalam tahap ini adalah untuk memperkirakan kebutuhan besarnya sumber daya (khususnya jam kerja) yang dibutuhkan dalam pengembangan, pengadaan, dan penerapan software. Jika, sebagai contoh, s/w di buat di rumah (in house), manajemen harus berusaha untuk memperkirakan berapa jumlah baris kode (program) yang di ketik atau banyaknya fungsi yang di buat.

Jika suatu software akan dikembangkan dan diimplementasikan secara in-house, manajemen harus memanfaatkan lima teknik perencanaan biaya yang di buat oleh Boehm (1984) sbb :

1. **Algorithmic Models** (model algoritma) : model ini akan memperkirakan jumlah sumber daya yang dibutuhkan berdasar pada faktor biaya, sebagai contoh memperkirakan jumlah instruksi yang harus di ketik (di tulis), bahasa pemrograman yang digunakan, dan perubahan pada permintaan kebutuhan. Dengan menggunakan model COCOMO (Boehm's (1981).
2. **Expert Judgment** (penilaian seorang ahli), seorang ahli dapat memperkirakan kebutuhan sumber daya yang diperlukan dalam proyek / pembuatan program. Menurut penelitian Vicinanza et al's (1991), seorang ahli dapat menjadi pembuat perkiraan yang lebih baik untuk menentukan sumber daya jika dibanding dengan model algoritma.
3. **Analogy** (analogi) : jika proyek software yang sama pernah dibuat, penentuan sumber daya yang dibutuhkan dapat di dasarkan pada pengalaman sebelumnya.
4. **Top-Down Estimation** (Perkiraan atas-bawah) : proyek di pecah kedalam beberapa tugas (pekerjaan), dan penentuan sumber daya yang dibutuhkan oleh setiap tugas tersebut baru dibuat.
5. **Bottom-Up Estimation** (Perkiraan bawah-atas) : jika tugas-tugas sudah di buat terlebih dahulu, kebutuhan sumber daya untuk masing-masing dapat diperkirakan dan di satukan / dikumpulkan untuk keperluan seluruh kebutuhan proyek.

Selain memperkirakan kebutuhan sumber daya, manajemen juga harus memutuskan tujuan dari keputusan penting yang dibuat selama fase perencanaan seperti :

Keputusan	Keterangan
Design Approach (pendekatan desain)	Jika program akan di kembangkan di rumah, manajemen harus memilih pendekatan desain yang akan digunakan seperti : prototyping atau tipe top-down atau bottom-up.
Implementation approach (pendekatan implementasi)	Software dapat dikembangkan di rumah, pengembang dapat membangun atau mengemas s/w untuk dijual. Jika s/w akan dibuat koding, harus dibuat keputusan tentang bahasa pemrograman yang akan dipakai.
Integration and testing approach (pendekatan testing dan integrasi)	Pada saat pengetesan akan dibutuhkan sumber daya yang khusus, seperti simulator atau hardware tertentu untuk memonitor jalannya program.
Project team organization (pengorganisasian tim proyek)	Saat tim proyek harus di bentuk untuk pengembangan atau pengadaan s/w, maka anggota dan struktur tim harus ditentukan.

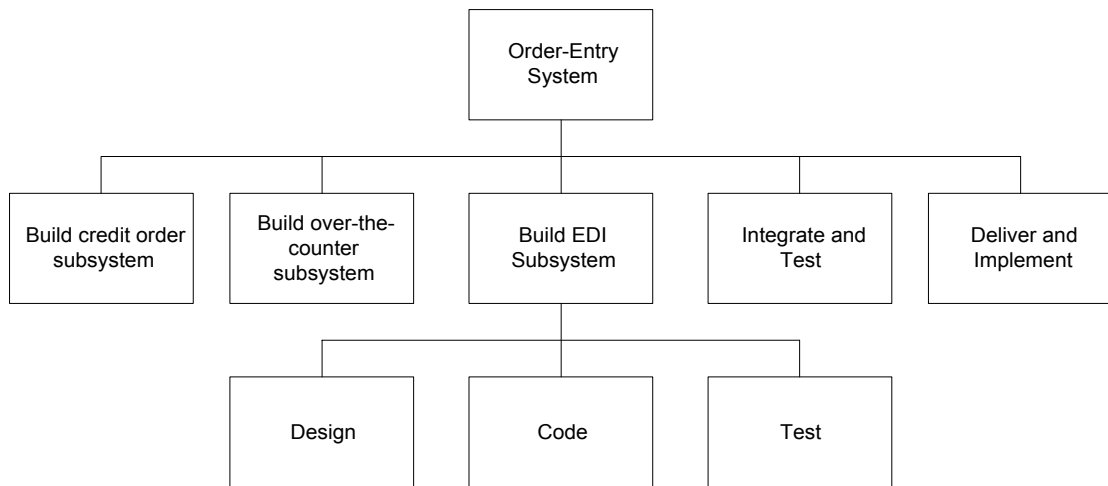
**Pengendalian (Control)**

Pada tahap kontrol ini, ada dua tujuan utama yaitu :

1. Untuk memonitor kemajuan dan beberapa tahap pada siklus hidup s/w agar tidak bertentangan dengan rencana awal.
2. Mengontrol tugas pengembangan, pengadaan dan implementasi s/w, agar s/w dapat di produksi secara autentik, akurat dan lengkap.

Untuk memonitor agar kontrol tidak bertentangan dengan rencana awal, beberapa teknik dapat digunakan seperti :

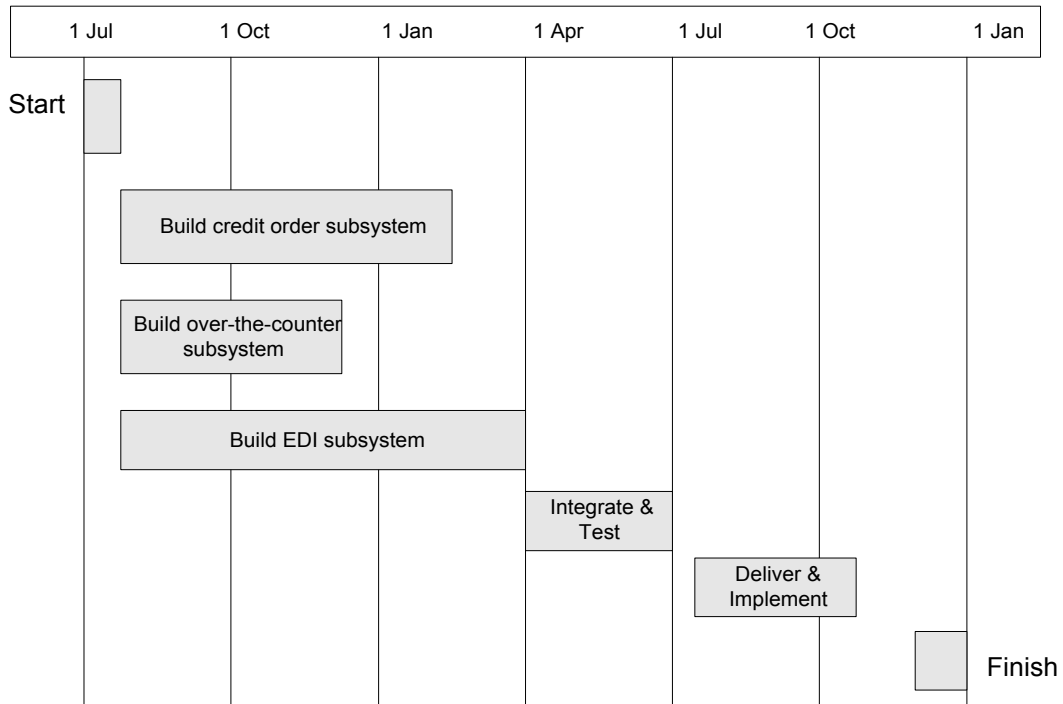
- a. Work Breakdown Structures (WBS), dengan teknik ini kita dapat mengidentifikasi tugas-tugas yang spesifik untuk pengembangan, pengadaan, dan implementasi s/w yang dibutuhkan. (Mc.Leod and Smith 1996). Lihat gbr. 4.2.



Gbr.4.2 Pemecahan struktur untuk order-entry system



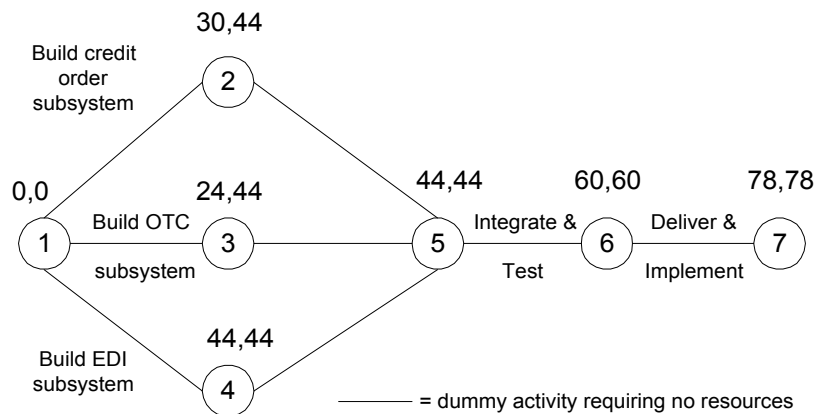
- b. Gantt Chart, dapat digunakan untuk membantu mengatur tugas (schedule) (lihat gbr.4.3). Teknik ini akan menunjukkan kapan tugas harus dimulai dan diselesaikan, tugas apa yang harus dibuat bersama-sama, dan tugas apa yang harus dihasilkan secara serial.



Gbr.4.3 Gantt Chart untuk order-entry system



- c. Program Evaluation and review technique (PERT), menunjukkan tugas-tugas yang harus diselesaikan, bagaimana hubungannya, kebutuhan sumber daya apa untuk setiap tugas-tugasnya. (lihat 4.4)



Gbr.4.4 PERT Chart untuk order-entry system

Seorang auditor harus mempunyai dua perhatian khusus pada kendali, pada tahap kontrol ini yaitu :

1. Auditor harus dapat mengevaluasi apakah fungsi dari aktivitas kontrol dapat diterapkan juga pada software yang berbeda.
2. Seorang auditor harus dapat mengumpulkan bukti apakah prosedur dari suatu kontrol sudah dijalankan dengan benar dan dapat dipercaya.

Perancangan (Design)

Dalam tahap desain, seorang programmer bertugas untuk menspesifikasikan struktur dan operasi dari program untuk menemukan artikulasi yang dibutuhkan selama tahap proses informasi sistem desain dari pengembangan sistem.

Selama tahap ini, perhatian utama seorang auditor adalah untuk menentukan apakah programmer menggunakan suatu tipe khusus dari pendekatan sistematis untuk desain. Auditor harus mengubah keinginannya berdasarkan beberapa faktor seperti ukuran dan bahan dari suatu program.

Seorang auditor juga dapat memperoleh bukti dari proses desain dengan melakukan interview, observasi, dan review dari dokumentasi. Mereka dapat berkomunikasi dengan programmer, apakah mereka dapat memahami tentang kebutuhan dengan menggunakan pendekatan yang sistematis untuk desain, jika ya, bagaimana menggunakannya.

Auditor juga dapat mengamati apakah programmer menggunakan pendekatan sistematis untuk mendesain program.

Mereka juga dapat meninjau dokumentasi program, apakah memiliki struktur chart sebagai bukti programmer menggunakan pendekatan yang sistematis untuk mendesain.



Pengkodean (Coding)

Tahap koding (pengetikan / penulisan program) dilakukan pada saat s/w akan dibuat atau dimodifikasi. Selama tahap ini, programmer akan menulis dan mendokumentasikan source code (program sumber) dalam bahasa pemrograman untuk mengimplementasikan desain program.

Strategi Implementasi modul dan integrasi

Tiga strategi utama dari implementasi modul dan integrasi adalah sbb :

1. Top-Down, strategi ini digunakan jika, modul level atas (high-level modules) dibuat (coding), di test, dan diintegrasikan sebelum modul level bawah (low-level modules). Keuntungannya adalah kesalahan pada modul level atas dapat teridentifikasi lebih dini, kerugiannya adalah pada saat uji coba program akan menemui kesulitan ketika modul level bawah menemukan kesalahan fungsi input-output yang sangat sulit.
2. Bottom up, strategi ini digunakan jika, modul level bawah di buat (coding), di test, dan diintegrasikan sebelum modul level atas di buat. Keuntungannya adalah modul level rendah yang merupakan operasi yang paling sulit di implementasikan dan diuji terlebih dahulu. Kerugiannya adalah pendekatan ini sangat sulit untuk di teliti seluruh operasinya, sebelum programnya selesai dibuat.
3. Threads (rangkain / untaian), strategi ini digunakan jika, keputusan dibuat terlebih dahulu untuk fungsi program yang akan dibuat, kemudian modul yang akan mendukungnya baru dibuat dan kemudian diimplementasikan untuk menghasilkan fungsi yang penting. Keuntungannya adalah fungsi yang paling penting di implementasikan terlebih dahulu. Kerugiannya adalah integrasi dari modul yang berikutnya mungkin akan lebih sulit, jika dibandingkan dengan pendekatan top-down atau bottom-up.

Auditor perlu mencari bukti yang benar dengan cara uji coba oleh manajemen program dalam memilih strategi implementasi modul dan integrasi. Khususnya pada program yang besar, penggunaan strategi yang salah (jelek) dapat mengakibatkan program yang dihasilkan menjadi kurang berkualitas. Auditor dapat melakukan wawancara untuk menguji apakah manajemen menggunakan pendekatan sistematis untuk memilih strategi implementasi modul dan integrasi. Mereka juga dapat menguji dokumentasi program untuk memperoleh bukti tipe strategi yang telah di adopsi (di pilih).

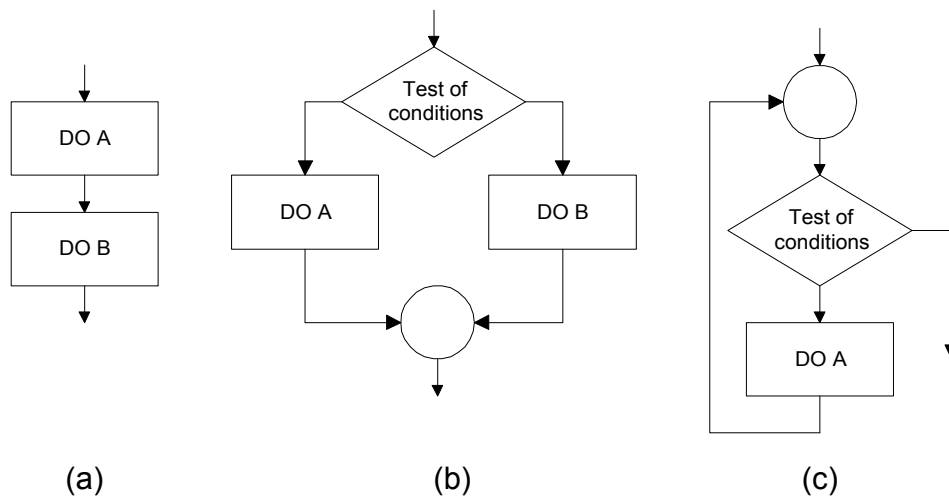
Strategi Coding

Menurut konvensi (kesepakatan) program terstruktur, terdapat tiga dasar struktur utama dalam struktur kontrol yaitu (lihat gbr.4.5) :

1. Urutan sederhana (simple sequence - SEQUENCE)
2. Pemilihan dengan seleksi (selection based on a test – IF-THEN-ELSE) dan
3. Pengulangan kondisi (conditional repetition-DO WHILE)

Jika konvensi pemrograman terstruktur di penuhi, dapat dipastikan bahwa para programmer akan membuat source-code yang tingkat kesalahannya kecil, mudah untuk dimengerti dan mudah untuk dirawat.

Auditor dapat mencari bukti untuk memastikan apakah manajemen programming di jamin di buat oleh programmer mengikuti struktur programming yang telah di sepakati. Mereka dapat melakukan wawancara dengan manager atau programmer tentang tugas dan cara yang dilakukannya dalam membuat program.



Gbr.4.5 (a) Simple Sequence (b) Selection (c) Repetition

Auditor juga dapat mengecek apakah programmer dalam membuat programnya menyediakan fasilitas otomatis sebagai alat bantu untuk mereka. Beberapa tipe penggunaan fasilitas koding otomatis antara lain :

- Shorthand preprocessor, memungkinkan programmer untuk menulis kode secara singkat, juga dapat menerjemahkan kode singkat ini dalam sintak yang lebih lengkap, contoh COBOL.
- Decision-table preprocessor, memindahkan bentuk teks program ke dalam bentuk source-code menggunakan bahasa pengolahan compiler.
- Copy facility, memungkinkan penggunaan kode secara berulang
- Editor, yang memungkinkan kode di ciptakan, di format, dan dimodifikasi secara mudah.
- User-interface management system, memungkinkan desain dari implementasi yang cepat, seperti windows, icons, menus, dan dialog boxes.
- CASE tools, berisi bermacam-macam fasilitas yang dapat membantu proses koding.

Strategi Dokumentasi

Pedoman untuk menghasilkan dokumentasi yang berkualitas adalah sbb :

1. Sediakan petunjuk yang menunjukkan proses pembuatan program ke dalam beberapa tahap dan komponen secara keseluruhan dan hubungan antara komponen-komponen tersebut.
2. Gunakan baris komentar dalam program secara bebas untuk menerangkan jalannya (logika) program.
3. Beri nama untuk variabel, konstanta tipe, paragraf, modul, dan seksi yang berarti kepada para pembaca source-code program.
4. Buat lay-out dari source-program sehingga mudah untuk dibaca.
5. Kelompokkan tipe kode yang saling berhubungan.



Tugas Kelompok :

Buat tulisan tentang testing (pengujian) program mencakup :

- a. Static analysis test (kel.1)
 1. Desk-checking
 2. Structured walk-throughs
 3. Design and code inspections
- b. Dynamic analysis test
 1. Black-box test (kel.2)
 2. White-box test (kel.3)
- c. Integration Testing (kel.4)
 1. Top-down test
 2. Bottom-up test
 3. Regression test
- d. Validation Test (kel.5)
- e. Basis path test (kel.6)
- f. Control structure test (kel.7)
- g. System test (kel.8)

Buku referensi yang dapat digunakan :

1. Ron Weber, "**Information Systems Control and Audit**", Prentice-Hall, USA., 1999.
2. Roger S.Pressman, Ph.D, "**Software Engineering: A. Practitioner's approach, fifth edition**", Mc-Graw Hill, USA,2001,



Pengetesan (Testing)

< materi dijadikan tugas kelompok >

Pengoperasian dan Pemeliharaan (Operation and Maintenance)

Dalam sudut pandang Sistem Audit, perhatian utama pada operasional program adalah bagaimana performance program tersebut dapat dimonitor setiap saat. Seseorang harus bertanggung jawab untuk mengidentifikasi apabila program perlu perawatan, kemungkinan lain adalah identifikasi dari kebutuhan perawatan mungkin tidak terjadi. Akibatnya, bisa terjadi kekeliruan pada database program, kegagalan dalam pencapaian keinginan user, atau operasi program tidak efisien.

Mekanisme formal dalam monitoring status operasional program sangat diperlukan, ketika pengguna dari program adalah seluruh anggota organisasi yang terdiri dari berbagai macam latar belakang.

Ada 3 macam tipe dari perawatan (maintenance) yang diperlukan agar program tetap beroperasi:

1. Repair-maintenance-errors, perawatan dengan cara memperbaiki kesalahan.
2. Adaptive maintenance-users needs, perawatan dengan mengadaptasi pada keinginan user.
3. Perfective maintenance, perawatan dengan maksud agar diperoleh program yang sempurna.

Perhatian utama seorang auditor pada fase operation & maintenance adalah untuk memastikan bahwa fase ini berjalan dengan efektif dan pelaporan secara berkala dapat dilakukan, serta proses perawatan bisa di kontrol dengan baik.

Auditor harus bisa mencari bukti bawa manajemen telah meninjau sistem dengan baik dan bertanggungjawab didalam monitoring status dari operasional program. Caranya dengan melakukan interview (wawancara), observasi, tinjauan pada dokumen yang menunjukkan bahwa sistem telah beroperasi dengan baik. Selanjutnya mereka harus fokus pada kualitas dari kontrol proses maintenance.

3. Pengorganisasian Tim Programming

Cara seorang programmer dalam menangani pekerjaan mereka sangat berpengaruh pada kualitas software yang mereka buat. Alternatifnya, para programmer bisa diorganisasikan sebagai satu kesatuan team. Mereka bekerja untuk periode waktu tertentu untuk menyelesaikan suatu proyek, dimana keputusannya dibicarakan diantara anggotanya. Hal ini sangat bermanfaat bila proyek yang ditangani sangat kompleks dan tidak jelas.

Proses pengembangan, penerapan, dan implementasi dari software, untuk saat ini banyak dilakukan secara team. Dari segi audit, perhatian/tujuan utamanya adalah bahwa manajemen telah memilih struktur team dengan hati-hati dilihat dari segi proyek, tingkat kompleksitasnya, dan tingkat keterlambatan dari jadwal proyek agar kemampuan dan kualitas mereka bisa diorganisasikan dalam bentuk team dimana mereka harus bekerja.

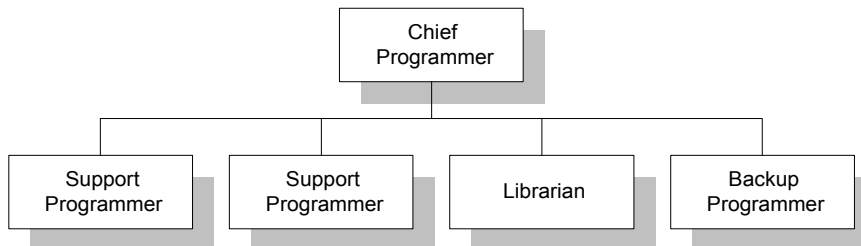
Untuk itu ada 3 struktur team yang digunakan untuk mengorganisasikan para programmer:

1. Chief Programmer Teams
2. Adaptive Teams dan
3. Controlled-Decentralized Teams



3.1. Ketua Tim Programmer (*Chief Programmer Team*)

Lihat gambar 4.6.



Gbr.4.6 Struktur Organisasi Chief Programmers Team

Fungsi dan Cirinya :

Chief Programmer :

- Bertanggung jawab secara total/penuh untuk sistem dimana team bekerja
- Harus seorang ahli
- Seorang programmer yang sangat produktif
- Bertanggungjawab dalam mendesain, coding, dan mengintegrasikan bagian yang kritis dalam sistem
- Memberikan perintah kerja pada bagian back-up dan support programmers.

Back-up Programmers :

- Seorang programmer senior yang bertanggungjawab dalam memberikan dukungan penuh pada chief programmer
- Harus bisa mengambil alih tugas chief programmer setiap saat

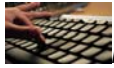
Support Programmers:

- Diperlukan pada saat proyek besar yang tidak bisa dikerjakan oleh chief programmer dan back-up programmer saja.
- Menyediakan dukungan
- Bekerja dalam pembuatan coding dan uji coba modul tingkat rendah (testing lower-level)

Librarian (penyedia data) :

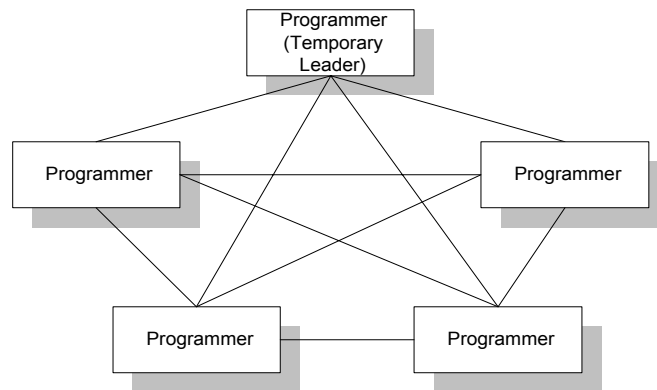
- Bertanggungjawab dalam perawatan program production library.
- Menyediakan input dan mengumpulkan keluaran untuk para programmer, file output dari hasil kompilasi dan ujicoba, mempertahankan agar source code dan object-code library tetap up to date.

Struktur “ The Chief Programmer team “ ini di desain untuk mengurangi kebutuhan proses informasi antara anggota team dan untuk meningkatkan kapasitas dari proses informasi.



3.2. Penyesuaian Tim (*Adaptives Teams*)

Lihat gambar 4.7.



Gbr.4.7 Struktur Organisasi Adaptive Team

Struktur ini diperuntukan untuk melayani 2 kebutuhan, yaitu:

1. Keinginan organisasi untuk meningkatkan kualitas program
2. Memenuhi kebutuhan sosial/ psikologi dari setiap anggota programmer dalam team.

Perbedaan dari struktur ini dengan struktur sebelumnya adalah:

- Adaptive team tidak punya tingkat otoritas, dimana kepemimpinan dalam team ada di tangan para anggota.
- Dalam Adaptive team, tugas diberikan pada anggota dari team daripada ditentukan lewat posisi.
- Adaptive team tidak mempunyai aturan formal librarian (penyedia data) dalam mengkoordinasikan fungsi team.

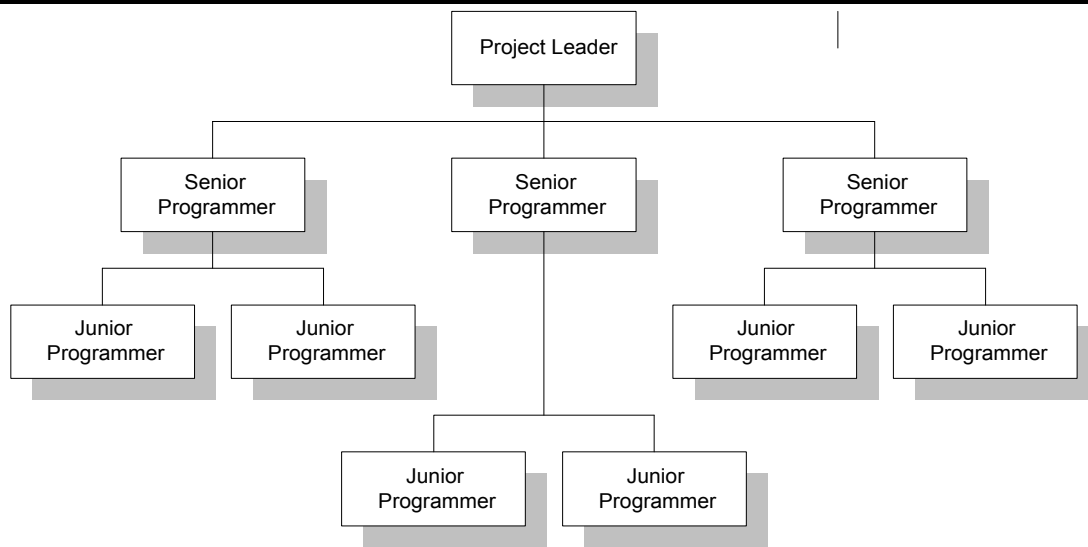
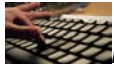
3.3. Desentraliasi Pengendalian Tim (*Controlled-Decentralized Teams*)

Struktur ini mempunyai junior programmer yang akan melaporkan hasil program pada senior programmer, kemudian oleh senior programmer dilaporkan juga pada ketua proyek.

Dengan struktur ini, manfaat/keuntungan dari struktur sebelumnya akan didapatkan. Keuntungannya : dapat memecahkan masalah yang kompleks, dimana struktur dari grup ini akan memfasilitasi pemecahan masalah.

Kerugian : struktur ini tidak bisa bekerja dengan baik apabila tugas dari programmer tersebut tidak bisa di bagi-bagi, dan dengan waktu deadline yang sangat ketat.

Lihat gambar 4.8.



Gbr.4.8 Struktur Organisasi Controlled-Decentralized

4. Pengelolaan Kelompok Sistem Programming

Para programmer sering diklasifikasikan menurut aplikasi programmer atau sistem programmer. Dahulu, programmer membangun dan merawat program untuk sistem aplikasinya. Tetapi kini, membangun dan merawat sistem software. Seperti sistem operasi, sistem manajemen database, dan komunikasi software.

- Mengontrol Masalah

Mengontrol sistem programmer adalah tugas yang berat, mereka biasanya memiliki keahlian yang tinggi dan sering bekerja sendiri atau ada di dalam grup yang kecil. Dengan menerapkan kontrol secara tradisional pada aktivitas mereka seperti pemisahan tugas, sangatlah sulit. Mereka biasanya bekerja pada situasi yang kritis.

- Mengukur Sistem Kontrol

Meskipun sulit untuk mengontrol sistem programmer, beberapa hal ini dapat diimplementasikan untuk mengontrolnya:

1. Pekerjakan staf sistem programming yang mempunyai kualitas yang tinggi.
2. Pisahkan tugas seluas mungkin, contohnya tanggung jawab untuk desain dan coding sistem program dipisah dari tanggung jawab untuk uji coba program.
3. Buat metode dokumen standar
4. Batasi wewenang sistem programmer, jadi seorang programmer hanya bekerja sesuai dengan aplikasi yang dikuasainya.
5. Jauhkan prosedur petunjuk manual dan kunci mesin dari aktivitas sistem programmer. Hal ini dimaksudkan agar aktivitas yang tidak diinginkan / sesuai dengan tugasnya tidak terjadi.
6. Pekerjakan konsultan dari luar untuk mengevaluasi pekerjaan programming.
7. Perintahkan programmer aplikasi untuk mengevaluasi pekerjaan sistem programmer secara berkala agar dapat dihasilkan program yang berkualitas.



BAB V

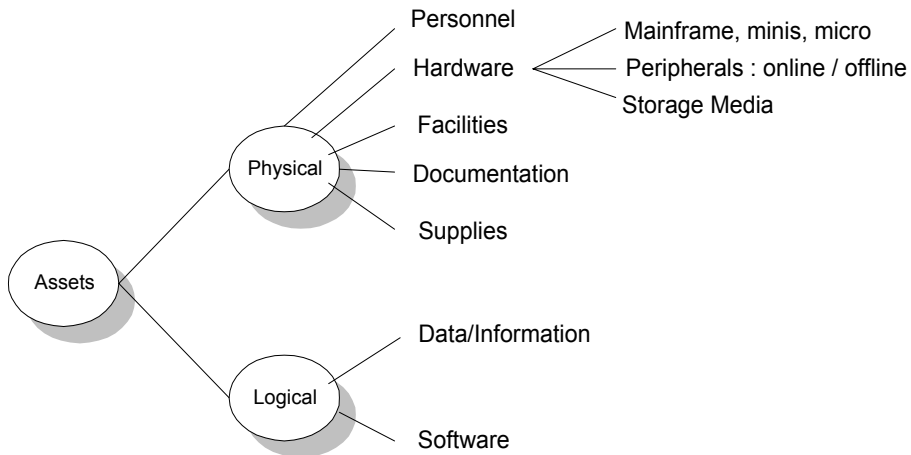
Manajemen Kontrol Keamanan *(Security Management Controls)*

1. Pendahuluan
2. Pelaksanaan Program Keamanan (*Conducting a Security Program*) :
 - 2.1. Persiapan Rencana Pekerjaan (*Preparation of a Project Plan*)
 - 2.2. Identifikasi Kekayaan (*Identification of asset*)
 - 2.3. Penilaian Kekayaan (*Valuation of asset*)
 - 2.4. Identifikasi Ancaman-ancaman (*Threats Identification*)
 - 2.5. Penilaian Kemungkinan Ancaman (*Threats Likelihood Assessment*)
 - 2.6. Analisis Ekspose (*Exposures analysis*)
 - 2.7. Penyesuaian Kendali (*Controls Adjustment*)
 - 2.8. Laporan Persiapan (*Report Preparation*)
3. Ancaman Utama Keamanan dan Ukuran Perbaikannya (*Major Security Threats and Remedial Measures*)
 - 3.1. Kebakaran Api (*Fire Damage*)
 - 3.2. Kerusakan oleh Air (*Water Damage*)
 - 3.3. Variasi Energi (*Energy Variations*)
 - 3.4. Kerusakan Struktural (*Structural Damage*)
 - 3.5. Polusi (*Pollution*)
 - 3.6. Gangguan akibat otorisasi yang tidak syah (*Unauthorized Intrusion*)
 - 3.7. Virus (*Virus and Worms*)
 - 3.8. Penyalahgunaan S/W, Data dan Pelayanan (*Misuse of Software, Data, and Services*)
 - 3.9. Hacking

▪ **Pendahuluan**

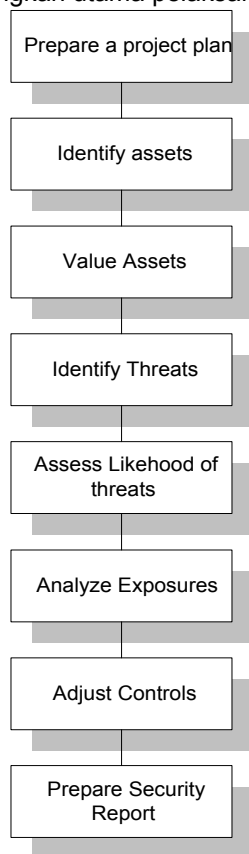
Aset Sistem Informasi yang harus di lindungi melalui sistem keamanan dapat diklasifikasikan menjadi 2 yaitu (lihat gbr. 5.1) :

1. Aset Fisik, meliputi :
 - a. Personnel
 - b. Hardware (termasuk media penyimpanan, dan periperalnya)
 - c. Fasilitas
 - d. Dokumentasi dan
 - e. Supplies
2. Aset Logika
 - a. Data / Informasi dan
 - b. Software (Sistem dan Aplikasi)



Gbr.5.1 Kategori Aset Sistem Informasi

2. Pelaksanaan Program Keamanan (*Conducting a Security Program*)
Langkah-langkah utama pelaksanaan Program keamanan yaitu (lihat gbr.5.2)



Gbr.5.2 Langkah utama dalam pelaksanaan program keamanan



Persiapan Rencana Pekerjaan (*Preparation of a Project Plan*)

Perencanaan proyek untuk tinjauan kemanan mengikuti item sbb :

- Tujuan Review
- Ruang Lingkup (Scope) Review
- Tugas yang harus dipenuhi
- Organisasi dari Tim Proyek
- Sumber Anggaran (Pendanaan) dan
- Jadwal untuk Menyelesaikan Tugas

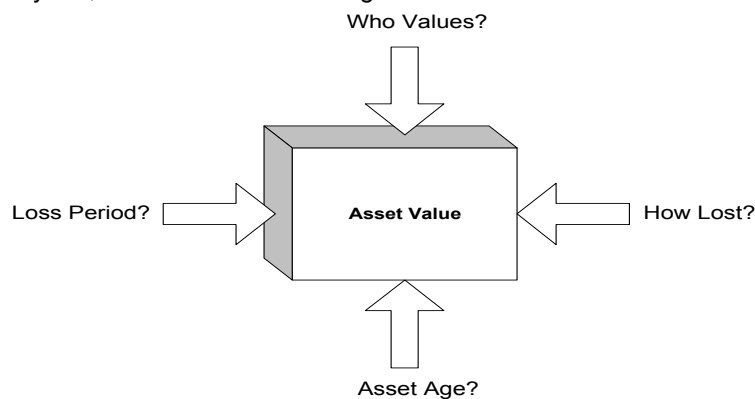
Identifikasi Kekayaan (*Identification of asset*)

Katagori asset :

- Personnel (end users, analyst, programmers, operators, clerks, Guards)
- Hardware (Mainframe, minicomputer, microcomputer, disk, printer, communication lines, concentrator, terminal)
- Fasilitas (Furniture, office space, computer room, tape storage rack)
- Dokumentasi (System and program doc., database doc., standards plans, insurance policies, contracts)
- Persediaan (Negotiable instrument, preprinted forms, paper, tapes, cassettes)
- Data/Informasi (Master files, transaction files, archival files)
- Software Aplikasi (Debtors, creditors, payroll, bill-of-materials, sales, inventory)
- Sistem Software (Compilers, utilities, DBMS, OS, Communication Software, Spreadsheets)

Penilaian Kekayaan (*Valuation of asset*)

Langkah ke tiga adalah penilaian kekayaan, yang merupakan langkah paling sulit. Parker (1981) menggambarkan ketergantungan penilaian pada siapa yang ditanya untuk memberikan penilaian, cara penilaian atas kekayaan yang hilang (lost), waktu periode untuk perhitungan atas hilangnya kekayaan, dan umur asset. Lihat gbr. 5.3



Gbr.5.3 Faktor efek penilaian SI keamanan

**Identifikasi Ancaman-ancaman (*Threats Identification*)**

Lihat gbr. 5.4 Lapisan jenis ancaman aset SI

		Nature of threat	
		Accidental (kebetulan/tdk sengaja)	Deliberate (sengaja)
Source of Threat	External	e.g. Acts of God	e.g. Hackers
	Internal	e.g. Pollution	e.g. Sabotage

Gbr.5.4 Lapisan jenis ancaman Aset SI

Sumber ancaman External :

1. Nature / Acts of God
2. H/W Suppliers
3. S/W Suppliers
4. Contractors
5. Other Resource Suppliers
6. Competitors (sabotage, espionage, lawsuits, financial distress through fair or unfair competition)
7. Debt and Equity Holders
8. Unions (strikes, sabotage,harassment)
9. Governmnets
10. Environmentalist (Harassment (gangguan), unfavorable publicity)
11. Criminals/hackers (theft, sabotage, espionage, extortion)

Sumber ancaman Internal :

1. Management, contoh kesalahan dalam penyediaan sumber daya, perencanaan dan control yang tidak cukup.
2. Employee, contoh Errors, Theft (pencurian), Fraud (penipuan), sabotase, extortion (pemerasan), improper use of service (penggunaan layanan yg tidak sah)
3. Unreliable system, contoh Kesalahan H/W, kesalahan S/W, kesalahan fasilitas.

Penilaian Kemungkinan Ancaman (*Threats Likelihood Assessment*)

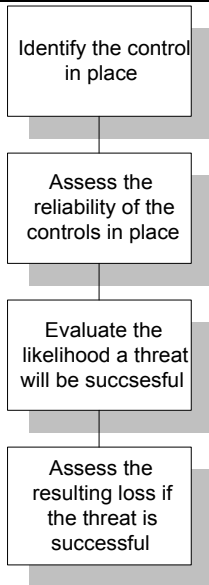
Contoh, perusahaan asuransi dapat menyediakan informasi tentang kemungkinan terjadinya kebakaran api dalam satu waktu periode tertentu.

Analisis Ekspose (*Exposures analysis*)

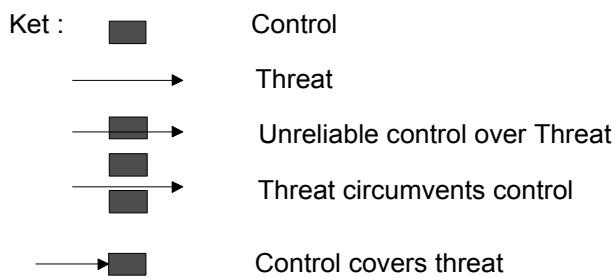
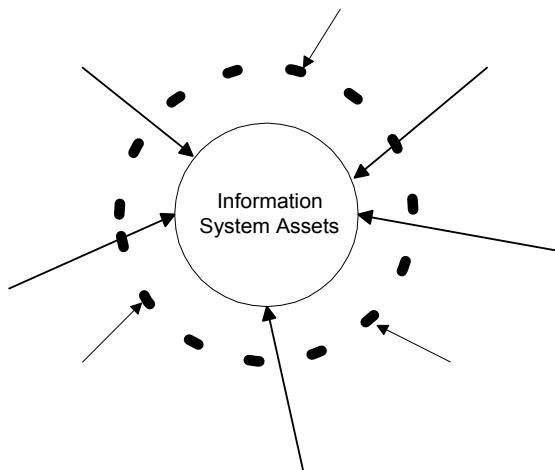
Tahap analisis ekspose terdiri dari 4 tugas yaitu :

1. Identification of the controls in place
2. Assessment of the reliability of the controls in place
3. Evaluation of the likelihood that a threat incident will be successful
4. Assess the resulting loss if the threat is successful

Lihat gbr. 5.5. Tugas utama tahap analisis ekspose dan gbr.5.6. Ancaman, keandalan kontrol, cakupan control, dan ekspose



Gbr. 5.5. Tugas utama tahap analisis ekspose



Gbr.5.6. Ancaman, keandalan kontrol, cakupan control, dan ekspose



BAB VI

Kendali / Kontrol Pembatasan

(Boundary Controls)

1. Pendahuluan
2. Cryptographic Controls
 - 2.1. Nature of Cryptography
 - 2.2.
3.
 - 3.1.
 - 3.2.

Kisi – kisi ujian akhir :

1. Bab II : Conducting an Information System Audit
2. Bab III : Systems Development Management Controls
3. Bab IV : Programming Management Controls
4. Bab V : Security Management Controls



by:sws

Versi-1

diktat – AKSI

Vocabulary Verb :

Carried = mengangkat, membawa
Obviously = dg terang / nyata
Sobering = yang menenangkan
Completion = penyelesaian
Assessment = penaksiran
Around = keliling, sekitar
Complexity = keruwetan, kerumitan
dealing = hubungan, transaksi, urusan
Conduct = perilaku
Conducting = memimpin
theft = pencurian
destruction = perusakan
violations = pelanggaran
harm = kerugian
safeguard = usaha penjagaan, usaha perlindungan
privilege = hak istimewa
determine = menetapkan, menentukan, memutuskan
whether = apakah
allows = mengizinkan/membolehkan
achieved = sangat sukses karena kemampuannya, terbukti keahliannya, menyelesaikan dengan baik
victim = korban
suffered = menderita = damage
intention = maksud
perpetrator = pelaku, pelaku kejahatan
gain = keuntungan
disrupt = mengacaukan, mengganggu
assignment = menyetujui
purpose = maksud
pertaining = menyanggung, mengenai
cease = berhenti
harm = kerugian / kejahatan
impact = pengaruh yg kuat
supplies = persediaan barang
perhaps = barangkali
undertake = melakukan
amount = jumlah
acquisition = didapatnya
attempt = usaha
volatility = berubah pendirian