
Perancangan Sistem Informasi (Output, Input, Proses, Basis Data, Kontrol, LAN)



SIKLUS HIDUP SISTEM

- **PROJECT DEFINITION:** Apakah ada masalah? Dapatkah permasalahan tersebut diselesaikan dengan project?
- **SYSTEM STUDY:** Menganalisis masalah dalam sistem yang ada saat ini; definisikan tujuan dan evaluasilah berbagai alternatif yang ada
- **DESIGN:** spesifikasi logik dan fisik untuk solusi sistem
- **PROGRAMMING:** mengembangkan software code



PENDEKATAN MEMBANGUN SISTEM

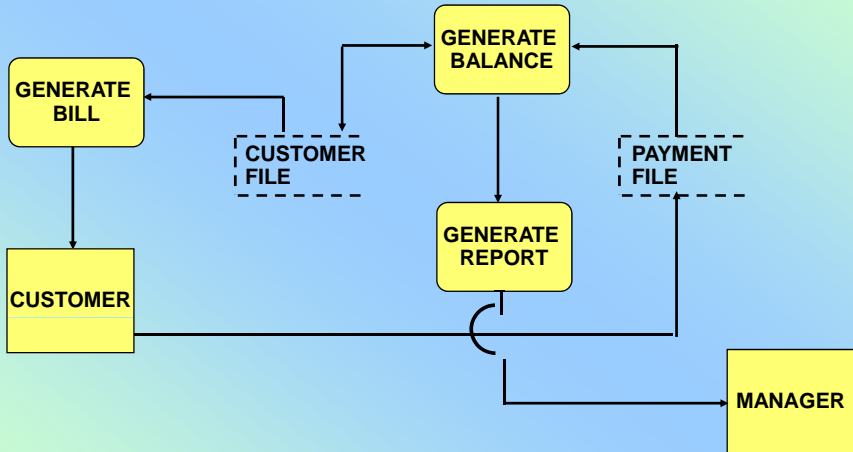


STRUCTURED ANALYSIS

- **DEFINES SYSTEM INPUTS, PROCESSES, OUTPUTS**
- **PARTITIONS SYSTEM INTO SUBSYSTEMS OR MODULES**
- **LOGICAL, GRAPHICAL MODEL OF INFORMATION FLOW**
- **DATA FLOW DIAGRAM: Graphical display of component processes, flow of data**

*

DATA FLOW DIAGRAM:





STRUCTURED DESIGN

**DESIGN RULES / TECHNIQUES TO
DESIGN SYSTEM, TOP DOWN IN
HIERARCHICAL FASHION**

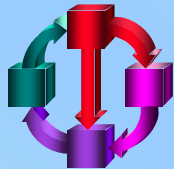
- **STRUCTURE CHART**
- **STRUCTURED PROGRAMMING**
- **MODULE**
- **SEQUENCE CONSTRUCT**
- **SELECTION CONSTRUCT**

*

PEMROGRAMAN TERSTRUKTUR:

- **DISCIPLINE TO ORGANIZE, CODE PROGRAMS**
- **SIMPLIFIES CONTROL PATHS**
- **EASY TO UNDERSTAND, MODIFY**
- **MODULE HAS ONE INPUT, ONE OUTPUT**

*



PROGRAM FLOWCHART SYMBOLS:



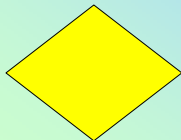
**BEGIN OR
END**



DIRECTION



PROCESS



DECISION

**INPUT OR
OUTPUT**



SUBROUTINE



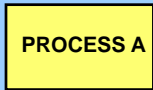
**MANUAL
OPERATION**



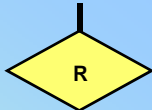
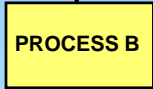
CONNECTOR



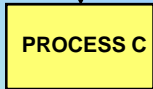
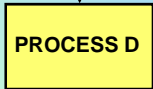
PROGRAM FLOWCHART:



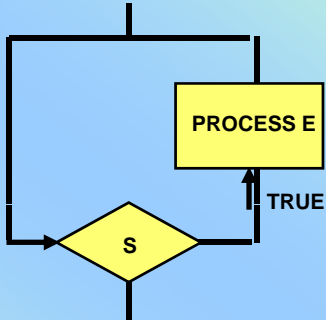
SEQUENCE



TRUE

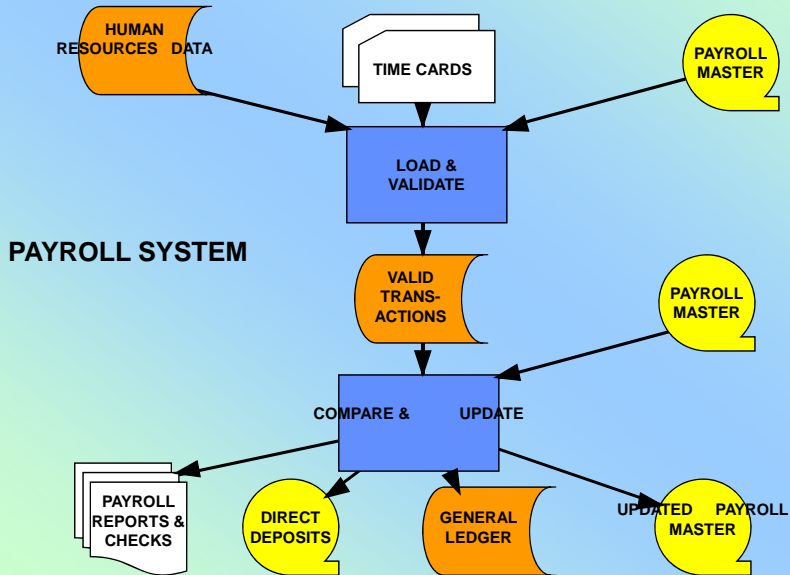


SELECTION



ITERATION

SYSTEM FLOWCHART:



Karakteristik Sistem

- **Komponen Sistem (*Components*)**
 - Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.
 - Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.
 - Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem.
- Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan/berfungsi sebagaimana mestinya. Tentunya sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

Bagian-Bagian Sistem (1)

■ Penghubung (*Interface*) Sistem

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

■ Masukan (*Input*) Sistem

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

Bagian-bagian Sistem (2)

■ Keluaran (*Output*) Sistem

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

■ Pengolah (*Process*) Sistem

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

Sasaran Sistem

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

Desain Sistem

- Tujuan:
 - Membuat spesifikasi rinci dari solusi berbasis komputer
- Solusi berbasis komputer dapat berupa
 - *turn-key system*: beli sistem jadi dan tinggal pakai
 - *customized system*: beli sistem jadi dengan penyesuaian di sana sini
 - *in-house development*: membuat sistem sendiri
- Output desain sistem hasil in-house development:
 - Desain proses
 - Desain basis data
 - Desain input
 - Desain output
 - Desain antarmuka dengan pemakai
 - Desain modul/unit
 - Desain jaringan komputer

Desain Output

- Jenis-jenis output:
 - Output internal: output bagi pemilik dan pemakai sistem dalam organisasi
 - Laporan rinci
 - Laporan rangkuman
 - Laporan pengecualian
 - Output eksternal: output bagi pihak-pihak di luar organisasi
- Metode implementasi output: printer, layar, terminal point-of sale (seperti di kasir), multimedia, e-mail, hyperlink, microfilm

Syarat-syarat Output

- Output harus mudah dibaca dan diinterpretasikan
 - ada judul
 - ada tanggal & waktu output dihasilkan
 - dalam output berbentuk form: seluruh item harus ada labelnya
 - tabel: tiap kolom harus ada labelnya
 - singkatan-singkatan ada keterangannya (legenda)
 - format seimbang (tidak terlalu padat atau kosong)
 - pemakai dapat menemukan dengan mudah
 - pemakai tidak harus mengedit manual agar output dapat bermanfaat bagi mereka
 - istilah teknis komputer sebaiknya dihindari dalam output maupun dalam pesan-pesan kesalahan
- Output harus sampai pada pemakai tepat waktu
- Distribusi atau akses ke output harus memadai bagi pemakai
- Output harus dapat diterima (*acceptable*) oleh pemakai, artinya sesuai dengan apa yang mereka butuhkan dan sesuai dengan rencana mereka untuk memanfaatkan output tersebut

Langkah-langkah Desain Output

- Identifikasi output sistem dan review kebutuhan pemakai
- Buatlah spesifikasi kebutuhan fisik output
- Desain, validasi, dan tes output menggunakan beberapa kombinasi alat bantu layout, prototyping, dan generator program

Desain Input

Tipe proses

- Input data bisa dilakukan pada waktu yang berbeda. Hal ini ditentukan oleh jenis pemrosesan yang dibedakan menjadi:
 1. *Batch processing*:
 - data tidak diproses langsung ketika datang, tetapi dikumpulkan dulu dalam satu *batch* (kumpulan), baru diproses
 2. *Online processing*:
 - Data diinputkan (biasanya melalui *workstation*) dan langsung diproses
 3. *Remote batch*:
 - Kombinasi batch dan online: data dimasukkan secara *online* atau *offline* kemudian dikumpulkan dalam batch sebelum akhirnya diproses.
-

Syarat-syarat Desain Input (1)

- Metode input: keyboard, mouse, touch screen, point of sale, sound and speech, penanda optis, tinta magnetis, transmisi elektromagnetik, smart card, biometric
- Beberapa syarat-syarat desain input:
 1. Yang diinputkan hanya data-data variabel (bukan konstanta)
 2. Tidak perlu menginputkan data yang dapat dihitung atau disimpan dalam program
 3. Gunakan kode untuk atribut-atribut yang sesuai
- Jika suatu dokumen dirancang untuk mengumpulkan data, gunakan hal-hal berikut:
 1. cantumkan instruksi pengisian form (dokumen)
 2. minimalkan jumlah tulisan tangan
 3. urutkan data yang harus diisi seperti urutan membaca buku (kiri→kanan, atas→ bawah)
 4. Jika mungkin, gunakan rancangan berdasar pada metafor (misal desain layar input penarikan rekening berdasar desain form standar penarikan rekening)

Syarat-syarat Desain Input (2)

- Kontrol internal dalam desain input:
 1. Jumlah input harus dimonitor, jangan sampai ada yang terlewat
 2. Pastikan bahwa data yang diinputkan valid :
 - eksistensi (harus ada/tidak)
 - tipe data
 - domain
 - nilai-nilai kombinasi
 - *self-checking digits*
 - format

Langkah-langkah Desain Input

- Langkah-langkah desain input:
 - Identifikasi input sistem dan review kebutuhan pemakai
 - Pilih kontrol GUI yang sesuai:
 - text box
 - radio button
 - check box
 - list box
 - drop-down list
 - combination box (combo box)
 - buttons
 - Desain, validasi, dan tes input menggunakan beberapa kombinasi alat bantu *layout* dan *prototyping*
 - Jika perlu, buat pula desain dokumen sumber (formulir yang dipakai untuk menyimpan data transaksi)

Desain Antarmuka (*Interface*) (1)

- Tipe pemakai komputer:
 - ahli
 - pemula
- Umumnya desain interface saat ini berasumsi pemakai adalah pemula yang sedang dalam proses menjadi ahli.
- Desain antarmuka perlu memperhatikan:
 - faktor pemakai
 - faktor human engineering
 - dialog dan istilah

Desain Antarmuka (*Interface*) (2)

- Beberapa masalah yang sering muncul dalam desain antarmuka:
 1. Penggunaan jargon atau singkatan-singkatan komputer yang berlebihan
 2. Desain yang tidak jelas
 3. Ketidakmampuan membedakan tindakan-tindakan alternatif yang harus dipilih pemakai
 4. Pendekatan *problem-solving* yang tidak konsisten
 5. Desain yang tidak konsisten
- Beberapa hal untuk memecahkan masalah di atas:
 1. Pahami *user* dan tugas mereka
 2. Libatkan *user* dalam desain antarmuka
 3. Uji sistem dengan melibatkan *user*
 4. Lakukan proses desain secara iteratif

Desain Antarmuka (*Interface*) (3)

- Syarat-syarat Desain Antarmuka

1. Layar harus diformat agar berbagai info, perintah, pesan muncul di area layar yang konsisten
2. Pesan, perintah, informasi jangan terlalu panjang
3. Jangan terlalu sering memakai atribut *display* yang mengganggu seperti *blinking*, *highlight*, dsb
4. Nilai *default* dibuat jelas
5. Antisipasi *error* yang mungkin dibuat *user*
6. Jika ada *error*, *user* mestinya tidak boleh melanjutkan tanpa memperbaiki *error* tsb
7. Jika *user* melakukan sesuatu yang membahayakan sistem, *keyboard* harus terkunci dan pesan utk meminta bantuan teknisi harus dimunculkan

Desain Antarmuka (*Interface*) (4)

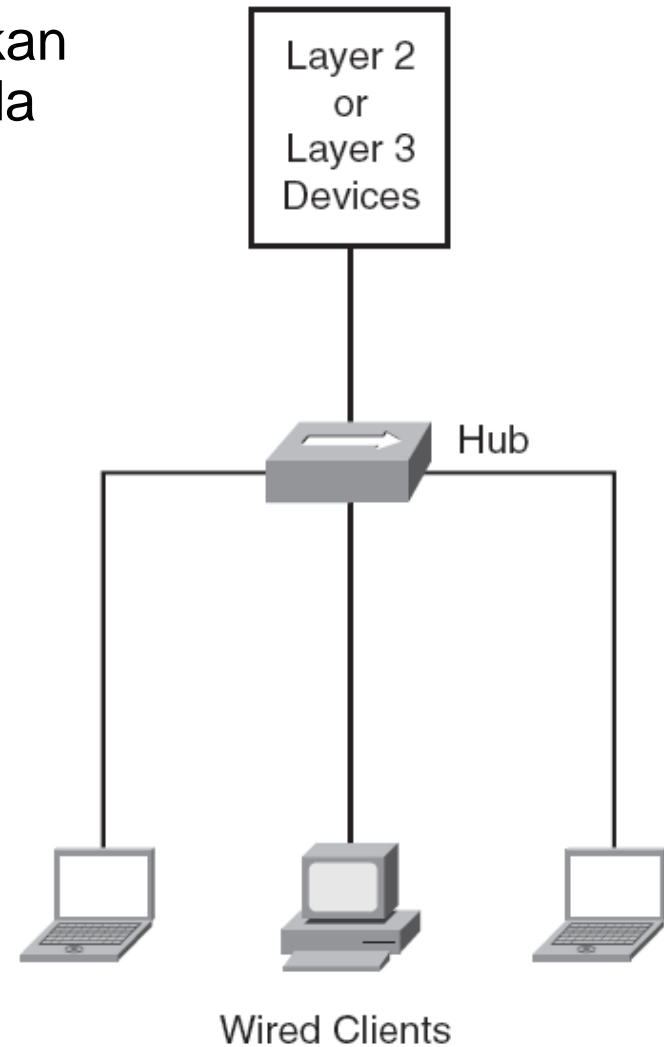
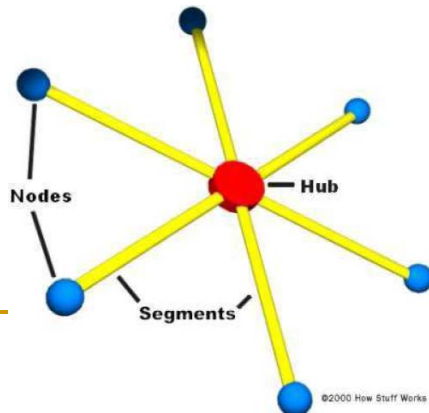
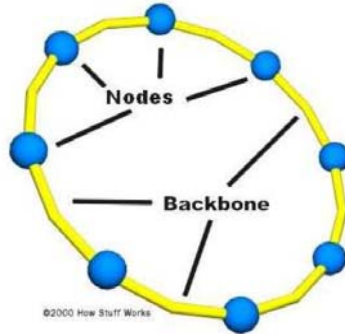
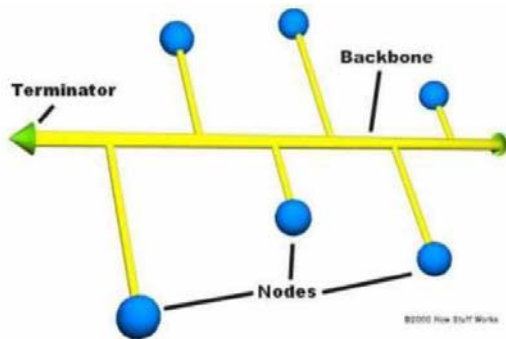
- Syarat-syarat dialog:
 1. Gunakan kalimat sederhana dan benar
 2. Jangan mencoba melucu
 3. Jangan menghina
- Syarat-syarat istilah:
 1. Jangan gunakan jargon komputer
 2. Hindari singkatan jika mungkin
 3. Konsisten dengan pemilihan istilah
 4. Pilihlah istilah yang tepat untuk perintah
- Proses perancangan antarmuka:
 1. Buatlah bagan dialog antarmuka (misal memakai diagram status/*state* diagram)
 2. Buatlah *prototype* dialog dan antarmuka
 3. Carilah umpan balik dari *user*
 4. Jika perlu, kembali ke langkah 1 atau 2

Desain Basis Data

- Merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut
- Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (database management system, DBMS)
- Memodelkan dengan ERD (*Entity Relationship Diagram*)
- Harus memenuhi aturan semantik (sesuai dengan kehidupan nyata)

Desain LAN

- LAN (*Local Area Network*) merupakan jaringan komputer yang berada pada lokasi fisik yang sama
- Kebutuhan akan konektivitas
- Arsitektur LAN: Bus, Ring, Star



WLAN (1)

- WLAN adalah LAN yang menggunakan frekuensi radio (*Radio Frequency (RF)*) untuk berkomunikasi tanpa menggunakan kabel
- Pertimbangan Menggunakan WLAN
 - Kebutuhan akan fleksibilitas
 - Kebutuhan penambahan produktivitas
 - Penghematan pengeluaran untuk kabel
- Kelemahan
 - Masalah keamanan
 - Masalah kecepatan yang tidak stabil

WLAN (2)

