



Computer Science, University of Brawijaya

Putra Pandu Adikara, S.Kom

Design Approaches

Interaksi Manusia dan Komputer



- ❖ Design Rules
 - Standards
 - Guidelines
 - Principles
- ❖ Prototipe
- ❖ Tingkat Prototipe
- ❖ Teknik Prototipe
- ❖ Pemodelan



Tujuan

- ❖ Memahami siklus hidup pengembangan perangkat lunak
- ❖ Memahami aturan desain berupa standar dan/atau garis pedoman
- ❖ Membuat dokumentasi spesifikasi dalam aktifitas siklus hidup perangkat lunak
- ❖ Membuat prototipe yang tepat dalam desain perangkat lunak
- ❖ Membuat desain perangkat lunak berdasarkan dokumen spesifikasi, prototipe dan aturan desain



Design Rules

Standard, Guidelines, Principles



Standard

- ❖ ditetapkan oleh badan-badan nasional atau internasional untuk memastikan kepatuhan yg dibuat komunitas besar dari standar desainer yang memerlukan teori yang mendasari perubahan teknologi yg berubah perlahan
- ❖ standar hardware dan perangkat lunak lebih umum daripada otoritas tinggi dan detail tingkat rendah
- ❖ ISO 9241 mendefinisikan daya guna sebagai efektivitas, efisiensi dan kepuasan saat pengguna menyelesaikan tugas



Guidelines

- ❖ lebih sugestif dan umum
- ❖ banyak buku teks yg melaporkan ttg pedoman
- ❖ pedoman abstrak (prinsip) berlaku selama siklus kehidupan awal kegiatan
- ❖ rincian pedoman (gaya panduan) berlaku selama aktivitas siklus hidup kemudian
- ❖ pemahaman bantuan pedoman dalam menyelesaikan konflik



Norman's Guidelines

❖ Beberapa prinsip (singkat) utk desain yg baik dr Norman adalah:

- Visibility
- Feedback
- Constraints
- Good mappings
 - Recall, elevator
- Consistency



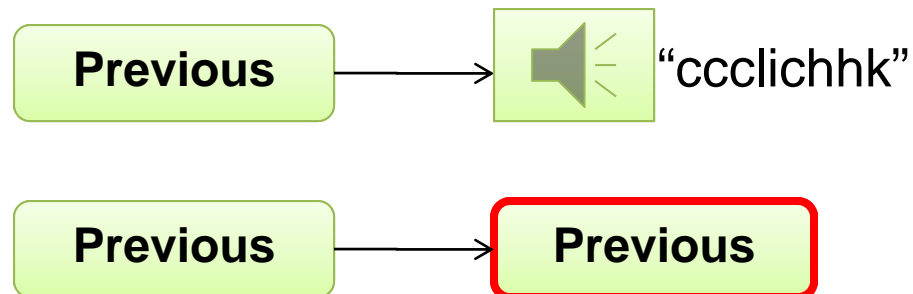
Visibility

- ❖ Buat bagian-bagian yang relevan terlihat dgn baik oleh pengguna
- ❖ Buat apa yang harus dilakukan oleh pengguna dgn jelas
- ❖ Misal:
 - Menyediakan sebuah pesan suara, yang mengatakan apa yang harus dilakukan (dgn bahasa apa?)
 - Menyediakan label besar



Feedback

- ❖ Mengirim informasi kembali kepada pengguna tentang apa yang telah dilakukan
- ❖ Termasuk suara, highlighting, animasi dan/atau kombinasinya
- ❖ Misal ketika mengklik tombol, keluar feedback berupa suara atau highlight merah.





Constraints

- ❖ Membatasi tindakan-tindakan yang mungkin dapat dilakukan oleh pengguna
- ❖ Membantu mencegah pengguna dari salah memilih pilihan

- ❖ Tiga jenis utama (Norman, 1999)
- ❖ Fisik
 - Lihat cara objek fisik membatasi pergerakan hal-hal
misal: hanya satu cara Anda dapat memasukkan kunci ke kunci
 - Berapa banyak cara yang dapat Anda memasukkan CD atau DVD disk ke komputer?
 - Bagaimana membatasi tindakan ini secara fisik?
 - Bagaimana hal itu berbeda dengan cara memasukkan disket ke dalam komputer?
- ❖ Logis
 - Memanfaatkan penalaran akal sehat sehari-hari mengenai cara dunia bekerja
Contohnya adalah mereka hubungan logis antara layout fisik perangkat dan cara kerjanya (slide berikutnya)
- ❖ Budaya
 - Belajar, tanda-tanda tertentu



Logical or ambiguous design?



- ❖ Where do you plug the mouse?
- ❖ Where do you plug the keyboard?
- ❖ top or bottom connector?
- ❖ Do the color coded icons help?



How to design more logically



A provides direct adjacent mapping between icon and connector



B provides color coding to associate the connectors with the labels



Cultural constraints

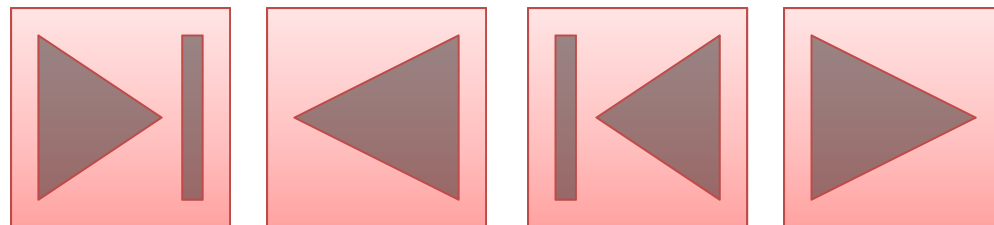
- ❖ Pelajari konvensi yg bisa berubah
Misal: segitiga merah untuk peringatan
- ❖ Dapat universal atau spesifik thdp budaya tertentu





Mapping

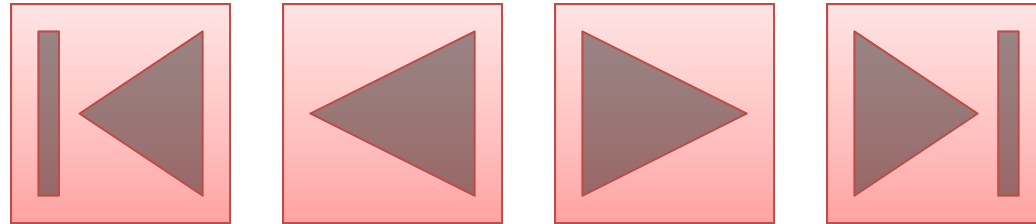
- ❖ Hubungan antara kontrol dan gerakan mereka serta hasil dari aksi yg dilakukan
- ❖ Mengapa contoh berikut ini merupakan pemetaan yg buruk dari tombol kontrol?





Mapping

- ❖ Mengapa contoh berikut ini pemetaan yang lebih baik?



- ❖ Tombol kontrol dipetakan lebih baik sesuai urutan aksi *fast rewind*, *rewind*, *play*, dan *fast forward*



Constistency

- ❖ Desain antarmuka yg memiliki operasi serupa dan menggunakan elemen serupa untuk tugas serupa

Contoh:

- Selalu gunakan tombol Ctrl ditambah awal pertama dari perintah operasi: Ctrl + C, Ctrl + S, Ctrl + O
- ❖ Manfaat utamanya adalah konsistensi antarmuka serta lebih mudah untuk dipelajari dan dipergunakan
- ❖ Ketika konsistensi rusak
 - Apa yang terjadi jika ada lebih dari satu perintah yang diawali dengan huruf yang sama?
 - Misal: Save, Spelling, Select, Style
 - Harus menemukan inisial atau kombinasi tombol lain , sehingga melanggar aturan konsistensi
 - Misal: Ctrl + S, Ctrl + Sp, Ctrl + Shift + L
 - Meningkatkan beban belajar pengguna, membuat mereka lebih rentan terhadap kesalahan



Internal and External Consistency

❖ Internal Consistency

- Merancang operasi yg mempunyai aksi sama dalam suatu aplikasi
- Sulit dicapai pada antarmuka yang kompleks

❖ Eksternal konsistensi

- Merancang operasi, antarmuka, dan sebagainya, harus sama di seluruh aplikasi dan perangkat
- Sangat jarang terjadi, berdasarkan preferensi desainer yang berbeda



External Inconsistency

❖ layout keypad angka

(a) phones, remote controls

1	2	3
4	5	6
7	8	9
	10	

(b) calculators, computer keypads

7	8	9
4	5	6
1	2	3
0		



Jakob Nielsen's 10 Heuristics

- ❖ Match the real
- ❖ Consistency
- ❖ Help and Documentation
- ❖ User Control and Freedom
- ❖ Visibility of System Status
- ❖ Flexibility and Efficiency
- ❖ Error Prevention
- ❖ Recognition, Not Recall
- ❖ Error Reporting, Diagnosis, Recovery
- ❖ Aesthetic and Minimalist Design



Shneiderman's 8 Golden Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*



Tog's 16 Principles

- ❖ Beberapa prinsip (yg baru):
- ❖ Anticipation
- ❖ Defaults
- ❖ Explorable interfaces
- ❖ Learnability
- ❖ Protect users' work
- ❖ Readability
- ❖ Track state
- ❖ Visible Navigation



Prototipe



Prototipe

- ❖ Prototipe merupakan simulasi atau animasi dari beberapa fitur dari bakal sistem.
- ❖ Tujuannya adalah untuk mengevaluasi tingkat kegunaan dari seluruh sistem dimana terdapat versi pertama.
- ❖ Prototipe membantu mengidentifikasi masalah, menganalisis alasan kenapa terjadi dan mengimplementasikan solusi pada prototipe berikutnya



Jenis Prototipe

- ❖ Pelbagai jenis prototipe antara lain:
 - Throw-away (sekali pakai, buang)
 - Incrementasi (bertingkat)
 - Evolutionary (evolusi)
- ❖ Dalam pengembangan, prototipe harus dipilih sesuai dengan masalah manajemen, antara lain:
 - Waktu
 - Perencanaan
 - Fitur non-fungsional
 - Kontrak



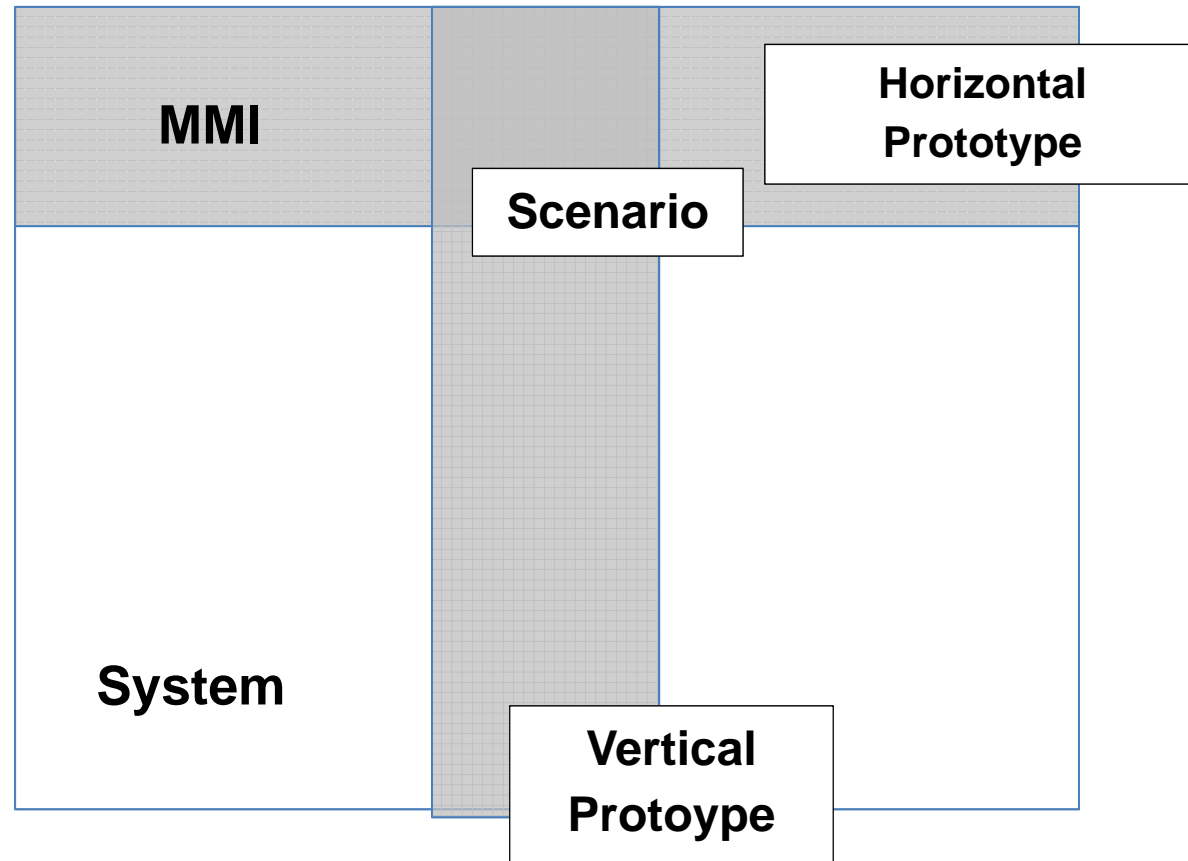
Tingkatan Prototipe

❖ Prototipe Horizontal

- hanya merupakan antarmuka manusia-mesin atau *man-machine interface (MMI)*.
- Prototipe ini bisa berupa sketsa pada kertas.
- Prototipe ini merupakan ‘antarmuka permukaan’, fungsionalitas tidak bekerja, namun memungkinkan untuk mencapai **uji persepsi (*perception test*)**.

❖ Prototipe Vertical

- dilakukan setelah prototipe horizontal
- mengimplementasikan sekumpulan fungsionalitas dari aplikasi untuk memungkinkan interaksi lebih pengguna untuk mencapai skenario khusus dalam penggunaan.
- Tahap kedua dari fase prototipe terdiri dari serangkaian **uji pengguna (*user test*)**





Teknik Prototipe

Metode non-komputer

❖ Sketsa/mock-up

- Gambaran antarmuka berbasiskan kertas
- Bagus untuk brainstorming
- Berfokus pada gagasan desain high-level
- Tidak bagus untuk untuk mengilustrasikan alur dan detail
- Cepat dan murah

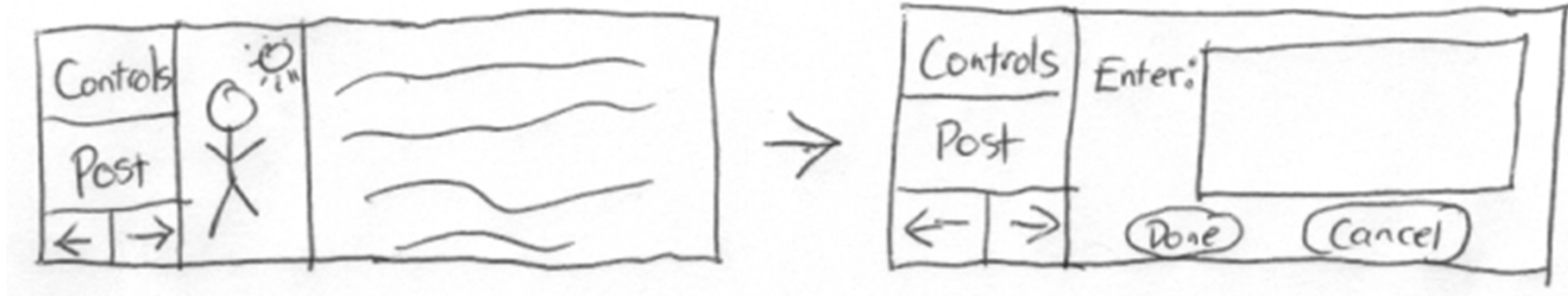


Teknik Prototipe

Metode non-komputer

❖ Papan cerita (*Storyboard*)

- Seperti sketsa/mock-up, mensimulasikan jalannya tampilan atau fungsionalitas sistem.
- Menggunakan diagram/gambar sekuensial
- Tidak perlu harus berbasis komputer.
- Dapat dianimasikan.
- Cepat dan mudah





Teknik Prototipe Metode non-komputer

❖ Skenario

- Penggunaan hipotesis atau situasi fisik.
- Biasanya melibatkan beberapa orang, peristiwa, situasi, dan lingkungan.
- Memberikan konteks operasi.
- Sering dalam bentuk narasi, tetapi juga bisa dalam bentuk sketsa atau bahkan video.
- Memungkinkan desainer untuk melihat masalah dari sudut pandang orang lain
- Memfasilitasi umpan balik dan opini
- Bisa sangat futuristik, kreatif, dan menarik



Teknik Prototipe Metode Komputer

❖ Program gambar/paint

- Menggambarkan tiap layar, bagus untuk tampilan
- Alat bantu seperti Adobe Photoshop, Corel Draw, bisa juga Word processor seperti Microsoft Word





Teknik Prototipe Metode Komputer

❖ Simulasi script/slide show fungsionalitas terbatas

- Biasa disebut *chauffeured prototyping*
- Beberapa bagian dari fungsionalitas sistem disediakan oleh desainer
- Alat bantu seperti HyperCard, PowerPoint, Macromedia Director, Flash, Dreamweaver, HTML
- Contoh Apple Hypercard sebagai alat bantu prototyping untuk mensimulasikan UI dengan kelakuan kompleks

```
on mouseUp
  play "boing"
  wait for 3 seconds
  visual effect wipe left very fast to black
  click at 150,100
  type "goodbye"
end mouseUp
```



Macromedia Director

The screenshot displays the Macromedia Director 8 interface with the following components:

- Movie Script 4: All Functions:** Contains the following code:

```
end repeat
sprite(63).loch = endh
DataTabState = 0
else
--move into scene

currenth = 918
endh = 648

repeat while sprite(63).loch-3 > endh
if myvartime < the ticks then
sprite(63).loch=(1.0/5)*(endh-current
currenth = sprite(63).loch
myvartime = the ticks + 5
updatestage
end if
```
- InfoArt Stage (100%):** Shows a yellow duck sprite on a stage. A tooltip for the duck reads: "duck [Internal] Bitmap" and "Sprite 62: (144,70,183,116) Copy, 100%".
- Internal Cast:** Lists various elements including PicTab, DataTab, ToolsTab, duck, and various actions like "on exitFrame me go to the Stop" and "on mousedown me MoveData".
- InfoArt Score:** A timeline showing member numbers 60 through 68. Members 62, 63, 64, and 65 are highlighted, corresponding to the duck, PicTab, DataTab, and ToolsTab respectively.

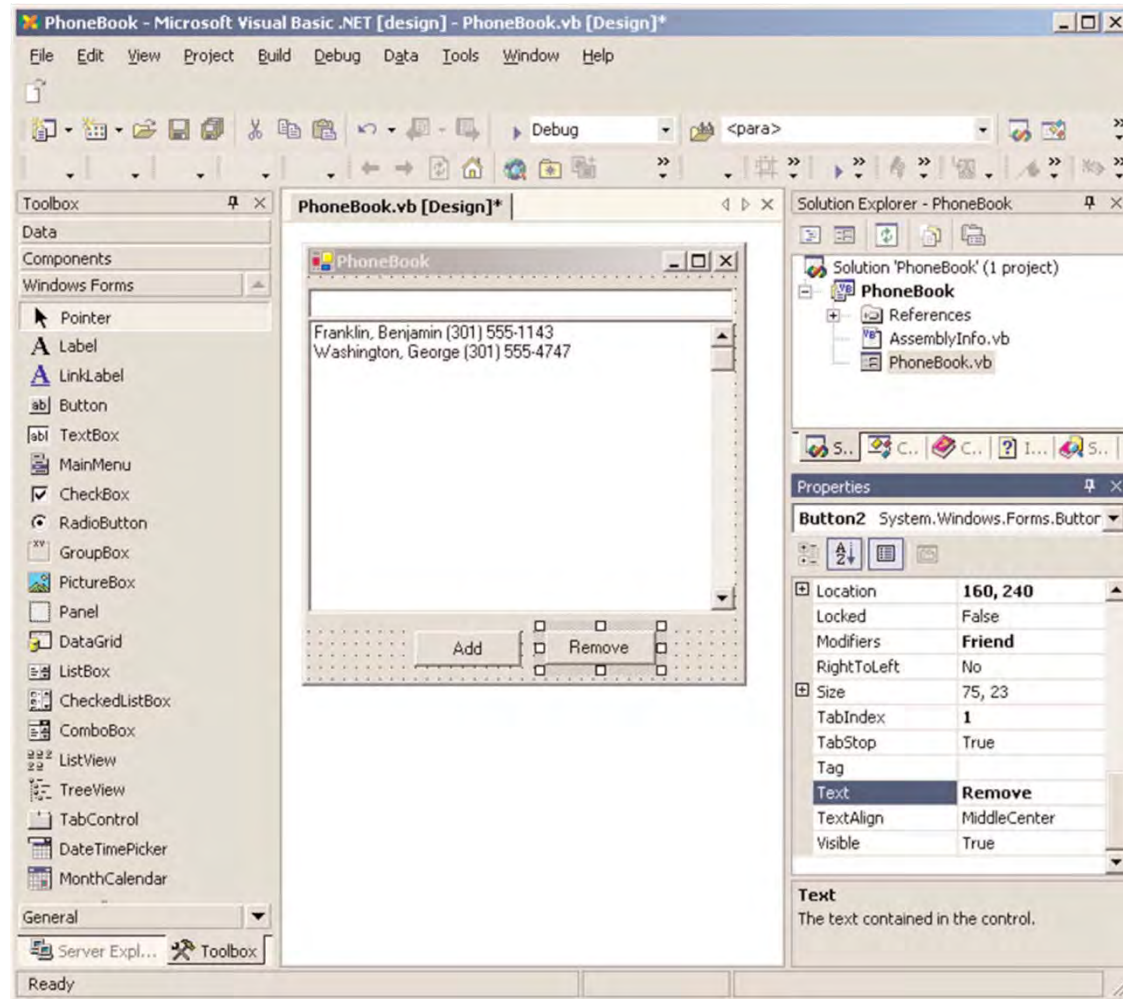


Teknik Prototipe Metode Komputer

- ❖ **Interface builder/Rapid prototyping tool,**
 - Prototyping terkomputerisasi dengan visual editing contohnya:
 - Alat bantu: Microsoft Visual Studio, JBuilder, UIMX, Delphi



Visual Studio





JBuilder

The screenshot shows the JBuilder 9 IDE with the following components:

- Title Bar:** JBuilder 9 - C:/Documents and Settings/fekete.PPC-FEKETE/jbproject/PhoneBook/src/PhoneBook.java
- Menu Bar:** File, Edit, Search, View, Project, Run, Team, Wizards, Tools, Window, Help
- Toolbar:** Standard IDE icons for file operations and execution.
- Project Explorer:** Shows the project structure for 'PhoneBook', including 'PhoneBook.jpx' and 'PhoneBook.java'.
- Component Palette:** Displays various Swing components like 'OK', 'ON', 'label', 'text', and 'T'.
- Design View:** A visual representation of the 'PhoneBook' application. It features a list box containing two entries: 'Franklin, Benjamin (301) 555-4321' and 'Washington George (301) 555-1234'. Below the list are two buttons labeled 'Add' and 'Remove'.
- Properties Window:** A table of properties for the selected 'panel' component.
- Component Hierarchy:** A tree view showing the UI structure: 'panel (BorderLayout)' contains 'borderLayout1', which contains 'entry', 'jScrollPane1', and 'sub'. 'sub' contains a '<FlowLayout>' with 'add' and 'remove' components.
- Bottom Bar:** Includes tabs for 'Source', 'Design', 'Bean', 'Doc', and 'History'.

name	panel
actionMap	
alignmentX	0.5
alignmentY	0.5
background	Control
border	
debugGraphic...	<default>
doubleBuffered	True
enabled	False
font	"MS Sans Serif"
foreground	Black
inputVerifier	
layout	BorderLayout
maximumSize	2147483647, 21
minimumSize	143, 80
nextFocusabl...	
opaque	True
preferredSize	196, 196
requestFocus...	True
toolTipText	
verifyInputWh...	True



Pemodelan

- ❖ Perangkat pemodelan merupakan salah satu ciri pendekatan terstruktur,
- ❖ Digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan mengkomunikasikan secara konseptual dan fungsionalitas.



Peran Perangkat Pemodelan

❖ Komunikasi

- sebagai alat komunikasi antara pengguna dan analis sistem dalam pengembangan sistem

❖ Eksperimentasi

- pengembangan sistem bersifat trial and error.

❖ Prediksi

- meramalkan bagaimana sistem akan bekerja.



Jenis Perangkat Pemodelan

- ❖ Jenis perangkat pemodelan antara lain:
 - **Diagram Arus Data (DFD):** menunjukkan proses yang dijalankan data dalam sistem
 - **Kamus Data:** definisi elemen data dalam sistem
 - **Entity Relationship Diagram (ERD):** model penyimpanan data dalam DFD
 - **State Transition Diagram (STD):** menunjukkan keadaan tertentu dimana suatu sistem dapat ada dan transisi yang menghasilkan keadaan tertentu yang baru. STD digunakan untuk sistem yang real time.
 - **Bagan Struktur:** menggambarkan suatu hierarki modul program perangkat lunak termasuk dokumentasi interface antar modul.



Jenis Perangkat Pemodelan

- **Diagram Alur Program Terstruktur (Structured Program Flowchart):** menggambarkan alur dan logika program.
- **Alat Spesifikasi Proses:** memberikan deskripsi yang lengkap tentang proses-proses yang ditemukan dalam diagram alur data tingkat dasar. Contoh:
 - Bahasa Inggris terstruktur
 - Tabel keputusan
 - Pohon keputusan
 - Persamaan
- **Diagram Warnier-Orr (WOD):** menunjukkan penguraian hierarki proses atau data.
- **Diagram Jackson:** membuat model struktur program perangkat lunak dari struktur data.