Rekayasa Perangkat Lunak
Perancangan Perangkat Lunak

Teknik Informatika
UNIKOM

Perancangan P/L

1. Kenapa butuh tahap perancangan?
2. Definisi perancangan perangkat lunak
3. Fase-Fase perancangan perangkat lunak
4. Penjelasan langkah-langkah perancangan
5. Aturan perancangan perangkat lunak
6. Rules of thumbs perancangan antarmuka
7. DPPL (Deskripsi Perancangan P/L)

Kenapa Butuh Tahap Perancangan?

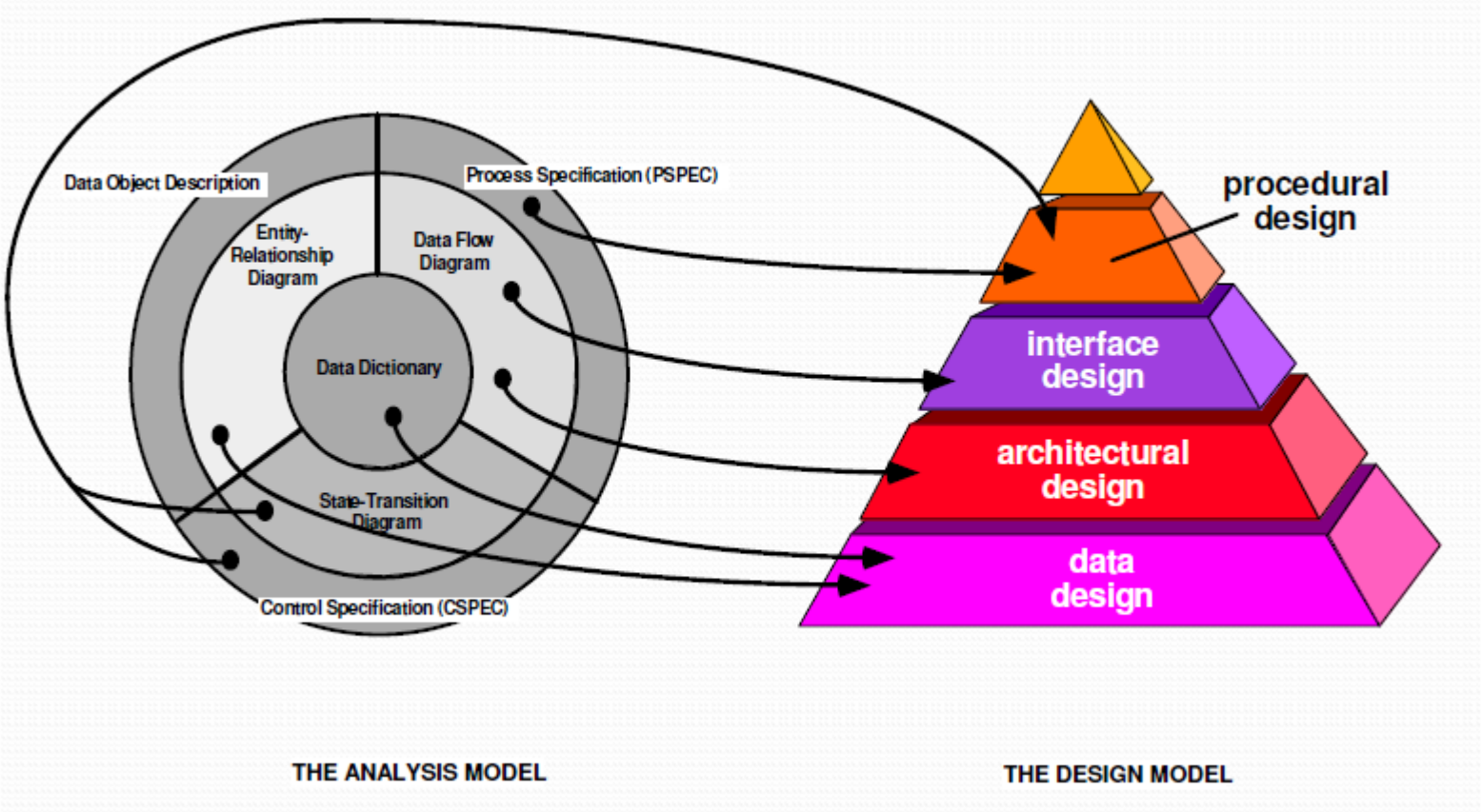
"...And that, in simple terms, is what's wrong with your software design."

Kenapa Butuh Tahap Perancangan?

Definisi Perancangan Perangkat Lunak

1. Menurunkan solusi yang berasal dari kebutuhan perangkat lunak.
2. Sekumpulan prinsip-prinsip, konsep-konsep, dan praktek.
3. Sebuah proses yang menghasilkan sebuah model atau representasi yang menampilkan ketegasan, komoditas, dan kemudahan untuk dipahami. [Presmann, 7th edition]

Prinsip Analisis Menuju Desain



Perancangan yang Baik

1. Mengimplementasikan keseluruhan kebutuhan yang eksplisit.
2. Mudah dibaca dan dimengerti oleh coder/tester.
3. Memberikan gambaran lengkap tentang perangkat lunak yang dibangun: data, fungsional, perilaku.

BUT HOW...

Prinsip-Prinsip Perancangan

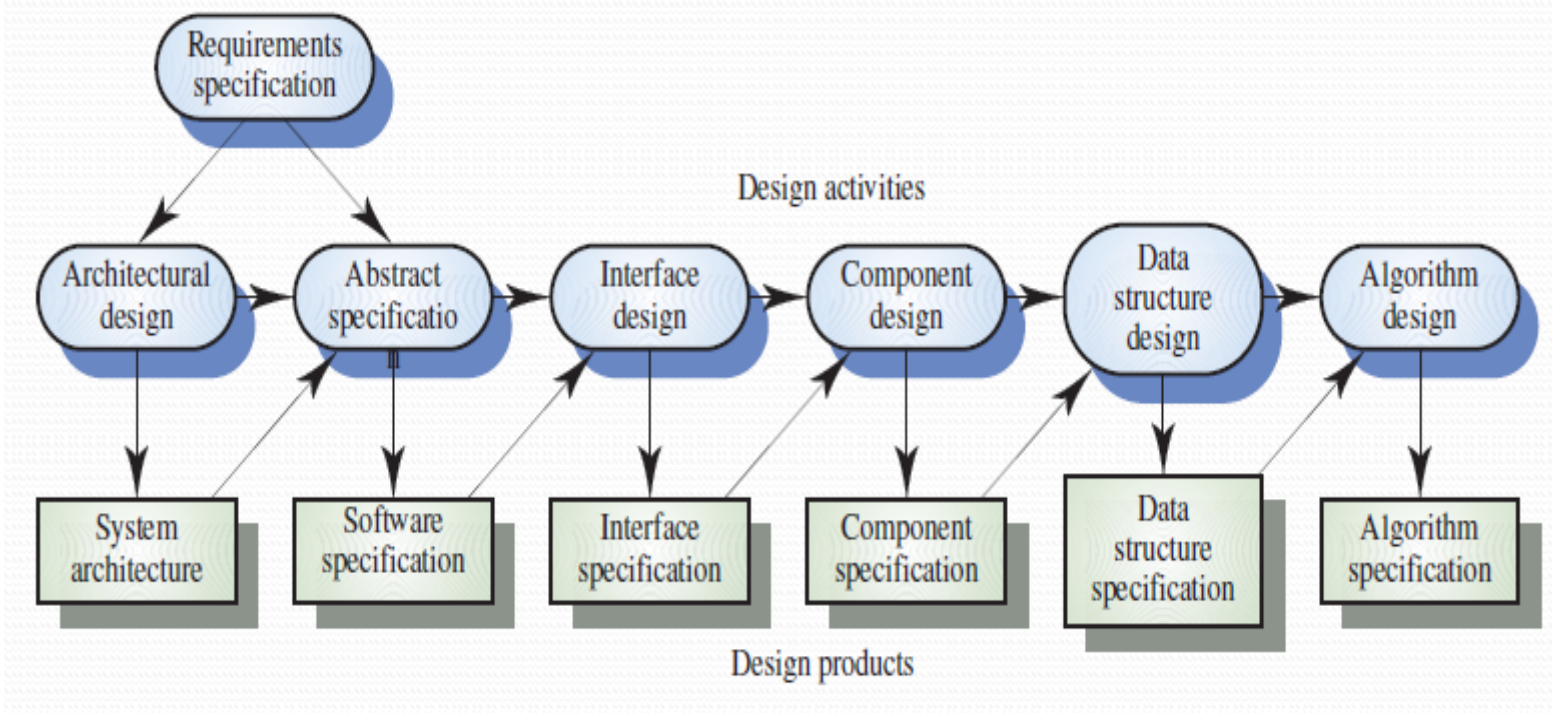
1. Proses perancangan tidak boleh menggunakan konsep “tunnel vision”.
2. Perancangan yang dibuat harus bisa ditelusuri pada model analisis.
3. Hasil perancangan harus original.
4. Bisa mengurangi jarak antara proses perangkat lunak dengan proses dunia nyata.

Prinsip-Prinsip Perancangan

5. Desain harus seragam dan terintegrasi.
6. Perancangan bukan coding dan coding bukan perancangan.
7. Desain harus terstruktur dalam menghadapi perubahan.
8. Desain yang dibuat harus bisa dinilai dan direview untuk melihat kesalahan semantik.

Prinsip-Prinsip Perancangan

Fase-Fase Perancangan

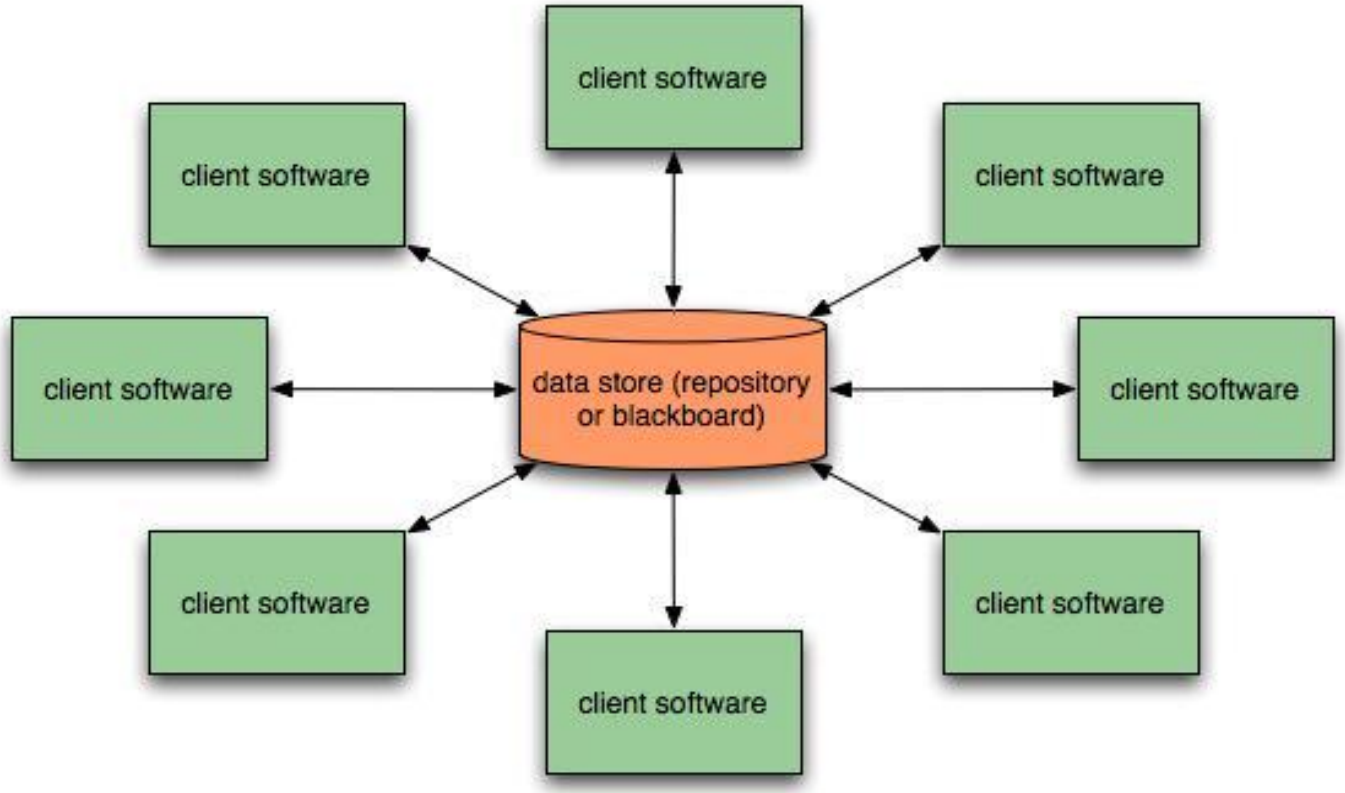


PERANCANGAN ARSITEKTURAL

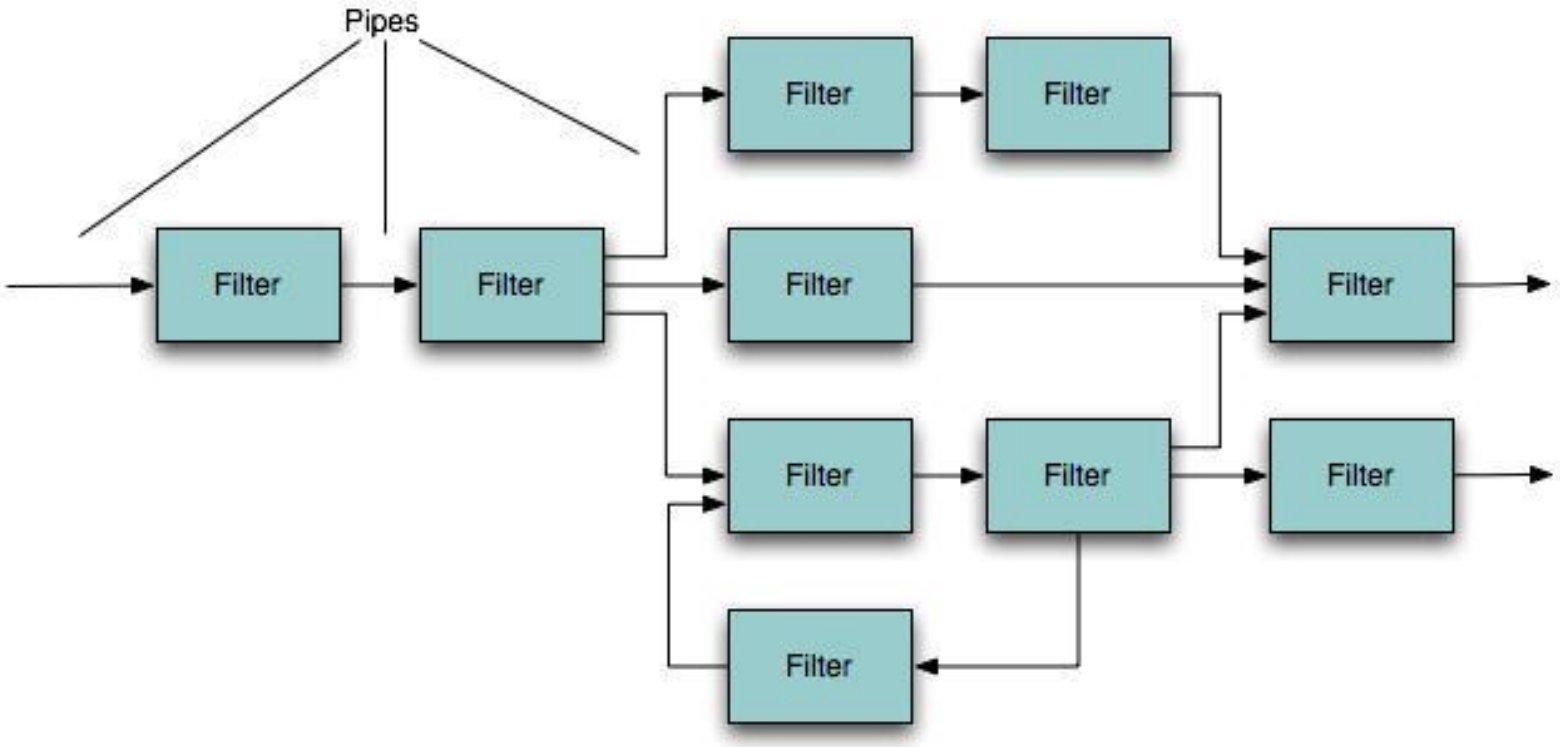
Perancangan Arsitektural

1. Arsitektur Data-centered
2. Arsitektur Data Flow
3. Arsitektur Call and Return
4. Arsitektur Object-Oriented
5. Arsitektur Layered

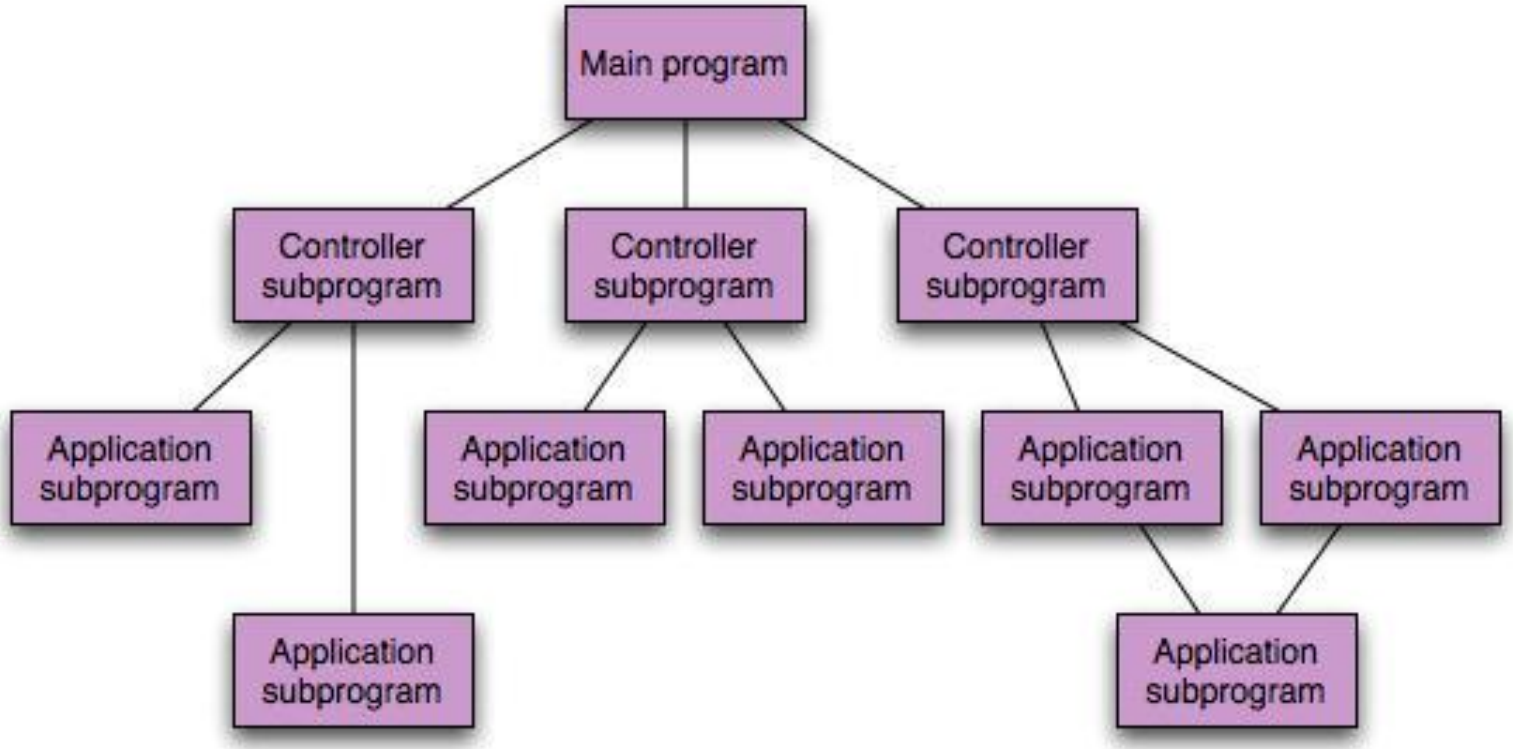
Arsitektur Data-Centered



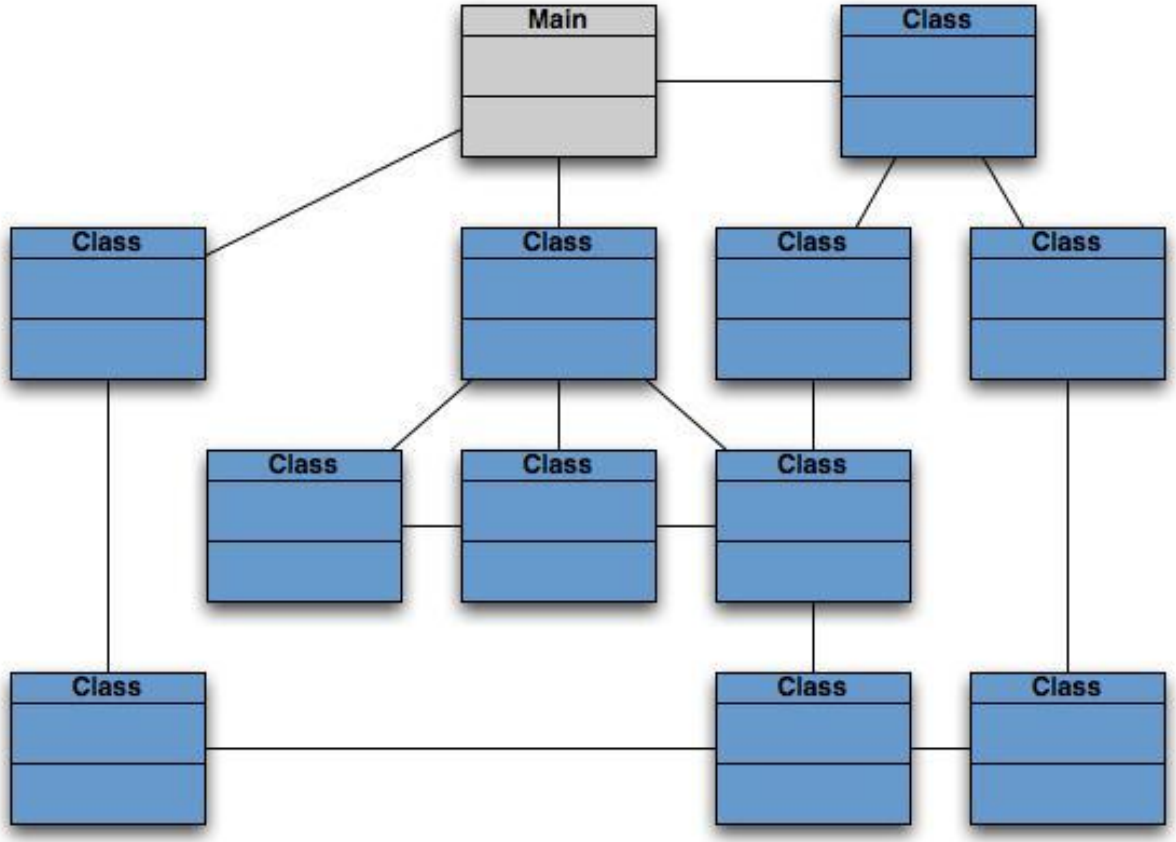
Arsitektur Data Flow



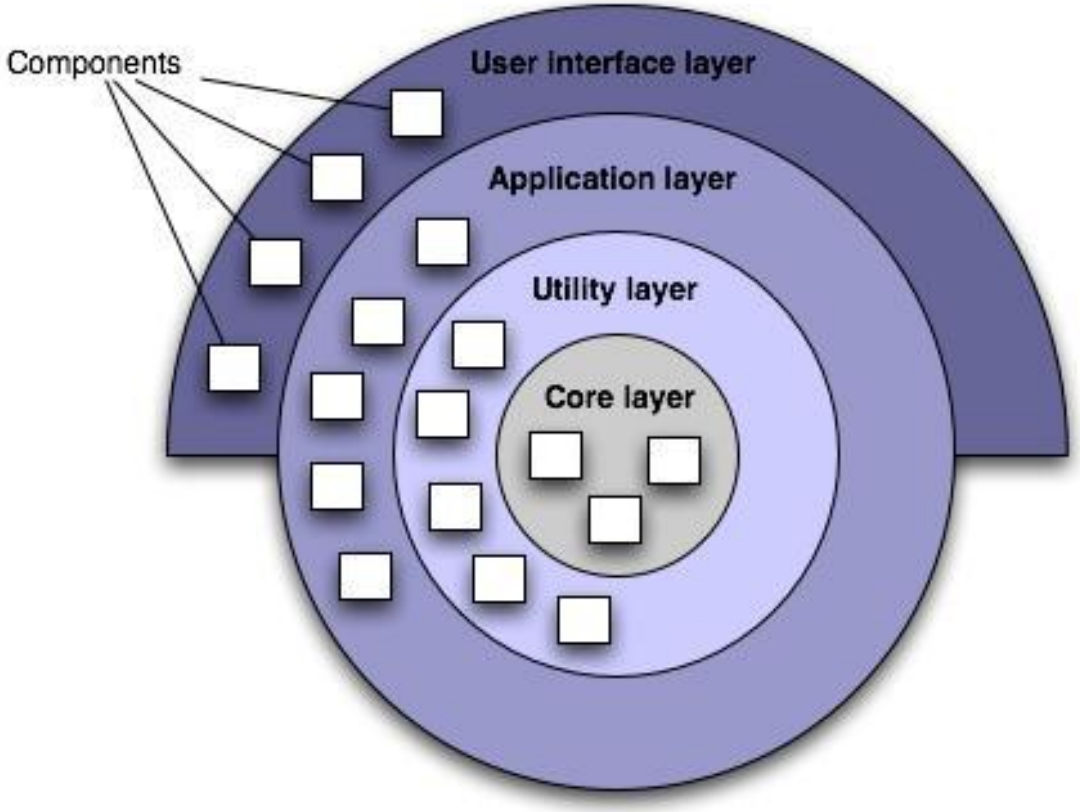
Arsitektur Call and Return



Arsitektur Object-Oriented



Arsitektur Layered



Arsitektur Menu

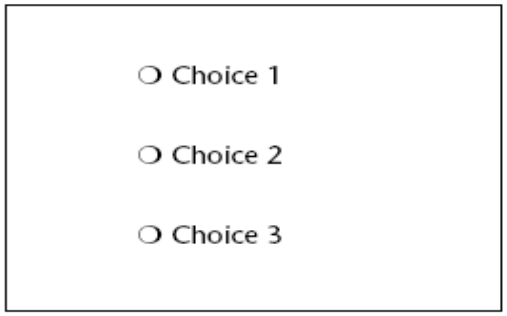


Figure 4.1: Single menu.

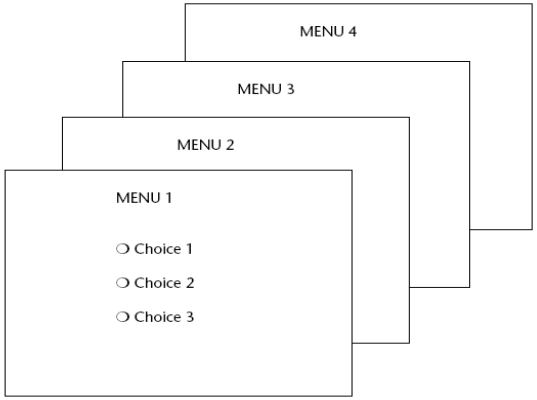


Figure 4.2: Sequential linear menus.

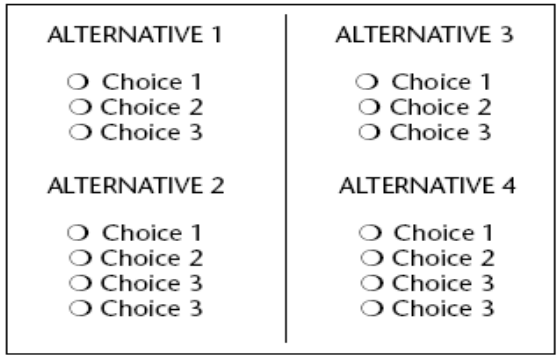


Figure 4.3: Simultaneous menus.

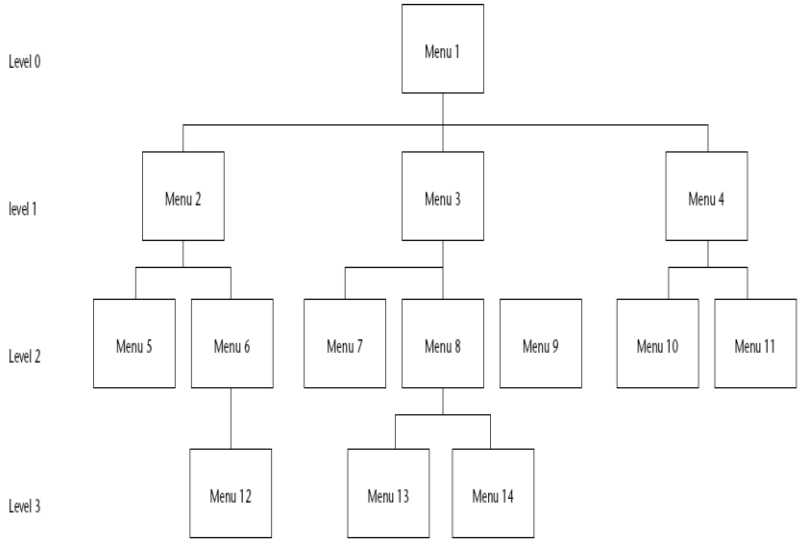


Figure 4.4: Hierarchical or sequential menus.

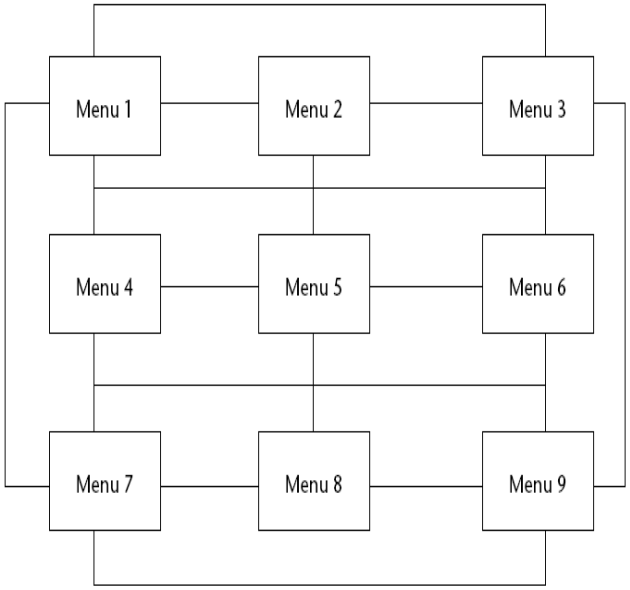
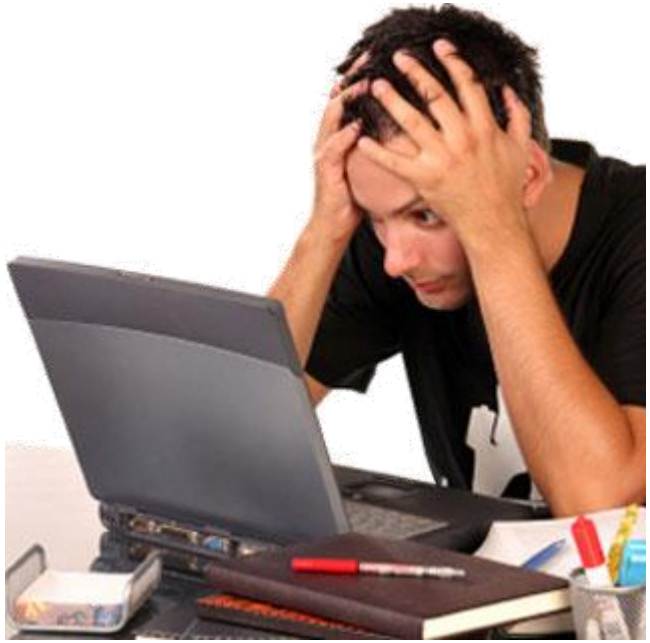


Figure 4.5: Connected menus.

PERANCANGAN ANTARMUKA

Perancangan Antarmuka

1. Mudah digunakan
2. Mudah dipelajari
3. Mudah dipahami



Aturan Perancangan Antarmuka

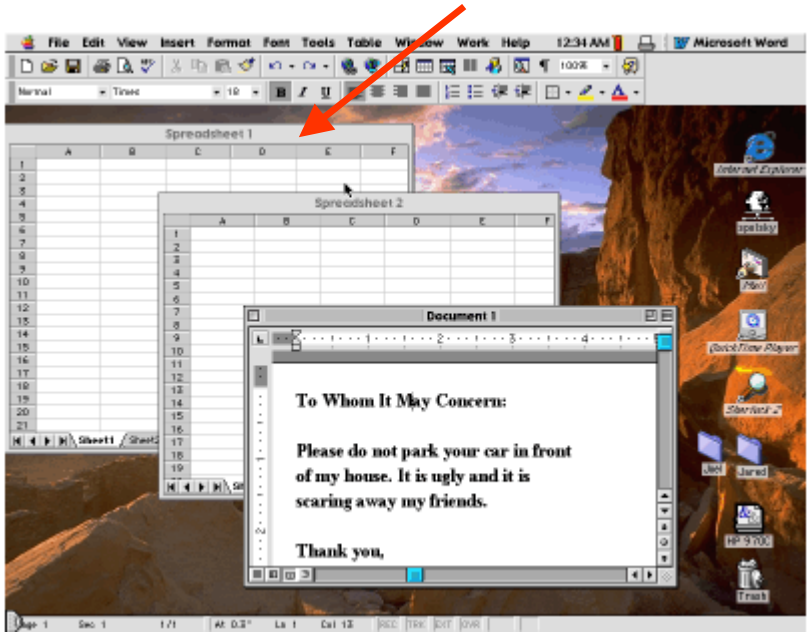
1. Place the user in control
2. Reduce the user's memory load
3. Make the interface consistent

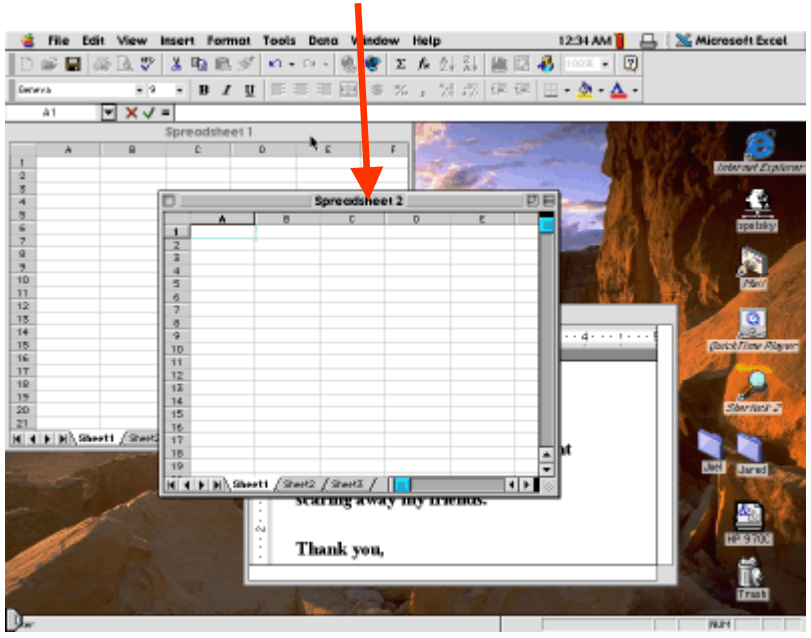
PRAKTIKAL UNTUK PERANCANGAN ANTARMUKA

Model Pengguna Sederhana

Click here

This window comes to top!



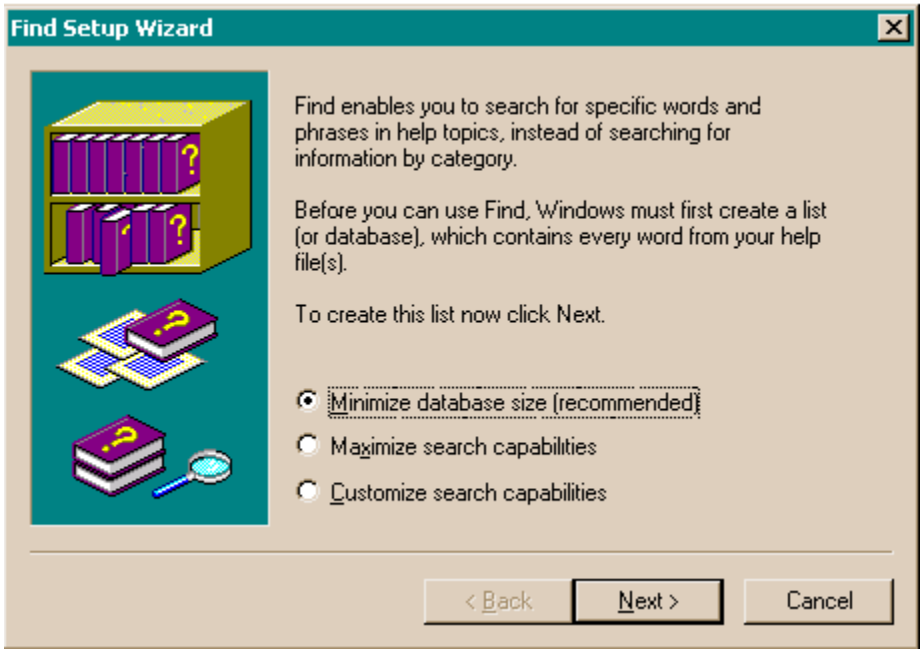


(“invisible sheets” in Excel)

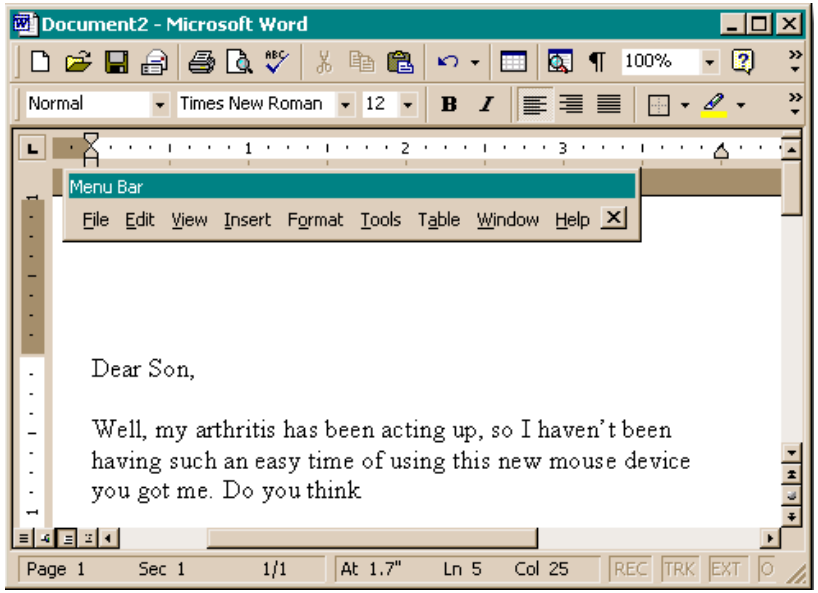
Pilihan

"Every time you provide an option, you're asking the user to make a decision." – Joel Spolsky

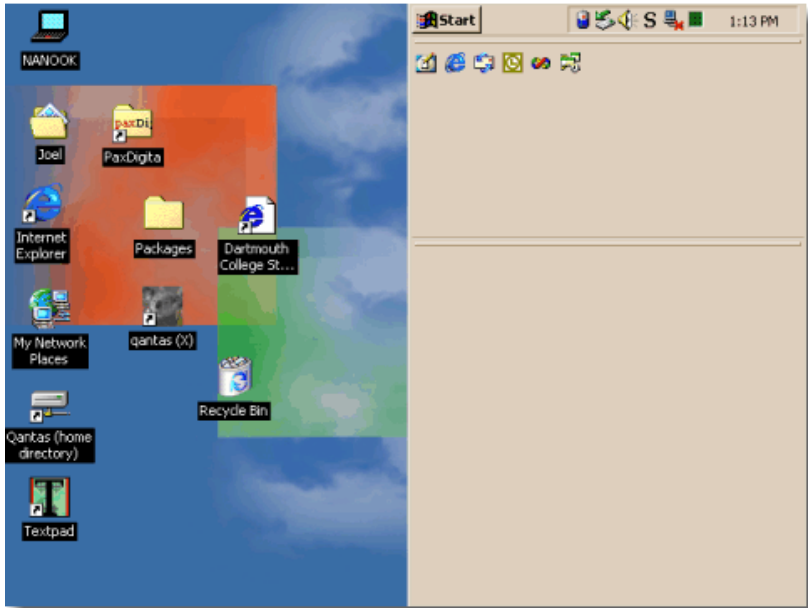
This is "unequivocally the most moronic 'wizard' dialog in the history of the Windows operating system. This dialog is so stupid that it deserves some kind of award. A whole new *category* of award."



Terlalu banyak kebebasan itu berbahaya



floating menu bar

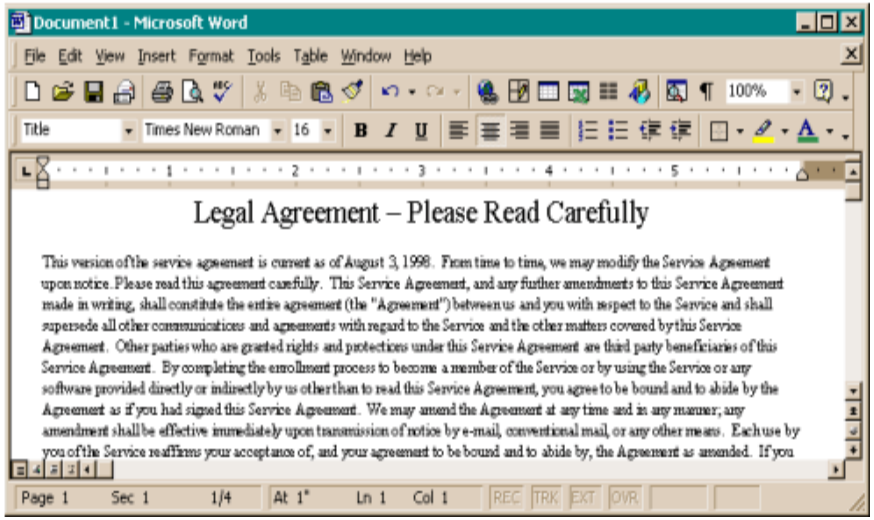


huge system tray

How many users want these?

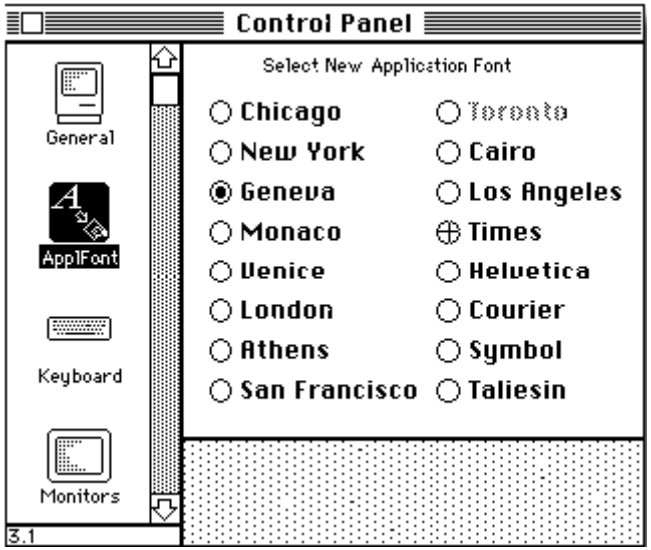
Metafora

vs

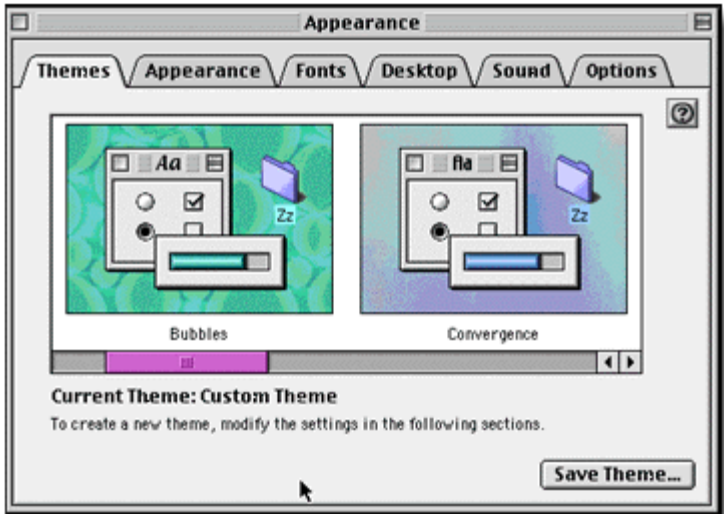


Also desktop, folders, paintbrush, ...

Keberdayagunaan



vs.



(30% usability)

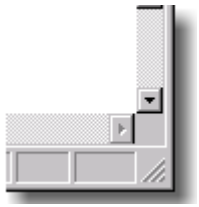
(100% usability)

Keberdayagunaan

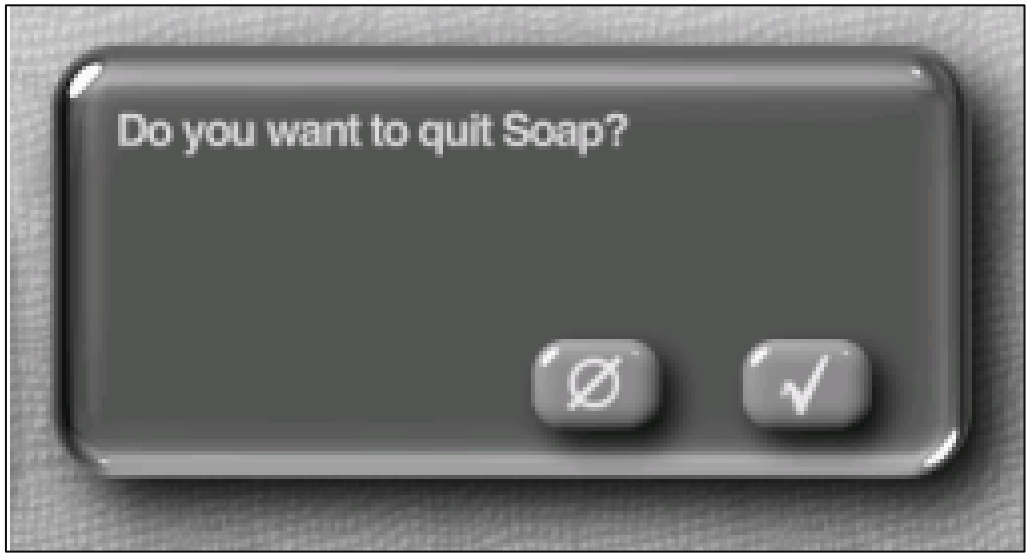
Where to grab?

Where to click?

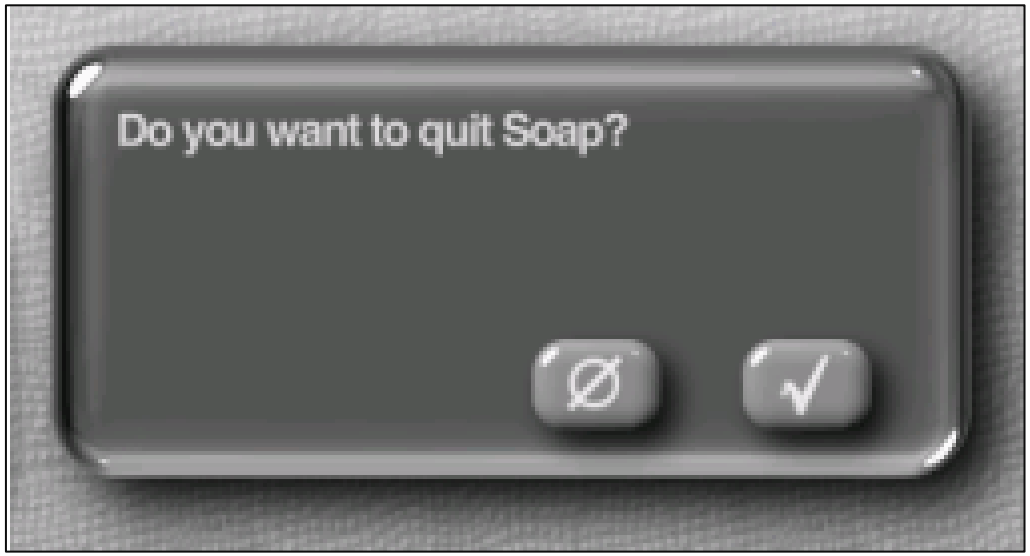
What to drag?



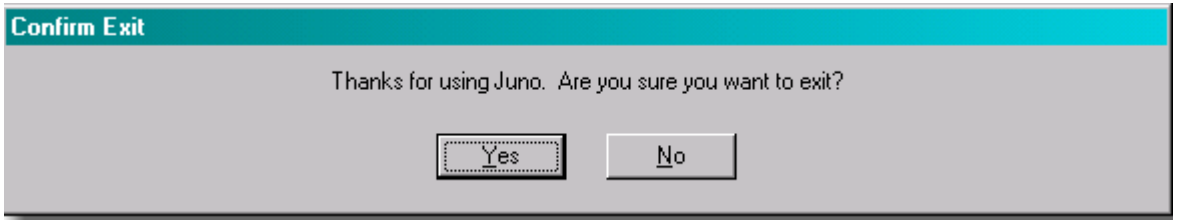
Consistency, not Creativity



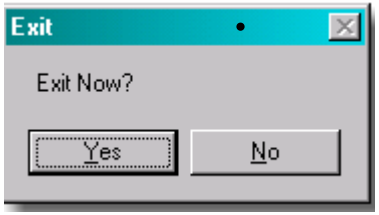
Consistency, not Creativity



Jangan Mengintimidasi Pengguna



vs



vs

(no dialog)

Which is better for an intimidated user?

Pemilihan Warna yang Tepat

Driving at night in San Jose, where the street lights are yellow

traffic light is green

traffic light is yellow

PERANCANGAN DATA

Perancangan Data

1. Skema relasi
2. Diagram relasi
3. Struktur File

Skema Relasi dan Diagram Relasi

1. Notasi untuk menggambarkan tabel, atribut tiap tabel, dan hubungan atribut kunci antar tabel yang berelasi.
2. Atribut kunci harus digarisbawahi.
3. Semua atribut harus dituliskan pada skema relasi.
4. Skema relasi merupakan bentuk konkrit dari ERD.
5. Diagram Relasi adalah representasi grafis dari skema relasi.

Skema Relasi

Format:

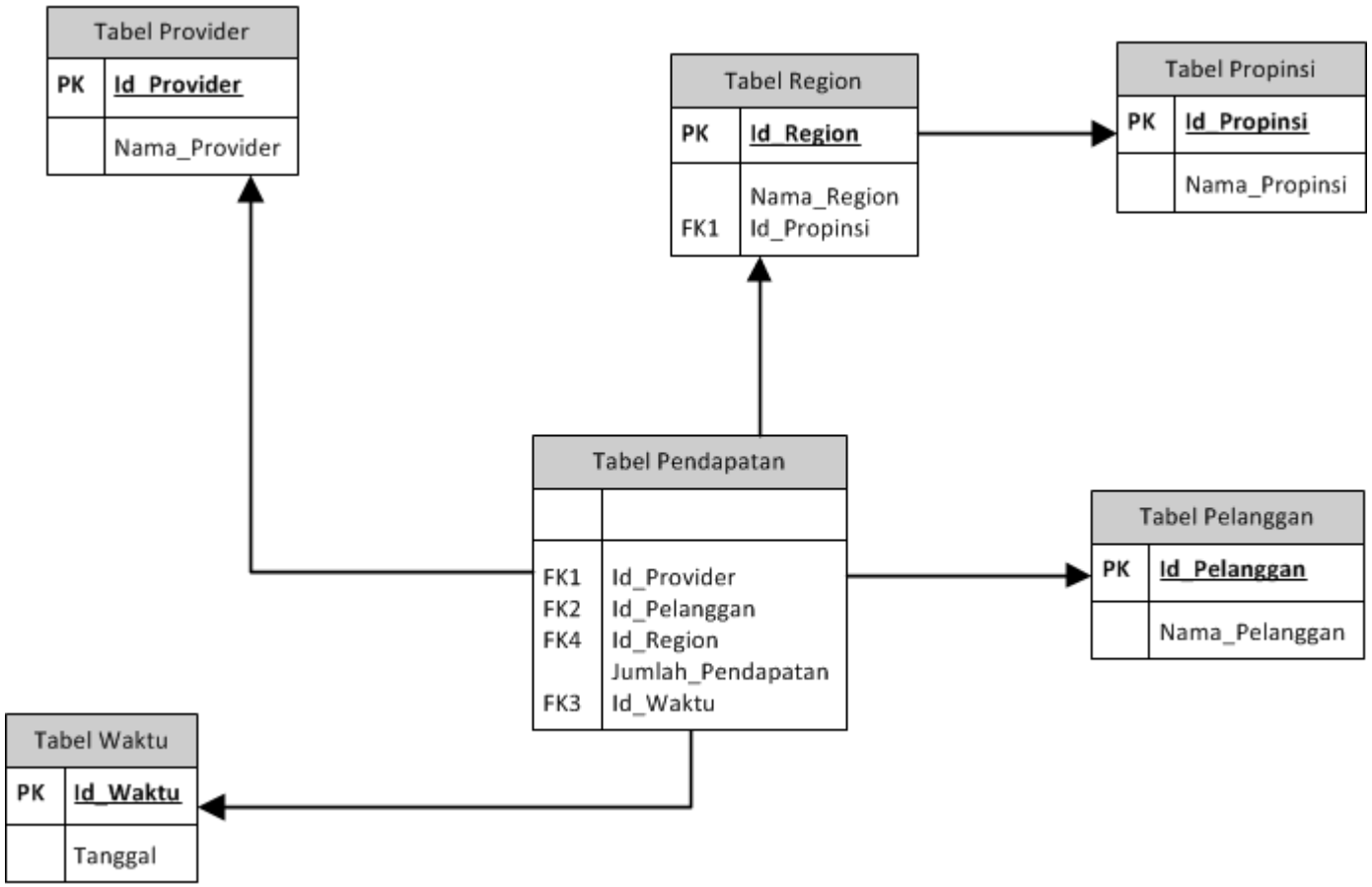
<nama_tabel> =

(<u><atribut_1></u>,<atribut_2>,...,<atribut_n>)

Contoh:

Dosen = (<u>NIP</u>, Nama_Dosen, Alamat_Dosen)

Diagram Relasi



Struktur File

1. Menggambarkan file yang akan mewakili satu tabel.
2. Ekstensi file disesuaikan dengan DBMS yang digunakan.
3. Jumlah tabel maupun atribut pada struktur file harus sama dengan skema relasi.

Struktur File

Format:

<no_tabel> . <nama_tabel>

Nama file : <nama_file>.<ekstensi_file>

Tempat penyimpanan: <media_penyimpanan>

Nama Field	Tipe	Panjang	Kunci	Keterangan

Struktur File

Contoh:

1. Tabel Dosen

Nama file : Dosen.accdb

Tempat penyimpanan: Harddisk

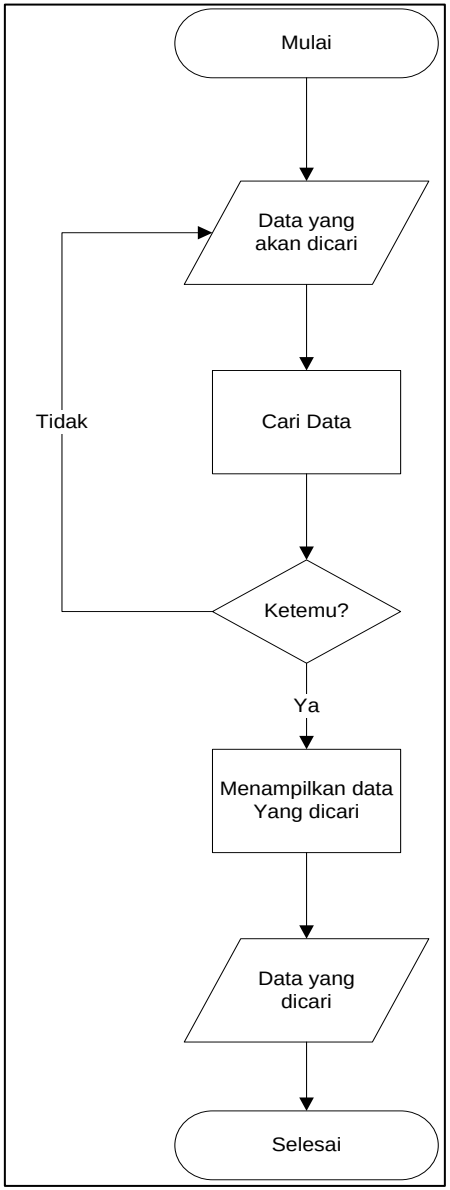
Nama Field	Tipe	Panjang	Kunci	Keterangan
NIP	varchar	10	Primary Key	Not null, Unique
Nama	Varchar	31		Not null, default="....."

PERANCANGAN PROSEDURAL / ALGORITMA

Perancangan Prosedural

1. Berisi perancangan perilaku dan kinerja suatu prosedur atau fungsi di dalam perangkat lunak.
2. Merupakan bentuk konkrit dari logika proses pada tahap analisis.
3. Bisa digambarkan dalam bentuk flowchart ataupun pseudo code.

Perancangan Prosedural



DPPL

DPPL (Deskripsi Perancangan Perangkat Lunak)

1. Dokumen yang diperuntukkan untuk menuliskan hasil desain perangkat lunak yang dibuat.
2. Aturan pembuatan sama dengan aturan pembuatan SKPL.
3. Template disediakan.

SELESAI...