



**PROGRAM STUDI
TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH
REKAYASA PERANGKAT LUNAK LANJUT

Rekayasa Perangkat Lunak Lanjut

Overview: Perancangan Terstruktur

Disusun Oleh:
Tim RPLL



Silabus Mata Kuliah

1. Pendahuluan
2. *Overview: Analisis Terstruktur*
3. **Overview: Perancangan Terstruktur – Arsitektur, Interface, Data**
4. Analisis Berorientasi Objek
5. Perancangan Berorientasi Objek
6. Pengenalan Web App. + Requirement Web App.
7. Konsep Web Engineering

Overview Perancangan Terstruktur

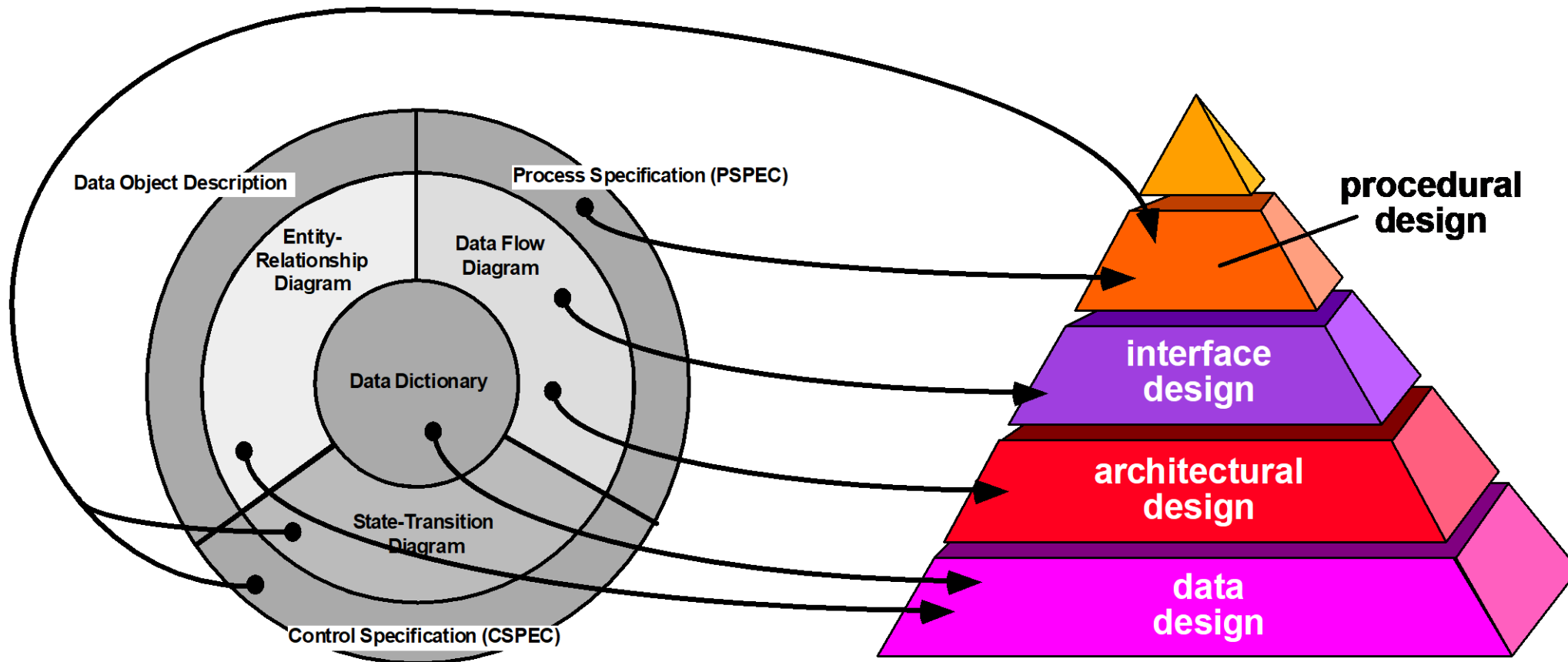
Apa tujuan perancangan?

- ✓ Memodelkan **SOLUSI** agar siap diimplementasikan (**dibuat program-nya**)

Apa yang dimodelkan pada tahap perancangan?

- ✓ Perancangan Arsitektural: Struktur Modul
- ✓ Perancangan Antarmuka:
 - Antarmuka antar-modul
 - Antarmuka dengan S/W lain atau H/W
 - Antarmuka dengan pengguna
- ✓ Perancangan Data: Struktur Data dan Skema Basis Data
- ✓ Perancangan Prosedural / Component-Level: Algoritma

Analysis to Design



THE ANALYSIS MODEL

THE DESIGN MODEL

Design Principles

Should consider alternative approaches

- Harus mempertimbangkan pendekatan alternatif

Should be traceable to the analysis model

- Harus dirujuk dari model analisis

Should “minimize the intellectual distance”

- Haruskah "meminimalkan jarak intelektual"

Should exhibit uniformity and integration

- Harus menampilkan keseragaman dan integrasi

Should be structured to accommodate change

- Harus terstruktur untuk mengakomodasi perubahan

Should be structured to degrade gently

- Harus terstruktur untuk mendegradasi lembut

Design is not coding, coding is not design

- Desain bukan coding, coding bukan desain

Should be assessed for quality

- Harus dinilai untuk kualitas

Should be reviewed to minimize conceptual (semantic) error

- Harus direview untuk meminimalkan konseptual (semantik) error

Review Arsitektural Design – Struktur Modul

- ❑ Kesalahan umum:
 - Perancangan modul dibuat tanpa mengacu pada DFD
- ❑
 - Tiap modul belum dilengkapi dengan daftar fungsi yang harus disediakan modul
 - ✓ Cari batas incoming flow dan outgoing flow pada DFD
 - ✓ Gambarkan hirarki modulnya
- ❑ Sebutkan elemen struktur modul!
 - ✓ Modul
 - ✓ Keterkaitan antar-modul

Why Architecture?

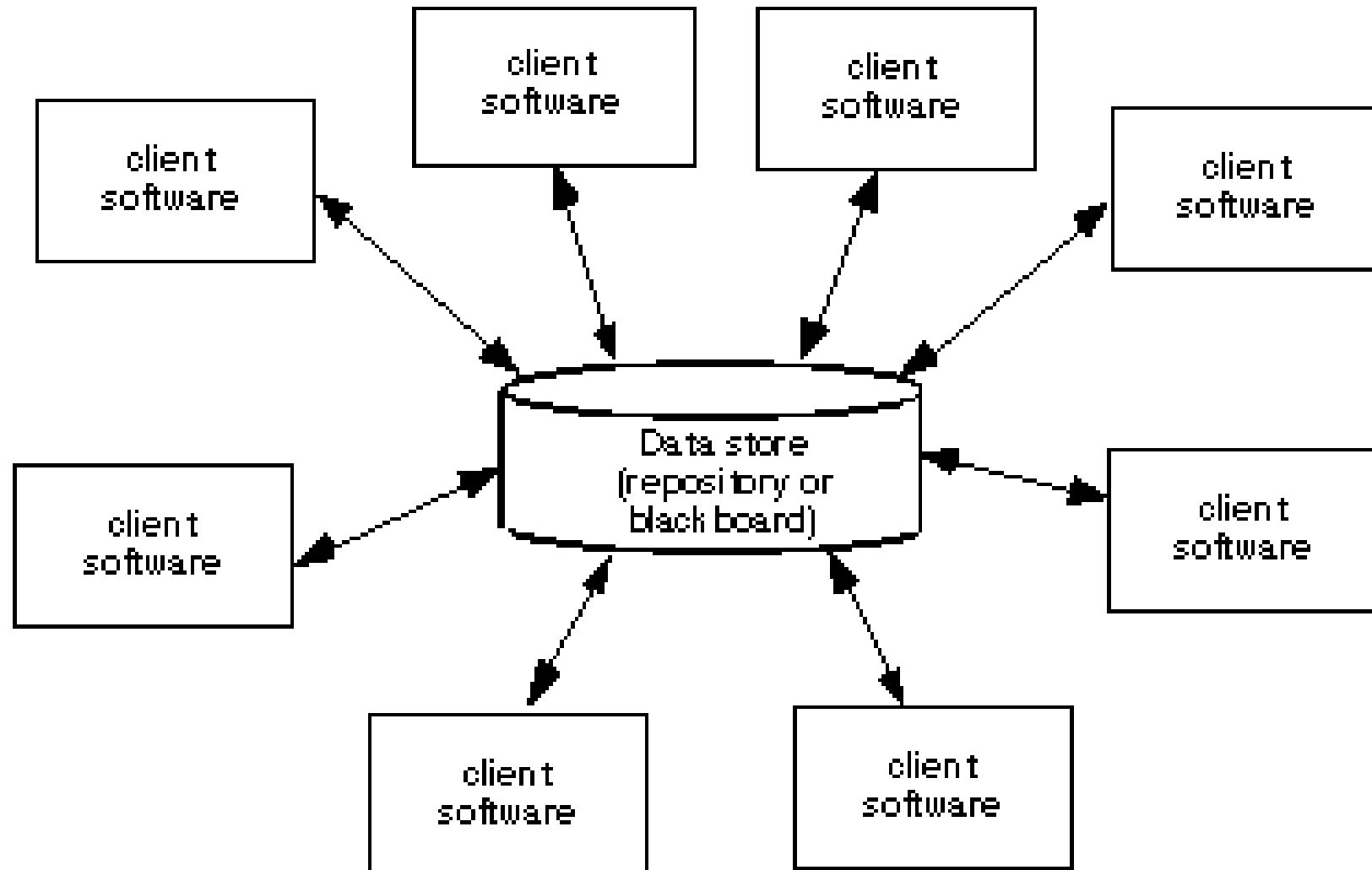
Arsitektur bukanlah operasi perangkat lunak. Sebaliknya, itu adalah representasi yang memungkinkan seorang insinyur perangkat lunak untuk:

- (1) menganalisis efektivitas desain dalam memenuhi persyaratan yang dinyatakannya,
- (2) mempertimbangkan alternatif arsitektur pada tahap ketika membuat perubahan desain masih relatif mudah, dan
- (3) mengurangi risiko yang terkait dengan pembangunan perangkat lunak.

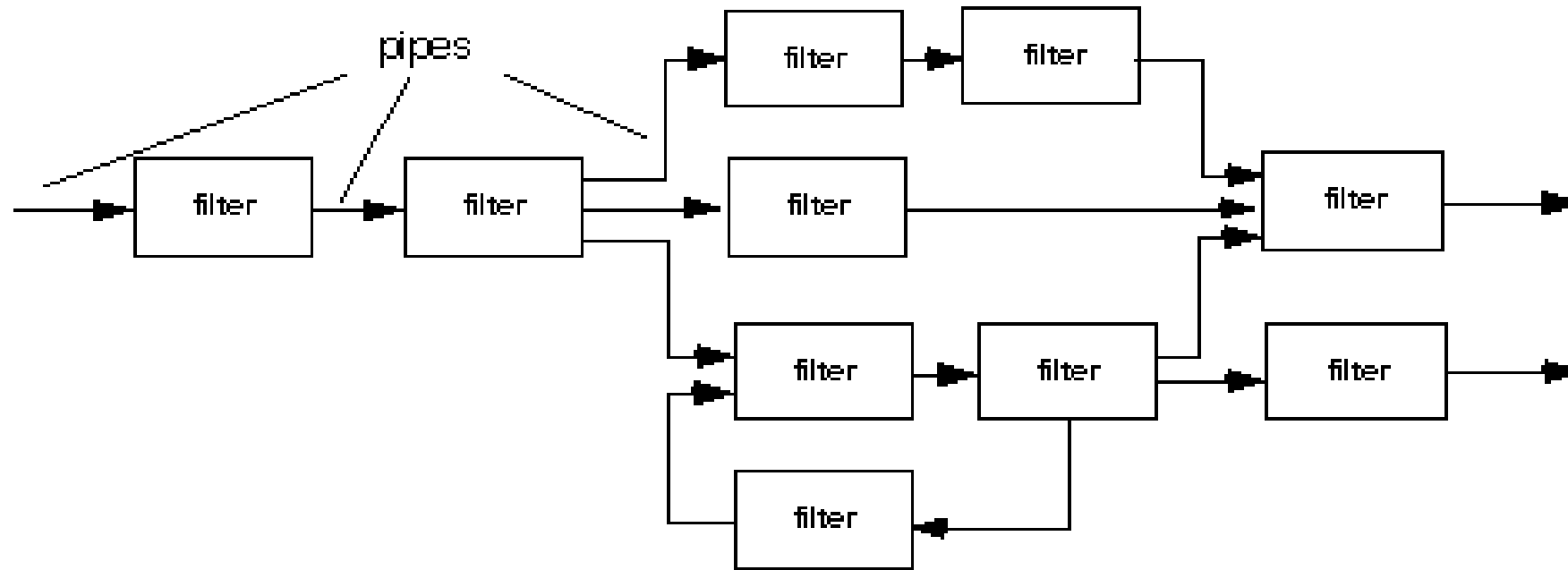
Architectural Styles

- Data-centered architectures
- Data flow architectures
- Call and return architectures
- Object-oriented architectures
- Layered architectures

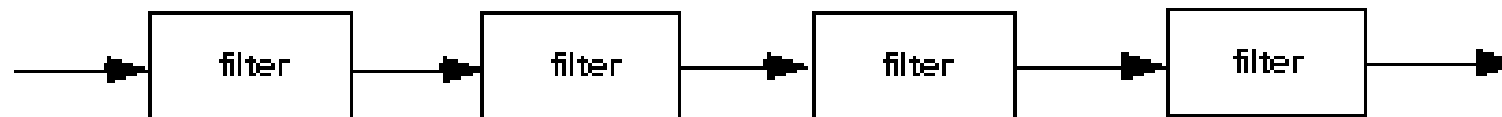
Data-Centered Architecture



Data Flow Architecture

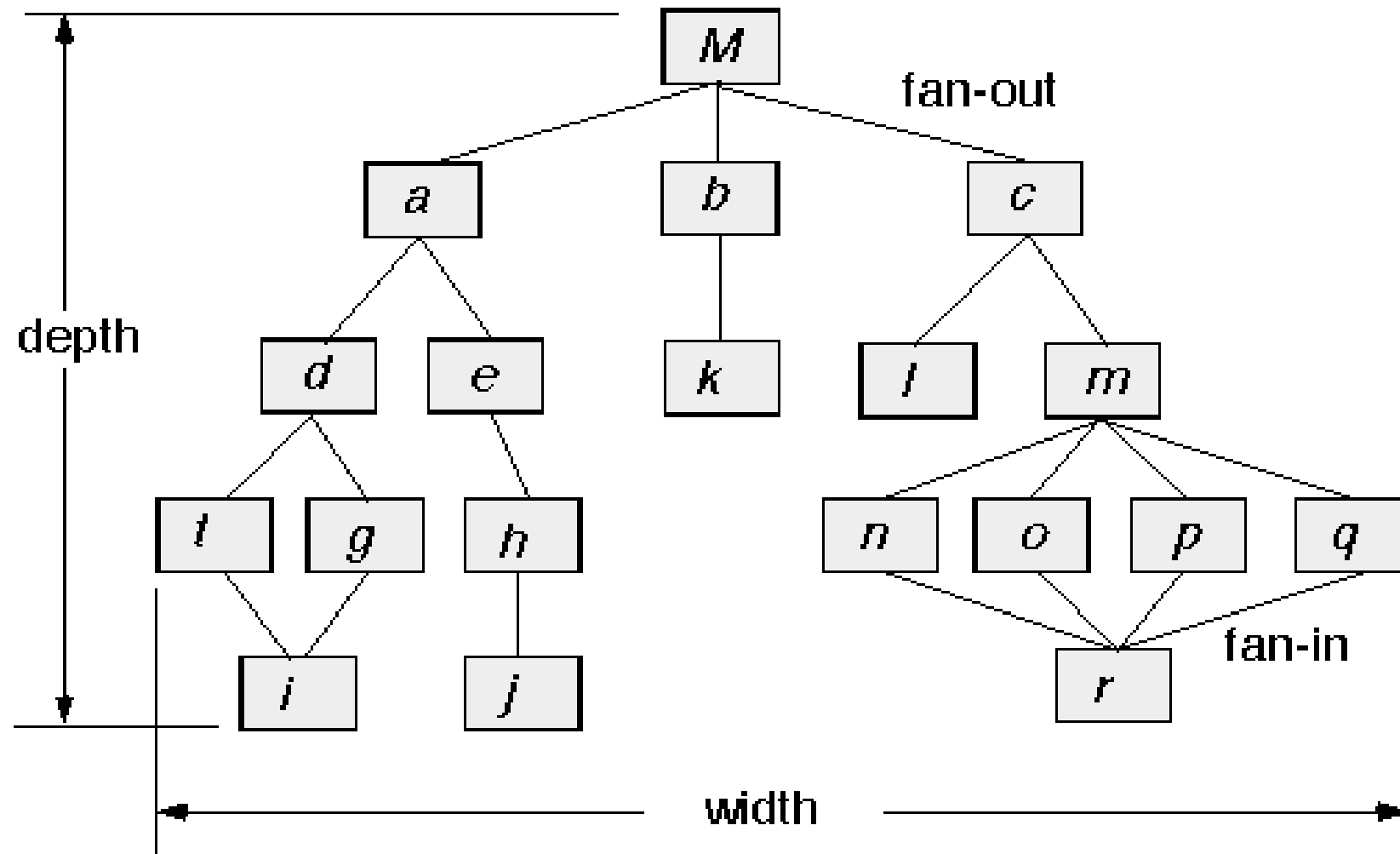


(a) pipes and filters

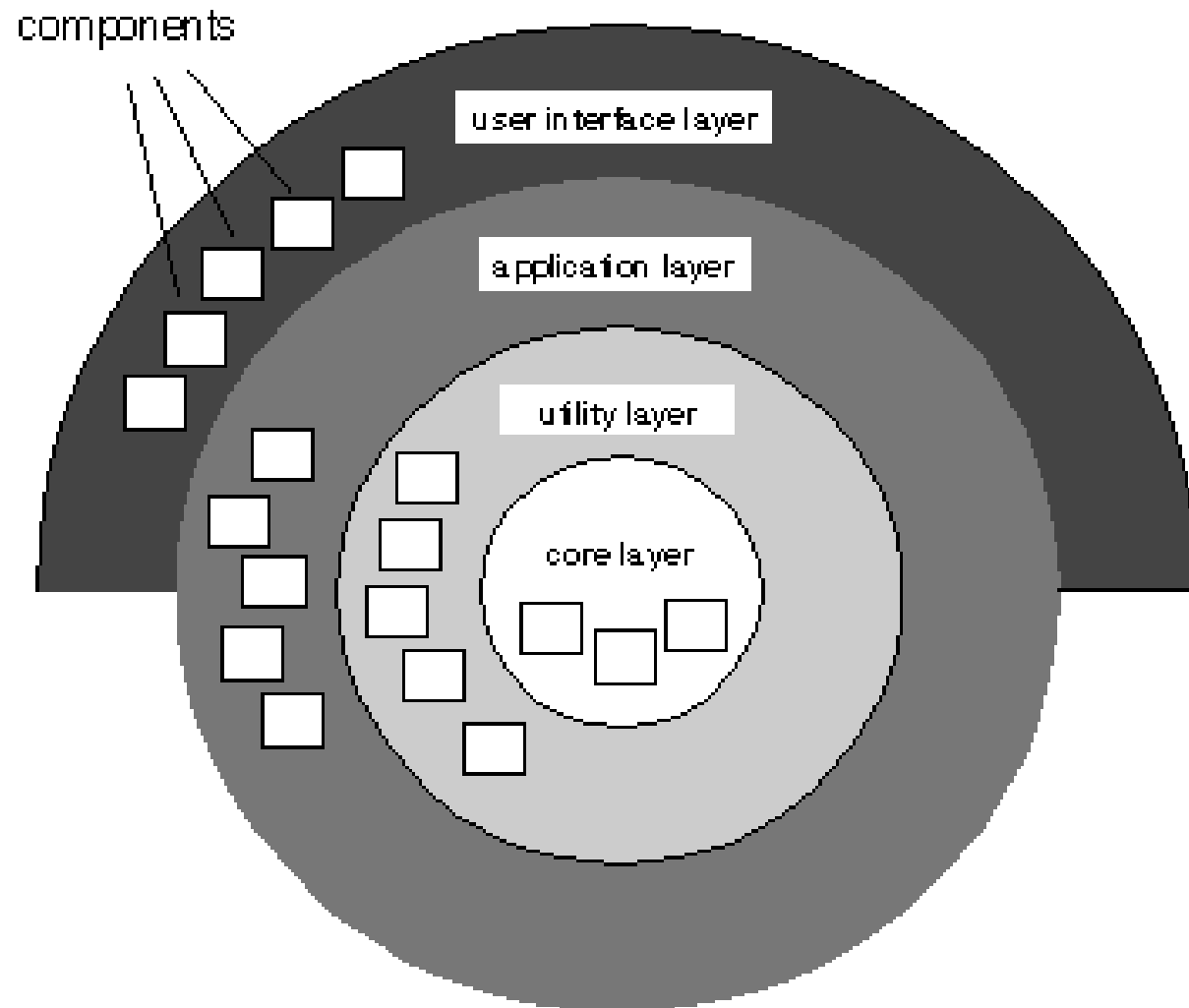


(b) batch sequential

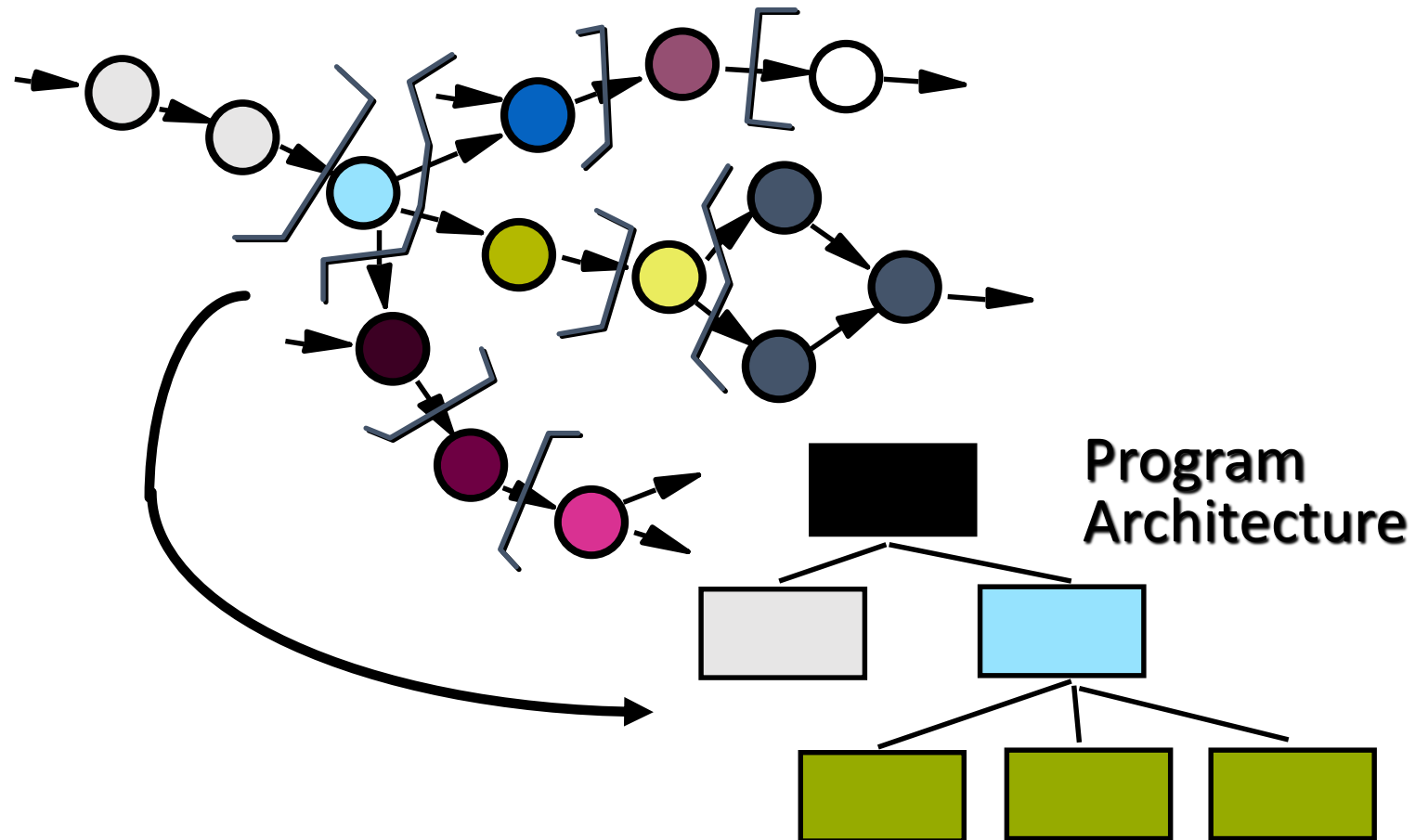
Call and Return Architecture



Layered Architecture

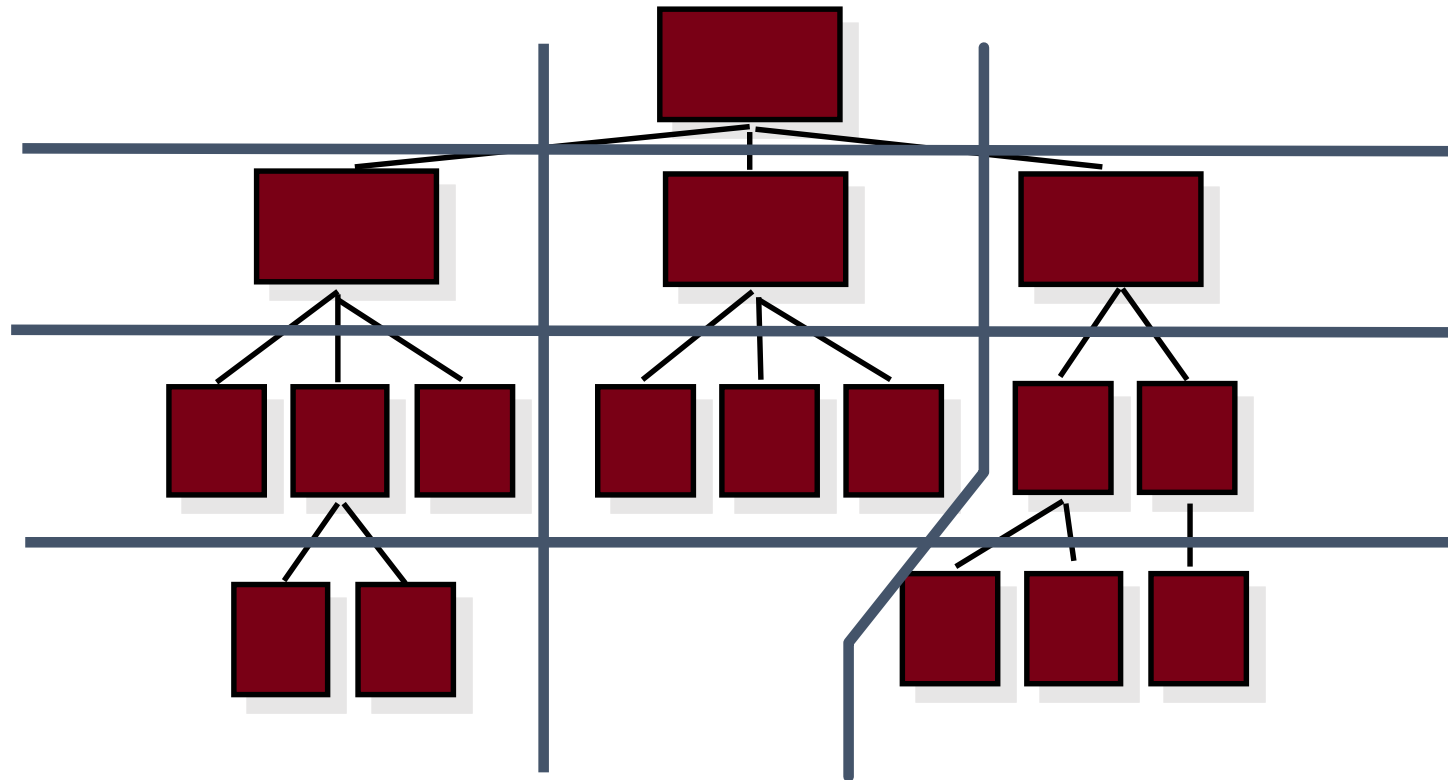


Deriving Program Architecture



Partitioning the Architecture

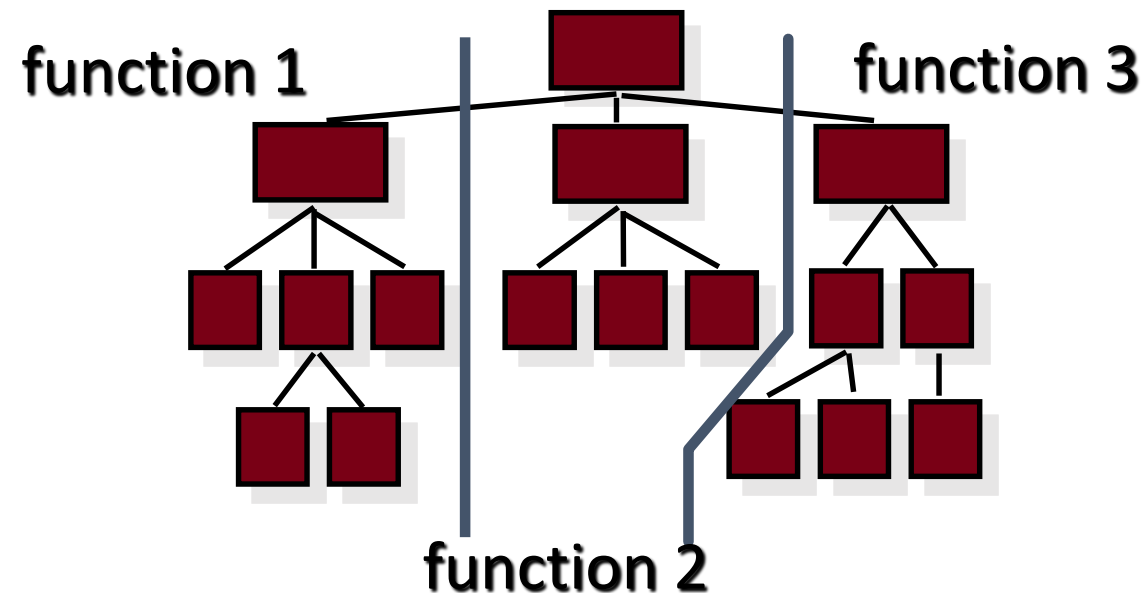
"Horizontal" and "vertical" partitioning are required



Horizontal Partitioning

Mendefinisikan cabang terpisah dari hirarki modul untuk setiap fungsi utama

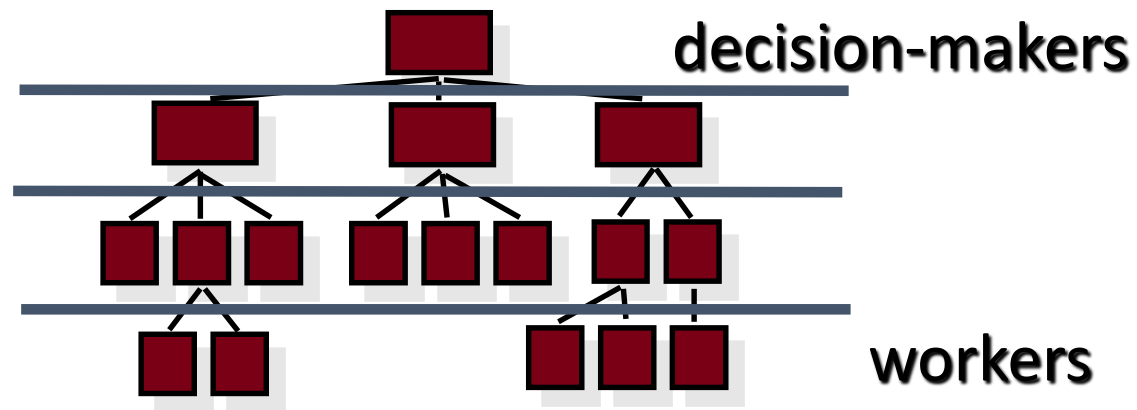
Gunakan modul kontrol untuk mengkoordinasikan komunikasi antara fungsi



Vertical Partitioning: Factoring

Pengambilan keputusan dan kerja yang bertingkat

Pengambilan keputusan modul harus berada di atas arsitektur



Why Partitioned Architecture?

- Menghasilkan software yang lebih mudah diuji
- Menunjuk pada software yang lebih mudah dipelihara.
- Menghasilkan propagasi dari sedikit efek samping
- Menghasilkan software yang lebih mudah diperluas.

Review: Perancangan Data

➤ Apa yang dihasilkan pada perancangan Data ?

- ✓ Struktur data
- ✓ Skema basis data
- ✓ Rancangan detil tiap tabel:

Nama, deskripsi, volume, field, key, dll

➤ Bagaimana tahapan merancang Data ?

- ✓ Review ERD
- ✓ Petakan menjadi skema basis data

Entity → tabel

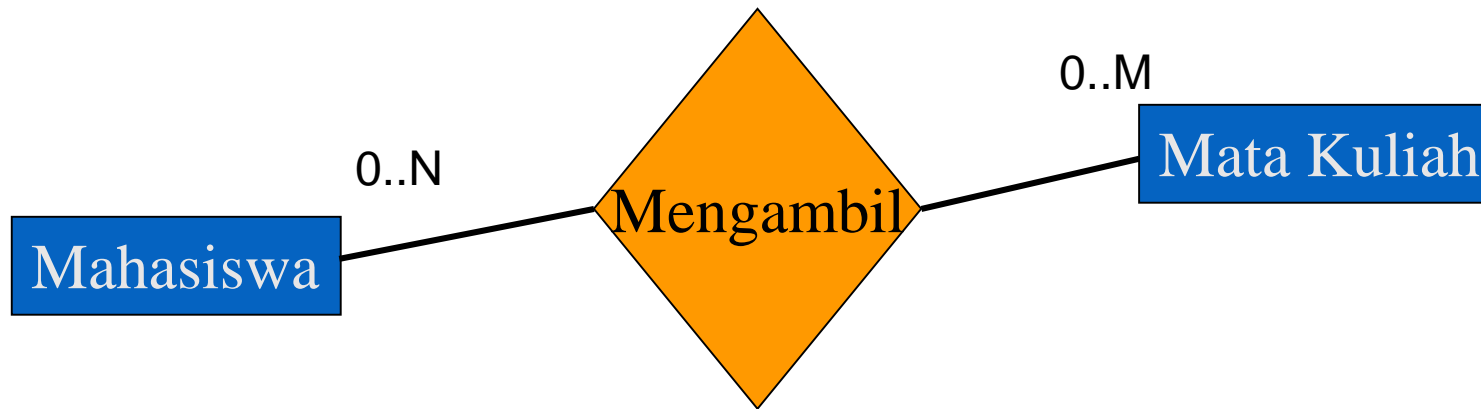
■ N ke M → jadi tabel

Relasi:

■ 1 ke N → jadi tabel

■ 1 ke 1 → dititipkan

Review: Perancangan Data



Skema Basis Data

Tabel Mahasiswa

NIM, Nama, Alamat, ...

Tabel Mata Kuliah

KD_Kul, Nama_Kul, SKS

Tabel Pengambilan_Kuliah

Sem, Thn, NIM, Kd_Kul, Nilai

Review: Perancangan Antarmuka

➤ Apa yang dihasilkan pada perancangan antarmuka ?

- ✓ Inter-modular interface design
- ✓ External interface design
- ✓ Human-computer interface design

➤ Apa arahan untuk merancang antarmuka ?

Three golden rules – Theo Mandel :

- ✓ Place user in control
- ✓ Reduce the user's memory load
- ✓ Make the interface consistent

Interface Design

Elemen-elemen perancangan interface untuk perangkat lunak menjelaskan **Bagaimana arus informasi masuk dan keluar dari sistem**, dan bagaimana arus informasi tersebut berkomunikasi diantara komponen2 yang didefinisikan sebagai bagian dari arsitektur.

User Interface Design

Tiga aturan emas → Theo Mandel

1. Menempatkan user di dalam kontrol

- “ Apa yang saya inginkan adalah sebuah sistem yang membaca pikiran saya. Dia tahu apa yang ingin saya lakukan sebelum saya butuhkan dan membuat mudah saya untuk melakukannya”

2. Mengurangi muatan memori user,

- Semakin banyak user harus mengingat, semakin banyak interaksi kesalahan dengan sistem
- Sistem seharusnya mengingat

3. Membuat Interface yang Konsisten

Review: Perancangan Prosedural

- Apa yang dihasilkan dari perancangan prosedural ?
 - ✓ Rancangan detil tiap modul dan fungsi
- Notasi apa yang bisa digunakan ?
 - ✓ Flowcharts
 - ✓ Box diagrams (Nassi-Scheidnerman charts)
 - ✓ Decision table
 - ✓ Program Design Language

Review: Assessment Criteria

● Modularity

- ❖ Notasi mendukung pengembangan PL

● Overall simplicity

- ❖ Mudah di pelajari, digunakan dan di tulis

● Ease of editing

- ❖ Mudah untuk di modifikasi Ketika terjadi perubahan

● Machine readability

- ❖ notation dapat diinput langsung kedalam system

● Maintainability

- ❖ Maintenance konfigurasi melibatkan pemeliharaan desain procedural

Review: Assessment Criteria (2)

● Structure enforcement

- ❖ Menggunakan pemrograman terstruktur

● Automatic processing

- ❖ Mengizinkan perancang untuk memverifikasi kebenaran dan kualitas perancangan

● Data representation

- ❖ Kemampuan untuk merepresentasikan data lokal dan global secara langsung

● Logic verification

- ❖ Verifikasi logika otomatis meningkatkan kecukupan pengujian

● Easily converted to program source code

- ❖ Membuat pembuatan kode lebih cepat

Ada Pertanyaan





Terima Kasih