

Rekayasa Perangkat Lunak

Desain Arsitektur

Oleh : Triastana Anang, ST, MT
Jurusan Teknologi Informasi
Politeknik Negeri Malang

fppt.com

Agenda

- Desain Arsitektur
- Proses Desain Arsitektur
- Architectural Pattern

Rekayasa Perangkat Lunak

fppt.com

Desain Arsitektur

- Desain arsitektur PL merupakan struktur komponen data & program yang diperlukan untuk membangun sebuah perangkat lunak
- Tahap awal dari proses desain sistem
- Model Desain Arsitektur bisa di re-use
 - Dapat diterapkan pada desain sistem lain
 - Abstraksi yang memungkinkan pengembang perangkat lunak untuk menggambarkan bagaimana organisasi dari sub system perangkat lunak

fppt.com

Proses Desain Arsitektur

- Tahapan Desain Arsitektur
 - Pembuatan desain data
 - Derivasi dari satu atau lebih struktur arsitektur sistem
 - Analisis gaya arsitektur alternatif untuk menyesuaikan kebutuhan pelanggan dan atribut kualitas
 - Elaborasi dari arsitektur berdasarkan arsitektur pattern yang dipilih
- Desainer database membuat desain arsitektur data untuk sistem yang mewakili komponen data
- System architect memilih arsitektur yang tepat sesuai dengan system requirement analysis

fppt.com

Atribut Arsitektur

- Performance
 - Seberapa cepat performance yang dibutuhkan fine-grain components vs coarse grained
- Security
 - Menggunakan arsitektur berlapis dengan aset penting dalam lapisan dalam yang terlindung
- Safety
 - Berhubungan dengan data integrity dan failure handling
- Availability
 - Seberapa besar fault tolerance
- Maintainability
 - Modular dan replaceable components.

fppt.com

Architectural Pattern yang sering digunakan

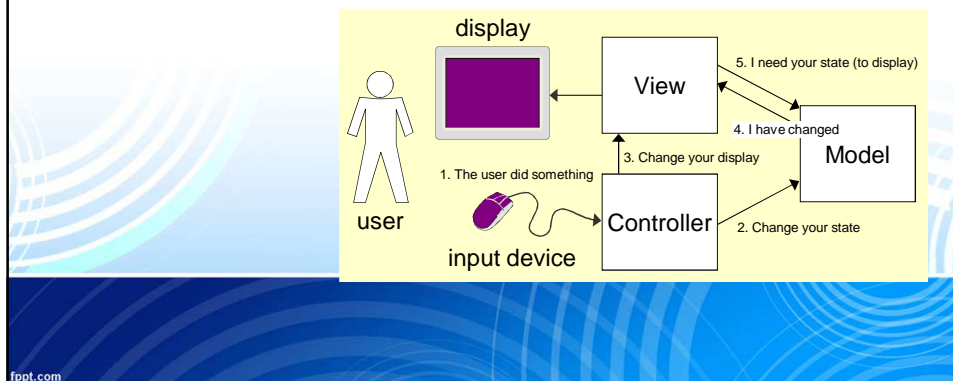
- Model-View-Controller
- Layered (or Multi-Tiered)
- Central Repository (database)
- Client/Server
- Pipe-and-Filter
- Peer-to-Peer

fppt.com

6

Model-View-Controller

- ◆ **Model:** menangani semua data dan state. Tidak mengenal cara kerja View ataupun Controller. Menyediakan API untuk mengambil data dan state (juga trigger)
- ◆ **View:** memberikan presentasi dari data Model
- ◆ **Controller:** Menghandle user input

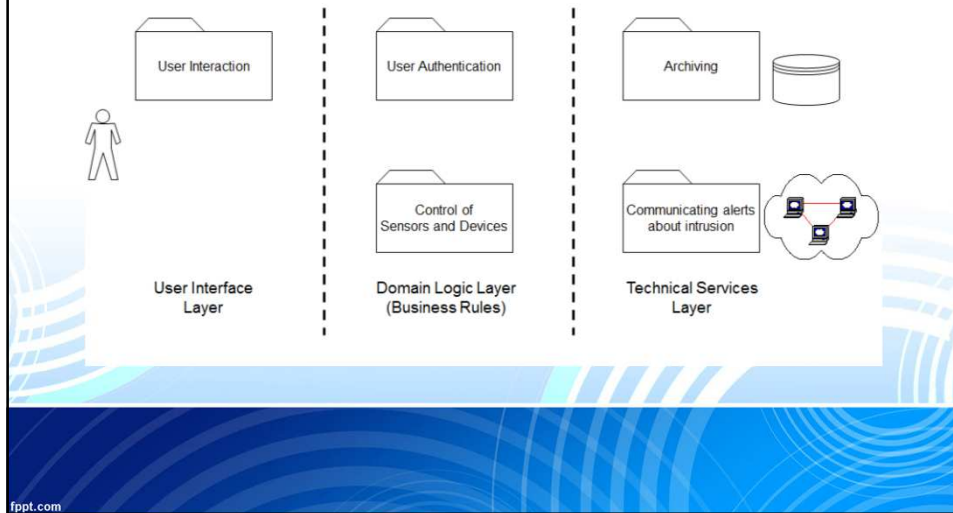


Model-View-Controller

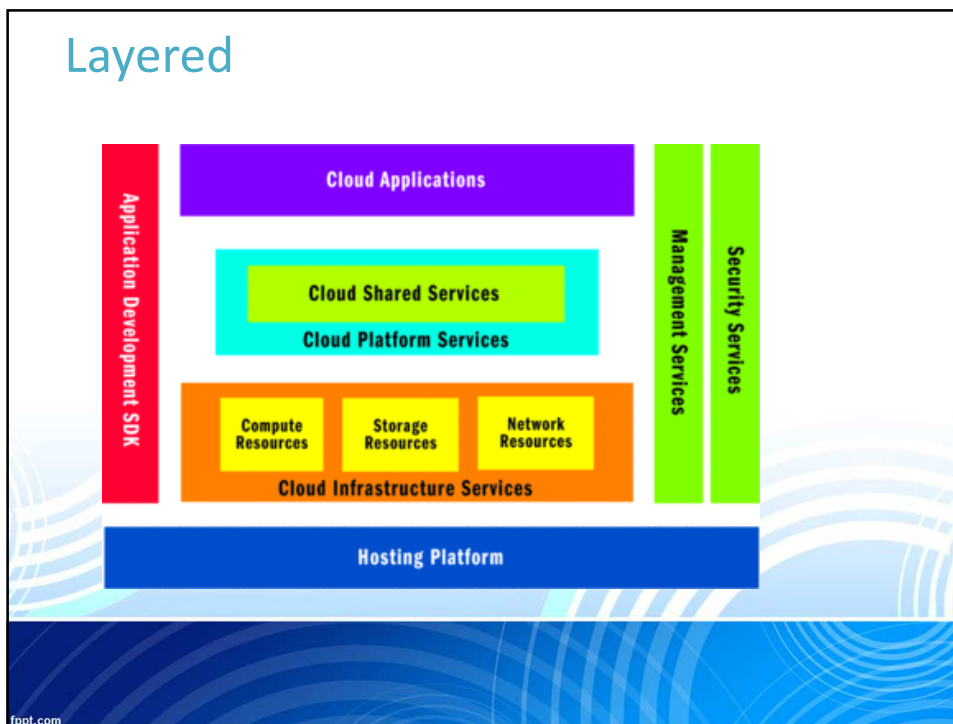
- Kelebihan
 - Memungkinkan independensi antara data dan representasi nya
 - Penyajian “data yang sama” dengan “cara yang berbeda” dengan perubahan di bagian presentasi
- Kekurangan
 - Menambah kompleksitas dan kode tambahan ketika "model data dan interaksi yang digunakan cukup sederhana"

Layered

Tiered Software Architecture



Layered



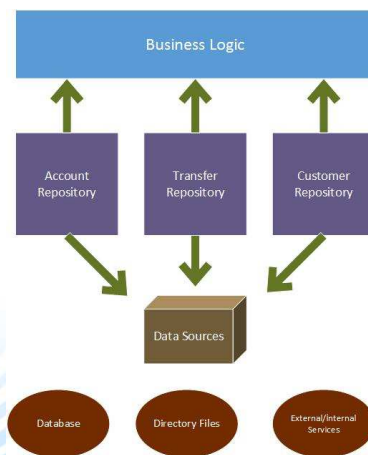
Layered

- Kelebihan
 - Memungkinkan penggantian satu lapisan selama interface dipertahankan tetap sama, tidak akan mengganggu lapisan lain
 - Tetap bisa menggunakan redundansi
- Kekurangan
 - Performance issue
 - memberikan pemisahan antar lapisan seringkali sulit
 - Lapisan mungkin harus berinteraksi langsung dengan lapisan yang lebih rendah

11

fppt.com

Repository



Data terpusat dengan model komunikasi klien atau sub-sistem lain yang berbeda

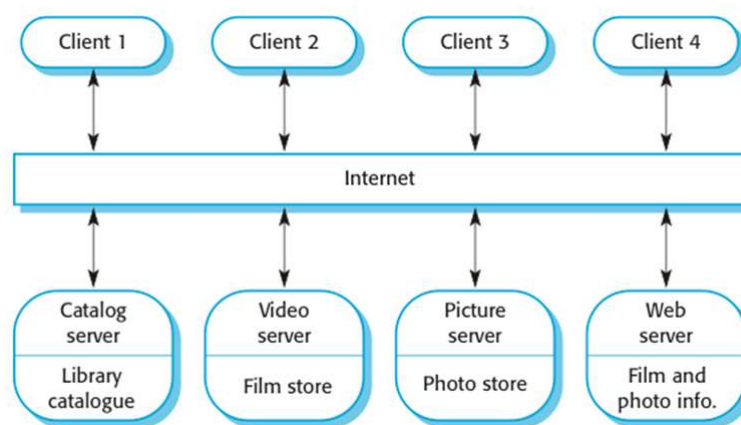
fppt.com

Repository

- Kelebihan
 - Konsistensi Data bisa terjaga
 - Perubahan dari satu komponen bisa diinformasikan ke komponen lain
 - Komponen indepen satu dengan yang lain
- Kekurangan
 - Database merupakan single point of failure
 - Tidak efisien bila seluruh data harus menggunakan data terpusat

13

Client Server



14

Client Server

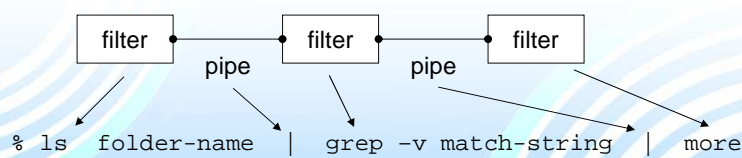
- Kelebihan
 - Client bisa merupakan perangkat yang sederhana
 - Perubahan logic bisa dengan mudah dilakukan di server tanpa perlu merubah client
 - Server bisa di distribusikan di jaringan
- Kekurangan
 - Server dan Jaringan merupakan single point of failure
 - Performance dari system tidak terprediksi

15

fppt.com

Pipe-and-Filter

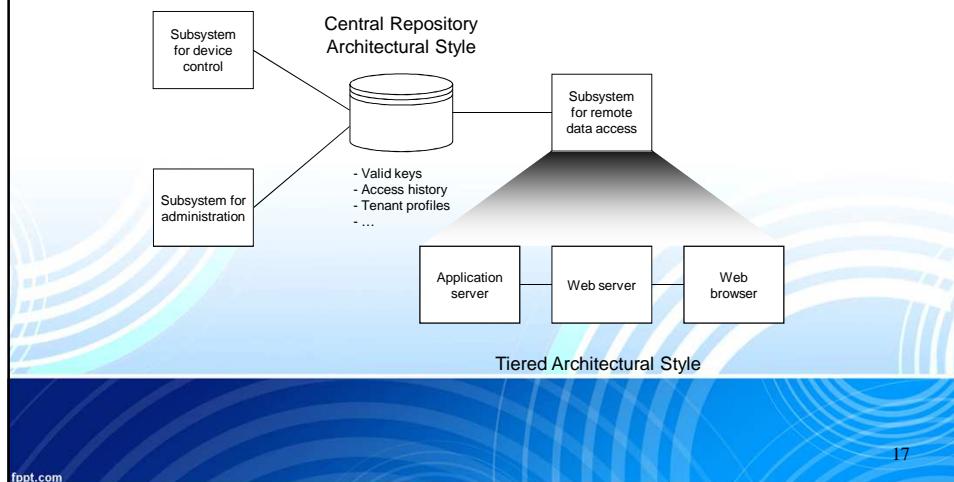
- Components: **Filters** merubah input menjadi output
- Connectors: **Pipe /** menyalurkan aliran data
- Contoh : UNIX shell commands



16

fppt.com

Sistem merupakan kombinasi dari beberapa Arsitektur



17

Merancang keseluruhan Arsitektur

- Interface Specification
 - sebagai kontrak antara penyedia komponen dan client, antarmuka harus
 - Terdokumentasi
 - Diperjelas, bukan hanya contoh
 - Mudah dimengerti dan tidak ambigu
- Menambahkan semantik
 - Deskripsi informal
 - Model desain (contoh: UML interaction diagrams)
 - pre/post conditions

18

Pertanyaan

fppt.com