

Arsitektur Perangkat Lunak

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi
Politeknik Negeri Malang

Tujuan

- Mengapa dibutuhkan arsitektur pada perangkat lunak
- Mengetahui pola arsitektur pada perangkat lunak

Arsitektur Perangkat Lunak

Berfungsi untuk menjelaskan bagaimana suatu sistem perangkat lunak diorganisir dengan menggambarkan struktur dari software secara keseluruhan.

Arsitektur perangkat lunak harus memodelkan struktur suatu sistem atau perangkat lunak serta bagaimana data dan komponen-komponen prosedural berkolaborasi satu dengan yang lainnya.

Mengapa Arsitektur Software Sangat Penting

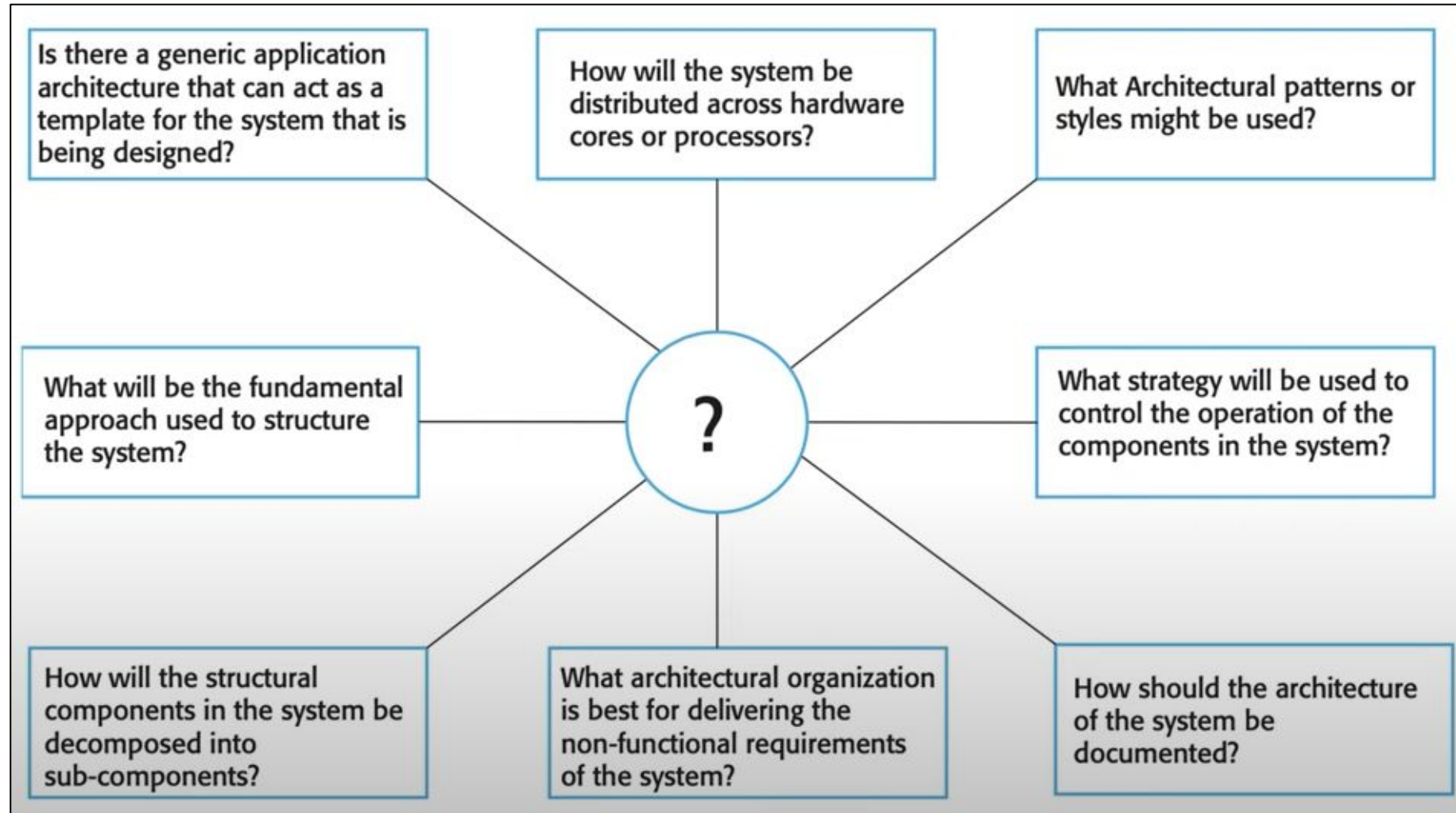
- Rancangan arsitektur dapat digunakan untuk mengkomunikasikan perangkat lunak kepada pihak-pihak yang terkait yaitu anggota tim Rekayasa Perangkat Lunak
- Arsitektur sistem menentukan keberhasilan sistem maupun menentukan pekerjaan Rekayasa Perangkat Lunak selanjutnya
- Arsitektur menggambarkan ilustrasi model intelektual suatu sistem yang distrukturkan untuk mudah dipahami

Membuat Rancangan *Software*

Perancangan Arsitektur sistem atau software adalah sebuah proses kreatif dimana kita membuat rancangan organisasi dari sebuah sistem yng harus memenuhi kebutuhan fungsional maupun kebutuhan non-fungsional dari sistem tersebut.

Oleh karena itu, kebutuhan fungsional dan kebutuhan non-fungsional harus sudah diidentifikasi terlebih dahulu pada tahap analisis.

Contoh diskusi dalam pencapaian Desain Arsitektur



Architectural View

Logical View

Menunjukkan abstraksi kunci dari sebuah sistem dalam bentuk objek atau kelas objek.

Process View

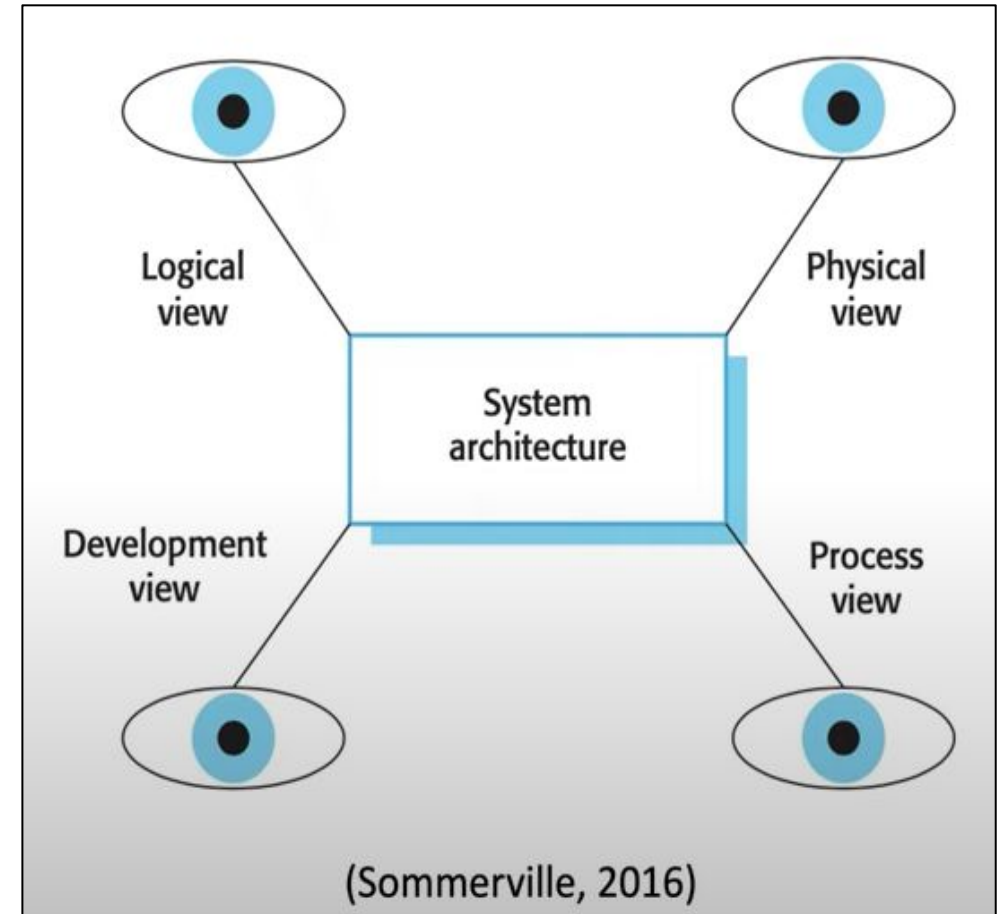
Menunjukkan bagaimana sistem menjalankan suatu proses tertentu.

Development View

Menunjukkan bagaimana sistem diuraikan untuk keperluan pengembangannya. Biasanya tergambarkan komponen-komponen sistem.

Physical View

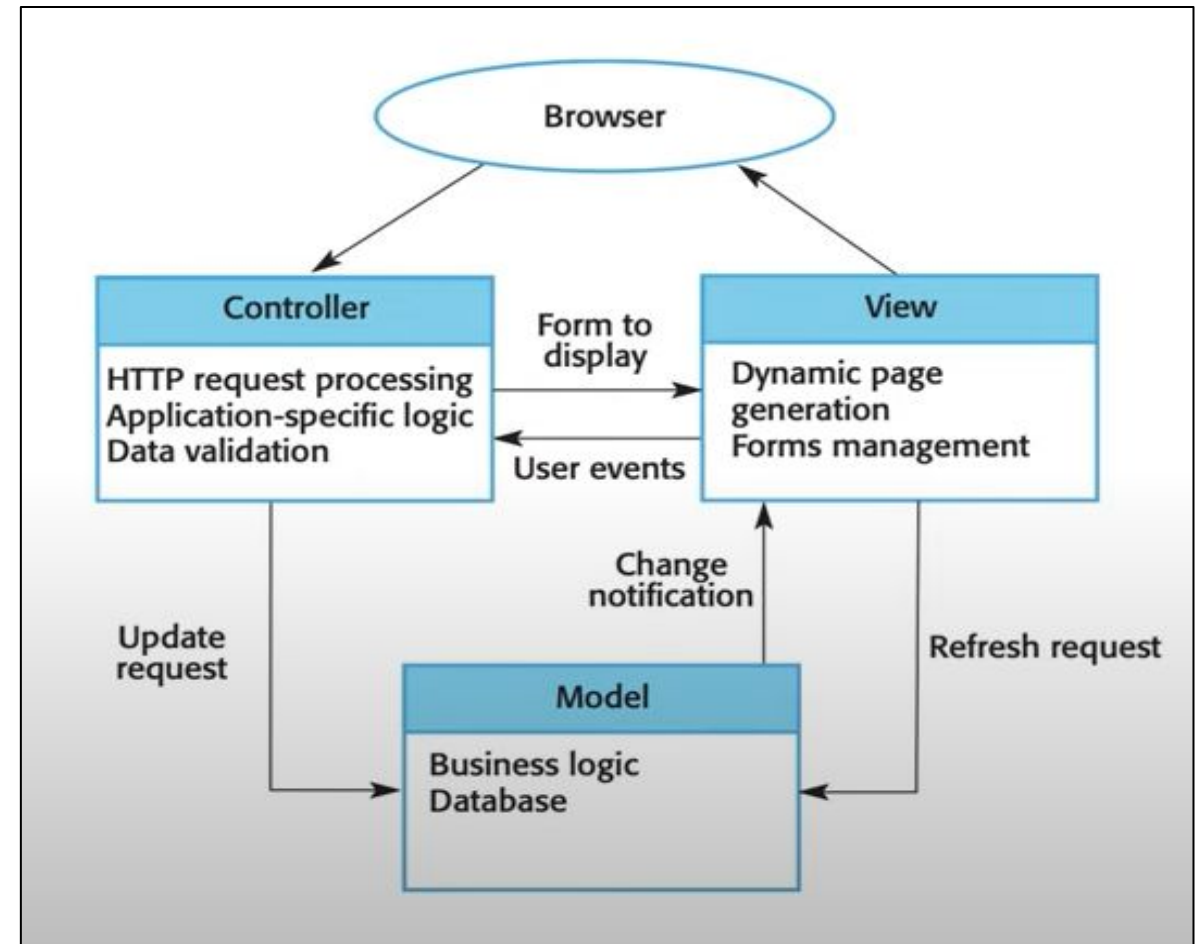
Menunjukkan perangkat keras yang berkaitan dengan sistem.



Pola Arsitektur

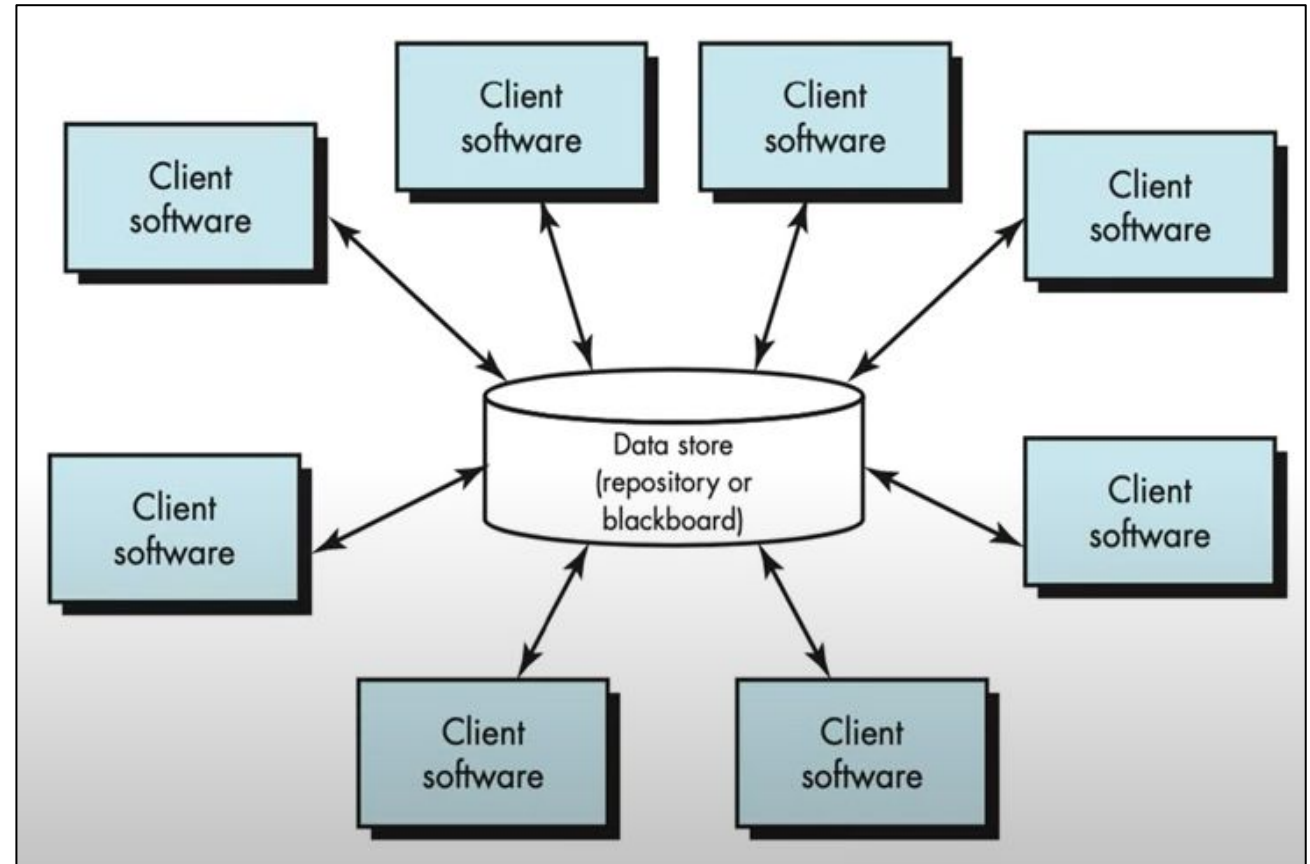
Ide dasar penggunaan pola arsitektur adalah sebagai sebuah cara untuk menggunakan kembali pengetahuan yang sudah ada sebelumnya, untuk permasalahan yang sama.

Contohnya:
Arsitektur MVC (Model View Controller) dapat digunakan sebagai pola untuk mengatasi permasalahan: “Bagaimana membuat struktur kode program untuk sistem berbasis web”.



Pola *Data Center Architecture*

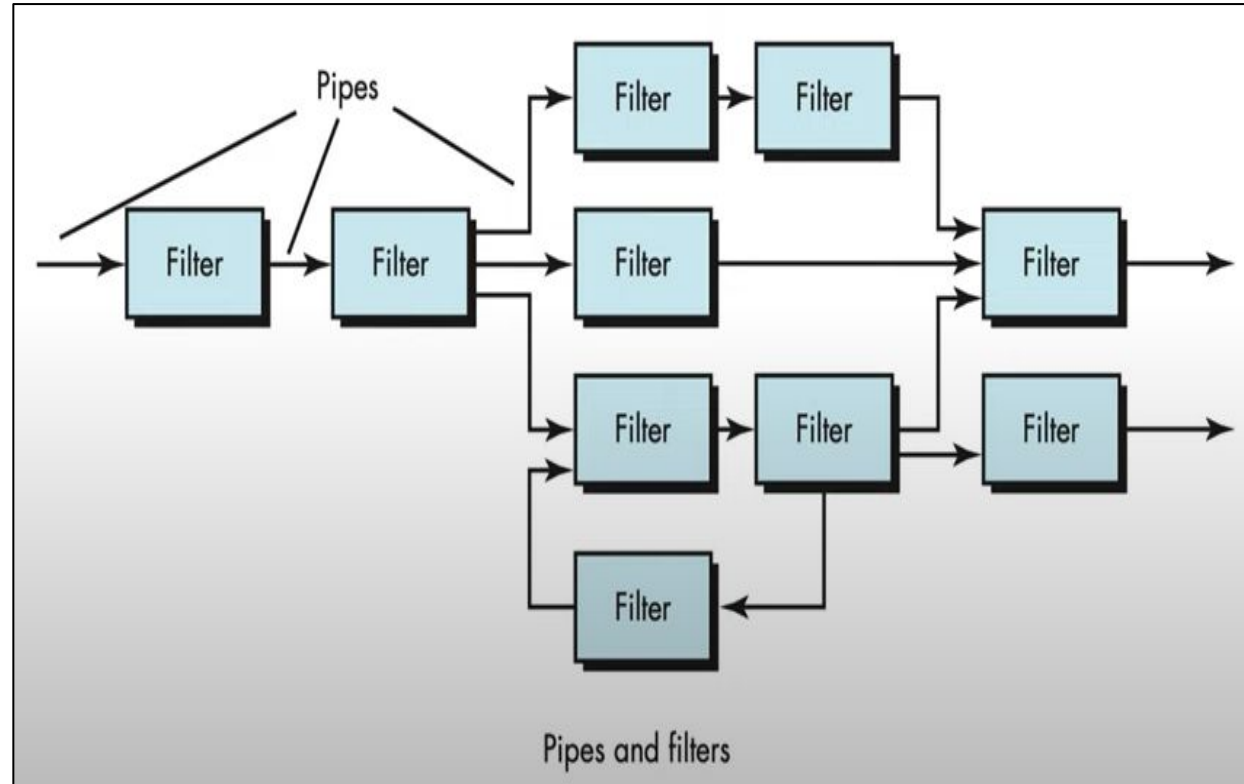
Sebuah arsitektur yang berpusat pada data. Sebuah tempat penyimpanan data berada pada pusat arsitektur, komponen-komponen yang menggunakan atau mengelola data tersebut berada di sekelilingnya



Pola Data Flow Architecture

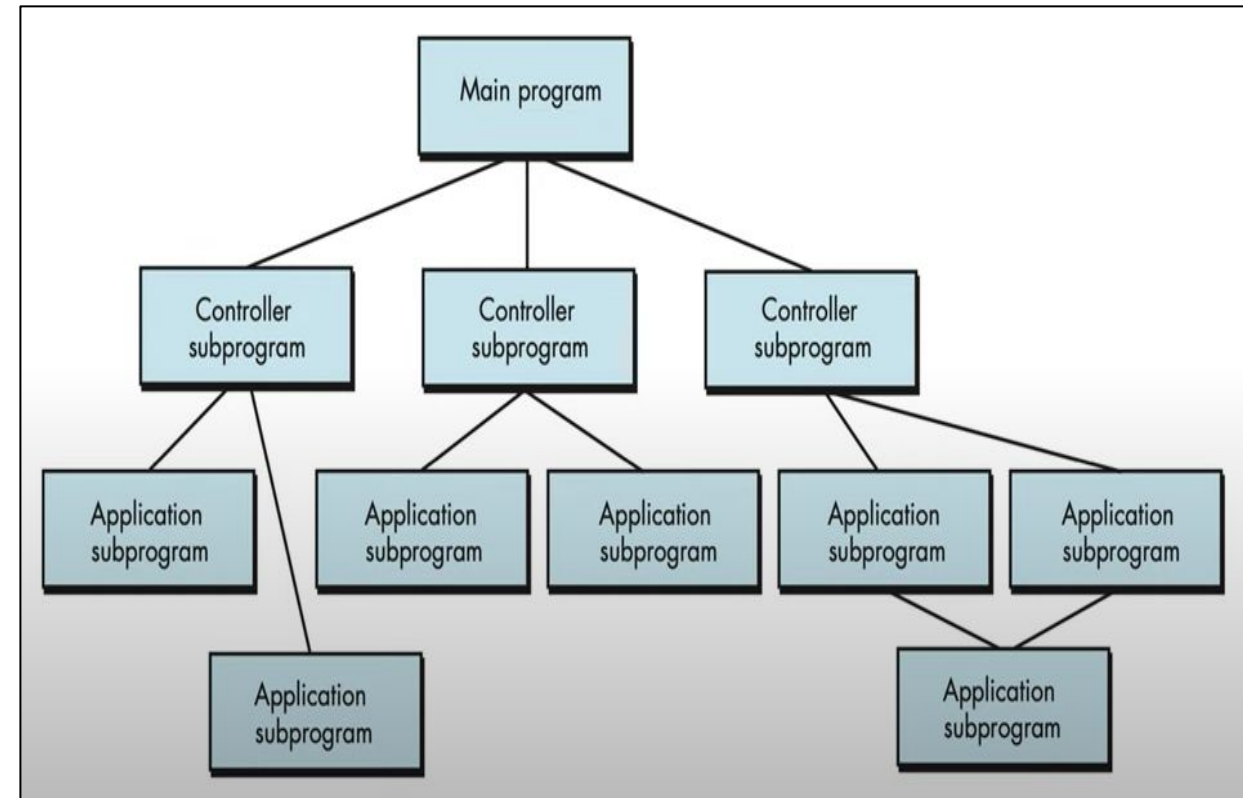
Sebuah arsitektur yang menggambarkan aliran data, biasanya diilustrasikan dengan Pipe and Filter.

Filter berguna untuk memproses data, sedangkan Pipe menunjukkan kemana data yang telah diproses diteruskan.



Pola Call and Return Architecture

Dengan menggambarkan struktur program yang relatif mudah untuk dimodifikasi dan diubah ukurannya.
Dilakukan dengan memecah fungsi ke dalam beberapa komponen atau sub komponen.

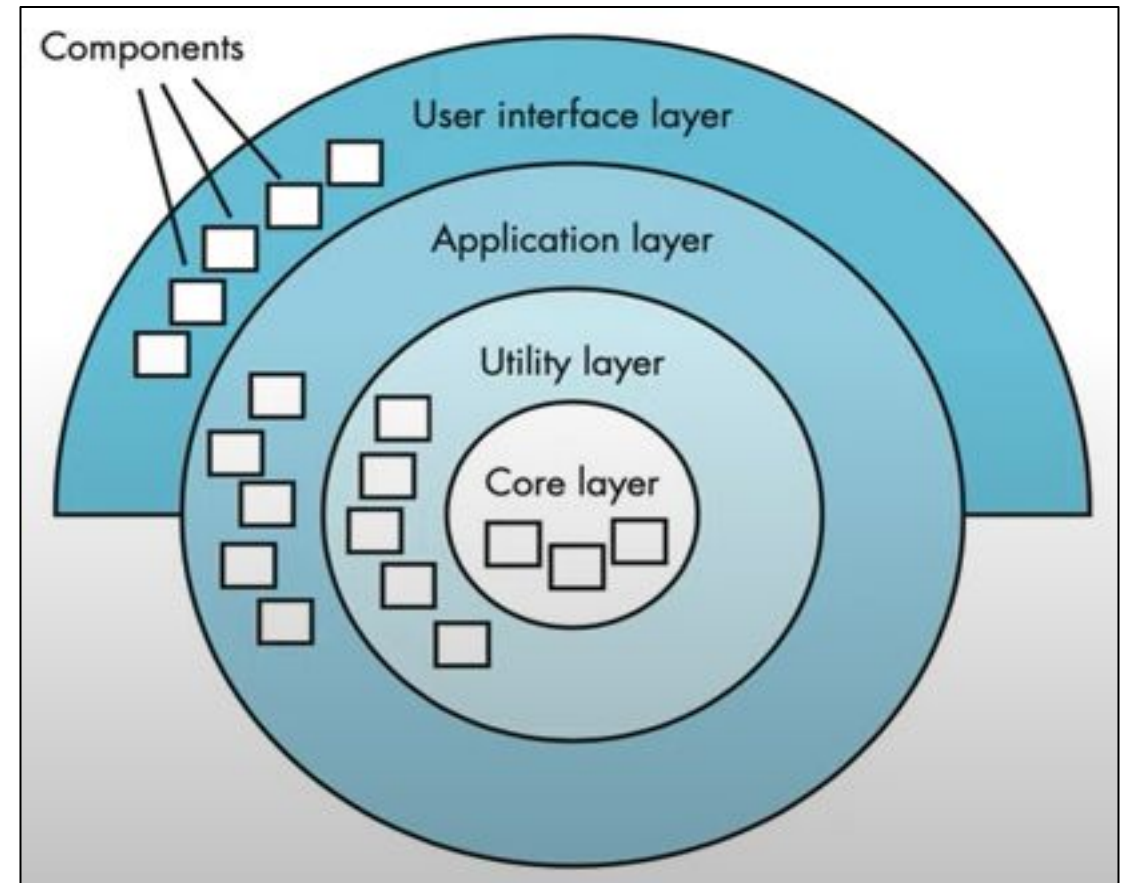


Pola *Layered Architecture*

Sebuah arsitektur yang membagi komponen ke dalam lapisan-lapisan. Biasanya setiap lapisan atau layer memiliki fungsi-fungsi tertentu.

Komponen pada layer terluar hingga layer paling dalam:

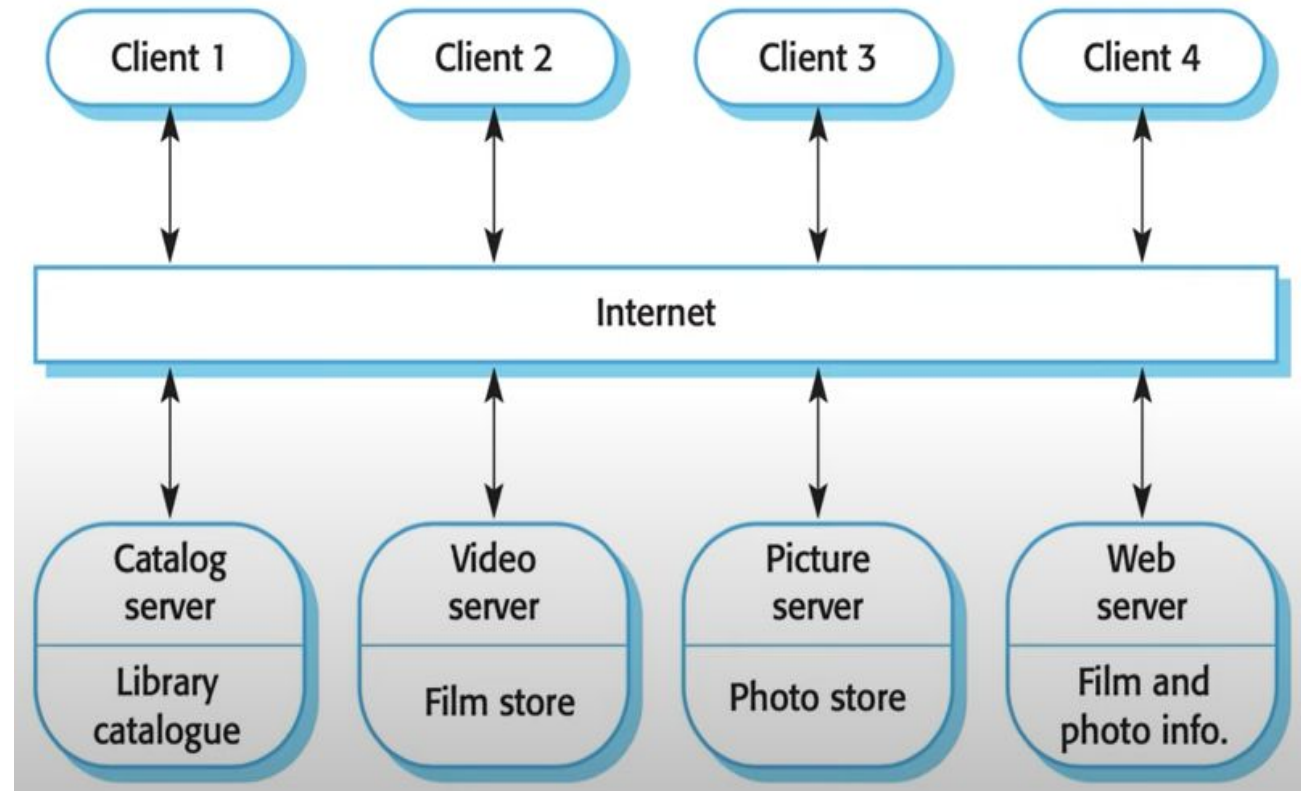
- User interface atau Atarmuka
- User interface mangement Authentication and Authorization
- Core business logic/application functionality system utilities
- System Support (OS, Database, etc.)



Pola *Client-Server Architectures*

Sistem digambarkan sebagai satu set servis yang diberikan oleh server terpisah.

Client adalah pengguna dari service tersebut.



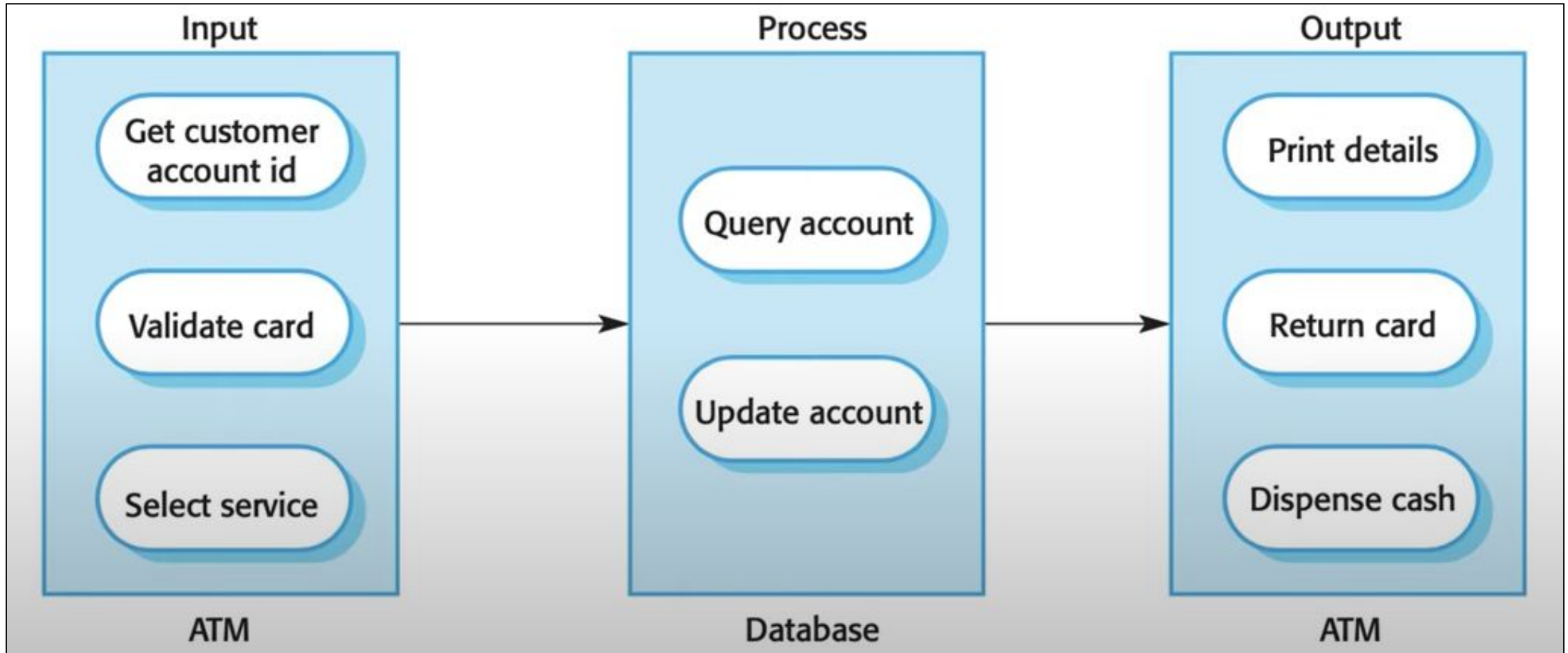
Application Architecture

Sebagai seorang software engineer, kita dapat memanfaatkan pola atau model arsitektur dalam berbagai cara:

1. Sebagai *starting point* dalam proses perancangan arsitektur.
2. Sebagai design checklist.
3. Sebagai dasar pembagian tugas tim.
4. Sebagai bahan pertimbangan untuk menggunakan kembali atau reuse komponen yg ada pada sistem
5. Sebagai vocabulary saat membicarakan aplikasi

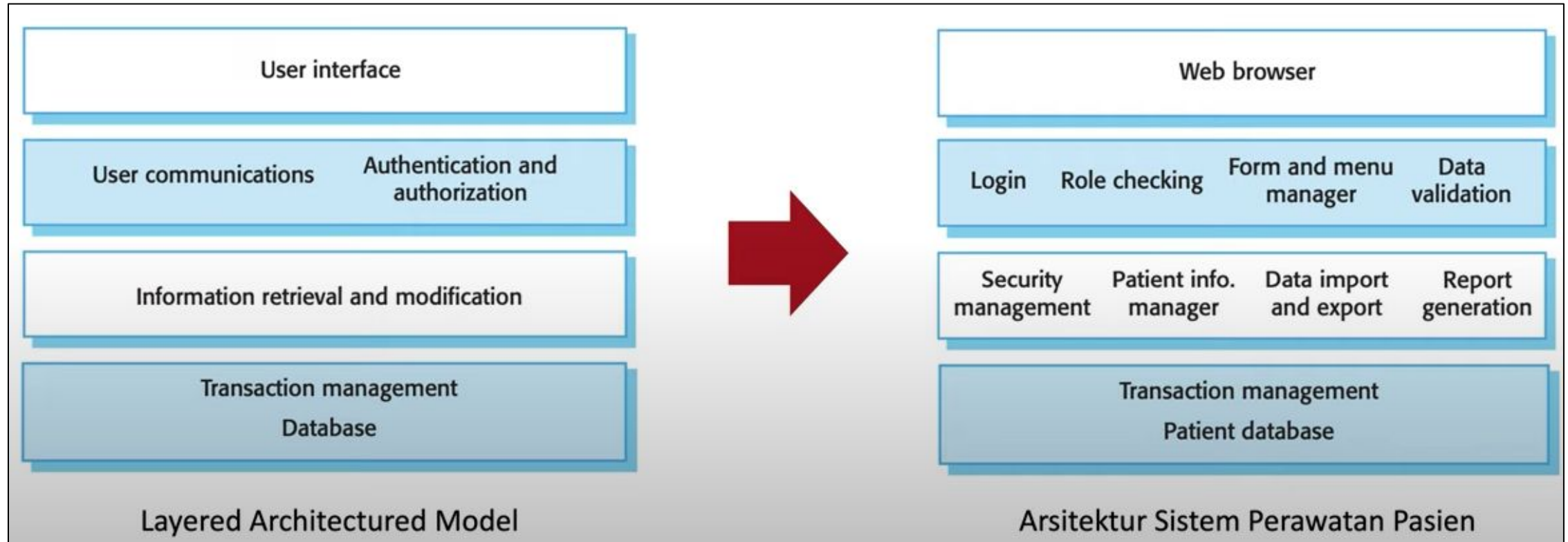
Contoh:

Arsitektur Mesin ATM (Automatic Teller Machine)



Contoh:

Arsitektur Sistem informasi perawatan pasien



Referensi

- Pressman, R.S. & Maxim, B.R. 2015. Software Engineer: A Practitioner's Approach 8th Edition. McGraw-Hill Education, New york.
- Sommerville, I. Software Engineering, 10th edition, Global Edition. Pearson Education Limited.

Any questions?