



Perancangan Arsitektur -Desain dan Analisis Sistem -

NISA'UL HAFIDHOH, M.T.

PRODI DIII – TEKNOLOGI INFORMASI

Arsitektur

Level sederhana: keseluruhan bentuk struktur fisik PL

Kenyataanya, arsitektur juga menggambarkan cara berbagai komponen PL diintegrasikan membentuk satu kesatuan.

Arsitektur PL adalah struktur yang terdiri atas komponen PL, properti komponen yang tampak dari luar dan hubungan antar komponen. [Bass, Clements, Kazman '03]

Pentingnya Arsitektur

Sejak program pertama dibagi menjadi modul, PL telah memiliki arsitektur dan programmer bertanggung jawab terhadap interaksi antar modul dan property global yang menyusunnya. [Shaw & Garlan '96]

3 kunci penting [Bass '03]

- Arsitektur PL menyediakan suatu representasi yang memfasilitasi komunikasi antar seluruh stakeholder.
- Arsitektur mendasari keputusan awal desain PL yang mempunyai pengaruh kuat dalam seluruh tahap rekayasa PL yang mengikutinya.
- Arsitektur mendasari model mengenai sistem distrukturkan dan bagaimana komponen bekerja bersama.

Elemen Arsitektur Perangkat Lunak

3 elemen yang saling berkaitan erat tentang arsitektur perangkat lunak *Erl (2009)*, yakni :

- ❖ ***Arsitektur teknologi***, yaitu desain fisik dari suatu perangkat lunak.
- ❖ ***Infrastruktur teknologi***, yaitu lingkungan pendukung yang termasuk di dalamnya perangkat keras dan perangkat lunak.
- ❖ ***Perangkat lunak itu sendiri***.

Gaya Arsitektur (Architectural Style)

Gaya arsitektur menggambarkan kategori sistem yang mencakup [Bass '03] :

1. Sekumpulan komponen (DB, modul) yang melakukan fungsi yang diperlukan sistem
2. Sekumpulan penghubung yang memungkinkan komunikasi, koordinasi dan kerjasama antar komponen
3. Batasan yang mendefinisikan komponen dapat diintegrasikan untuk membangun sistem
4. Model semantik yang memungkinkan perancang untuk memahami seluruh properti sistem

Macam Gaya Arsitektur

Arsitektur Data-Centered

Arsitektur Data Flow

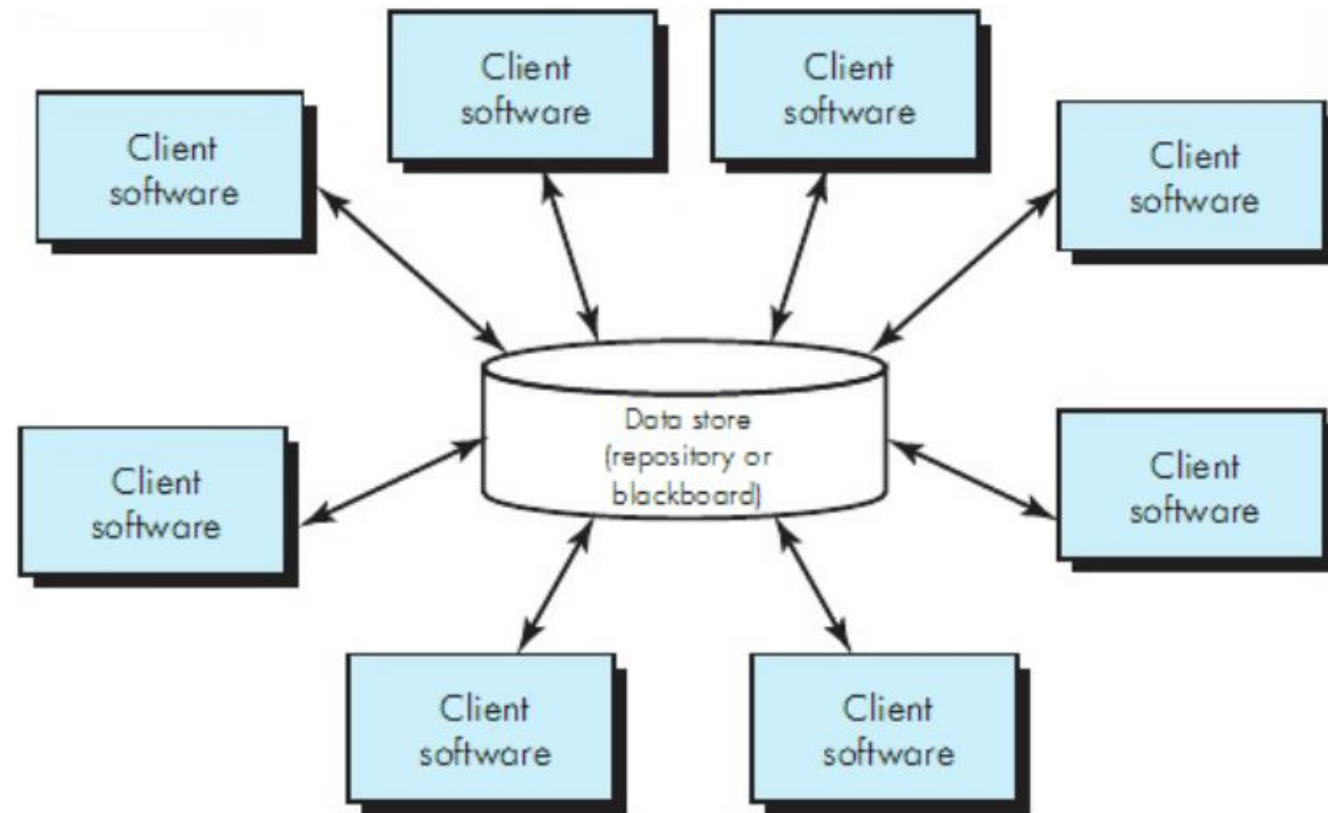
Arsitektur Call & Return

Arsitektur OO

Arsitektur berlayer

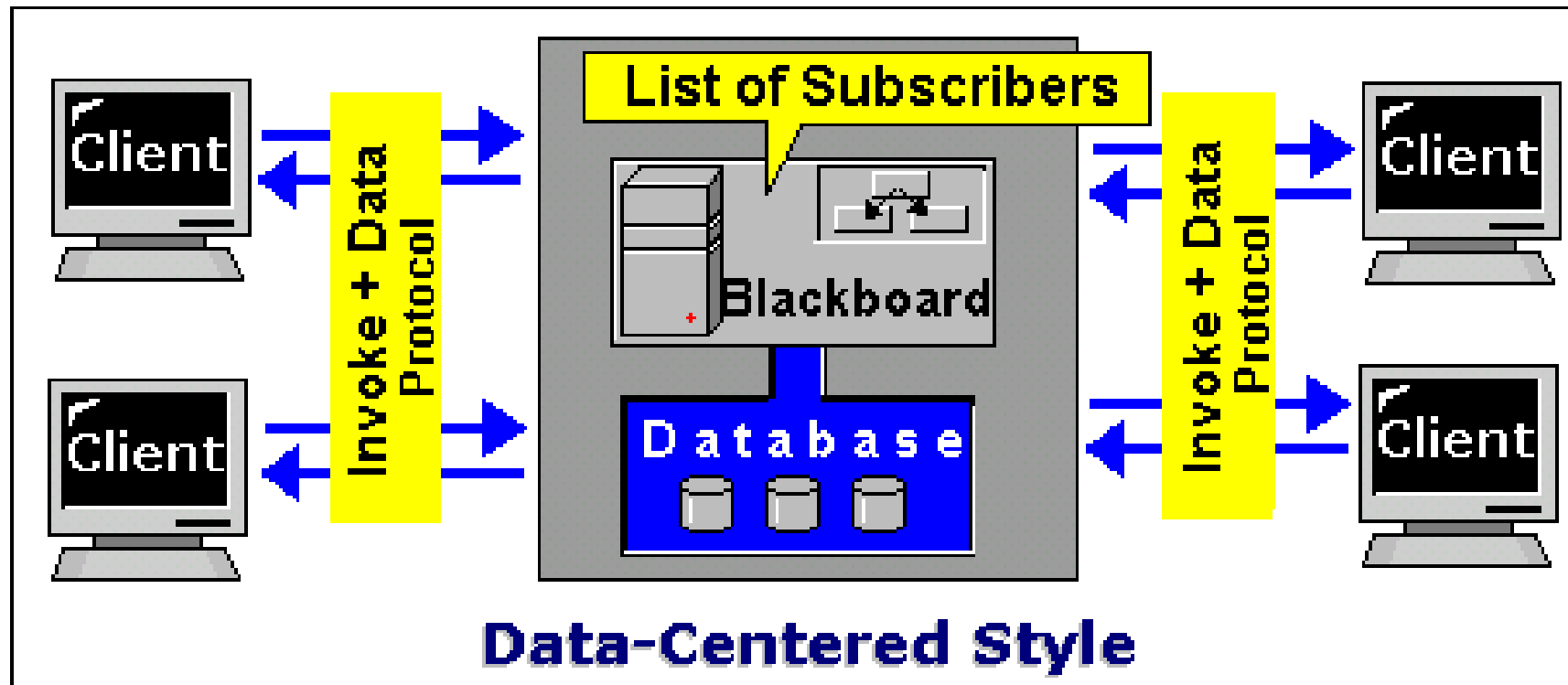
Arsitektur Data-Centered

Suatu data store (file / database) diletakan di tengah arsitektur & sering diakses oleh komponen lain yang mengubah, menambah atau menghapus data di dalam data store.



Contoh Arsitektur Data-Centered

Client dapat mengakses data dengan mekanisme blackboard



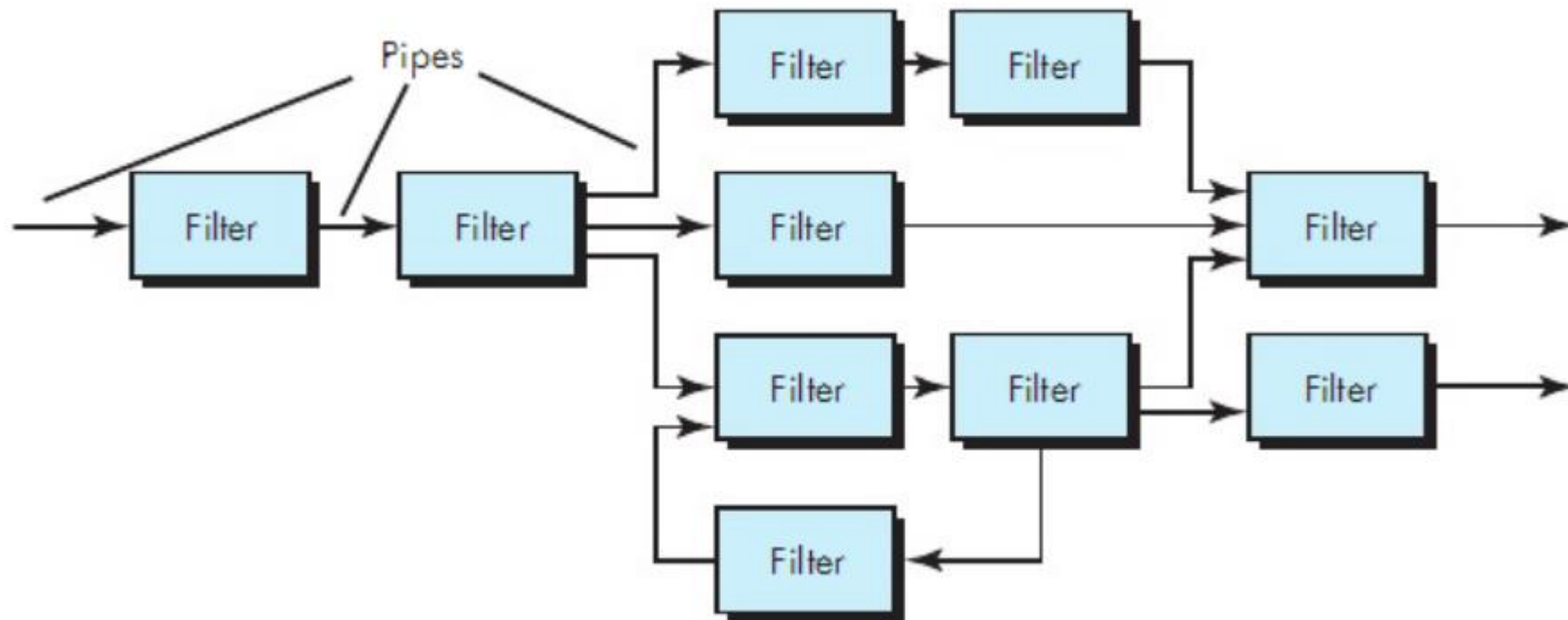
Arsitektur Data Flow

Gaya arsitektur ini diterapkan ketika data masukan ditransformasikan melalui serangkaian komponen komputasi atau manipulatif menjadi data keluaran.

Pola *pipe-and-filter* memiliki serangkaian komponen yang disebut *filter* dan dihubungkan dengan *pipe* yang mengirimkan data antar komponen.

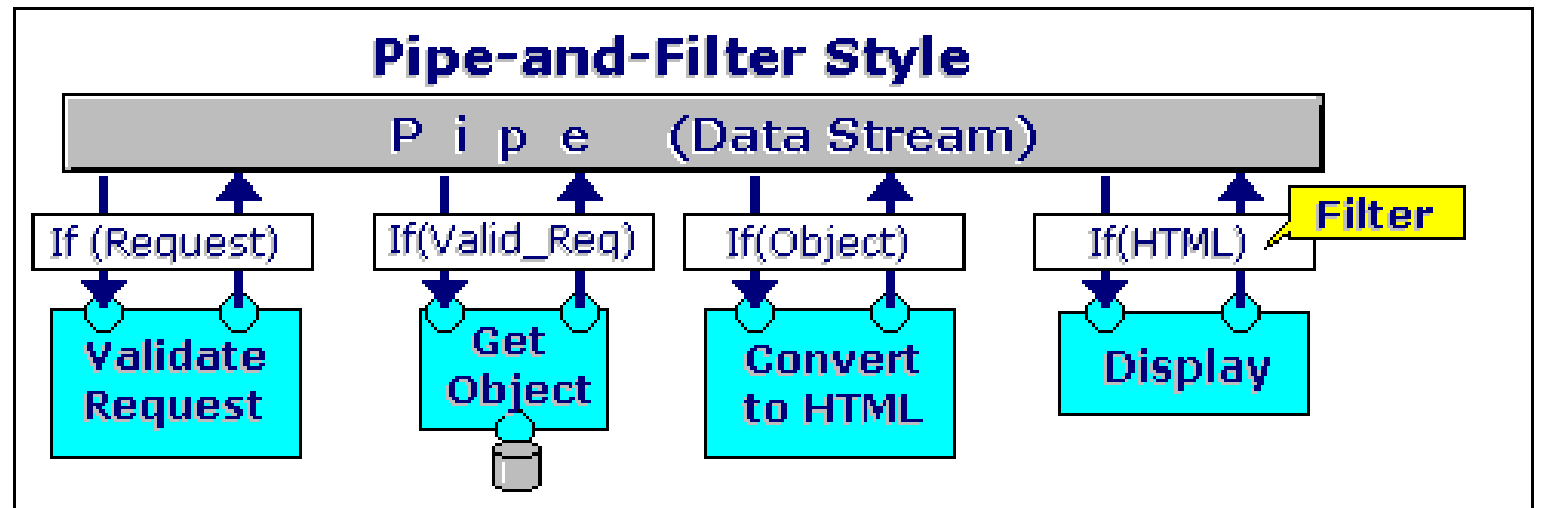
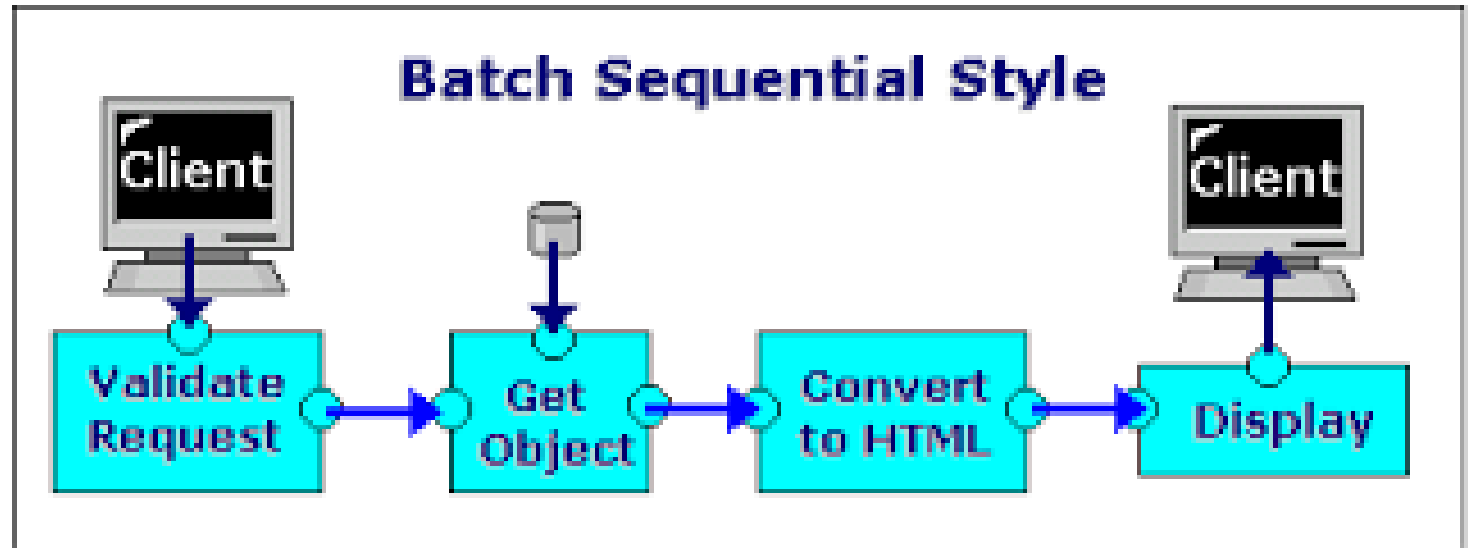
Jika data mengalir dalam satu aliran transformasi disebut *batch sequential*. Struktur ini menerima data dan menjalankan serangkaian komponen berurutan (*filter*) untuk merubahnya.

Pola *Pipe-and-filter*



Tiap filter bekerja independen dengan data masukan dari suatu form dan menghasilkan data keluaran (untuk filter berikutnya) dalam form tertentu.

Contoh *Pipe-and-filter*



Arsitektur Call & Return

Memungkinkan struktur program mudah dimodifikasi dan berubah

Ada dua jenis:

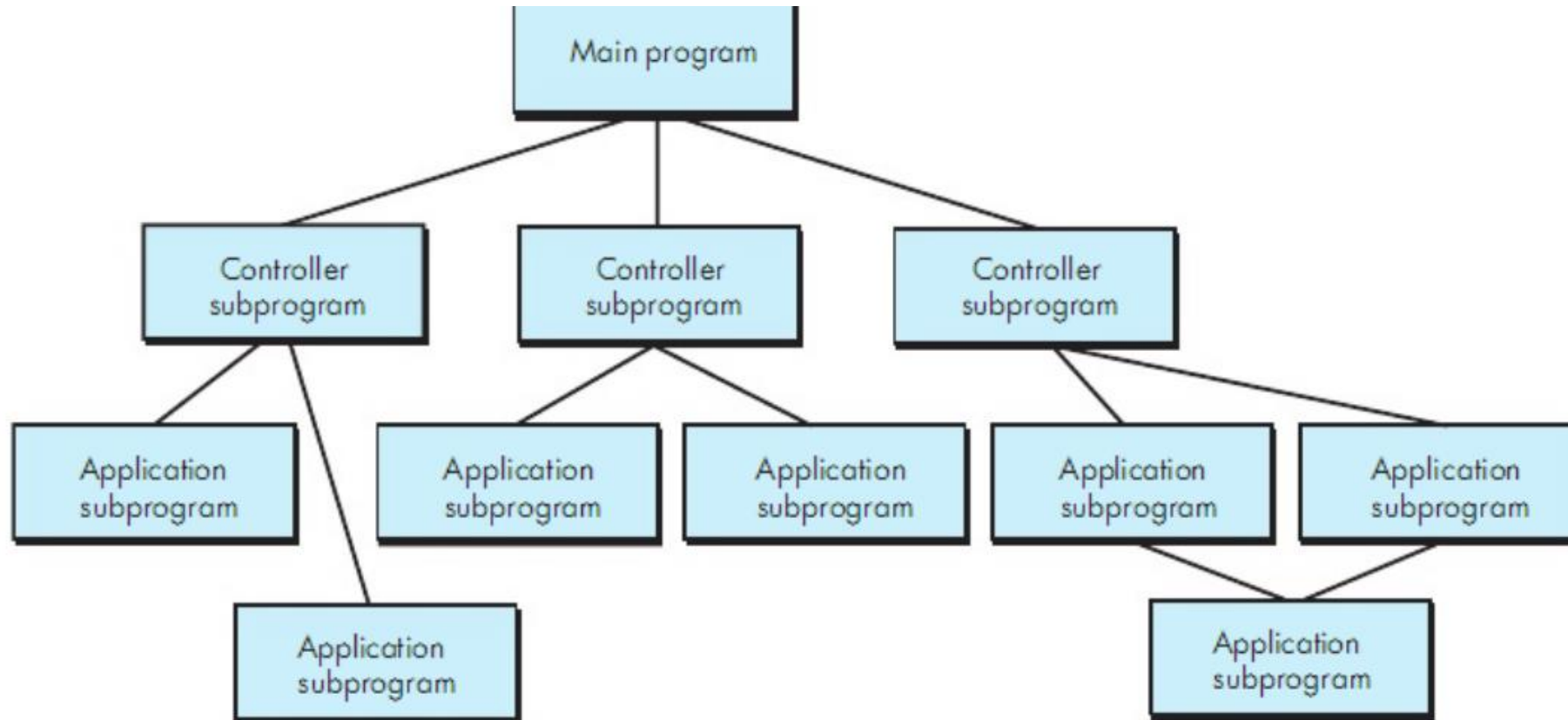
- Main program / subprogram arsitektur:

Struktur program klasik yang membagi fungsi dalam hirarki kontrol di mana “main program” memanggil sejumlah komponen program, & dapat memanggil komponen lainnya

- Remote procedure call arsitektur:

Komponen dari program utama / subprogram didistribusikan di beberapa komputer pada suatu jaringan

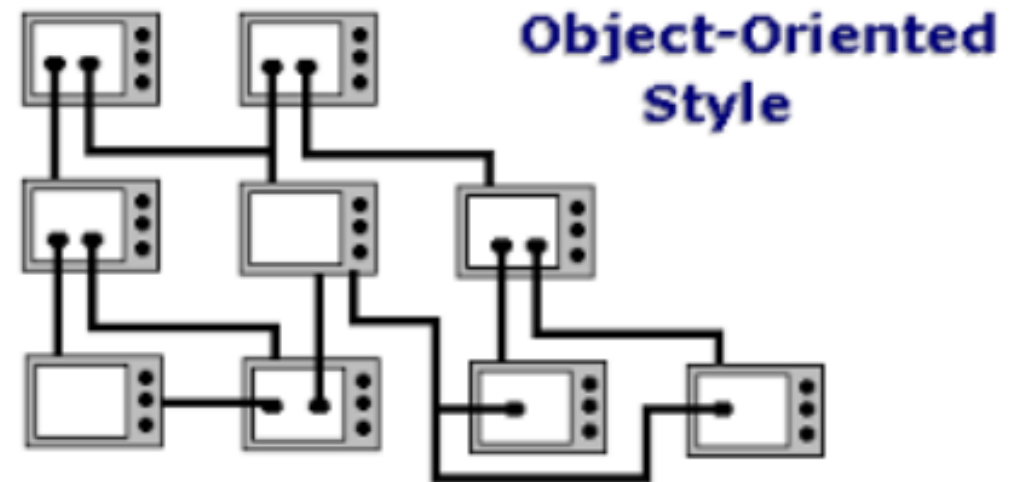
Main program / subprogram



Arsitektur Object Oriented

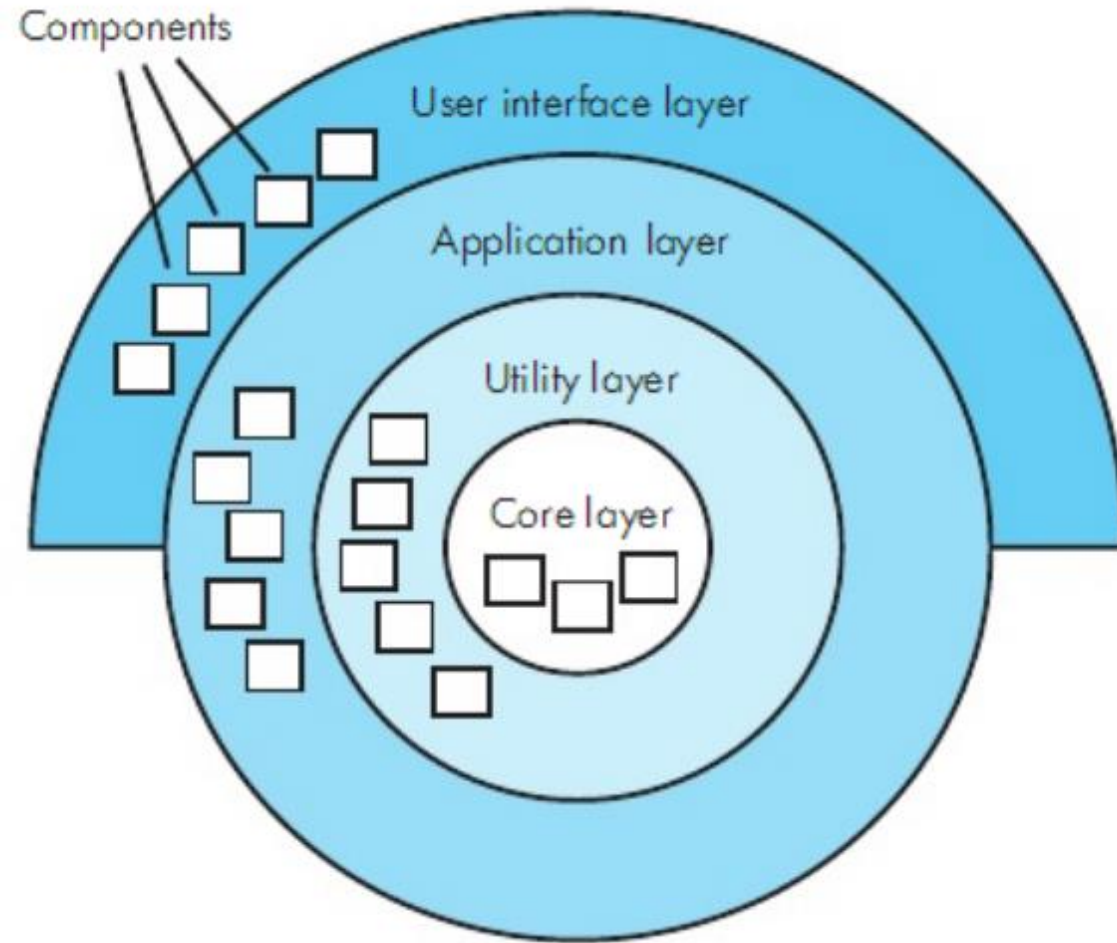
Komponen dari sistem mengenkapsulasi data dan operasi yang harus diterapkan untuk memanipulasi data.

Komunikasi dan koordinasi antara komponen dilakukan melalui message-passing.



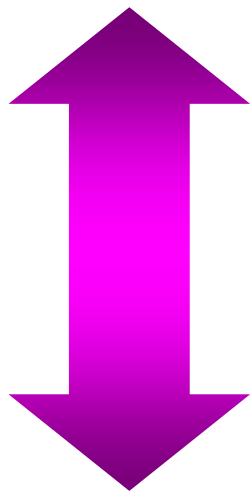
Arsitektur Berlayer

Sejumlah layer didefinisikan dalam level abstraksi berbeda. Layer teratas terkait dengan antarmuka, hingga terdalam berhubungan dengan mesin / implementasi.

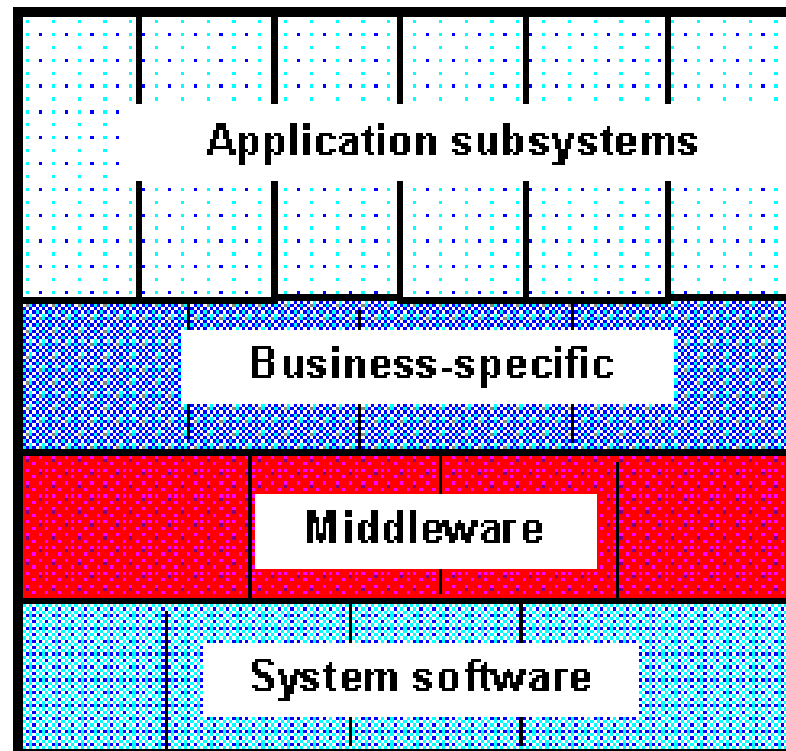


Contoh Arsitektur Berlayer [1]

Specific
functionality



General
functionality



Distinct application subsystem that make up an application - contains the value adding software developed by the organization.

Business specific - contains a number of reusable subsystems specific to the type of business.

Middleware - offers subsystems for utility classes and platform-independent services for distributed object computing in heterogeneous environments and so on.

System software - contains the software for the actual infrastructure such as operating systems, interfaces to specific hardware, device drivers and so on.

Contoh Arsitektur Berlayer [2]

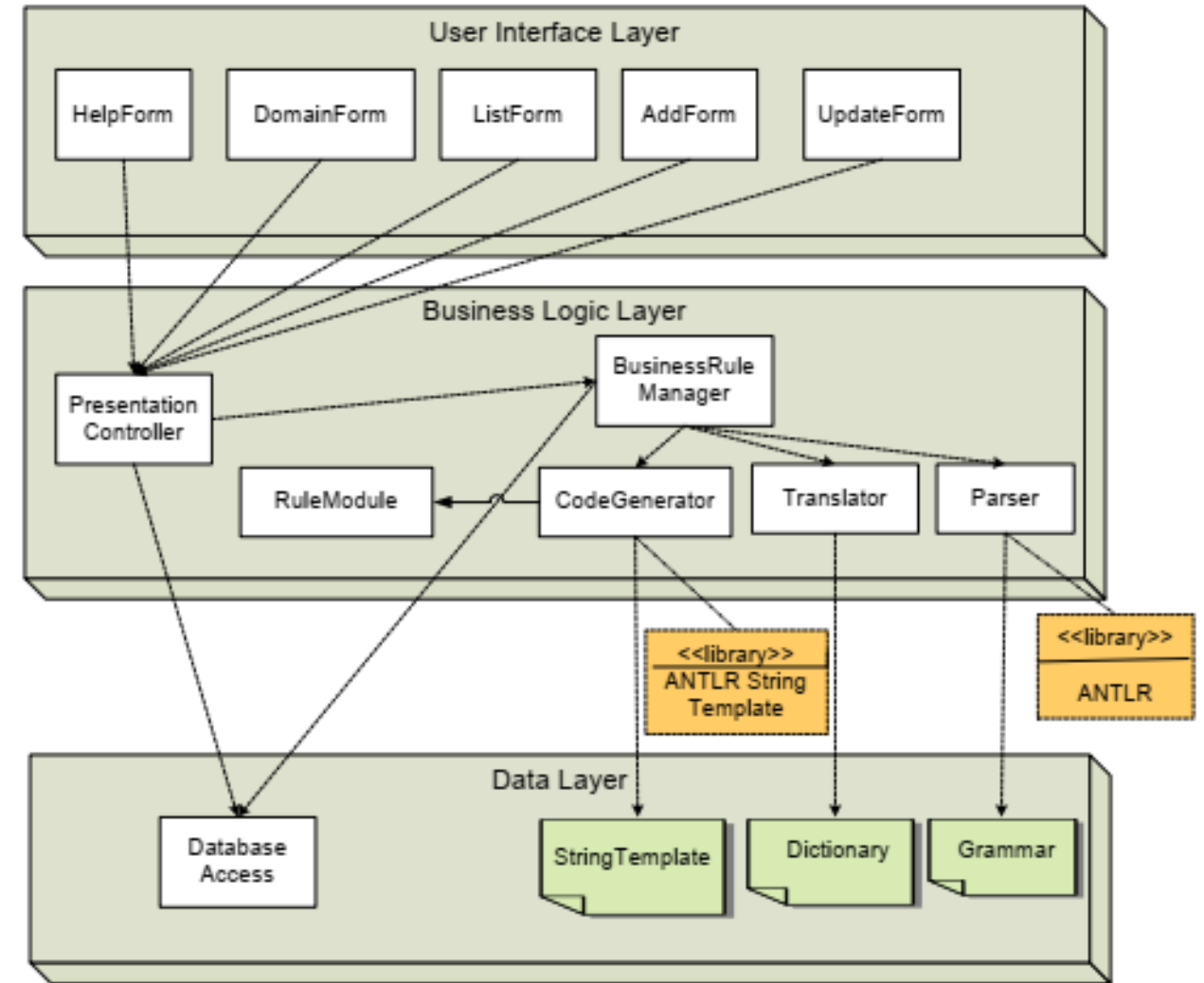


Figure 7. Business Rules Generator Achitecture

Hafidhoh (2015)