Kuliah

Rekayasa Perangkat Lunak (Software Engineering)

Bagian 2

Software Engineering: A Practitioner's Approach, 6/e

Bab 10
Desain Arsitektur

Kenapa Arsitektur ?

Arsitektur bukanlah PL operasional, namun dia merupakan representasi yang memungkinkan pengembang PL untuk : (1)menganalisa efektivitas desain dalam memenuhi kebutuhan,

(2) Mengetahui alternatif2x arsitektur pada keadaan dimana membuat perubahan desain masih relatif lebih mudah, dan

(3) Mengurangi resiko terkait dengan konstruksi PL.

Mengapa Arsitektur Penting?

- Representasi dari arsitektur PL adalah enabler bagi komunikasi antar pihak (stakeholder) yang tertarik dengan pengembangan sistem berbasis komputer.
- Arsitketur menyoroti keputusan desain awal yang akan mempunyai pengaruh yang sangat besar pada pekerjaan RPL yang mengikutinya, dan keberhasilan pada entitas sistem operasional.
- Arsitektur membangun model yang relatif kecil dan mudah digenggam secara intelektual tentang bagaimana sistem distrukturkan dan bagaimana komponen2x bekerja sama [BAS03].

Desain Data

- Pada level arsitektur …
  - Desain satu atau lebih database untuk mendukung arsitektur aplikasi
  - Desain method untuk ‘mining’ isi dari berbagai database
    - Navigasi melalui database2x yang ada dalam usaha untuk mengambil informasi level bisnis yang sesuai
    - Desain sebuah data warehouse—sebuah database besar, independen yang mempunyai akses pada data yang disipan dalam database yang melayani sekelompok aplikasi yang dibutuhkan bisnis

Desain Data

- Pada level komponen …
  - Mengambil objek2x data dan mengembangkan satu set abstraksi data
  - Implementasi atribut2x objek data sebagai satu atau lebih struktur data
  - review struktur data untuk memastikan bahwa relasi yang tepat sudah dibuat
  - Sederhanakan struktur data sesuai dengan kebutuhan

Desain Data—Level Komponen

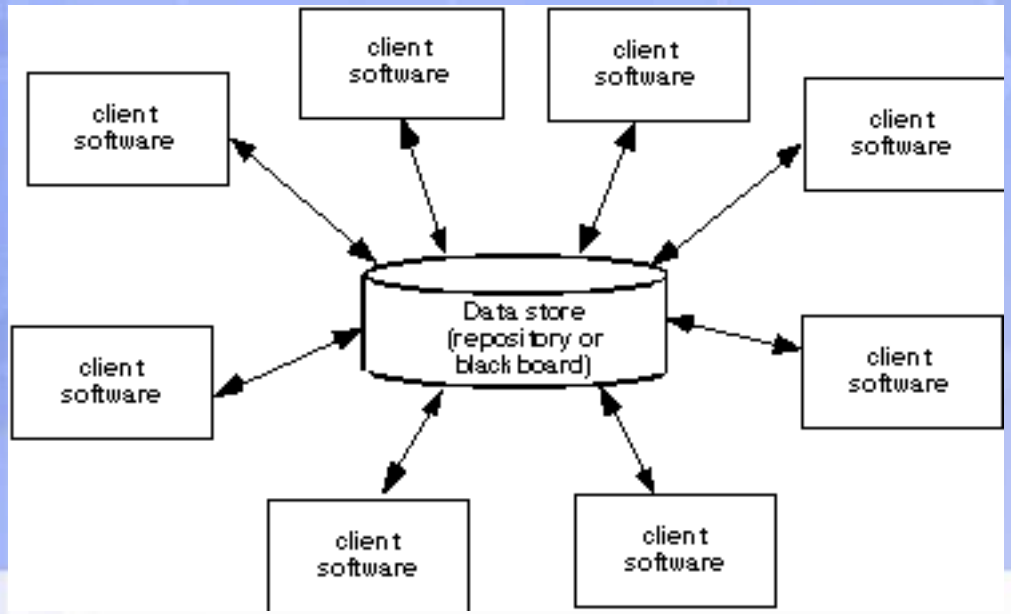
1. Prinsip2x analisis semantik yang diterapkan pada fungsi dan perilaku harus juga dapat berjalan pada data.
2. Seluruh struktur data dan operasi yang akan dilakukan harus dapat diidentifikasi.
3. Sebuah data dictionary harus dibuat dan digunakan untuk menentukan desain program dan data.
4. Keputusan desain data level rendah harus ditunda hingga akhir proses desain.
5. Representasi struktur dara harus diketahui oleh modul yang menggunakannya langsung dalam struktur tersebut (enkapsulasi).
6. Sebuah pustaka struktur data dan operasi yang memungkinkan untuk diterapkan harus dikembangkan.
7. Desain PL dan bahasa pemrograman harus mendukung spesifikasi dan realisasi dari tipe data abstrak.

Ragam Gaya Arsitektur

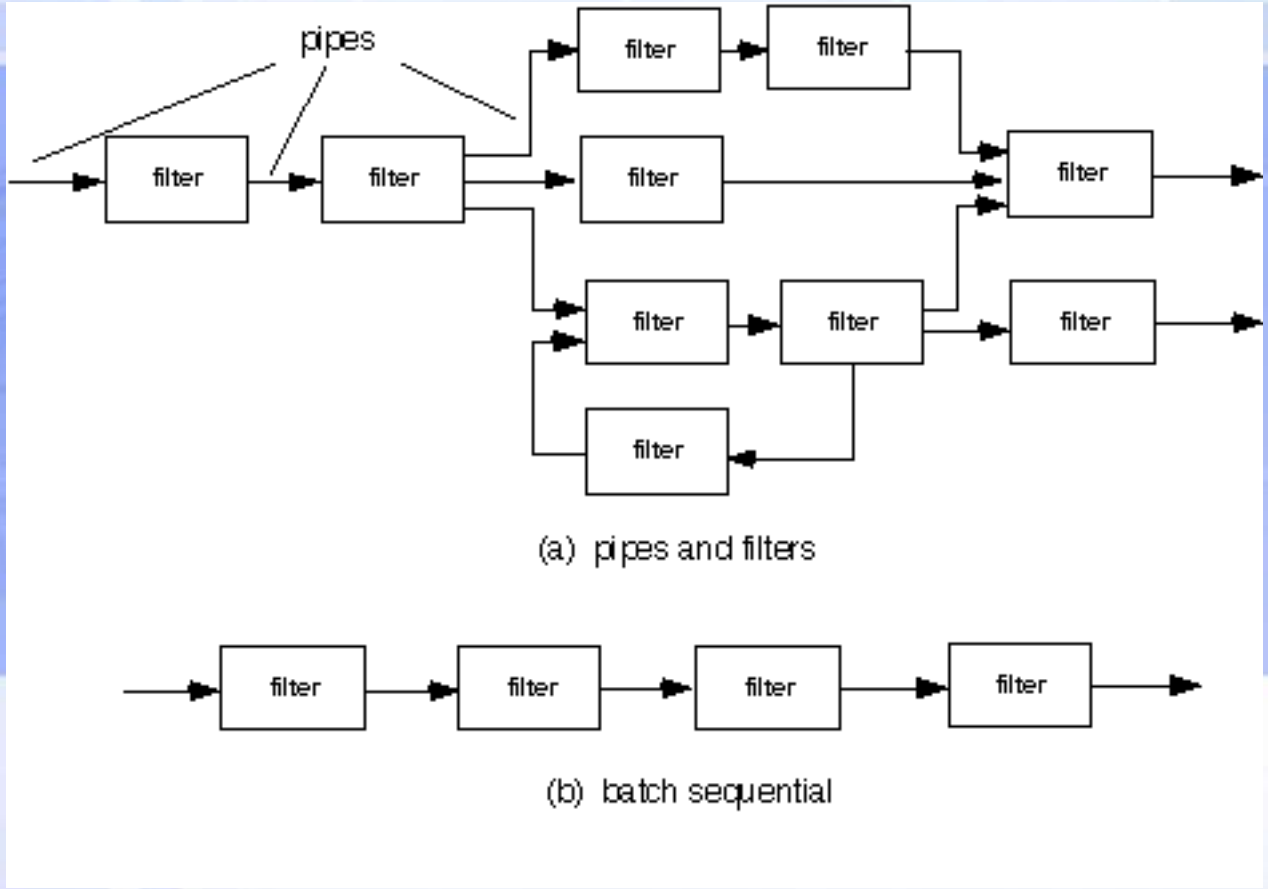
Masing2x menggambarkan kategori sistem yang menunjukkan : (1) a sekumpulan komponen (mis database, modul komputasi) yang menunjukkan fungsi yan dibutuhkan sistem, (2) sekumpulan connectors yang memungkinkan komunikasi, koordinasi dan kerjasama antar komponen components, (3) batasan yang menentukan bagaimana komponen dapat diintegrasikan untuk membentuk sistem, dan (4) smodel semantik yang memungkinkan desainer untk memahami properti keseluruhan dari sistem dengan menganlisai properti dalam bagian2x di dalamnya.

- 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