Software Architecture

Budi Irawan


facebook.com/deerawan

@masbugan

blog.budiirawan.com

Kenapa Harus Pusing Mikirin Software Architecture?

Arsitektur yang baik adalah kunci dari suksesnya suatu software

Apa itu Software Architecture?

Rancangan sistem dalam bentuk komponen-komponen

serta bagaimana komponen2 tersebut berinteraksi

Contoh Komponen Rumah

Pintu

Jendela

Genteng

Tembok

Lantai

Perabotan

Tujuan dari Software Architecture

- Alat komunikasi antar stakeholders
- Membantu dalam proses desain software
- Mengidentifikasi komponen software serta interaksinya dalam sistem
- Untuk digunakan kembali (jika bidangnya sama misal: rumah sakit)

Yg Mempengaruhi Arsitektur

- Lingkungan development misalnya penggunaan hardware, integrasi dengan software lain
- Kemampuan dari sang arsitek

Kaitan Software Requirement dengan Software Architecture

Saya mau buat fitur login yang terintegrasi dengan Facebook dan Twitter

User

Oh! Berarti nanti desain arsitekturnya harus ada hubungan dengan Facebook dan Twitter

Team

Saya mau buat fitur login terintegrasi dengan forum Kaskus sehingga saya login bisa pakai loginnya Kaskus

User

Hmm, susah nih karena arsitektur si kaskus tidak bisa diintegrasikan dalam fitur login. Saya mau nego ulang fitur ini.

Team

Dalam requirement engineering, kita harus selalu memikirkan apakah arsitekturnya dapat dibuat apa tidak

Architectural Design

Architectural Design merupakan sekumpulan **keputusan** terkait desain

Sang Arsitek Ridwan Kamil (RK)

Rancangan Denah Rumah RK

Tempat tidur saya taruh dekat taman biar nanti ada udara segar yang masuk

Kamar Mandi saya taruh di tengah2 dua kamar untuk nanti memudahkan ketika ke kamar mandi

Keputusan2 Sang Arsitek RK

Jika kita ingin mendokumentasikan arsitektur berarti kita **mendokumentasikan keputusan2** tersebut. Apa alasan dibalik keputusan tsb.

Biasanya dokumentasi keputusan desain jarang dilakukan

3 Alasan Keputusan Desain Tidak Terdokumentasi

- **Keputusan bersifat implisit:** biasanya karena pengalaman arsitek
- **Keputusan bersifat eksplisit tapi tidak terdokumentasi:** biasanya karena mepet waktu
- **Keputusan eksplisit dan tidak terdokumentasi eksplisit:** biasanya karena si arsitek cari aman di posisinya biar ga kena pecat

Cara Dokumentasi Keputusan Desain

Elemen	Deskripsi
Issues	Isu yang sedang ingin diselesaikan
Decision	Keputusan yang diambil
Status	Pending atau disetujui
Assumptions	Asumsi terkait environment pengembangan
Alternatives	Alternatif lain dari keputusan yang diambil
Rationale	Alasan kuat dari keputusan yang diambil
Implications	Implikasi dari keputusan yang diambil
Notes	Catatan jika ada

Contoh Dokumentasi Keputusan Desain

Elemen	Deskripsi
Issues	Sistem yang dibuat harus menggunakan database open source
Decision	Menggunakan MySQL
Status	Disetujui
Assumptions	Sistem tidak diakses oleh puluhan juta pengguna
Alternatives	PostgreSQL namun banyak hosting yang tidak support
Rationale	Programmer sudah familiar, banyak hosting yang mendukung dan lebih mudah digunakan
Implications	Kinerja MySQL harus diuji coba sekuat apa untuk sistem ini
Notes	Tidak ada

Architectural View

Software Architecture digunakan untuk komunikasi diantara para stakeholders

Siapa Stakeholder?

End User

Arsitek itu Sendiri

Pengguna sistem lain

manager

Pakar security

programmer

Tiap Stakeholder memiliki kepentingan yang berbeda

Contoh Perbedaan

- **Programmer** ingin tahu dimana dia bisa implementasi suatu fitur
- **Pakar security** ingin tahu keterkaitan sistem ini dengan jaringan gimana

Butuh untuk membuat tampilan gambar arsitektur yang berbeda untuk stakeholder yang berbeda. Ini yang disebut **Architectural View**

Elemen Dasar Software Architecture (IEEE standard 1471)

1. **Stakeholder:** pihak yang berkepentingan dengan sistem
2. **View:** representasi sistem dari perspektif kepentingan stakeholder
3. **Viewpoint:** teknik untuk membangun view

"4+1 Model" Viewpoints

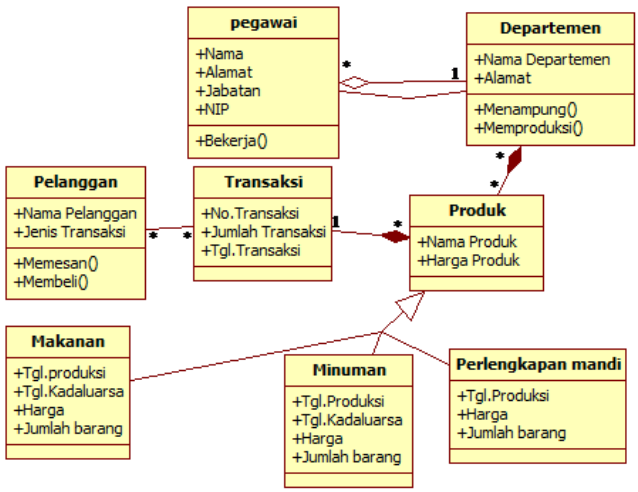
1. Logical viewpoint
2. Development viewpoint
3. Process viewpoint
4. Physical viewpoint
5. Scenarios

1) Logical Viewpoint

Terkait dengan fungsionalitas yang diberikan sistem ke pengguna

Logical view direprentasikan dengan diagram UML seperti Class Diagrams, Sequence Diagram, dsb

Contoh Class Diagram

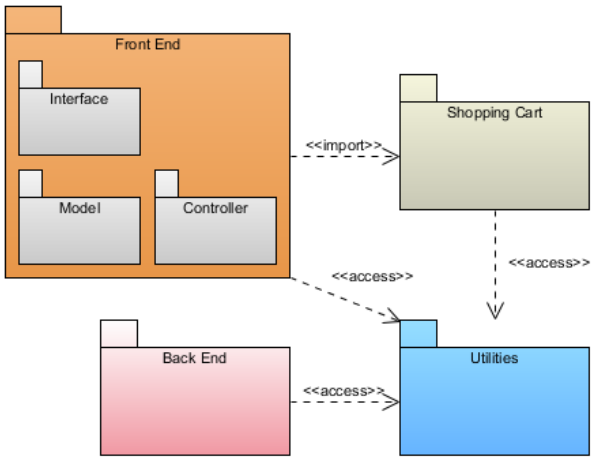


2) Development Viewpoint

Ilustrasi sistem dari perspektif programmer dan terkait dengan software management

Development view direprentasikan dengan diagram UML seperti Package Diagram atau Component Diagram

Contoh Package Diagram

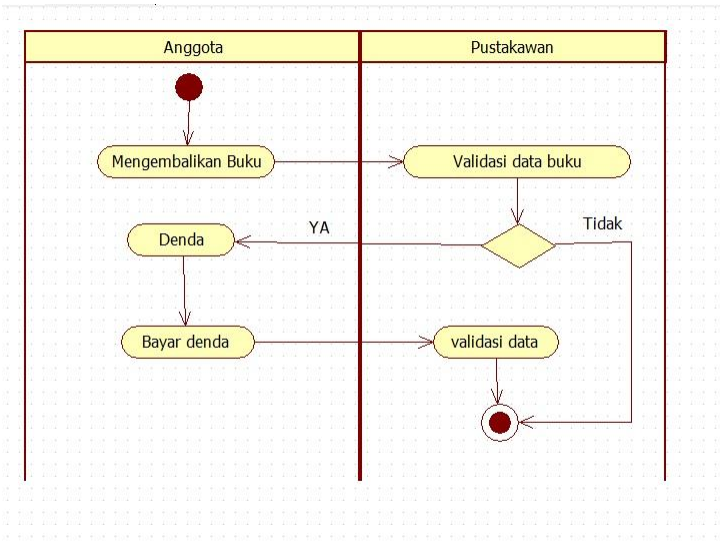


3) Process Viewpoint

Menggambarkan aspek dinamis dari sistem meliputi proses dan interaksinya

Process view direprentasikan dengan diagram UML seperti Activity Diagram

Contoh Activity Diagram

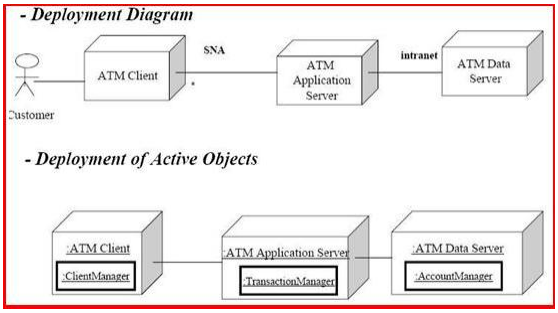


4) Physical Viewpoint

Menggambarkan topologi komponen dari software dalam physical layer (bentuk fisik seperti server dsb)

Physical view direprentasikan dengan diagram UML seperti Deployment Diagram

Contoh Deployment Diagram

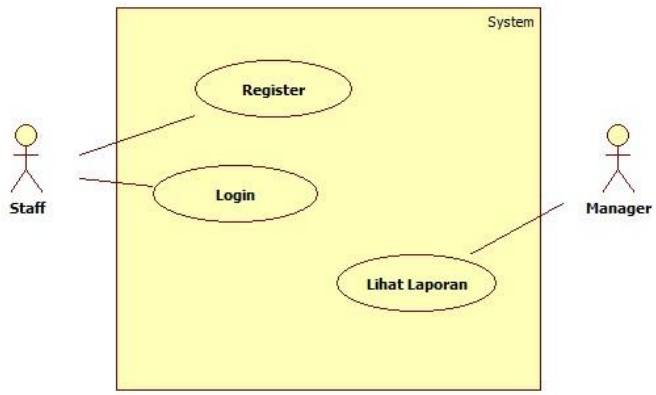


5) Scenarios

Menggambarkan interaksi antara proses dengan objek (objek bisa user, sistem, dsb)

Scenario direprentasikan dengan diagram UML seperti Use Case Diagram

Contoh Use Case Diagram



Class Diagram untuk Stakeholder siapa?

programmer

Use Case Diagram untuk Stakeholder siapa?

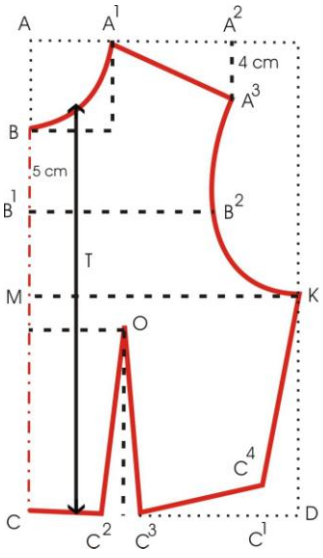
End user

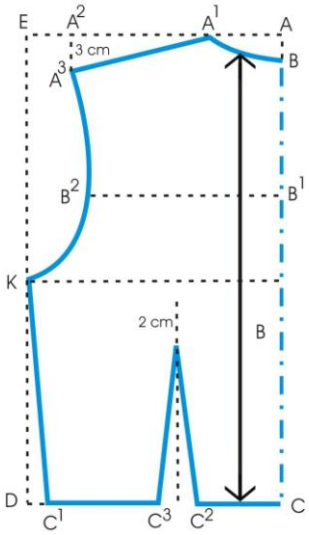
Architectural Pattern/Styles

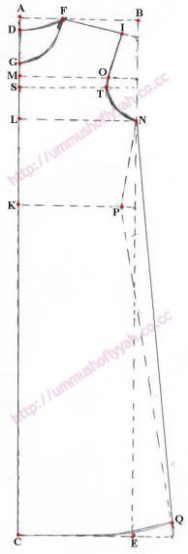
Architecture Pattern

Pola arsitektur yang menjadi solusi untuk memecahkan masalah desain software

Analogi Pola Baju







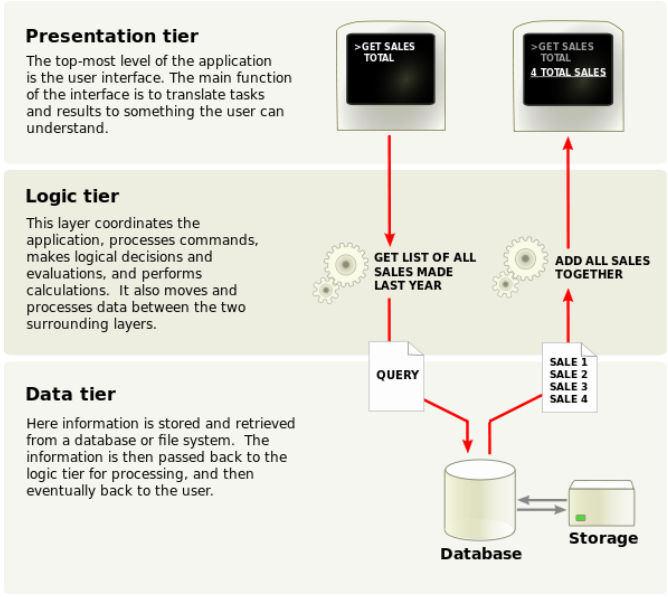
Contoh Architecture Pattern

1. Three-tier Architecture
2. MVC architecture
3. dsb….total ada 21 arsitektur bo!

1) Three-tier Architecture

Memperlihatkan adanya pemisahan (separation) modules

Diagram Three-tier Architecture



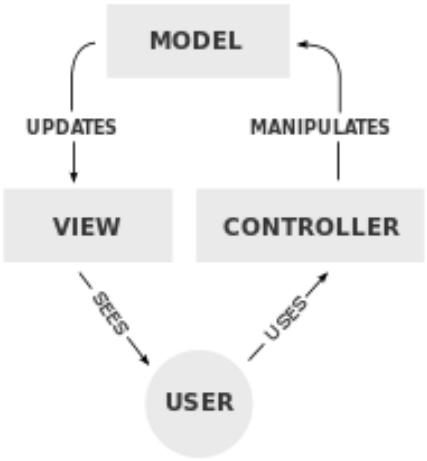
Three Tier

1. Presentation tier: ngurusin tampilan
2. Logic tier: ngurusin kalkulasi, pemrosesan data
3. Data tier: ngambil dan nyimpan data

2) MVC Architecture

Memperlihatkan adanya pemisahan (separation) interaksi

Diagram MVC Architecture



MVC

1. Model: ngambil dan nyimpan data, logika aplikasi
2. Controller: penghubung antara model dan view
3. View: menampilkan tampilan ke user

What we have learned?

What we Have Learned?

- Apa itu Software Architecture? Kaitannya dengan Rumah apa?
- 4+1 Viewpoints?
- Activity Diagram masuk ke viewpoint apa?
- Contoh architectural pattern

Review Mission

The Mission

- Misi Quiz Kelompok Use Case Diagram
- Misi 4

Your Mission

The Mission

- James Bond butuh persiapan misi penting yaitu misi UTS
- Misi UTS hanya one time mission

Thank You