Desain Software

Arna Fariza
PENS

Materi

- Apakah desain software itu?
- Apakah modularisasi itu?
- Model

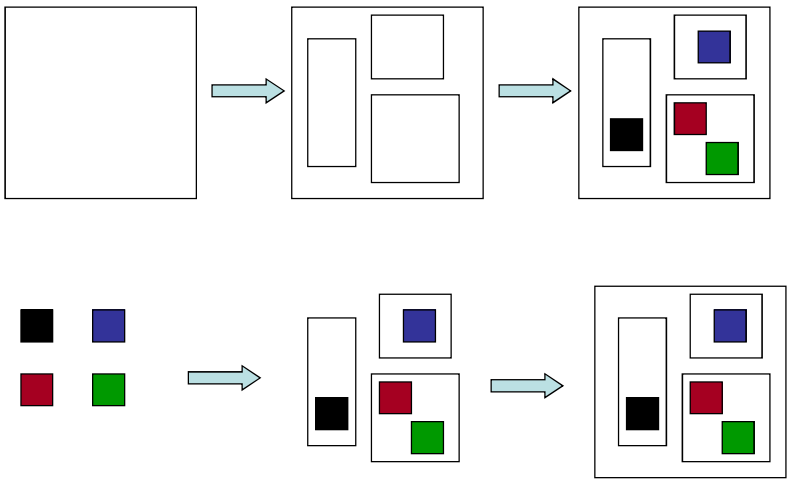
Apakah Desain Software itu?

- ***Desain*** adalah proses mengubah persyaratan sistem ke dalam produk yang lengkap
- ***Desain Software*** adalah output dari proses yang
  - Dekomposisi sistem dalam beberapa baian
  - Menentukan tanggung jawab setiap bagian
  - Memastikan bagian-bagian tersebut dapat mencapai tujuan global
- Desain terdiri dari aktifitas dan hasil dari aktifitas
- Desain adalah aktifitas iteratif
  - Mulai dari titik inisial
  - Berkembang dan beriterasi dengan menambahkan fitur/kapabilitas
  - Mencapai state yang tepat berdasarkan
    - Obyektif/tujuan desain
    - Input/feedback dari konsumen

Apakah Desain Software itu?

- Software dikembangkan secara terinci sampai cukup detail untuk mendukung implementasi
  - Sistem dibagi ke dalam modul- modul (komponen, paket, class dll)
  - Dekomposisi Top-Down
    - Modul besar (komponen, paket, class) didekomposisi ke dalam bagian yang lebih kecil
    - Bagian yang lebih kecil tsb mengimplementasikan bagian yang lebih besar
    - Bagian yang lebih besar dikomposisi dari bagian yang lebih kecil
  - Dekomposisi Bottom-Up
    - Modul level yang lebih rendah didefinisikan terlebih dahulu
    - Modul-modul tsb dikombinasikan untuk membentuk modul yang lebih besar
- Baik TD atau BU dapat mencapai hasil yang sama

Contoh Desain TD dan BU



Apakah Desain Software itu?

- Desain Software adalah
  - Aktifitas yang menjembatani antara persyaratan dan implementasi software
  - Aktifitas yang memberikan struktur benda yang dihasilkan
    - Dokumen spesifikasi persyaratan harus didesain
    - Struktur dapat memudahkan untuk mengerti dan mengembangkannya
- Aktifitas Desain Software
  - Menentukan dekomposisi sistem ke dalam modul
  - Menghasilkan dokumen desain software
    - Menggambarkan dekomposisi sistem ke dalam modul-modul
  - Biasanya *Arsitektur Software* dihasilkan sebelum desain software

Apakah Arsitektur Software itu?

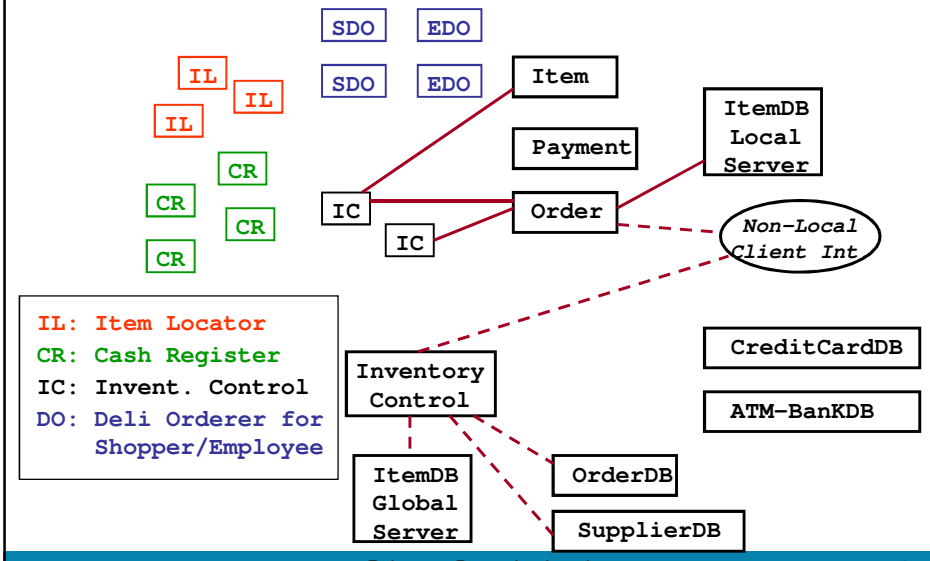
- Arsitektur Software
  - Bentuk kasar dari sistem yang ditentukan
  - Terdiri dari deskripsi
    - Komponen utama sistem
    - Relasi antar komponen
    - Alasan dekomposisi ke dalam komponen-komponen
    - Batasan yang harus dipatuhi dalam mendesain komponen
  - Petunjuk pengembangan desain

Contoh Arsitektur Sistem

High-Tech Supermarket System (HTSS)

- Otomatisasi fungsi dan aksi
  - Update kasir dan inventori
  - Meletakkan bahan makanan secara user friendly
  - Pelacakan delivery order dengan cepat
  - Mengontrol inventory
- Antar muka sistem user
  - Scanner Cash Register
  - GUI untuk kontrol Inventory
  - Shopper menghubungkan Locator dan Orderer
  - Antar muka deli untuk pekerja delivery order

Arsitektur Software HTSS

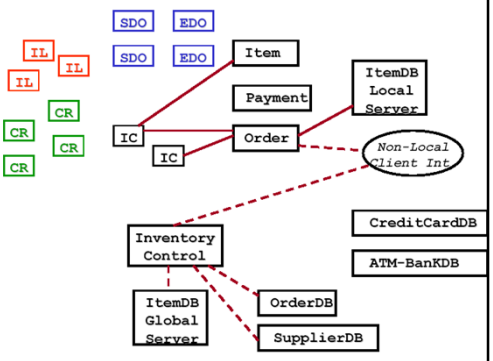


Apakah Modularisasi itu?

- Modularisasi mengungkit ide dari arsitektur software
  - Arsitektur software mendekomposisi sistem ke dalam komponen yang berpengaruh
  - Biasanya, komponen tersebut merupakan program independent terhadap platform hardware
  - Modularisasi mendekomposisi komponen tersebut ke dalam modul-modul yang berpengaruh
- Modul:
  - Adalah komponen Well-defined (terdifinisi dengan baik) dari sistem software
  - Bagian dari sistem yang menyediakan layanan ke modul-modul lain
    - Layanan adalah elemen komputasi yang tersedia untuk digunakan oleh modul lain
    - Modul memberikan layanan dan mengambil layanan

Contoh Modularisasi

- Memanggil komponen Cash Register dari HTSS
- CR terdiri dari modul
  - UPC Scanner
  - Menyimpan Item yg terjual
    - Mengurangi inventory
    - Menambah ke Nota
    - Menambah porfil konsumen
  - Pembayaran
    - Kupon
    - Cash/Debit/Credit
  - Nota
- Modul-modul tersebut berinteraksi dengan komponen HTTS lain



Model

- Kita membuat model untuk mendapatkan pemahaman yang lebih baik dari entitas aktual yang akan dibangun.
- Ketika entitas adalah hal fisik (misalnya, bangunan, pesawat, mesin), kita dapat membangun sebuah model yang identik dalam bentuk tetapi skala lebih kecil.
- Ketika entitas yang akan dibangun adalah perangkat lunak, model kita harus mengambil bentuk yang berbeda.

Model

- Harus mampu mewakili
  - informasi transformasi software,
  - arsitektur dan fungsi yang memungkinkan transformasi terjadi,
  - fitur yang diinginkan pengguna, dan
  - Perilaku sistem dimana transformasi terjadi

Dua Kelas Model

- Dalam software engineering, dapat dibuat 2 kelas model yaitu:
  - Model persyaratan dan
  - Model desain
- *Model persyaratan (atau analysis model)* merepresentasikan persyaratan konsumen dengan menggambarkan software dalam 3 domain yang berbeda:
  - Domain infomrasi
  - Domain fungsional
  - Domain perilaku

Dua Kelas Model

- *Model desain merepresentasikan karakteristik software yang membantu praktisi membangun dengan efektif:* arsitektur, user interface dan detail component-level

Prinsip Pemodelan

1. Tujuan utama dari tim software adalah membangun software, bukan membuat model
2. Travel light—jangan membuat model lebih dari yang dibutuhkan
3. Berusaha untuk menghasilkan model sederhana yang akan menjelaskan masalah atau perangkat lunak.
4. Membangun model yang memungkinkan untuk perubahan
5. Mampu menyatakan suatu tujuan secara eksplisit untuk setiap model yang dibuat.

Prinsip Pemodelan

6. Sesuaikan model yang Anda kembangkan untuk sistem yang dibangun.
7. Cobalah untuk membangun model yang berguna, tetapi lupakan membangun model yang sempurna.
8. Jangan menjadi dogmatis terhadap sintak dari model. Jika dapat menjelaskan konten dengan baik, representasi adalah hal sekunder.
9. Jika naluri Anda mengatakan model ini tidak tepat meskipun tampaknya baik-baik saja di atas kertas, Anda mungkin punya alasan untuk khawatir.
10. Dapatkan umpan balik secepat yang Anda bisa.

Prinsip Model Persyaratan

1. Domain informasi dari masalah harus diwakili dan dipahami.
2. Fungsi apa yang dilakukan software harus didefinisikan.
3. Perilaku software (sebagai akibat event eksternal) harus direpresentasikan.
4. Model yang menggambarkan informasi, fungsi, dan perilaku harus dipartisi dengan cara mengungkapkan detail dalam layer berlapis (atau hirarkis).
5. Task analisis harus bergerak dari informasi penting menjadi detail implementasi

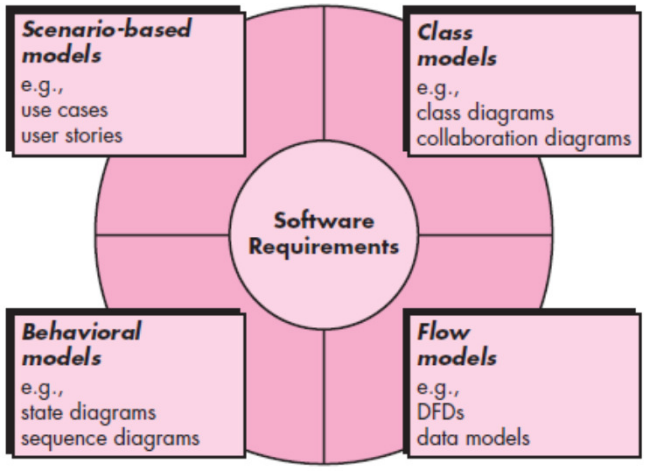
Elemen Model Persyaratan

- Elemen Scenario-based.
  - Sistem ini dijelaskan dari sudut pandang pengguna menggunakan pendekatan berbasis skenario
  - Elemen Scenario-based dari model persyaratan seringkali menjadi bagian pertama dari model yang dikembangkan.
  - Dengan demikian, mereka berfungsi sebagai input untuk pembuatan elemen pemodelan lainnya.
- Elemen Class-based.
  - Setiap penggunaan skenario menyiratkan satu set objek yang dimanipulasi sebagai aktor yang berinteraksi dengan sistem.
  - Obyek ini dikategorikan ke dalam kelas- yaitu sekumpulan hal-hal yang memiliki atribut yang sama dan perilaku yg umum.
  - Kelas perlu berkolaborasi dengansatu sama lain dan hubungan dan interaksi antara kelas

Elemen Model Persyaratan

- Elemen Perilaku.
  - Perilaku sistem berbasis komputer dapat dapat memiliki efek mendalam pada desain yang dipilih dan pendekatan implementasi yang diterapkan.
  - Oleh karena itu, model persyaratan harus menyediakan unsur-unsur pemodelan yang menggambarkan perilaku.
- Elemen Flow-oriented.
  - Informasi berubah karena mengalir melalui sebuah sistem berbasis komputer.
  - Sistem ini menerima masukan dalam berbagai bentuk, menggunakan fungsi untuk mengubahnya, dan menghasilkan output dalam berbagai bentuk.

Elemen Model Persyaratan



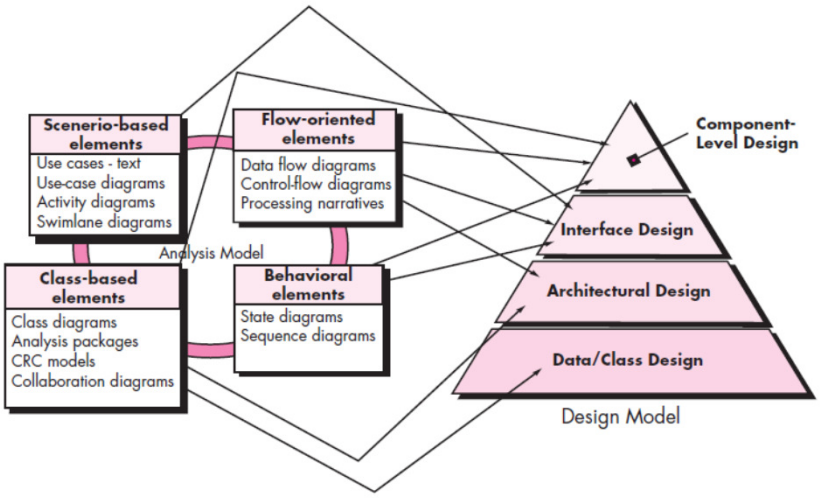
Model Desain

- Model desain perangkat lunak dianalog dengan rencana arsitek untuk rumah.
- Dimulai dengan mewakili keseluruhan yang akan dibangun (misalnya, render tiga dimensi dari rumah) dan perlahan-lahan menyempurnakan hal untuk memberikan bimbingan untuk membangun setiap detail (misalnya, tata letak pipa).
- Demikian pula, model desain yang dibuat untuk perangkat lunak menyediakan berbagai pandangan yang berbeda dari sistem.

Model Desain

- Desain adalah apa yang engineer ingin lakukan.
- Adalah tempat di mana kreativitas aturan-pemangku kepentingan persyaratan, persyaratan bisnis, dan pertimbangan teknis semua menjadi perumusan suatu produk atau sistem.
- Desain membuat representasi atau model perangkat lunak, tetapi tidak seperti model persyaratan (yang fokus pada penggambaran data yang dibutuhkan, fungsi, dan perilaku), model desain menyediakan detail tentang arsitektur perangkat lunak, struktur data, antarmuka, dan komponen yang diperlukan untuk mengimplementasikan sistem.

Dari Model Persyaratan ke Model Desain



Prinsip Model Desain

1. Desain harus dapat dilacak dari model persyaratan.
2. Selalu mempertimbangkan arsitektur sistem yang akan dibangun.
3. Desain data sama pentingnya dengan desain fungsi pengolahan.
4. Antarmuka (baik internal maupun eksternal) harus dirancang dengan hati-hati.
5. Desain antarmuka pengguna harus disesuaikan dengan kebutuhan pengguna akhir. Namun, dalam setiap kasus, harus ditekankan pada kemudahan penggunaan.

Prinsip Model Desain

6. Desain level komponen harus independen secara fungsional.
7. Komponen harus mudah digabungkan satu sama lain dan dengan lingkungan eksternal.
8. Representasi desain (model) harus mudah dimengerti.
9. Desain harus dikembangkan secara iteratif. Dengan setiap iterasi, desainer harus berusaha untuk menyederhanakan.

Prinsip Model Desain

- Ketika prinsip-prinsip desain diterapkan dengan benar, Anda membuat desain yang menunjukkan faktor kualitas baik eksternal dan internal.
- *Faktor kualitas eksternal* adalah sifat dari perangkat lunak yang dapat dengan mudah diamati oleh pengguna (misalnya, kecepatan, kehandalan, ketepatan, kegunaan).
- *Faktor kualitas internal* yang penting bagi engineer perangkat lunak.
  - Mengarah pada desain berkualitas tinggi dari perspektif teknis.
  - Untuk mencapai faktor mutu internal, desainer harus memahami konsep desain dasar