Pemrograman Web Berbasis Framework

Pertemuan 13 :
Pengembangan Project (Bag. 1)

Hasanuddin, S.T., M.Cs.
Prodi Teknik Informatika UAD
hasan@uad.ac.id

Pokok Bahasan

- Pendahuluan
- Requirement atau penelusuran kebutuhan
- Analisis dan Pemodelan Sistem
- Perancangan sistem

TIK :

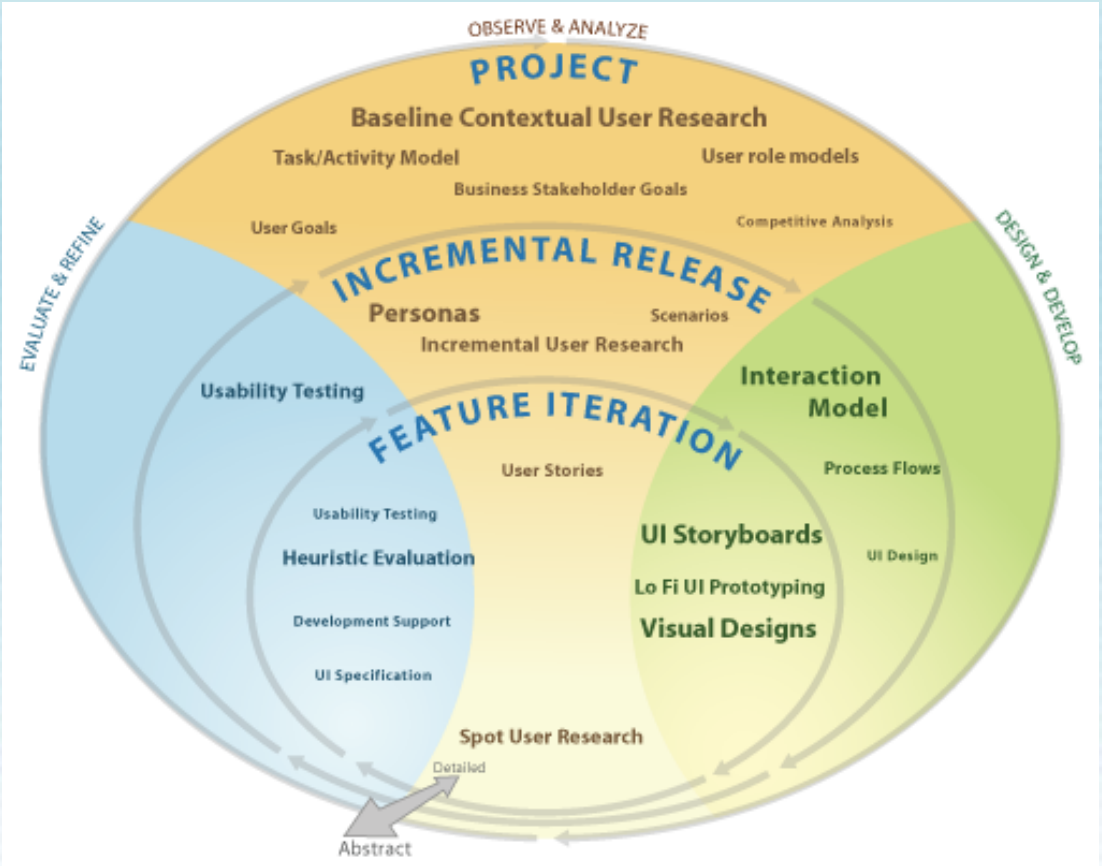
Setelah mengikuti kuliah ini mahasiswa dapat memahami dan menerapkan tahapan pengembangan aplikasi pada Web Framework

Pendahuluan

- Dalam dunia *software engineering*, metodologi paling klasik terdiri dari 5 fase yaitu:
  - Fase requirement atau penelusuran kebutuhan
  - Fase analisa dan pemodelan
  - Fase perancangan
  - Fase pengembangan
  - Fase installasi
- Fase diatas adalah fase standar yang bersifat dinamis. Tidak selamanya tiap fase yang dilewati akan ditinggalkan

Pendahuluan (2)

- Contoh tahapan pengembangan sistem dengan metode *Agile Development* :



Fase Requirement atau Penelusuran Kebutuhan

- Pada fase ini dilakukan pengumpulan data/informasi
- Informasi/data tersebut harus dapat menjawab pertanyan utama berikut :
  - apa yang dibutuhkan?
  - apa tujuan dari aplikasi ini?
  - apa yang ingin dicapai?
  - apakah ada referensi atau contoh?
  - siapa sasaran penggunaan aplikasi ini?

Analisa dan Pemodelan

- Pada fase ini dilakukan analisis kebutuhan sistem termasuk penentuan *feature* dan fungsi apa saja yang akan dibuat untuk memenuhi kebutuhan yang ada
- *Feature* dibagi menjadi *feature* utama atau bagian besar terlebih dahulu baru kemudian didefenisikan menjadi bagian yang lebih kecil
- Misalnya pada aplikasi CMS salah satu *feature* utama adalah blog, kemudian *feature* blog didefenisikan seperti *feature* tagging, category, comment, dsb.
- Pada tahapan ini yang perlu diperhatikan adalah apakah daftar *feature* dan fungsi yang dibuat telah memenuhi kebutuhan yang diinginkan.

Analisa dan Pemodelan (2)

- Analisis Kebutuhan
  - Menentukan karakteristik operasional Perangkat Lunak (PL)
  - Menunjukkan antarmuka PL dengan elemen sistem yang lain
  - Membuat batasan yang harus dipenuhi PL
- Analisis Kebutuhan memungkinkan Software Engineer untuk :
  - Memperinci kebutuhan dasar yang dibuat kapada rekayasa kebutuhan sebelumnya
  - Membangun model yang dapat menggambarkan skenario user, aktivitas fungsional, class masalah dan relasinya, sistem dan perilaku class, dan aliran data ketika ditransformasikan.

Analisa dan Pemodelan (3)

Aturan-aturan :

- Model harus fokus pada masalah atau domain bisnis. Tingkat abstraksinya relatif harus lebih tinggi.
- Setiap elemen model analisis sebaiknya memberikan tambahan pada pemahaman keseluruhan kebutuhan PL dan menyediakan wawasan pada domain informasi, fungsi dan perilaku sistem.
- Tunda semua konsideran infrastruktur dan model non fungsional hingga fase desain.
- Minimalisasi rangkaian melalui sistem.
- Pastikan model analisis menyediakan nilai untuk semua stakeholder.
- Jaga model sesederhana mungkin.

Analisa dan Pemodelan (4)

Pemodelan data:

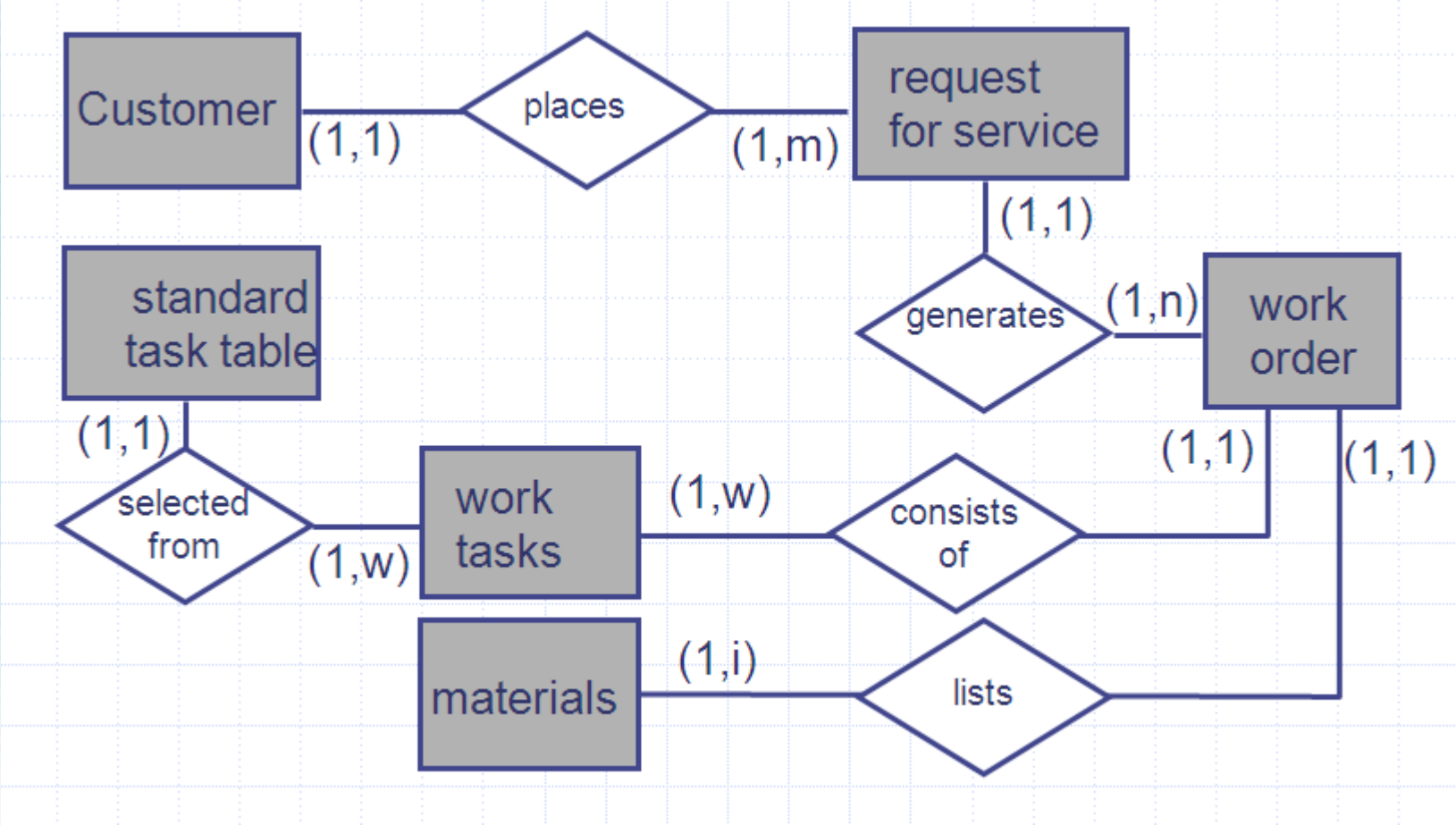
- Memeriksa objek data secara independen terhadap proses
- Fokus perhatikan pada domain data
- Membuat sebuah model pada abstraksi level konsumen
- Mengindikasikan bagaimana objek data berhubungan satu dengan yang lain

Analisa dan Pemodelan (5)

- Pemodelan secara umum dilakukan menggunakan :
  - Diagram ERD
  - Diagram DFD
  - Pemodelan Berorientasi Objek

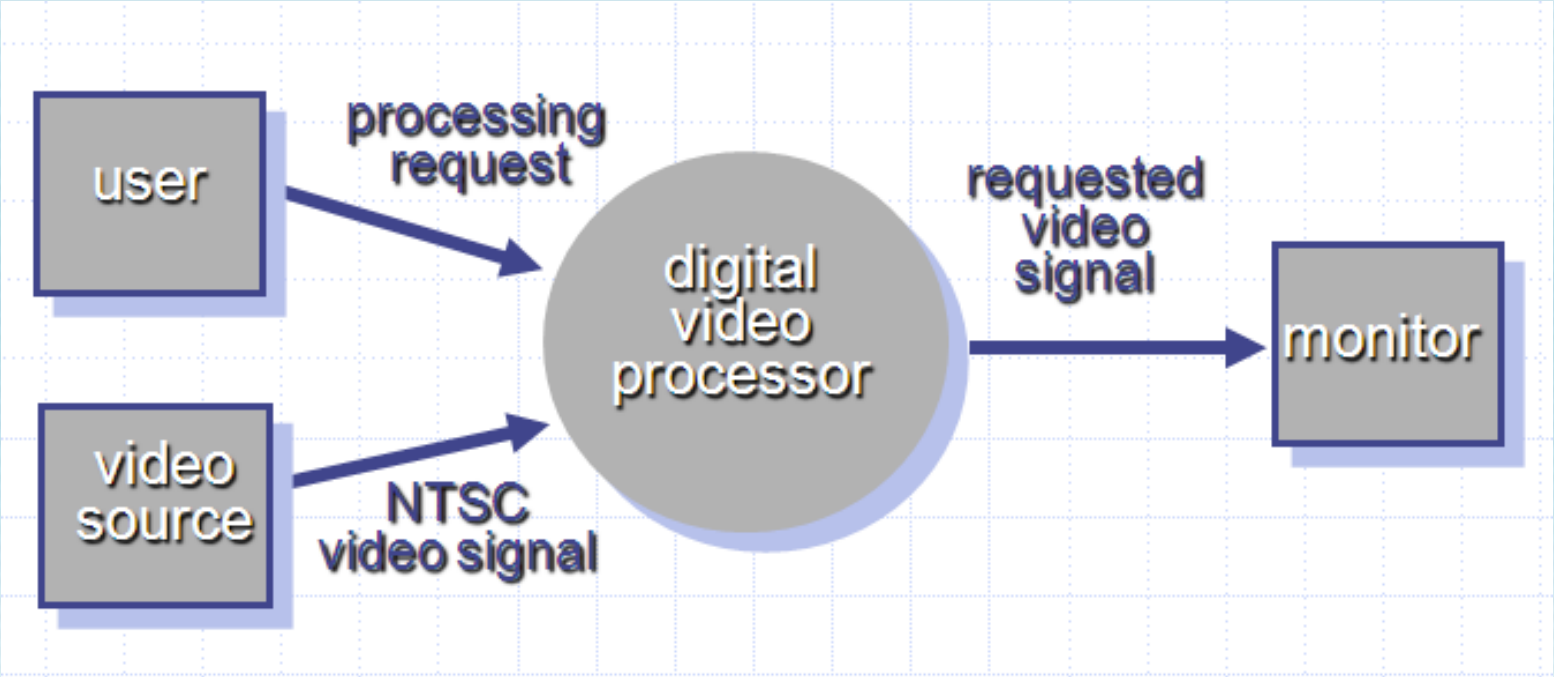
Analisa dan Pemodelan (6)

- Contoh penggunaan diagram ERD



Analisa dan Pemodelan (7)

- Contoh penggunaan diagram DFD



Perancangan Sistem

Desain dan kualitas :

- Desain harus mengimplementasikan semua kebutuhan eksplisit yang ada dalam model analisis, dan dia harus mengakomodasi semua kebutuhan implisit yang diinginkan oleh konsumen.
- Desain harus dapat berupa panduan yang dapat dibaca dan dipahami oleh orang-orang yang akan membuat kode, dan mereka yang menguji serta nantinya mendukung PL tersebut.
- Desain harus menyediakan gambaran utuh dari PL, menggambarkan domain data, fungsional, dan perilaku dari perspektif implementasi.

Perancangan Sistem (2)

Prinsip-prinsip desain:

- Proses desain tidak boleh berjalan dengan “kacamata kuda”
- Proses desain harus bisa dirujuk dari model analisis.
- Proses desain tidak boleh mengulang penemuan-penemuan dasar.
- Desain harus dapat meminimalkan jarak intelektual antara PL dan permasalah yang ada di dunia nyata.
- Desain harus menampakkan keseragaman dan integrasi.
- Desain harus terstruktur untuk mengakomodasi perubahan.
- Desain harus terstruktur untuk turun secara bertahap, walaupun ketika data, event, atau kondisi operasi yang menyimpang ditemui.
- Desain bukan coding dan coding bukan desain.
- Desain harus dapat dipantau kualitasnya mulai dari dia dibuat, bukan setelah jadi.
- Desain harus direview untuk meminimalkan kesalahan semantik (konseptual).

Perancangan Sistem (3)

Elemen Model Desain:

- Elemen-elemen Data
  - Data model --> struktur data
  - Data model --> arsitektur database
- Elemen-elemen arsitektur
  - Domain aplikasi
  - Class-class analisis, relasinya, kolaborasi dan perilaku diubah menjadi realisasi desain
  - Patterns dam “styles”

Perancangan Sistem (4)

Elemen Model Desain (lanjutan):

- Elemen-elemen interface
  - user interface (UI)
  - Interface external pada sistem lain, piranti-piranti, jaringan-jaringan atau produsen maupun konsumen informasi lainnya
  - Interface internal antara komponen-komponen desain**.**
- Elemen-elemen komponen
- Elemen-elemen deploy

Perancangan Sistem (5)

Konsep Desain *Object Oriented* (OO) :

- Desain Class
  - Entity classes
  - Boundary classes
  - Controller classes
- Inheritance—semua tanggung jawab superclass akan diwarisi oleh semua subclassnya
- Messages—stimulasi beberapa perilaku yang dapat terjadi pada objek penerima pesan
- Polymorphism—sebuah karakteristik yang mengurangi usaha yang dibutuhkan untuk memperluas desain

Perancangan Sistem (6)

Desain Class :

- Analisis class disempurnakan dalam desain untuk menjadi class-class entitas
- Class-class boundary dikembangkan selama desain untuk membuat interface(mis. Layar interaktif atau laporan cetak) yang dilihat pengguna dan berinteraksi.
  - Class-class boundary didesain dengan tanggungjawab untuk mengelola cara objek entitas ditampilkan kepada user.
- Class-class control didesain untuk mengelola :
  - Pembuatan atau perubahan objek entitas;
  - Instansiasi object boundary dengan mengambil informasi dari objek entitas;
  - Komunikasi kompleks antara sekelompok objek;
  - Validasi data yang dikomunikasikan antar objek atau antar pengguna dan aplikasi.

Perancangan Sistem (7)

Inheritance:

- Pilihan-pilihan desain :
  - Class dapat didesain dan dibangun dari nol. Jika demikian, inheritance tidak digunakan.
  - Hierarki class dapat dicari untuk mencari kemungkinan jika sebuah class yang lebih tinggi pada hierarki (superclass) memiliki atribut-atribut dan operasi-operasi yang paling banyak dibutuhkan. Class baru ini diturunkan dari superclass dan beberapa tambahan dapat diberikan jika dibutuhkan.
  - Hierarki class dapat di restrukturisasi sehingga atribut-atribut dan operasi-operasi yang dibutuhkan dan diturunkan oleh class baru tersebut.
  - Karakteristik dari class yang sudah ada dapat di override dan atribut-atribut dan operasi-operasi dengan versi berbeda dapat diimplementasikan untuk class baru tersebut.

Referensi :

- Pressman, RS., 2008, *Software Engineering: A Practitioner's Approach*, New York: McGraw-Hill
- Sommerville, I, 2007, *Software Engineering*, Addsion Wesley