

## **MATERI PEMODELAN PERANGKAT LUNAK KELAS XI RPL**

*Oleh : Samsul Arifin, S.Kom*

*Email : samsul.skom@gmail.com*

### **Konsep Pemodelan Perangkat Lunak (PL)**

Konsep rekayasa PL.

Suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

Komponen dan karakteristik PL.

Komponen PL : Terdiri dari berbagai macam hal seperti perencanaan dalam membuat aplikasi.

Karakteristik PL : Dalam perangkat lunak kita harus disiplin dalam melakukan tahap-tahap membahas aspek produksi.

Prinsip Analisis dan Desain.

Prinsip analisis : Untuk mencari kebutuhan pengguna dan organisasi serta menganalisa kondisi yang ada (sebelum diterapkan sistem informasi yang baru).

Desain : Bertujuan menentukan spesifikasi detail dari komponen-komponen sistem informasi (manusia, hardware, software, network dan data) dan produk-produk informasi yang sesuai dengan hasil tahap analisis.

Ragam pemodelan PL.

The Waterfall Model, Prototyping, dan Unified Process (UP).

### **Model proses pengembangan perangkat lunak**

Tahapan proses pengembangan PL.

Dalam pengembangan PL pertama kita harus menentukan kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

Ragam model proses pengembangan PL.

Linear sequential Model (Model Sekuensial Linear)/Model Waterfall.

Prototyping Model.

RAD Model (Rapid Application Development).

Incremental Model.

Spiral Model/Spiral Boehm.

Component Assembly Model (CAM/Model Perakitan Komponen).

The Concurrent Development Model.

Formal Method Models.

Fourth Generation Techniques/Model Teknik Generasi ke-4/4GT.

Model Air terjun.

Model Air terjun adalah model klasik yang mengusung pengembangan perangkat lunak yang sistematis, berurutan/sekuensial dimulai pada tingkat dan kemajuan system pada seluruh persyaratan dalam analisis, perancangan (desain), pengkodean, pengujian (testing), hingga ke tahap pemeliharaan dalam membangun software (perangkat lunak).

Model Prototipe.

Model Prototipe menyajikan gambaran yang lengkap dari sistem, terdiri atas model kertas, model kerja dan program. Pihak pengembang akan melakukan identifikasi kebutuhan pemakai, menganalisa sistem dan melakukan studi kelayakan serta studi terhadap kebutuhan pemakai, meliputi model interface, teknik prosedural dan teknologi yang akan dimanfaatkan.

Model RAD.

Model RAD adalah proses pembangunan Perangkat Lunak yang menekankan pada siklus pengembangan yang pendek dan singkat. Model ini mengawinkan model waterfall dan model component based construction.

Model Spiral.

Model Spiral mengadaptasi dua model perangkat lunak yang ada yaitu model prototyping dengan pengulangannya dan model waterfall dengan pengendalian dan sistematikanya.

### **Rekayasa kebutuhan PL.**

Tipe kebutuhan dan penggunaanya.

Ini biasanya harus sesuai dengan permintaan pelanggan, seperti kebutuhan kantor, atau untuk mesin kasir di minimarket

Ukuran kebutuhan.

Diukur dari kebutuhan si pemakai atau sesuai permintaan pelanggan.

Tahapan proses rekayasa kebutuhan.

Tahapan rekayasa kebutuhan yaitu menentukan jenis keperluan, menentukan aplikasi.

Teknik-teknik analisa kebutuhan.

Pertama menentukan jenis kebutuhan, menentukan jenis aplikasi.

Perancangan kebutuhan perangkat lunak.

Perancangan kebutuhan merancang sebuah aplikasi apabila semua aspek-aspek perangkat lunak terpenuhi

### **Diagram alur data (DFD).**

Fungsi dan Komponen DFD.

Fungsi : Menyajikan gambaran apa yang manusia, proses dan prosedur lakukan untuk mentransformasi data menjadi informasi.

Komponen : Komponen DFD terdiri dari Data flow dan lain sebagainya.

Tingkatan level DFD.

Tingkatan Level DFD :

- a. Diagram konteks : Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan system dengan lingkungannya.
- b. Diagram level Zero : Diagram ini adalah dekomposisi dari diagram konteks. Merupakan diagram yang menggambarkan proses-proses utama system dan alur datanya.
- c. Diagram level satu : Diagram ini merupakan dekomposisi dari diagram level zero.
- d. DFD level dua,tiga, ... : Diagram ini merupakan dekomposisi dari level sebelumnya.
- e. Entity Relationship Diagram : Model Entity Relationship adalah suatu penyajian data dengan menggunakan Entity dan Relationship.

Spesifikasi proses.

Spesifikasi proses terdiri External entity, data flow, Context diagram.

Tahapan pembuatan DFD.

Identifikasi seluruh External Entity yang terlibat > identifikasi seluruh proses yang terlibat > tentukan Data Flow > Buat Context-Diagram > Buat DFD level 0 > Buat DFD Level Berikutnya (jika perlu).

### **Diagram hubungan antar entitas (ERD).**

Model data logik (Conceptual Data Model).

Conceptual Data Model (CDM) merupakan representasi platform-independen dari sistem, memberikan pandangan abstrak struktur basis data.

Model data fisik (physical data model (PDM).

Physical Data Model (PDM) merupakan representasi dari database nyata dan obyek terkait yang berjalan pada server melalui informasi lengkap mengenai struktur data fisik, seperti tabel, kolom, referensi, trigger, prosedur, view dan indeks.

Transformasi model data fisik ke bahasa SQL.

Transformasi model data fisik ke SQL harus menentukan sistem pengodean.

### **Antar muka pengguna (user interface).**

Tujuan dan manfaat.

Tujuan : Mengkomunikasikan fitur-fitur sistem yang tersedia agar user mengerti dan dapat menggunakan sistem tersebut.

Manfaat : User / pengguna dapat dengan mudah mengoperasikan sebuah PC, dengan adanya user interface.

Prinsip desain antar muka.

Perangkat lunak adalah program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras, diartikan sebagai 'penterjemah' perintah-perintah yang dijalankan pengguna komputer untuk diteruskan ke atau diproses oleh perangkat keras.

Interaksi pengguna.

Pengguna dapat berinteraksi dengan mudah dan memahami semua intruksi program menggunakan teknologi Interface/tatapmuka.

Penyajian informasi.

Dalam penyajian ditampilkan secara luas atau Detail sehingga User dapat memahaminya dengan mudah.

Perancangan user interface.

User Interface merancang tampilan agar dapat di pahami pengguna dengan mudah.

Arsitektur perangkat lunak.

Pengenalan arsitektur perangkat lunak.

Arsitektur komputer juga berkaitan dengan atribut-atribut yang mempunyai dampak langsung pada suatu sistem komputer, maksud dari atribut-atribut itu adalah arsitektur komputer mempunyai kebutuhan yang fungsional sebagai perangkat keras dalam desain suatu program komputer.

Layering.

Ragam arsitektur.

Composite Structure Diagram, Interaction Overview Diagram dan lain sebagainya.

Pengenalan struktur chart diagram.

Struktur chart adalah bagian dari Pemodelan Perangkat Lunak

Transformasi DFD ke struktur chart diagram.

Interaksi komponen.

### **Pemodelan sistem berorientasi obyek (UML).**

Prinsip analisis dan desain sistem berorientasi obyek.

Secara umum, metode analisis berorientasi objek mencakup representasi kelas dan hirarki kelas, model hubungan objek, dan model perilaku objek.

Pemodelan menggunakan UML.

Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat

dengan use case dan actor.

Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagrams.

Menggambarkan representasi struktur statik sebuah sistem dalam bentuk class diagrams.

Membuat model behavior yang menggambarkan kebiasaan atau sifat sebuah sistem dengan state transition diagrams.

Menyatakan arsitektur implementasi fisik menggunakan component and development diagrams.

Menyampaikan atau memperluas fungsionalitas dengan stereotypes.

Definisi UML.

Unified Modeling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. Unified Modeling Language (UML) adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri software untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak.

Sejarah UML.

Tahun 1994, Grady Boch dan James Rumbaugh bergabung untuk menggunakan metode berorientasi objek. Ivan Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat suatu bahasa pemodelan objek standar sebagai ganti dari pendekatan atau metode objek standar.

Ragam diagram UML.

Use Case Diagram.

Class Diagram.

State machine diagram.

Activity Diagram.

Sequence Diagram.

Component diagram.

Langkah-langkah pembuatan model UML.

Membuat Functional requirement

Membuat Use Case Diagram

Membuat Scenario

Membuat Class Diagram

Membuat Sequence Diagram.

Membuat Activity Diagram.

### **Kebutuhan sistem berbasis obyek.**

Use Case diagram.

Use Case diagram untuk menggambarkan prosedur-prosedur perilaku perangkat lunak.

Definisi.

Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Kata lain Use Case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu.

Manfaat.

Pengguna dapat dengan mudah mengetahui prosedur perilaku perangkat lunak (software).

Notasi / simbol.

Spesifikasi use case.

Langkah-langkah pembuatan use case diagram.

Mengidentifikasi proses-proses bisnis dan kebutuhan pemakai dan mengekspresikan dengan ‘use-case’.

Sebenarnya bukan merupakan aktivitas analisis berorientasi objek, karena tidak membicarakan pembahasan tentang objek.

Diperlukan karena dapat menjelaskan aktivitas-aktivitas apa saja yang harus dikerjakan oleh sistem, dan menjelaskan juga perilaku dari komponen-komponen sistem.

Ada diagram tertentu yang dapat merepresentasikan model kebutuhan dari “use-case” yang diperoleh.

### **Alur kerja sistem berorientasi obyek.**

Pengenalan activity diagram.

Salah satu cara untuk memodelkan aliran kerja (workflow) dari business use case dalam bentuk grafik.

Diagram ini menunjukkan langkah-2 dalam aliran kerja, titik-titik keputusan dalam aliran kerja, siapa yang bertanggungjawab menyelesaikan masing-masing aktivitas dan objek-objek yang digunakan dalam aliran kerja.

Notasi activity diagram.

Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses

Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis

Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur

Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara

keseluruhan

Activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram

Langkah-langkah pembuatan activity diagram.

- Langkah pertama kita buka aplikasi star UML, kemudian akan muncul pilihan pada kotak dialog dan pilih Empty Project.

- Setelah lembar kerja terbuka maka kita sudah bisa membuat activity diagram sesuai dengan tema yang telah ditentukan. Pada activity diagram terdapat elemen-elemen yaitu :

- a. ActionState : digunakan untuk aktivitas yang dilakukan oleh user

- b. InitialState : digunakan untuk memulai suatu workflow pada sebuah Activity Diagram

- c. FinalState : digunakan untuk mengakhiri suatu workflow pada sebuah Activity Diagram biasanya pada beberapa activity diagram ditemukan lebih dari satu FinalState.

- d. Synchronization : terdapat dua fungsi yaitu fork dan join untuk menunjukkan kegiatan yang dilakukan secara paralel.

- e. Decision : sebagai pilihan untuk mengambil keputusan.

- f. Transition : menunjukkan kegiatan berikutnya setelah suatu kegiatan sebelumnya.

## **T U G A S**

### **STUDI KASUS :**

1. Sebutkan dan jelaskan kelebihan serta kekurangan dari model proses yg ada.
2. Anda adalah seorang manajer yang mendapatkan proyek sebuah perangkat lunak pengelola database yg sangat sederhana, tetapi waktu yg ditentukan oleh pelanggan sangat ketat. Pertanyaan; model pengembangan perangkat lunak mana yang anda pilih dan jelaskan alasannya.
3. Anda dipilih menjadi seorang manajer proyek yg mendapat proyek membangun sebuah aplikasi yang sangat mirip dengan aplikasi lain yg sudah pernah dibangun sebelumnya, meskipun yang satu ini lebih besar dan kompleks, syarat syaratnya sudah didokumentasikan dengan teliti oleh pemakai. Pertanyaan; model pengembangan perangkat lunak mana yg anda pilih dan jelaskan alasannya.
4. Buat sebuah program dan kirimkan folder source programnya ke alamat email.

*NB: Tolong kirimkan jawabannya ke alamat email berikut :  
[samsul.skom@gmail.com](mailto:samsul.skom@gmail.com)*

**SELAMAT BELAJAR**