

BAB 3

METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Waktu penelitian dilakukan pada semester genap tahun ajaran 2013/2014.

3.2 Spesifikasi *Hardware* dan *Software* yang Digunakan

Pengembangan sistem verifikasi mata kuliah ini menggunakan satu unit *notebook* dengan spesifikasi sebagai berikut.

1. Spesifikasi *hardware* yang digunakan dalam pengembangan sistem ini adalah sebagai berikut.
 1. *Processor* Intel(R) Core(TM) i7-2670QM CPU @2.20 GHz.
 2. RAM DDR2 4 GB.
 3. HDD 750 GB.
 4. VGA NVIDIA GEFORCE GT540M.
2. Spesifikasi *software* yang digunakan adalah sebagai berikut.
 1. Sistem Operasi Windows 7 Ultimate.
 2. Aplikasi *local server* XAMPP 1.8.1.
 3. Notepad++
 4. *Browser* Mozilla Firefox

3.3 Data

Data yang diambil dan yang digunakan pada sistem ini adalah sebagai berikut.

1. Data Kurikulum Jurusan Ilmu Komputer.
2. Data Transkrip Siakad Unila.

3.4 Metode Penelitian

Rational Unified Process(RUP) adalah proses rekayasa perangkat lunak yang menggunakan pendekatan disiplin untuk menetapkan tugas dan tanggung jawab dalam pengembangan sistem (Manalil, 2010). RUP diciptakan, dikembangkan dan dikelola oleh *Rational Software* sekarang IBM. Tujuan dari RUP adalah menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna dan dapat diprediksi penjadwalan dan biaya pengembangannya (Kruchten, 2003). Fase dan iterasi pengembangan perangkat lunak dengan metode *Rational Unified Process* antara lain sebagai berikut.

1. Fase

Dalam *Rational Unified Process*(RUP) siklus pengembangan perangkat lunak dibagi ke dalam empat fase, yaitu:

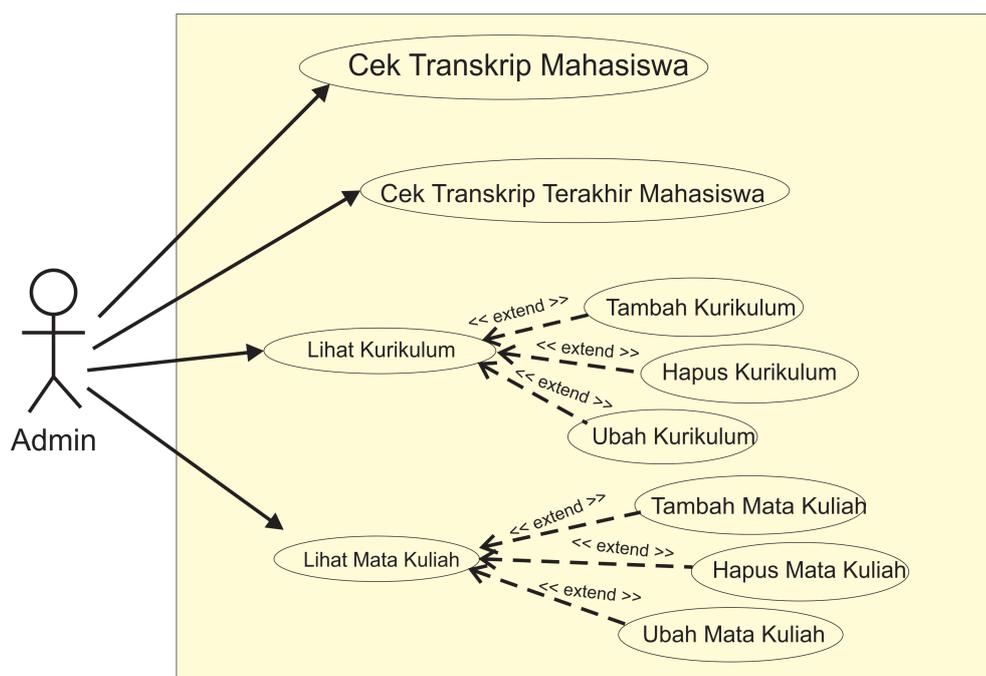
1. *Inception*

Pada fase awal atau *inception* pengembang mendefinisikan batasan kegiatan, menentukan bisnis proses, melakukan analisis kebutuhan pengguna, melakukan perancangan awal perangkat lunak (perancangan *use case*). Fase ini akan menghasilkan proses bisnis, *use case* awal.

2. *Elabration*

Tujuan dari fase elaborasi (*Elaboration*) adalah untuk menganalisa permasalahan utama, membangun arsitektur dasar, mengembangkan ren-

cana proyek, dan menghilangkan resiko tertinggi dari proyek. Hasil dari fase elaborasi ini adalah sebuah *use case* yang sudah, deskripsi arsitektur perangkat lunak, persyaratan non-fungsional, dan gambaran umum pengembangan perangkat lunak. Contoh rancangan *use case* yang diperlihatkan pada gambar 3.1.



Gambar 3.1: Rancangan *Use Case*

3. Construction

Fase konstruksi terfokus pada pengembangan perangkat lunak baik komponen utama maupun fitur-fitur pendukung dengan melakukan sederet iterasi. Di setiap iterasi terdapat proses analisa, desain, implementasi, dan pengujian. Dalam proses pengembangannya dapat menggunakan konstruksi paralel agar mempercepat hasil perangkat lunak. Hasil yang diharapkan dari fase ini adalah sebuah produk perangkat lunak yang siap digunakan oleh *end-user*, yaitu sebuah produk perangkat lunak yang terintegrasi dengan *platform* yang memadai dan penjelasan dari produk tersebut.

4. *Transition*

Fase transisi merupakan fase dimana perangkat lunak hasil dari pengembangan didistribusikan kepada pengguna akhir (*end-user*). Kegiatan-kegiatan yang dilakukan pada fase ini antara lain: *beta testing* untuk memvalidasi sistem baru terhadap harapan pengguna, konversi *database* operasional, pelatihan bagi pengguna dan pengelola, dan pemasaran produk.

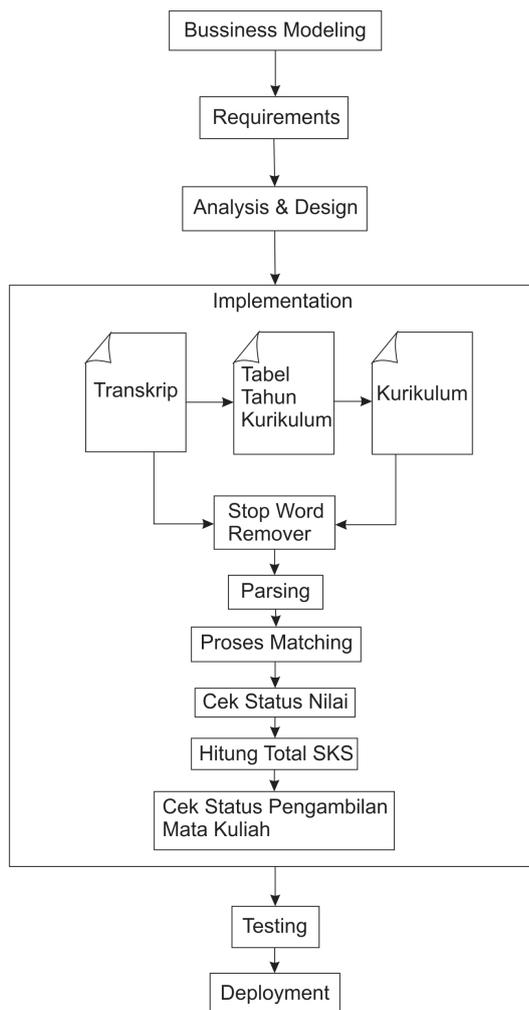
2. Iterasi

Setiap fase dalam *rational unified process* selanjut dibagi menjadi iterasi. Iterasi adalah *loop* pengembangan yang lengkap sehingga dalam rilisnya (internal atau eksternal) dari *executable* produk, bagian dari produk akhir yang sedang dikembangkan, menjadi sistem akhir secara bertahap dari iterasi ke iterasi. Manfaat penggunaan pendekatan iterasi dibanding pendekatan *water fall*, sebagai berikut:

1. Resiko telah diantisipasi lebih awal.
2. Pengelolaan perubahan lebih mudah.
3. Tingkat penggunaan kembali lebih tinggi.
4. Pengembang dapat belajar diproses pengembangan proyek.
5. Secara keseluruhan memiliki kualitas yang baik.

3.4.1 Tahapan Penelitian

Dalam *Rational Unified Process*(RUP) ada sembilan proses inti, yang merupakan bagian dari seluruh pekerja dan aktifitas. Kategori pertama, enam inti rekayasa yang ditunjukkan pada Gambar 3.2.



Gambar 3.2: Bagan Aktifitas Kategori Pertama

1. *Business Modeling* (Pemodelan Bisnis)

Salah satu masalah utama dalam rekayasa perangkat lunak adalah komunikasi yang tidak baik antara rekayasa perangkat lunak dan rekayasa bisnis. Namun *Rational Unified Process* mengatasinya dengan menyediakan bahasa dan proses dari kedua bagian tersebut, serta menunjukkan bagaimana cara menciptakan dan memelihara ketertelusuran langsung antara bisnis dan perangkat lunak. Dalam pemodelan bisnis, bisnis proses didokumentasikan ke dalam *business use case*. *Business use case* merupakan analisis untuk memahami bagaimana bisnis harus mendukung proses bisnis.

2. *Requirement* (Persyaratan)

Tujuan dari alur kerja persyaratan *requirement workflow* adalah untuk menggambarkan kebutuhan sistem dan deskripsi kesepakatan antara pengembang dan pelanggan. Dokumen bayangan dibuat, dan kebutuhan *stakeholder* dimunculkan. Identifikasi aktor, menyatakan pengguna dan sistem lain yang berinteraksi dengan sistem yang sedang dikembangkan. *Use case* diidentifikasi, menyatakan perilaku sistem, dan dikembangkan sesuai dengan kebutuhan aktor. Setiap *use case* dijelaskan secara rinci, deskripsi dari *usecase* menunjukkan bagaimana sistem berinteraksi dengan aktor dan apa yang dapat sistem lakukan. Kebutuhan non-fungsional dijelaskan dalam spesifikasi tambahan. Fungsi *usecase* adalah sebagai benang pemersatu siklus pengembangan sistem.

3. *Analysis and Design* (Analisis dan desain)

Tujuan dari analisis dan desain adalah membuat rancangan sistem yang akan direalisasikan dalam bentuk implementasi. Hasil dari analisis dan desain adalah model desain dan model analisis opsional. Model desain berguna sebagai abstrak dari *source code* program, sebagai acuan bagi *programer* dalam mengimplementasikan sebuah sistem.

4. *Implementation* (Implementasi)

Tujuan implementasi, sebagai berikut.

1. Untuk mendefinisikan pengelompokan kode, dalam pengertian pengimplementasian subsistem yang dikelompokkan dalam lapisan-lapisan.
2. Untuk mengimplemenatsikan kelas dan objek dalam pengertian komponen seperti *Source code*, *binary*, atau *executable*.
3. Untuk menguji komponen yang dikembangkan sebagai unit-unit.
4. Untuk mengintegrasikan hasil yang diproduksi oleh pengimplementasi individu atau tim menjadi sistem yang dapat dieksekusi.

5. *Test* (Pengujian)

1. Untuk memastikan interaksi antarobjek.
2. Untuk memastikan kesatuan integrasi dari semua komponen perangkat lunak.
3. Untuk memastikan seluruh kebutuhan yang telah diimplementasikan.
4. Untuk mengidentifikasi dan memastikan kekurangan pada perangkat lunak.

6. *Deployment* (Penyebaran)

Tujuan dari alur penyebaran adalah untuk memproduksi rilis perangkat lunak sehingga siap untuk dipakai dan didistribusikan.

Kategori kedua dalam proses pelaksanaan RUP adalah 3 inti pendukung, yaitu :

1. *Project Management* (Manajemen Proyek)

Pendesripsian berbagai macam strategi untuk bekerja dengan proses iterasi.

2. *Configuration and Change Management* (Konfigurasi dan Manajemen Perubahan)

Pengendalian perubahan pada proyek dan pemeliharaan integrasi artefak proyek dan aktifitas manajemen.

3. *Environment* (Lingkungan)

Mencakup seluruh kebutuhan infrastruktur yang dibutuhkan untuk mengembangkan suatu sistem.