BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian, yang membantu penelitian dalam pengumpulan dan menganalisis data. Dan, dalam penelitian ini penulis menggunakan metode penelitian deskriptif dengan dua metode yaitu : metode pengumpulan data dan metode pengembangan perangkat lunak.

Metode deskriptif adalah suatu metode dalam penelitian status kelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran, atau pun suatu kelas peristiwa pada masa sekarang. Menurut Whitney (1960) berpendapat, metode deskriptif adalah pencarian fakta dengan interpretasi yang tepat. Penelitian deskriptif mempelajari masalah-masalah dalam masyarakat, sertatat acara yang berlaku dalam masyarakat serta situasi-situasi tertentu, termasuk tentang hubungan, kegiatan-kegiatan, sikap-sikap, pandangan-pandangan serta proses-proses yang sedang berlangsung dan pengaruh-pengaruh dari suatu fenomena.

3.2 Metodologi Penelitian

Metodologi penelitian adalah sekumpulan peraturan, kegiatan, dan prosedur yang digunakan oleh pelaku suatu disiplin ilmu.Metodologi juga merupakan analisis teoritis mengenai suatu cara atau metode. Penelitian merupakan suatu penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban.

3.3 Jenis Dan Metodologi Pengumpulan Data

Metode pengumpulan yang di lakukan oleh penulis untuk membuat *game* RPG (*Role Playing Game*) adalah sebagai berikut :

1.4 Studi Literatur

Studi literature adalah metode pengumpulan data yang di lakukan dengan mengumpulkan jurnal, informasi, dari internet dan buku yang terkait dengan pembuatan skripsi penulis.

2.4 Obervasi

Metode pengumpulan data dengan cara melakukan penelitian terhadap permasalahan yang di ambil.

3.3.1 Metode Pendekatan dan Pengembangan Sistem

3.3.1.1 Metode Pendekatan Sistem

Metode pendekatan sistem adalah bagaimana untuk melihat masalah pada sistem yang di buat dengan menggunakan alat alat dan peraturan-peraturan yang melengkapi satu atau lebih dalam tahapan-tahapan pengembangan sistem informasi. Metode pendekatan sistem yang di gunakan oleh penulis adalah pendekatan berorientasi objek dengan menggunakan alat bantu UML yang terdiri dari :*use case, sequence, collaboration, activity.*

3.3.1.2 Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang di ambil adalah model waterfall yang meliputi dari :

1. **Requirement**

   Pemodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan kedalam bentuk software. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dsb. Tahap ini sering disebut dengan Project Definition.

2. Analysis

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada pelanggan.

3. Design

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi kedalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

4. Coding

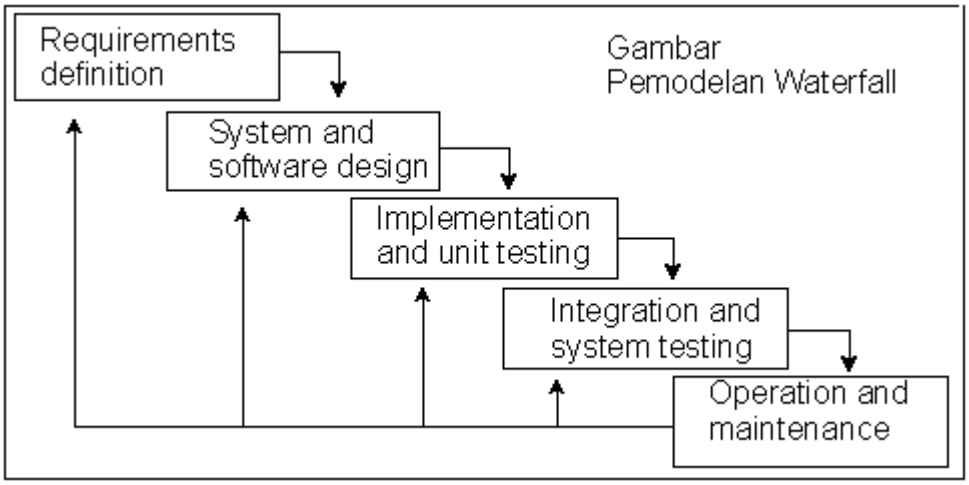
Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknisnantinya dikerjakan oleh programmer.

5. **Testing**

   Berfungsi untuk mencoba software yang telah di rancang agar dapat mengetahui kesalahan dalam pemograman sehingga dapat di perbaiki. Sehingga software benar – benar sesuai dengan kebutuhan yang sudah di definisikan sebelumnya.

6. **Maintenance**

   Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada errors kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.



Gambar 3.1 Waterfall Model

http://tonyjustinus.wordpress.com/2007/11/11/waterfall-process-model/

3.3.1.3 Alat Bantu Pengembangan Sistem

Alat bantu yang adagunakanpadaaplikasiiniadalahsebagaiberikut :

1. Use Case Diagram

Menggambarkan sekelompok use cases danaktor yang disertai dengan hubungan diantaranya.Diagram usecases ini menjelaskan dan menerangkan kebutuhanyang pengguna, serta sangatberguna dalam menentukan struktur organisasi dan model daripada sebuah sistem.

1.2 Karakteristik Use Case Diagram

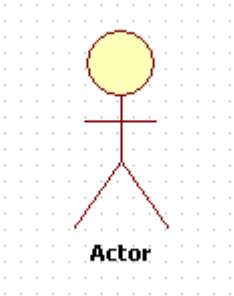
Terdapat beberapa karakteristik yang terdapat pada *Use Case Diagram* yaitu, sebagai berikut :

a. Use cases adalah interaksi atau dialog antara sistem dan actor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem.
b. Use cases dilakukan oleh actor dan mungkin melibatkan peran actor lain.Use cases harus menyediakan nilai minimal kepada satu actor.
c. Use cases bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi.
d. Use case class memiliki objek use case yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal.

1.3 Aktor

Pada dasarnya actor bukanlah bagian dari use case diagram, namun untuk dapa tterciptanya suatu use case diagram diperlukan beberapa aktor. Aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah actor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem.

Actor hanya berinteraksi dengan use case, tetapi tidak memiliki control atas use case. Actor digambarkan dengan stick man .Actor dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan relationship. Gambar actor dapat di lihat dari gambar di bawah ini.



Gambar 3.2 Aktor Pada Use Case

http://mitanovia.wordpress.com/belajar-yuk/uml/use-case-diagram/

1.4 Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan

sistemyang akan dibangun. Dan cara untuk menentukan sebuah *use case* adalah sebagai berikut :

a. Pola perilaku perangkat lunak aplikasi.
b. Gambaran tugas dari sebuah aktor.
c. Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada actor.
d. Apa yang dikerjakan oleh suatu perangkat lunak.

2. Class Diagram

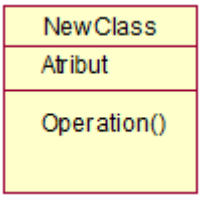
Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam system. Class diagram memberikan gambaran system secara statis dan relasi antar mereka. Biasanya, dibuat beberapa class diagram untuk system tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya.

Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap system yang dibangun. Class diagram adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur system sebelum kode ditulis, dan membantu untuk memastikan bahwa system adalah desain terbaik.

a. Class

Kelas adalah sesuatu yang membungkus informasi dan perilaku. Secara tradisional, system dibangun dengan ide dasar bahwa akan menyimpan informasi

pada sisi baris data dan data perilaku pengolahnya pada sisi aplikasi. Salah satu perbedaan terstruktur dengan pendekatan berorientasi obyek adalahpada berorientasi obyek menggabungkan informasi dan perilaku pengolah informasi dan menyembunyikan semua kedalam sesuatu yang disebut kelas. Dalam UML, kelas ditunjukkan menggunakan notasi sebagai berikut.



Gambar 3.3 Class pada Class Diagram

http://stanlysk.blogspot.com/2012/10/menggambar-uml-class-diagram.html

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan/prilaku objek pada proses dengan mendeskripsikann waktu hidup objek dan pesan yang dikirimkan dan diterimaantar objek. Oleh karena itu untuk menggambar Sequence Diagram maka harus diketehui objek-objek yang terlibat dalam sebuah proses beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya Sequence Diagram yang harus digambarkan adalah sebanyak pendefinisian proses yang memiliki proses sendiri atau yang penting semua proses yang didefinisikan interaksi jalannya pesan sudah dicakup pada Sequence Diagram sehingga makin banyak proses yang didefinisikan maka Sequence Diagram yang harus dibuat juga semakin banyak.

4. Activity Diagram

Activity diagram memiliki pengertian yaitu lebih focus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip flowchart atau data flow diagram pada perancangan terstruktur. Memiliki pula manfaat yaitu apa bila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan activity dibuat berdasarkan sebuah atau beberapa use case pada use case diagram. Beberapa hal yang penting untuk di ketahui dalam pembuatan activity diagram yaitu, sebagai berikut :

a. Activity mengambarkan sebuah pekerjaan atau tugas dalam workflow.
b. Pada UML, activity digambarkan dengan symbol kotak.
c. Start state dengan tegas menunjukan dimulainya suatu workflow pada sebuah activity diagram.
d. Hanya adas atu start state dalam sebuah workflow.
e. Pada UML, start state digambarkan dengan symbol lingkaran yang solid.
f. End state menggambarkan akhir atau terminal dari pada sebuah activity diagram.
g. Bisa terdapat lebih dari satu end state pada sebuah activity diagram
h. Pada UML, end state digambarkan dengan symbol sebuah bull's eye

i. State transition menunjukan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.
j. Pada UML, state transition digambarkan oleh sebuah solid line dengan panah.
k. Decision adalah suatu titik atau point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

3.4 Metode Pengujian

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak,mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Dan salah satu metode pengujian yang dapat di lakukan dengan menggunakan metode pengujian *black box*.

Pengujian black box merupakan pendekatan komplementer dari teknik whitebox, karena pengujian black box diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik white box. Pengujian black boxberfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratanfungsional suatu program.

Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian

black box merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Metode yang digunakan oleh penulis adalah dengan metode *black box* karena sesuai dengan aplikasi yang dibuat yaitu dengan melihat fungsi dari aplikasi tersebut, karena yang dilihat dari aplikasi yang dibuat adalah output dari aplikasi yang dibuat oleh penulis.