BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Menurut Sugiono (2009:38) menyatakan bahwa, definisi objek penelitian adalah, sebagai berikut:

"Objek penelitian merupakan Suatu atribut atau sifat atau nilai dari orang, objek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk di pelajari dan kemudian ditarik kesimpulannya."

Berdasarkan definisi diatas, dapat ditarik kesimpulan bahwa objek penelitian merupakan sesuatu hal yang akan diteliti dengan mendapatkan data untuk tujuan tertentu dan kemudian dapat ditarik kesimpulan.

Tetapi karena dalam penelitian ini tidak terfokus dalam satu objek penelitian maka disini tidak akan dilampirkan objek penelitiannya.

3.2 Metode Penelitian

Menurut Sugiono (2009:2) menyatakan bahwa, definisi metode penelitian adalah, sebagai berikut:

"Metode penelitian merupakan cara ilmiah untuk mendapatkan data yang valid dengan tujuan dapat ditemukan, dibuktikan dan dikembangkan suatu pengetahuan sehingga pada gilirannya dapat digunakan untuk memahami, memecahkan dan mengantisifikasi masalah."

Berdasarkan definisi diatas, dapat ditarik kesimpulan bahwa metode penelitian merupakan cara ilmiah yang mendapatkan data dengan tujuan dan kegunaan tertentu.

3.2.1 Desain Penelitian

Dalam melakukan suatu penelitian sangat diperlukan perencanaan dan perancangan dalam penelitian, agar penelitian dapat berjalan dengan lancar, baik dan sistematis.

Menurut Jonathan Sarwono (2006:79) pengertian desain penelitian memiliki pengertian sebagai berikut:

"Desain penelitian bagaikan sebuah peta jalan bagi peneliti yang menuntun serta menentukan arah berlangsungnya proses penelitian secara benar dan tepat sesuai dengan tujuan yang telah ditetapkan."

Berdasarkan definisi diatas maka dapat ditarik kesimpulan bahwa desain penelitian merupakan rencana dan struktur penyelidikan terhadap pengumpulan data sehingga dapat menjawab pertanyaan dalam penelitian.

Dalam melakukan penelitian diperlukan melakukan perancangan dan perencanaan. Maka langkah-langkah dalam penelitian ini adalah sebagai berikut:

1. Menetapkan judul yang akan diteliti, sehingga dapat diketahui apa yang akan diteliti dan menjadi masalah dalam penelitian.
2. Menetapkan masalah-masalah yang akan dianalisis.
3. Memberi definisi terhadap pengukuran variable
4. Memilih teknik pengumpulan data.

Teknik pengumpulan data yang digunakan dalam penelitian ini dengan menggunakan 2 cara, yaitu pengumpulan data melalui penelitian lapangan seperti wawancara, observasi, dokumentasi dan penelitian kepustakaan atau data yang di peroleh dari sumber lain, seperti buku, literatur, ataupun catatan-catatan perkuliahan.

5. Pelaporan hasil penelitian termasuk proses penelitian dan interpretasikan data.

3.2.2 Jenis dan Metode Pengumpulan Data

Dalam prosedur pengumpulan data untuk penelitian ini penulis memperoleh data-data dari dua sumber yaitu :

1. **Sumber Data Primer**

   Data primer adalah data yang langsung dapat dan dijadikan sebagai sumber dari penelitian dan pengamatan secara langsung, dimana dilakukan dengan cara penelitian lapangan melalui observasi dan wawancara dengan pihak yang berkepentingan.

2. **Sumber Data Sekunder**

   Data sekunder adalah data tidak langsung yang didapat dan dijadikan sebagai sumber informasi, dimana data sekunder ini dapat penulis proleh dari studi kepustakaan dan pengumpulan data dari literature-literature serta sumber lain yang berhubungan dengan masalah yang sedang diteliti oleh penulis.

3.2.3 Metode Pendekatan

Dalam melakukan penelitian perlu melakukan metode pendekatan, metode pengembangan sistem dan memerlukan alat bantu analisis.

3.2.3.1 Metode Pendekatan Sistem

Proses pemecahan masalah secara sistematis bermula dari John Dewey, seorang professor pilosofi di Columbia Univercity pada awal abad ini. Dalam bukunya 1910 diidentifikasi ada 3 (tiga) seri penilaian dalam memecahkan suatu kontoversi yang memadai:

1. Mengenali kontroversi
2. Menimbang klaim alternative
3. Membentuk penilaian

Metode pendekatan sistem yang akan digunakan dalam penelitian ini adalah pendekatan OOP (Object Oriented Programming). Pendekatan OOP (Object Oriented Programming) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek, dimana setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Pendekatan objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem.

Pendekatan Objek memiliki beberapa keuntungan, antara lain:

1. Maintenance. Program lebih mudah dibaca dan dipahami.

2. Pengubahan program (berupa penambahan ataupun penghapusan fitur tertentu). Perubahan ini antara lain menyangkut penambahan dan penghapusan dalam suatu database program misalnya.
3. Dapat digunakannya objek-objek sesering yang diinginkan.

3.2.3.2 Metode Pengembangan Sistem

Dalam perancangan sebuah sistem hal yang sangat penting dan harus diperhatikan agar menghasilkan suatu sistem yang baik dan dapat melengkapi informasi yang dibutuhkan merupakan kajian teknis formal. Untuk itu digunakan metode-metode perancangan sistem. Pengembangan sistem yang penulis gunakan dalam penelitian ini adalah dengan menggunakan pendekatan model SDLC (*System Development Life Cycle*).

Dikutip dari http://id.wikipedia.org/wiki/SLDC/09_Januari_2012 *"Siklus hidup pengembangan sistem (Systems Development Life Cycle) adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi."*

Secara konseptual siklus pengembangan sebuah sistem informasi adalah sbb :

1. **Analisis Sistem**: menganalisis dan mendefinisikan masalah dan kemungkinan solusinya untuk sistem informasi dan proses organisasi.

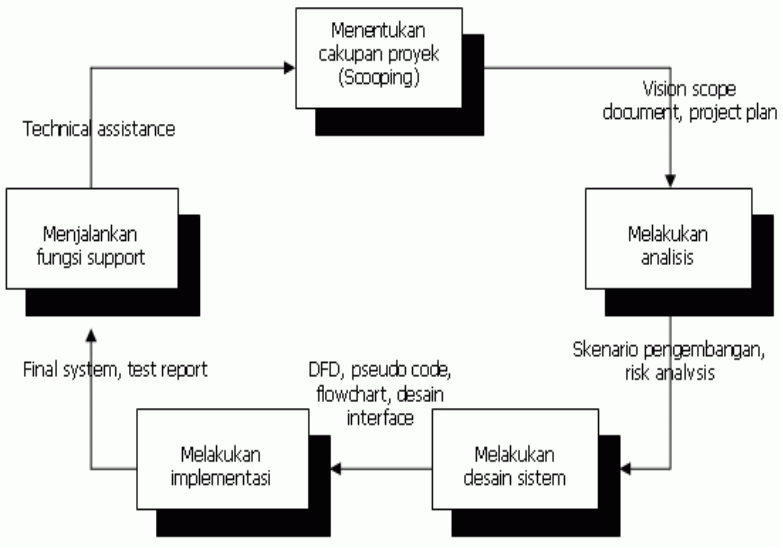
2. **Perancangan Sistem**: merancang output, input, struktur file, program, prosedur, perangkat keras dan perangkat lunak yang diperlukan untuk mendukung sistem informasi
3. **Pembangunan dan Testing Sistem**: membangun perangkat lunak yang diperlukan untuk mendukung sistem dan melakukan testing secara akurat. Melakukan instalasi dan testing terhadap perangkat keras dan mengoperasikan perangkat lunak
4. **Implementasi Sistem**: beralih dari sistem lama ke sistem baru, melakukan pelatihan dan panduan seperlunya.
5. **Operasi dan Perawatan**: mendukung operasi sistem informasi dan melakukan perubahan atau tambahan fasilitas.
6. **Evaluasi Sistem**: mengevaluasi sejauih mana sistem telah dibangun dan seberapa bagus sistem telah dioperasikan.

Siklus tersebut berlangsung secara berulang-ulang. Siklus di atas merupakan model klasik dari pengembangan sistem informasi. Model-model baru, seperti prototyping, spiral, 4GT dan kombinasi dikembangkan dari model klasik di atas.

Dalam penerapan tahapan pengembangan sistem informasi penulis menggunakan sistem pengembangan *prototype*. *Prototyping* merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya

mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detal output apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efesiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer.

Untuk mengatasi ketidakserasian antara pelanggan dan pengembang , maka harus dibutuhakan kerjasama yanga baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan dengan tidak mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalm menyelasaikan system yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.



Gambar 3.1 Proses Prototype

(Sumber : http://www.ristinet.com/2010/04/16)

1. **Pengumpulan kebutuhan**

   Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. **Membangun *prototype***

   Membangun prototyping dengan membuat perancangan sementara yang berfokuspada penyajian kepada pelanggan (misalnya dengan membuat input dan format output)

3. **Evaluasi *protoptyping***

   Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah 4 akan diambil.Jika tidak prototyping direvisi dengan mengulang langkah 1, 2 , dan 3.

4. **Mengkodekan Sistem**

   Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai

5. **Menguji Sistem**

   Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box.

6. **Evaluasi Sistem**

   Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. **Menggunakan system**

   Perangkat lunak yang telah diuji dan diterima pengguna siap untuk digunakan .

Beberapa keunggulan prototype yang membuat penulis menggunakan metode ini, diantaranya :

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Pelanggan berperan aktif dalam pengembangan sistem
4. Lebih menghemat waktu dalam pengembangan sistem
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Adapun kelemahan dari prototype :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jenjang waktu lama.

2. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem .
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik

3.2.3.3 Alat Bantu Analisis dan Perancangan

Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegoisasikan, dapat dikatakan kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented adalah UML.

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah system blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam *'MS Word'* untuk kegunaan komunikasi. Sebuah

bahasa model adalah sebuah bahasa yang mempunyai vocabulary dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software.

UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (common).

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; *use-case*, *class*, *object*, *state*, *sequence*, *collaboration*, *activity*, *component*, dan *deployment diagram*. Diagram pertama

adalah *use case* menggambarkan sekelompok *use cases* dan aktor yang disertai dengan hubungan diantaranya. Diagram *use cases* ini menjelaskan dan menerangkan kebutuhan / requirement yang diinginkan/ dikehendaki *user*/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.

Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented adalah UML. Alasan mengapa UML digunakan adalah, pertama, scalability dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, dynamic modeling, dapat dipakai untuk pemodelan sistem dinamis dan real time.

Sebagaimana dalam tulisan pertama, penulis menjelaskan konsep mengenai obyek, OOA&D (Obyek Oriented Analyst/ Design) dan pengenalan UML, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana UML digunakan dalam merancang sebuah pengembangan software yang disertai gambar atau contoh dari sebuah aplikasi.

3.2.4 Pengujian Software

Software testing adalah investigasi yang dilakukan kepada software untuk mengetahui kualitas suatu produk. dilakukan untuk mengeliminasi defect yang terjadi setelah software diluncurkan, pengujian software sendiri tidak akan

pernah secara penuh mengenali defect pada suatu software.black box testing dan white box testing adalah metode yang dikenal untuk menguji software.

Black box testing mengambil perspektif dari luar untuk melakukan testing, tes ini bisa berupa fungsional test atau non-fungsional, dilakukan dengan cara memasukkan input dengan benar maupun salah, kemudian menguji apakah outputnya sesuai dengan spesifikasi atau tidak, hal ini dilakukan tanpa harus tahu apa yang terjadi didalam mengolah input untuk mengasilkan output. black box testing popular karena kesederhanaan metodenya.

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

Pengujian black box berusaha menemukan kesalahan dalam kategori :

1. fungsi-fungsi yang tidak benar atau hilang
2. kesalahan interface
3. kesalahan dalam struktur data atau akses database eksternal
4. kesalahan kinerja
5. inisialisasi dan kesalahan terminasi.