BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Penelitian ini dalakukan di Adolescence di jln. ir.H.juanda no 61 - 65. Lantai 3 no 8 Dago Plaza Bandung, pada bagian penjualan,Objek penelitian merupakan sasaran utama dalam kegiatan penilitian untuk dilakukan tahap-tahap atau langkah-langkah dalam metode pendekatan. Objek Penelitian pada perusahaan meliputi sejarah perusahaan, struktur organisasi, visi dan misi perusahaan serta deskripsi kerja yang ada di lingkungan Adolescence pada bagian penjualan yang akan dijelaskan di bawah ini.

3.1.1. Sejarah Singkat Perusahaan

Adolescence berdiri tahun 2005 dan memiliki moto shoping shoping shoping shoping sesuai dengan lokasi yang berada ditempat yang cukup strategis. Adolescence adalah distro untuk kaum laki-laki dan perempuan. Adolescence diambil dari bahasa ingris *adolescence* yang artinya keremajaan atau masa remaja untuk membedakan tempat belanja anak muda atau remaja.

Pada awalnya Adolescence ini bukan bernama Adolescence tapi bernama storage yang berasal dari bahasa ingris yang artinya penyimpanan. Storage ini berdiri pada tahun 2000, berganti nama dari storeage menjadi adolcscence sejak tahun 2005. Alasan berubahnya nama yaitu karena pemilik meminta diganti nama karena dianggap kurang ngetren dizaman yang sekarang ini. Adolescence ini dimiliki oleh pemilik tempat Adolescence sendiri yaitu pemilik dago plasa, bisa juga dikatakan Adolescence ini sebagai anak perusahaan dari dago plasa, yang

berada didalam dago plaza sendiri yang dipimpin dalam satu manajemen. Kantor Adolescence berada didalam dago plaza yang beralamat sama dengan dago plaza yaitu jln ir.H.juanda no 61 - 65. Dago Plaza Bandung bersama dengan ruangan pemilik dago plaza sendiri. Karena Adolescence ada dalam satu manajemen dago plaza sendiri, Adolescence deberikan perhatian lebih oleh pemilik dago plaza yang banyak menguntungkan. Adolescence dimliki oleh Bpk.Ardi Santoso yang merupakan pemilik tunggal, di dalamnya memperkerjakan beberapa orang yang bekerja yaitu ada satu supervisor dan ada lima shoop keepor.

3.1.2. Visi dan Misi Perusahaan

Visi

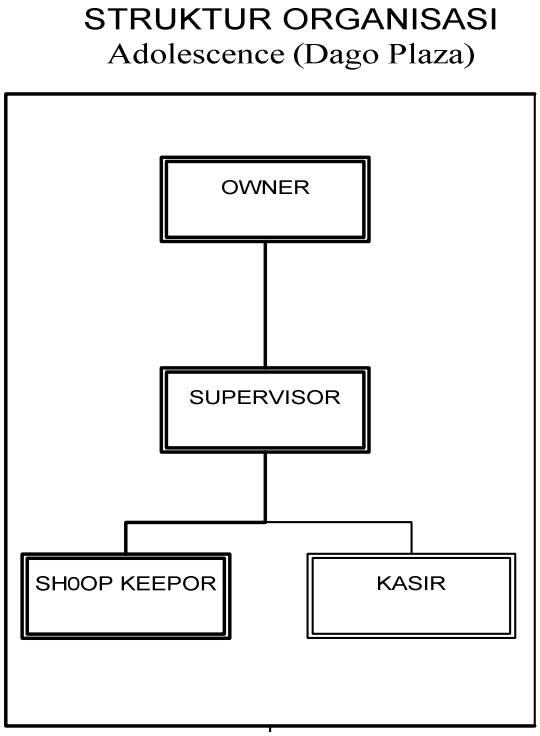
Untuk manjadikan perusahaan terkenal yang menghadirkan kualitas distro terbaik.

Misi

a. Untuk melebihi harapan pelanggan dengan menyediakan produk dan pelayanan terbaik dan berkualitas dengan harga yang seimbang dan sesuai.
b. Menetapkan suatu hubungan baik penyuplai dan pelanggan.

3.1.3. Struktur Organisasi Perusahaan

Struktur organisasi menunjukkan bagaiman bagian-bagian di dalam organisasi di koordinasikan bersama-sama melalui suatu jalur wewenang dan tanggung jawab. Struktur organisasi adalah penggambaran secara grafik yang menggambarkan struktur kerja dari suatu struktur organisasi. Berikut ini adalah struktur organisasi yang ada di Adolescence (Dago Plaza).



Gambar 3.1 Struktur Organisasi Adolescence

3.1.4. Deskripsi Tugas

Deskripsi Tugas *(Job description)* adalah suatu pernyataan tertulis yang berisi uraian atau gambaran tentang apa saja yang harus dilakukan oleh si pemegang jabatan, bagaimana suatu pekerjaan dilakukan dan mengapa pekerjaan tersebut dilakukan. Berikut ini adalah uraian kerja yang ada di Adolescence (Dago Plaza).

1. Owner

Adapun tugas owner yaitu:

a. Mengambil keputusan setiap kegiatan di Adolescence.

b. Menerima laporan penjualan dan persediaan barang.

2. Super Visor

Adapun tugas Super Visor yaitu:

a. Mengawasi kegiatan shoop keeper.
b. Melakukan pemesanan barang kepada supplier jika stok barang telah habis.
c. Melakukan pengemabalian barang kepada supplier jika stok barang yang tersisa untuk diretur.
d. Melakukan verifikasi laporan penjualan barang, laporan persediaan barang dan pemesanan barang.

3. Shoop Keeper

Adapun tugas Shoop Keeper yaitu:

a. Melayani Customer.
b. Mengelola persediaan dan pembelian barang.

4. Kasir

a. Melayani Customer.
b. Mengelola transaksi penjualan, persediaan dan pembelian barang.
c. Membuat laporan penjualan, persediaan dan pembelian barang.

3.2. Metode penelitian

Metode penelitian yang digunakan dalam pelaksanaan penelitian terdiri dari dua metode yaitu pengumpulan data dan metode Pendekatan dan pengembangan sistem.

3.2.1. Desain penelitian

Penyusunan penelitian ini penulis menggunakan metode *deskriptif* . Metode deskriptif adalah suatu metode dalam meneliti status sekelompok manusia, suatu

objek, suatu set kondisi, suatu sistem pemikiran, ataupun suatu kelas peristiwa pada masa sekarang. Tujuan dari penelitian deskriptif ini adalah untuk membuat deskripsi, gambaran atau lukisan secara sistematis, fakual dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antara fenomena yang diselidiki.

3.2.2. Jenis dan Metode Pengumpulan Data

Metode pengumpulan data merupakan suatu teknik atau cara untuk memperoleh, mencari, mengumpulkan dan mencatat data yang di gunakan untuk menyusun skripsi. Adapun data itu sendiri terbagi menjadi dua yaitu data primer dan data sekunder.

3.2.2.1. Sumber Data Primer

Data primer adalah data yang di kumpulkan oleh penulis secara langsung melalui objek penelitian, yaitu dengan cara observasi, wawancara, mengamati dan mencatat.

a. Observasi

Observasi adalah suatu cara untuk mengumpulkan data dengan melakukan penelitian secara langsung datang ke Di Adolescence (Dago Plaza) hal ini untuk mengamati dan pencatatan terhadap peristiwa yang sedang di selidiki pada objek penelitian, dan peneliti melakukan observasi dibagian penjualan dan mengamati sistem penjualan yang saat ini sedang berjalan.

b. Wawancara

Suatu cara pengumpulan data melalui tanya jawab secara langsung antara peneliti (pengumpul data) dengan responden (sumber data), dalam hal ini wawancara dikakukan dengan responden yaitu pada pegawai yang bekerja

sebagai penjaga toko (*shoop keeper*) Di Adolescence yang berhubungan langsung dengan sistem informasi penjualan yang sedang berjalan.

3.2.2.2. Sumber Data Sekunder

Data sekunder merupakan data primer yang telah diolah lebih lanjut dan disajikan dengan baik oleh pengumpul data primer atau pihak lain. Data primer disajikan antara lain data barang, data penjualan yang tersedia di Adolescence yang merupakan hasil sitem penjualan yang saat ini berjalan dalam bentuk tabel – tabel / diagram atau segala informasi yang berasal dari literatur yang ada hubungannya dengan teori – teori mengenai topik penelitian yang dalam bentuk buku – buku sebagai landasan teori yang dipakai.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan sistem yang digunakan oleh penulis untuk merancang sebuah sistem informasi penjualan secara oneline di Adolescence adalah dengan pendekatan berorientasi objek dan pengembangan secara *water fall*.

3.2.3.1 Metode Pendekatan Sistem

Menurut Bambang Hariyanto (2004:79) Metode adalah prosedur untuk melakukan sesuatu, sedang metodologi adalah studi mengenai metode. Metode – metode perancamgan sendiri berovolusi. Objek Oriented memandang software bagian perbagian dan menggambar satu bagian tersebut dalam satu objek. Pendekatan berorientasi objek terdiri dari analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD).

A. Teknik Dasar OOA/D *(Object-Oriented Analysis/Design)*

(http://www.ilmukomputer.com/*Architecture and Design : Unified Modeling Language (UML),/*14Mei 2010)

Dalam dunia pemodelan, metodologi implementasi obyek walaupun terikat kaidah-kaidah standar, namun teknik pemilihan obyek tidak terlepas pada subyektifitas *software analyst and designer*. Beberapa obyek akan diabaikan dan beberapa obyek menjadi perhatian untuk diimplementasikan di dalam sistem. Ada 3 (tiga) teknik/konsep dasar dalam OOA/D, yaitu pemodulan (*encapsulation*), penurunan (*inheritance)* dan *polymorphism.*

a. Pemodulan (*Encapsulation*)

Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan *menggunakan rice cooker*, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding*.

b. Penurunan (*Inheritance*)

Obyek-obyek memiliki banyak persamaan, namun ada sedikit perbedan. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dapat dikatakan sebagai obyek induk (*parent*). Sedangkan

minibus dikatakan sebagai obyek anak (*child*), hal ini juga berarti semua operasi yang berlaku pada mobil berlaku juga pada minibus.

c. *Polymorphism*

Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup / pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

B. *Object Oriented*

(http://www.cetus-links.org/oo_uml.html/ *Architecture and Design : Unified Modeling Language (UML),*/14Mei 2010)

Object Oriented Analysis dan *Design* (OOA&D) merupakan pendekatan yang menekankan pada solusi *logic* berbasis objek. Teknologi *Object Oriented* merupakan paradigma baru dalam rekayasa perangkat lunak yang di dasarkan pada objek dan kelas. Pendekatan ini dibagi menjadi 2 (dua) yaitu :

1. *Object Oriented Analysis* (OOA)

Analisis mempunyai beberapa definisi, yaitu :

a. Pemecahan, pembagian (dekomposisi) dari sesuatu yang bersifat menyeluruh/global *(whole)* menjadi bentuk bagian *(parts).*

b. Memisahkan atau memecahkan sejumlah *whole* ke dalam *part* sehingga dapat menemukan habitatnya, proporsi, fungsi dan relasi.
c. Kegiatan yang mempelajari suatu domain permasalahan, menemukan spesifikasi dan tingkah laku *eksternal* yang dapat diamati secara lengkap, konsisten dan *feasible* terhadap pernyataan yang diperlukan. Hal yang melingkupi fungsi dan sejumlah karakteristik oprasional.

Dalam *Object Oriented Analysis* (OOA) terdapat lima lapisan kegiatan (tidak bersifat *sequensial*), yaitu :

1. Menetukan lapisan subjek
2. Menetukan lapisan kelas dan objek
3. Menetukan lapisan struktur
4. Menetukan lapisan *attribute*
5. Menetukan lapisan service

2.*Object Oriented Design* (OOD)

Desain mempunyai beberapa definisi, yaitu:

a. Menandai, membuat rencana besar, sketsa, pola.
b. Kegiatan untuk mengambil dan menentukan spesifikasi dari prilaku luar yang dapat diamati dan menambah detail yang diperlukan untuk implementasi sistem komputer (secara nyata), termasuk interaksi manusia, pengolahan tugas/aksi, pengolahan data secara detail.

C. Konsep dasar OOAD

1. Objek

   Objek *(object)* adalah "benda", secara fisik atau konseptual yang dapat kita temui diseliling kita. *Hardware, software*, dokumen, manusia, dan bahkan konsep semuanya adalah contoh objek.

2. Kelas

   Kelas (*class*) adalah definisi umum (pola atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku dan atribut objek-objek tersebut. Objek adalah contoh dari sebuah kelas.

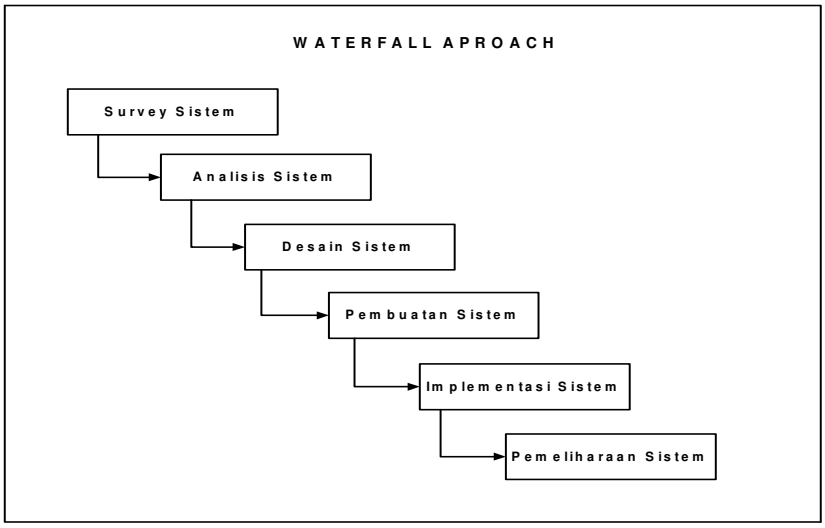
3. *Association* dan *aggregation*

   Association (asosiasi) adalah hubungan antar objek yang saling membutuhkan. Sedangkan, *aggregation* (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu objek dari objek lainnya.

3.2.3.2. Metode Pengembangan Sistem

Metodologi yang digunkan penulis dalam mengembngkan system informasi menggunakan metode *the Classic Life Cycle* atau pada umumnya dikatakan paradigma *water fall*. Pada metode ini terdapat 5 (lima) tahap untuk mengembangkan suatu perangkat lunak. Kelima tahapan itu tersusun dari atas kebawah , diantaranya *Analisis, Design, Testing,* dan *Maintenance*. Konsep dari metode ini adalah melihat bagaimana suatu masalah secara sistematis dan terstruktur dari atas kebawah.

Tahap-tahap pengembangan perangkat lunak metode *water fall* dapat dilihat dibawah ini:



Gambar 3.2. Tahap-Tahap Metode *Waterfall*

(Sumber : Al Bahra bin Ladjamuddin , Rekayasa Perangkat Lunak, 2006, Graha Ilmu: Yogyakarta)

Berikut ini akan diuraikan tahap-tahap pengembangan perangkat lunak dengan menggunakan metode *waterfall,* yaitu:

1. *Analisis* adalah tahapan menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan atau pengembangan *software*. Dalam hal ini analisis yang dilakukan dengan menganalisa dokumen-dokumen yang digunakan pada bagian penjualan dan prosedur pelaksanaanya.
2. *Desaign* adalah penterjemahan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai..

3. *Coding* adalah tahap penterjemah data/pemecahan masalah *oftwares* tang telah dirancang dalam bahasa pemograman yang telah ditentukan. Pada tahap ini bahasa pemograman yang digunakan dalam pembuatan sistem adalah *php*.
4. *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak untuk system informasi persediaan barang yang mencakup penjualan dan pembelian kemudian dalanjutkan pengujian terhadap modul-modul dan terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengans baik dan input yang deberikan hasilnya sesuai dengan yang diinginkan.
5. *Maintenance* adalah perangkat yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dipakai jika ada permintaaan penambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertubuhan dan perkembangan baik perngakat lunak maupun perangkat keras.

3.2.3.3 Alat Bantu Analisis dan Perancangan

Menurut Bambang Hariyanto (2004:369), dinyatakan oleh Tom DeMarco Analisis (sistem) adalah hal yang dapat membuat frustasi, penuh dengan hubungan antar manusia yang kompleks, tak tentu dan sulit. Dalam satu kata, ini menakjubkan. Sekali anda terlibat, kesenangan mudah dari pembangunan sistem yang lama tidak memadai untuk memuaskan anda.

Metode analisis yang digunakan adalah Coad-Yourdan yang memiliki langkah – langkah yang dilakukan adalah:

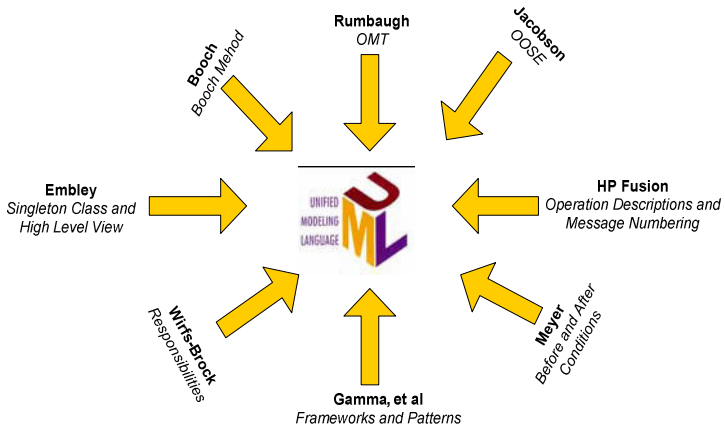
1. Identifikasi kelas – kelas dan objek – objek menggunakan kriteria "what to look for" – apa yang dicari.
2. Mendifinisikan struktur generalisasi/spesialisasi.
3. Mendifinisikan struktur whole-part.
4. Mengidentifikasikan subject (representasi dari subsistem)
5. Mendifinisikan atribut
6. Mendifinisikan layanan

A. *Unified Modelling Language* (UML)

Dalam penulisan skripsi kali ini penulis menggunakan alat bantu dengan UML dan perancangan dengan menggunakan perancangan berorientasi objek adlah analisis yang dilakukan Coad-Yourdon.

(http://www106.ibm.com/developerworks/rational/library/998.html/ *Introduction to The Unified Modeling Language* / 14 Mei 2010)

UML merupakan bahasa pemodelan yang menggunakan konsep orientasi *object*. UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacis (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuataan perangkat lunak) dari sistem perangkat lunak, seperti pemodelan bisnis dan sistem non perangkat lunak lainnya. Berikut ini adalah gambar unsure-unsur yang membentuk UML.



Gambar 3.3 Unsur-unsur pembentuk UML

(Sumber. Quatrani, T., (2003), *Introduction to The Unified Modeling Language*, DeveloperWorks, IBM, available from)

Bagian-bagian utama dari UML terdiri dari :

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstrak yang berisi sejumlah diagram. Beberapa jenis *view* yang terdapat dalam UML sebagai berikut:

1. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *View* ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams.*

2. Logical View

Logical view digambarkan dalam class diagram untuk struktur statis dan *view* ini digunakan untuk perancang dan pengembang.

3. Component View

Mendeskripsikan implementasi dan ketergantungan model. Komponen yang merupakan tipe lainnya dari kode model diperhatikan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi lainnya.

4. Concurrency View

Membagi sistem kedalam proses dan prosesor. Serta digunakan untuk pengembangan, pengintegrasi, dan penguji atau tester (contoh).

5. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat dan bagian hubungannya dengan lainnya.

b. Diagram

Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Adapun jenis diagram antara lain :

a. *Use Case* Diagram

Menggambarkan sejumlah *external actor* dan berhubungannya ke *use case* yang diberikan ke sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk *text* sebagai dokumentasi *dari use case* simbol namun dapat juga dilakukan dalam *activity diagrams.*

b. *Class* Diagram

Menggambarkan struktur statis *class* didalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem.

c. *State* Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa object lain yang mengirim pesan.

d. *Sequence* Diagram

Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaanya untuk menunjukan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. *Collaboration* Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams.* Dalam menunjukan pertukaran pesan, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks).

f. *Activity* Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu oprasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

g. *Component* Diagram

Menggambarkan struktur fisik kode dari komponen. komponen dapat berupa *source kode*, *component biner* atau *executable component*. Sebuah component berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

h. *Deployment* Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak. Menunjukan hubungan komputer dengan perangkat satu sama lain dan jenis hubungannya.

3.2.4 Pengujian Software

Setelah melakukan pengimplementasian aplikasi perangkat lunak, kegiatan selanjutnya adalah pengujian sistem yang telah dibangun. Pengujian ditujukan untuk menemukan kesalahan-kesalahan pada sistem dan memastikan sistem yang dibangun telah sesuai dengan apa yang direncanakan sebelumnya. Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi analisis, perancangan dari perangkat lunak itu sendiri. Pengujian perangkat lunak ini menggunakan metode pengujian *black box.*

Menurut Al Bahra bin Ladjamuddin (2006:379) Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat. Rancangan pengujian yang akan dilakukan adalah dengan menggunakan metode pengujian *black box*. Pengujian dengan metode *black box* menitikberatkan pada fungsi sistem . 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.