BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

TK AL - MAQOM Cimahi adalah tempat yang dijadikan objek penelitian oleh penulis. TK AL - MAQOM Cimahi beralamat di Jalan Pesantren VI No.223 kelurahan cibabat kecamatan cimahi utara 40513 Telp 022 – 76327412.

3.1.1 Sejarah TK AL - MAQOM

Taman Kanak-kanak AL _ MAQOM TK AL - MAQOM didirikan pada tahun 1995 yang tepatnya berlokasi dirumah sesepuh pesantren AL – MAQOM. Pendirinya antara lain : K H Sulaiman Kurnia (alm) dia adalah sesepuh pesantren AL – MAQOM, pendiri yang selanjutnya adalah R H Wawan Ridwan Syamsudin SPDI, ia adalah anak dari K H Sulaiman Kurnia, dan yang selanjutnya Ibu Rika Kartiani (alm), ia adalah istri dari R H Wawan Ridwan Syamsudin SPDI. Pertama kali sekolah ini berdiri di kepalai oleh R H Wawan Ridwan Syamsudin SPDI ,namun karena kesibukan jabatan tersebut diserahkan kepada istrinya Ibu Rika Kartiani pada tahun 2000. Alasan kenapa TK AL – MAQOM didirikan adalah banyaknya permintaan dari masyarakat yang ingin menyekolahkan anak usia pra sekolah dilingkungan sekitar.

3.1.2 Visi Misi

Adapun Visi Misi dari TK AL - MAQOM adalah :

Visi

"Membentuk generasi yang taqwa, berakhlakul karimah, islami, cerdas, berkualitas serta unggul dalam berprestasi."

Misi

1. Mampu menjalankan perintah-Nya dan menjauhi larangan-Nya.
2. Mampu berprilaku baik, benar dan sopan.
3. Dapat berkomunikasi dengan bahasa yang santun
4. Mampu melakukan kegiatan sendiri serta memiliki rasa percaya diri.
5. Mampu menulis dan membaca sesuai dengan usia, dalam rangka persiapan memasuki jenjang yang lebih tinggi.
6. Dapat mengoptimalkan tempat bermain dan belajar.

3.1.3 Tujuan TK AL - MAQOM

Secara umum TK AL – MAQOM mempunyai tujuan untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi dirinya untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, ahlak yang mulia serta keterampilan yang diperlukan dirinya, masyarakat, bangsa, dan negara. Sedangkan secara khusus yang menjad tujuan sekolah, adalah :

1. Membantu pemerintah dalam upaya peningkatan kualitas pendidikan di masyarakat.
2. Mencerdaskan kehidupan bangsa.

3. Menyediakan sarana pendidikan yang mudah dijangkau oleh masyarakat setempat, baik dalam hal jarak juga biaya.
4. Memperkenalkan suasana kepesantrenan kepada masyarakat umum.
5. Menciptakan generasi islami.
6. Menghasilkan anak agar siap memasuki jenjang pendidikan yang lebih tinggi (SD/MI).
7. Mencetak anak agar berakhlak karimah serta menjadi panutan bagi lingkungan sekitar.
8. Menciptakan kecerdasan anak yang seimbang antara SQ, EQ, IQ.
9. Menyediakan sarana bermain bagi anak yang bersifat mendidik.

Profil Sekolah

NamaTK	:	TK AL – MAQOM
Status Tk	:	Swasta
Alamat	:	Jl. Pesantren VI RT 06 RW 15 No. 223
Kode Pos	:	40513
Desa	:	Cibiru Wetan
Kecamatan	:	Cimahi Utara
Kelurahan	:	Cibabat
No. Tlp. Yang dapat dihubungi	:	022-76327412
A. Tahun Pendirian	:	1995
B. Tahun Beroperasi	:	1995

C. Jumlah anak didik dalam 2 tahun terakhir :

Tabel 3.1 Jumlah Anak Didik

Kelompok	Jumlah Anak Didik		Keterangan
	2010/2011	2011/2012	
A	16	16	
B	24	35	
Jumlah	40	51	

Tenaga Kependidikan

Tabel 3.2 Tenaga Kependidikan

Nama Guru	No NUPTK	Pendidikan Terakhir	Jabatan
1. R Leni Herliani	5144756658300063	SMK	Kepala Sekolah
2. H.R Wawan Ridwan	1258759652200033	S-1	Guru PAI
3. Jauhariah Ernawati	0839749651300082	SLTA	Guru kelas B
4. Siti Fatimah	8556765666300013	SLTA	Guru
5. Siti Rohmah	1041769671300003	SLTP	Guru Kelas A
6. Linda Handayani	-	SLTA	Guru
7. Lusinah	7563763665210343	Akademik	Guru Kelas C
8. Yosi Pratiwi	-	SLTA	Guru
9. Yuni Syahriah Khotimah	-	SMK	Guru

Potensi Yang Pernah Diraih oleh Siswa dan Guru

Tabel 3.3 Potensi Yang Pernah Diraih

No	Tahun	Jenis Lomba	Peserta	Prestasi	Keterangan
1	2006	Lomba gigi sehat	-	Juara ke 2	-
2	2006	Lomba mengisi pola dengan garis	-	Juara ke 1	LKA
3	2006	Lomba mengisi pola dengan garis	-	Harapan 3	LKA
4	2007	Lomba busana muslim putra	-	Juara ke 2	LKA
5	2008	Lomba Tahfidz putra	-	Juara ke 1	LKA
6	2008	Lomba Tahfidz putri	-	Juara ke 2	LKA
7	2009	Lomba Tahfidz putri	-	Juara ke 3	LKA
8	2010	Lomba cerdas cermat	-	Juara ke 1,2,3,4	Tingkat Cimut
9	2011	Lomba Tahfidz putri tingkat kota	Aluna	Juara ke 1	LKA

3.2 Metodologi Yang Digunakan

Metode penelitian adalah suatu teknik atau cara mengumpulkan, memperoleh data, baik berupa data primer maupun sekunder yang digunakan untuk keperluan menyusun suatu karya ilmiah dan kemudian menganalisa faktor-faktor yang berhubungan dengan pokok-pokok permasalahan sehingga terdapat

suatu kebenaran data-data yang akan diperoleh. Adapun pengertian metode penelitian menurut Sugiyono (2010 : 2), diantaranya adalah sebagai berikut : *" Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu"*.

3.2.1 Desain Penelitian

Dalam melakukan suatu penelitian sangat perlu dilakukan perencanaan dan perancangan penelitian, agar penelitian yang dilakukan dapat berjalan dengan baik dan sistematis.

Desain penelitian menurut Moh. Nazir (2003 : 84) memaparkan bahwa "Desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian".

Dari definisi di atas maka dapat dikatakan bahwa desain penelitian merupakan semua proses penelitian yang dilakukan oleh penulis dalam melaksanakan penelitian mulai dari perencanaan sampai dengan pelaksanaan penelitian yang dilakukan pada waktu tertentu.

3.2.2 Jenis dan Metode Pengumpulan Data

Metode pengumpulan data adalah penelitian lapangan (*field Research*), dilakukan dengan cara mengadakan peninjauan langsung pada instansi yang menjadi objek untuk mendapatkan data primer (data yang diperoleh dari TK AL - MAQOM) dan data sekunder.

3.2.2.1 Sumber Data Primer (Wawancara, Observasi)

Data primer merupakan data yang diambil secara langsung, data ini diperoleh dari kegiatan observasi yaitu pengamatan langsung pada objek penelitian dan mengadakan wawancara dengan pihak terlibat.

a. Observasi

   Dilakukan dalam bentuk pengamatan secara langsung terhadap pelaksanaan pembuatan sebuah aplikasi tutorial itu sendiri.

b. Wawancara

   Pengumpulan data yang dilakukan dengan cara tanya jawab dengan unsur individu (guru) dan (orang tua murid) yang terlibat guna mendapaktan data yang efesien dan akurat.

3.2.2.2 Sumber Data Sekunder (dokumentasi)

Dengan memperoleh dokumen – dokumen yang bersangkutan dengan obyek yang diteliti, yang dimaksudkan sebagai bukti bahwa penelitian benar – benar dilakukan pada perusahaan yang bersangkutan.

3.2.3 Metoda Pendekatan Sistem dan Pengembangan Sistem

3.2.3.1 Metode Pendekatan Sistem

Dalam melakukan penelitian ini, penulis menggunakan pendekatan sistem prototipe. Dimana di dalamnya terdapat identifikasi komponen – komponen yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau user mengenai sistem.

Adapun tahapan perancangan adalah agar pada akhir pengembangan perangkat lunak dapat memenuhi kebutuhan *user*, efektif, efisien, mudah digunakan, Tahapan perancangan:

1. Penyusunan Proposal
2. Pengumpulan Data
3. Analisis Sistem
4. Design Sistem
5. Codding
6. Testing
7. Implementasi

3.2.3.1.1 OOP (Object Oriented Programing)

OOP adalah suatu model pemrograman yang berorientasi pada obyek. Maksudnya adalah, bahwa setiap bagian dari apa yang anda program -visual maupun non visual- adalah sebuah obyek. Obyek, dalam konteks ini adalah "sesuatu", bisa dianalogikan dengan benda, yang padanya melekat attribut-attribut. Attribut tersebut bisa dikelompokkan ke dalam 2 jenis yaitu kelompok "property" dan kelompok "method" atau "event"(Y. Wibisono).

OOP adalah paradigma pemrograman yang cukup dominan saat ini, karena mampu memberikan solusi kaidah pemrograman modern. Meskipun demikian, bukan berarti bahwa pemrograman prosedural sudah tidak layak lagi (Efendi Hariyadi).

OOP diciptakan karena dirasakan masih adanya keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah, semua pemecahan masalah dibagi ke dalam objek. Dalam OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai objek. Proses perancangan atau desain dalam suatu pemrograman merupakan proses yang tidak terpisah dari proses yang mendahului, yaitu analisis dan proses yang mengikutinya. Pembahasan mengenai orientasi objek tidak akan terlepas dari konsep objek seperti inheritance atau penurunan, encapsulation atau pembungkusan.

Empat konsep yang dimiliki oleh pemograman yang berorientasi object adalah :

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

3.2.3.1.1.1 Encapsulation

Encapsulation adalah merupakan penyembunyian informasi melalui private dan protected. Pada dasarnya adalah kita tidak perlu mengetahui apa yang ada dalam objek itu (class), yang kita butuhkan adalah apa kegunaan, bagaimana cara memakainya dan apa yang akan terjadi.

3.2.3.1.1.2 Abstraction

Abstraction memiliki pengertian bahwa dalam perangkat lunak seorang programmer akan menggunakan atau menciptakan sejumlah objek dimana objek tersebut ada dan berprilaku pada sistem.

3.2.3.1.1.3 Inheritance

Inheritance atau pewarisan adalah merupakan suatu objek yang berasal dari turunan pertama atau lebih objek lainnya. Intinya adalah mengembangkan objek yang sudah ada. Jadi kita tidak perlu membentuk objek dari awal, objek yang sudah ada bisa dimodifikasi dan/atau ditambah fungsinya sesuai dengan kebutuhan.

3.2.3.1.1.4 Polymorphism

Polymorphism adalah kemampuan objek berubah menjadi objek yang lain, beberapa objek bertipe sama bereaksi secara berbeda terhadap “pesan” yang sama.

3.2.3.1.2 UML (Unified Modelling Language)

Menurut Bambang Hariyanto. Ir.,MT. (2004:17) UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi.

Tujuan UML

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan pertukaran model-model yang berarti.

2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independent bahasa pemprograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.

UML mendefinisikan diagram-diagram sebagai berikut:

1. *use case diagram*
2. *class diagram*
3. *statechart diagram*
4. *activity diagram*
5. *sequence diagram*

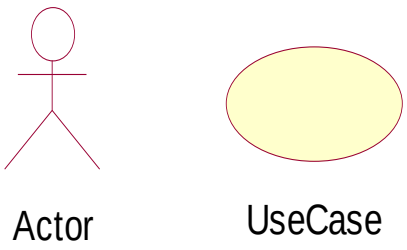
3.2.3.1.2.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

a. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di- include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal.
b. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common.
c. Sebuah use case juga dapat meng-extend use case lain dengan behavior-nya sendiri.
d. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain



Gambar 3.1. *Use Case Diagram*

(Munawar. 2005. *Pemodelan visual dengan UML*. Graha Ilmu. Yogyakarta.)

3.2.3.1.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda



Gambar 3.2 Contoh Class Diagram

(Booch Jcobson Tumbaugh, Martin Fowler , 2005, *UML Distilled* Edisi 2, Andi)

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

3.2.3.1.2.3 Statechhart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

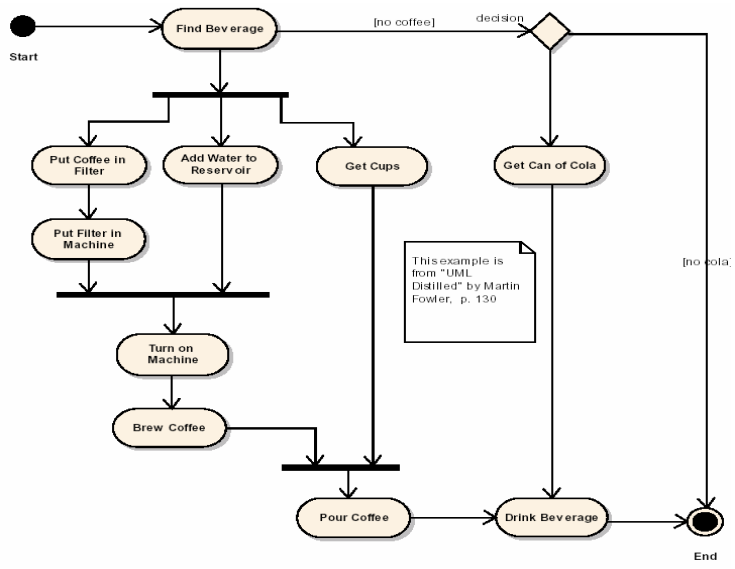
Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

3.2.3.1.2.4 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat

menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.



Gambar 3.3 Contoh Activiy Diagram

(Booch Jcobson Tumbaugh, Martin Fowler , 2005, *UML Distilled* Edisi 2, Andi)

3.2.3.1.2.5 Sequence Diagram

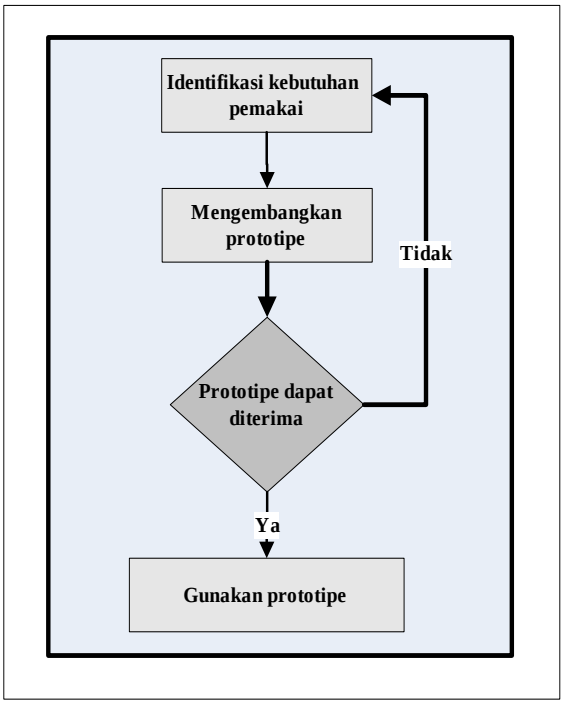
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

3.2.3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode Model Prototype yang merupakan mekanisme untuk mengidentifikasi kebutuhan perangkat lunak.



Gambar 3.4 Metode Pengembangan Prototype

(Sumber : Raymond McLeod, Jr. and Goorge P. Schell **.** 2007. *Management Information Systems* .New Jersey.)

a. Keuntungan prototype

1. Evaluasi dan feedback pada rancangan interaktif.
2. Stakeholder (dalam hal ini user) dapat melihat, menyentuh, berinteraksi dengan prototype.
3. Anggota tim dapat berkomunikasi secara efektif.
4. Para perancang dapat mengeluarkan ide-idenya.
5. Memunculkan ide-ide secara visual dan mengembangkannya.
6. Dapat menjawab pertanyaan dan membantu pemilihan di antara alternatif-alternatif.
7. Mengurangi biaya pengembangan dan pembiayaan

8. Dapat memperbaikimasalah penggunaan sebelum dibuat programnya

b. Kelemahan prototype

1. Prototype hanya akan berhasil jika pemakain bersungguh-sungguh dalam menyediakan waktu dan pikiran untuk mengerjakan prototype.
2. Kemungkinan dokumentasi terabaikan karena pengembangan lebih berkonsentrasi pada pengujian dan pembuatan prototype.
3. Mengingat target waktu yang pendek, ada kemungkinan sistem yang dibuat tidak lengkap dan bahkan kurang teruji.
4. Jika terlalu banyak proses pengulangan dalam membuat prototype, ada kemungkinan pemakai merasa jenuh dan memberikan reaksi negatif
5. Apabila tidak terkelola dengan baik, prototype menjadi tidak pernah berakhir. Hal ini disebabkan permintaan perubahan terlalu mudah untuk dipenuhi

3.2.3.3 Alat Bantu Analisis

OOP adalah merupakan teknik membuat suatu program berdasarkan objek. Untuk lebih memudahkan pemahaman tentang objek, saya beri sebuah contoh : Meja, Kursi, Jam Dinding dan yang lainnya merupakan contoh objek. Objek yang disebutkan tadi umumnya berada didalam sebuah ruangan, ruangan apabila kita melihatnya dari sebuah rumah bisa juga dikategorikan sebuah objek dan rumah jika dilihat dari cara pandang yang lebih luas lagi juga merupakan objek. Jadi

sebuah Objek merupakan suatu bentuk atau model yang tergantung dari cara pandang objek dimaksud.(Asep Ramdhani M).

3.2.4 Pengujian Software

Pengujian merupakan bagian penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dar perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang dapat diandalkan.

Dalam suatu program membutuhkan suatu teknik pengujian yang berguna dalam mengevaluasi program yang dibuat. Adapun teknik pengujian yang digunakan dalam perangkat lunak tersebut adalah *Black box.*

3.2.4.1 *Black box*

Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diiketahui kesalahan-kesalahannya

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya

menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

a. Fungsi – fungsi yang tidak benar atau hilang.
b. Kesalahan interface.
c. Kesalahan dalam struktur data atau akses database eksternal.
d. Kesalahan kinerja.
e. Inisialisasi dan kesalahan terminasi.