

## **BAB II**

### **LANDASAN TEORI**

Bab ini akan membahas mengenai tinjauan pustaka yang berisi tentang hasil penelitian yang pernah dilakukan, kerangka pemikiran, dan landasan teori yang membahas teori-teori dasar pendukung untuk penelitian ini.

#### **2.1 Tinjauan Pustaka**

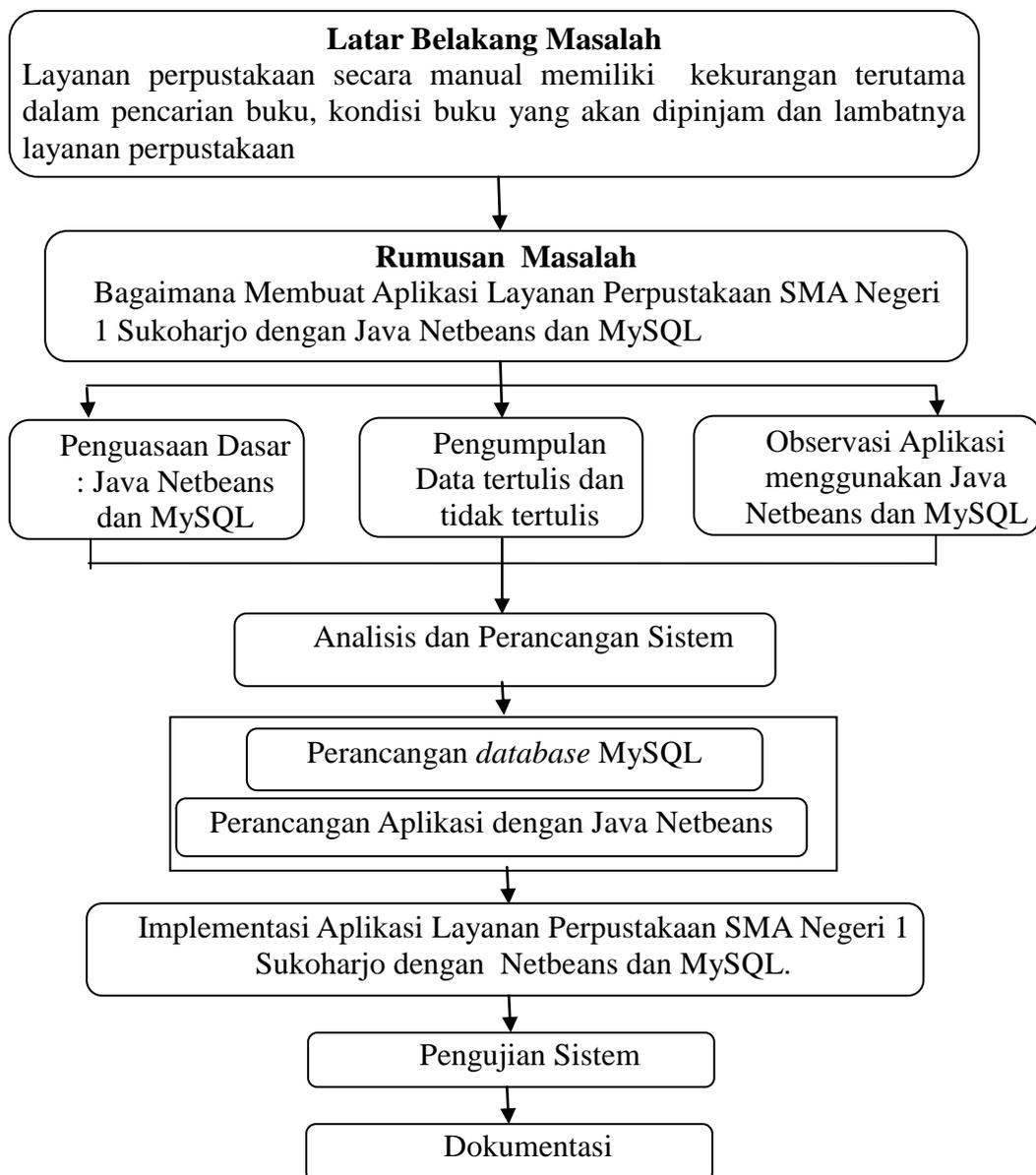
Berdasarkan hasil penelitian yang pernah dilakukan Hendra Kurniawan (2008) yang mengambil topik analisis dan perancangan sistem informasi perpustakaan SMP N 275 menggunakan Net Beans dengan database berbasis MySQL, menunjukkan bahwa dengan perancangan sistem perpustakaan dengan menggunakan Net Beans dengan database berbasis MySQL proses peminjaman buku dapat dilakukan dengan cepat, laporan transaksi dan data koleksi buku dapat disajikan secara cepat dan akurat.

Fitri Maqfiroh (2013), dalam penelitiannya yang berjudul “Analisis dan Perancangan Sistem Informasi Perpustakaan Pada MTs Negeri Ngemplak Sleman”, menjelaskan bahwa pemanfaatan sistem informasi perpustakaan menggunakan Netbeans dan MySQL dapat membantu dalam penyimpanan dan pengolahan data-data perpustakaan, seperti data peminjaman, data pengembalian, data buku, data anggota, dan data penerbit serta pembuatan laporan. Sistem ini dapat menjaga keakuratan data, mempermudah proses penyimpanan dan pembuatan laporan.

Bambang Triatmojo (2012) yang mengambil judul Analisis dan Perancangan Sistem Informasi Perpustakaan pada SMA Negeri 1 Sukoharjo, menunjukkan bahwa dengan adanya sistem informasi perpustakaan berbasis komputer mampu mengatasi masalah-masalah pada sistem lama, serta mampu menghasilkan informasi yang berkualitas sehingga dapat membantu pihak-pihak yang terkait dalam mengambil keputusan.

Berdasarkan hasil penelitian yang telah dilakukan sebelumnya, maka terlihat bahwa pembahasan penelitian hanya sebatas pada analisis dan rancangan sistem informasi perpustakaan. Maka penelitian ini akan mengembangkan dengan membuat aplikasi perpustakaan di SMA Negeri 1 Sukoharjo Menggunakan Netbeans dan MySQL.

## 2.2 Kerangka Pemikiran



Gambar 2.1 Diagram Kerangka Pemikiran

Gambar 2.1 merupakan diagram kerangka pemikiran yang dijalankan dalam penelitian ini. Keterangan dari Gambar 2.1 :

- 1) Latar belakang masalah  
Tahapan paling awal, yakni menelusuri latar belakang kenapa masalah yang akan diangkat menjadi penting untuk dipilih.
- 2) Rumusan masalah  
Penyimpulan latar belakang masalah menjadi suatu rumusan masalah yang akan diangkat untuk menjadi bahan penelitian.
- 3) Pengumpulan data tertulis dan tidak tertulis  
Pengumpulan data dilakukan baik dengan wawancara (*interview*), observasi, maupun studi literatur di perpustakaan.
- 4) Penguasaan dasar Netbeans dan MySQL  
Tahap untuk melakukan beberapa percobaan untuk membuat proyek sederhana yang melibatkan Netbeans dan MySQL dengan tujuan agar dapat lebih menguasai program - program yang akan digunakan untuk membangun aplikasi.
- 5) Observasi aplikasi Netbeans dan MySQL  
Merupakan tahap pengamatan sampel – sampel aplikasi yang telah ada, jurnal, buku, maupun karya ilmiah untuk kajian yang dapat dijadikan referensi untuk pembangunan aplikasi.
- 6) Analisis dan perancangan aplikasi berbasis objek  
Analisis dan perancangan aplikasi dilakukan pada program Netbeans dan database MySQL.
- 7) Implementasi aplikasi layanan perpustakaan sekolah dengan Netbeans dan MySQL.
  - a) Pembuatan Desain Form dan *Database*  
Pada tahap ini, pembuatan rancangan form login, anggota, buku, peminjaman, dan form pelaporan. Database dibuat dengan MySQL
  - b) Membuat *coding* setiap form, Pada tahap ini, pemberian *coding* setiap form dengan menggunakan Netbeans

c) Pembangunan aplikasi dengan Netbeans dan MySQL

Tahapan terakhir dalam implementasi, menggabungkan rancangan desain form yang telah diberi *coding* dengan rancangan database MySQL yang telah dibuat

8) Pengujian aplikasi

Pengujian sistem dengan *Black Box*, dilakukan pada data pustakawan, data siswa dan data buku, data peminjaman, data pengembalian, dan pelaporan untuk memastikan setiap bagian berjalan sesuai rancangan.

9) Dokumentasi

Tahapan terakhir, yakni tahap pendokumentasian seluruh poses penyusunan tugas akhir ke dalam laporan.

## 2.3 Landasan Teori

### 2.3.1 Program Aplikasi

Program adalah istilah yang biasa dipakai oleh pembuat program, sedangkan aplikasi adalah istilah dari sisi pemakai (*user*). Program adalah yang ditulis oleh pemrogram, sedangkan aplikasi adalah hasil terjemahan program (Abdul Kadir, 2013 :13). Menurut Raymond McLeod,Jr. (1998:232), program aplikasi yaitu program yang digunakan untuk memproses data perusahaan. Program aplikasi data diperoleh dengan dua cara, yaitu membuat sendiri (*custom programming*) atau memanfaatkan paket aplikasi yang sudah jadi (*prewritten package*).

Andi Purnomo (2007:29) berpendapat bahwa program aplikasi merupakan perangkat lunak penunjang untuk tujuan tertentu. Berdasarkan beberapa pendapat di atas, penulis menyimpulkan bahwa Program Aplikasi adalah *software* komputer yang dibuat khusus untuk melakukan tugas-tugas tertentu.

### 2.3.2 Layanan Perpustakaan Sekolah

Layanan perpustakaan adalah layanan informasi pustaka/sumber informasi yang dilakukan oleh perpustakaan untuk memenuhi kebutuhan pemustaka. Layanan ini merupakan aktivitas perpustakaan dalam memberikan jasa perpustakaan kepada pemustaka. Jenis/divisi layanan pemustaka cukup banyak.

Adapun penyelenggaraan semua jenis/divisi layanan tersebut disesuaikan dengan kondisi di setiap perpustakaan dan kebutuhan penmustakanya (Zulaekhah dkk : 2009 ).

Menurut Undang-Undang No. 43 tahun 2007 Perpustakaan adalah institusi pengelola koleksi “karya tulis, karya cetak, dan/atau karya rekam” secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Perpustakaan bertujuan memberikan layanan kepada pemustaka, meningkatkan kegemaran membaca, serta memperluas wawasan dan pengetahuan untuk mencerdaskan kehidupan bangsa.

### **2.3.3 Netbeans**

Java Netbeans atau Netbeans adalah sebuah proyek *software OpenSource*. Proyek Netbeans mulai diprakarsai oleh Perusahaan Sun Microsystems sejak bulan Juni 2000 dan terus berkembang hingga saat ini. NetBeans mengacu pada dua hal, yaitu NetBeans platform untuk pengembangan aplikasi desktop java dan sebuah *NetBeans Integrated Development Environment* (Lutfiana Rahmawati: 2012)

Netbeans merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface (GUI)*, suatu kode editor atau text, suatu *compiler* dan suatu *debugger*.

Netbeans memudahkan programmer untuk membuat program. Oleh karena itu dalam membuat proyek baru, skema dasar kode program telah disediakan dan programmer cukup melanjutkannya. Namun demikian, skema dasar tersebut harus difahami dan dihafalkan oleh seorang programmer Netbeans juga digunakan oleh sang programmer untuk menulis, meng-*compile*, mencari kesalahan dan

menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat *professional desktop, enterprise, web, dan mobile applications* dengan *Java language, C/C++*, dan bahkan *dynamic languages* seperti *PHP, JavaScript, Groovy, dan Ruby*.

### 2.3.3.1 Komponen Netbeans

Komponen Netbeans 7.4 terdiri dari *Graphic User Interface (GUI), Navigator Windows, Palette Windows, Properties Windows, dan Source Area*

#### 1) *Graphic User Interface (GUI)*

Pada aplikasi Netbeans, komponen *GUI* disimpan pada kontainer yang di sebut dengan form. Bahasa pemograman java mengumpulkan komponen antar muka dari form *GUI* yang telah terbentuk. Pada NetBeans komponen *GUI* ini telah tersedia. Hal ini mempermudah dalam merancang dan membangun form Java.

#### 2) *Design Area*

Area Desain adalah area di mana secara visual akan membangun *GUI*. Ini memiliki dua tampilan: melihat sumber, dan tampilan desain. Tampilan desain adalah default, seperti yang ditunjukkan di bawah ini. Anda dapat beralih antara tampilan setiap saat dengan mengklik tab masing-masing.

#### 3) *Navigator Windows*

Navigator windows merupakan jendela yang menampilkan pohon pewarisan dari semua komponen form yang di buka spserti button, label, menu, timer, dan sebagainya.

#### 4) *Palette Windows*

*Palette Windows* adalah jendela yang menampilkan daftar semua komponen swing yang akan dapat dimasukan ke dalam form. Komponen palette antara lain : *Swing Containers, Swing Control, Swing Menus, Swing Windows*, dan lain-lain.

Berikut ini bagian dari *Palette Windows* dan fungsinya :

a) *Swing Container*

Komponen yang tersedia dalam *Swing Container* antara lain Panel, Split Pane, Tool Bar, Internal Frame, Tabbed Pane, Scroll Pane, Desktop Pane, dan Layered Pane.

b) *Swing Control*

Komponen yang tersedia dalam *Swing Control* beserta fungsinya diperlihatkan oleh Tabel 2.1.

Tabel 2.1 *Swing Control*

No	Tools	Fungsi
1	Label	Memberikan keterangan tambahan atau indentitas pada tombol atau perintah lainnya
2	Radio Button	Meminta user agar memilih satu dari lebih dua pilihan
3	Combo box	Menyisipkan beberapa perintah, sehingga dengan Combo Box ini lebih menghemat tempat
4	Text Field	Membuat kolom agar bisa diisi perintah-perintah teks atau angka oleh user
5	Scroll Bar	Menggeser jendela (windows) secara vertikal
6	Editor Pane	Sebuah panel yang digunakan untuk mengedit tulisan
7	Tabel	Menampilkan atau menyisipkan tabel
8	Button	Kontrol yang tampil pada layar dengan bentuk tertentu yang mirip dengan tombol persegi panjang dengan tulisan di tengahnya
9	Text area	Tempat mengetikkan tulisan yang berada dalam kotak yang mempunyai fungsi scroll
10	Text pane	Menampilkan teks dan membolehkan user untuk mengeditnya

c) *Swing Menu*

Komponen yang tersedia dalam *Swing Menu* antara lain : Menu Bar, Menu, Menu Item, Radio Button, Popup Menu, dan Seprator

#### 5) *Properties Windows*

*Properties windows* merupakan jendela yang dapat di ubah memilih komponen yang akan di pergunakan.

#### 6) *Source Area*

*Source Area* merupakan jendela yang di gunakan untuk menambahkan kode program pada pemograman JAVA.

### **2.3.3.2 Membuat proyek baru**

Sebuah proyek baru dapat dibuat dengan cara :

- 1) Pilih Menu File > New Project.
- 2) Pilih salah satu jenis proyek, misalkan Java Application
- 3) Selanjutnya, Netbeans akan meminta nama dari proyek baru tersebut.

Secara default tampilan Netbeans terdiri dari 4 bagian utama, yaitu:

- a) Atas: Menu bar dan toolbar, digunakan sebagai menu navigasi Netbeans
- b) Kiri: Panel proyek dan file kode program, digunakan untuk mengatur proyek dan file kode program
- c) Kanan: Panel kerja/editor, digunakan sebagai editor, tempat menulis kode program
- d) Bawah: Panel output/keluaran program, digunakan untuk melihat hasil setelah program dijalankan

### **2.3.4 XAMPP**

Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, Mysql, phpmyadmin, PHP, Perl, Freetype2, dan lain-lain. Instalasi Mysql dan Apache langsung dapat dilakukan melalui Xampp tanpa menginstall aplikasi-aplikasi tersebut satu persatu. Setelah berhasil menginstall Xampp, kita dapat langsung mengaktifkan Mysql dengan cara mengaktifkan Xampp.

Menurut Bunafit Nugroho (2007 : 2) XAMPP adalah suatu bundel web server yang populer digunakan untuk coba-coba di Windows karena kemudahan instalasinya. Bundel program *open source* tersebut berisi antara lain server web Apache, interpreter PHP, dan basis data MySQL. Setelah menginstal XAMPP, kita bisa memulai membuka database MySQL. XAMPP, dapat di-*download* di situs resmi XAMPP: <https://www.apachefriends.org>

#### **2.3.4.1. Menjalankan XAMPP**

XAMPP dapat dijalankan dengan cara : Klik Menu START > All Programs > XAMPP > XAMPP Control Panel. Selanjutnya akan ditampilkan jendela XAMPP Control Panel. Semua control dilakukan melalui Jendela XAMPP Control Panel. Untuk mengaktifkan Module Apache dan MySQL klik start pada *Actions*, sehingga Port(s) akan aktif. *Port* Apache 80.443 dan MySQL 3306.

#### **2.3.5 Database MySQL**

Database merupakan sebuah *computerized recordkeeping system*. Artinya pemakai (*user*) dapat melakukan operasi pembuatan file baru, menambah data baru, menampilkan isi file, mengubah data dalam file, menghapus data dalam file, dan menghapus file (Bernaridho L Hutabarat: 2004 )

Database dibuat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumpulan data yang terorganisir untuk satu atau lebih penggunaan, dalam bentuk digital. database digital di-*manage* menggunakan *Database Management System (DBMS)*, yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain. Beberapa Database yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql. Menurut Didik Dwi Prasetyo (2004 :18) MySQL merupakan salah satu *database server* yang berkembang di lingkungan *open source* dan didistribusikan secara *free* (gratis) dibawah lisensi GPL. MySQL adalah salah satu jenis *database server* yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses databasenya.

MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql, anda dapat menggunakan *software* tertentu, di antara nya adalah phpmyadmin dan mysql yog. Pada kesempatan kali ini, kita akan menggunakan phpmyadmin, yang terdapat dalam *bundle* xampp, yang dapat diperoleh di [www.apachefriends.org](http://www.apachefriends.org).

Beberapa keunggulan dari MySQL yaitu :

1) Cepat, handal dan Mudah dalam penggunaannya

MySQL lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.

2) Didukung oleh berbagai bahasa

Database server MySQL dapat memberikan pesan *error* dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.

3) Mampu membuat tabel berukuran sangat besar

Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.

4) Lebih Murah

MySQL bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk UNIX *platform*, OS/2 dan Windows *platform*.

5) Melekatnya integrasi PHP dengan MySQL

Keterikatan antara PHP dengan MySQL yang sama-sama *software opensource* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan database server lainnya. Modul MySQL di PHP telah dibuat *built-in* sehingga tidak memerlukan konfigurasi tambahan pada file konfigurasi php.ini

Cara mengaktifkan Database MySql melalui Xampp adalah sebagai berikut :

a) Aktifkan Xampp, Klik Start – Exampp control Panel

Selanjutnya akan ditampilkan jendela Exampp control panel.

- b) Klik Start pada pilihan MySQL, sehingga akan ditampilkan pesan : Running. Selanjutnya klik Admin pada pilihan MySQL untuk menampilkan halaman phpmyadmin

### **2.3.5.1 Membuat Database**

Membuat *Database*, dapat di lakukan dengan cara :

- 1) Pada tampilan Halaman phpmyadmin, Klik Database
- 2) Selanjutnya akan ditampilkan jendela *Create new database*, ditunjukkan
- 3) Pada tampilan *Create new database*, ketik nama *database* yang akan dibuat ( misalnya : perpus )
- 3) Klik *Create*

### **2.3.5.2. Membuat Tabel**

Setelah *Database* terbentuk, maka selanjutnya dibuat tabel yang akan berisi data-data yang kita perlukan. Cara membuat table adalah sebagai berikut:

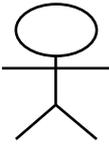
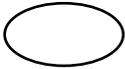
- 1) Pilih *Database*
- 2) Selanjutnya akan ditampilkan jendela *Create new table on database* perpus
- 3) Ketik nama *table* yang akan dibuat
- 4) Tentukan jumlah field, misalnya : 7
- 5) Selanjutnya akan ditampilkan table anggota dengan 7 field,
- 6) Masukkan nama Field, type data, dan ukuran tiap-tiap field sesuai data yang akan diisikan
- 7) Klik Save untuk menyimpan tabel
- 8) Untuk membuat sebuah primary key (field sebagai kunci) , pilih Index kemudian pilih PRIMARY.

## 2.4 Unified Modelling Language (UML)

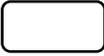
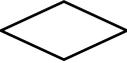
Menurut Munawar (2005:25) *Unified Modelling Language (UML)* adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerjasama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram yang memungkinkan bagi analis sistem untuk membuat cetak biru (*blue print*) sistem yang komprehensif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan.

UML terbagi menjadi beberapa jenis diagram yang masing-masing memiliki aturan-aturan tertentu dalam penyusunannya. Diagram-diagram tersebut tersusun atas sejumlah elemen grafis yang saling membentuk satu kesatuan dalam pemodelan *software*. Masing-masing diagram UML mempersatukan berbagai sudut pandang terhadap sistem dan mendefinisikan apa yang dikerjakan oleh sistem bukan bagaimana cara sistem bekerja. Simbol-simbol yang digunakan pada UML antara lain adalah *Use Case diagram* (Tabel 2.2), *Activity Diagram* (Tabel 2.3), dan *Class Diagram* (Tabel 2.4).

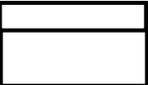
Tabel 2.2. Simbol-simbol *Use Case Diagram*

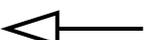
SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .
	<i>Use Case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>use case</i> dituliskan didalam elips tersebut
	<i>Relationship</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>use case</i> .
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

Tabel 2.3. Notasi *Activity Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Initial Node</i>	Titik Awal, memulai suatu aktivitas
	<i>Final Node</i>	Titik Akhir, mengakhiri suatu aktivitas
	<i>Activity</i>	Menandakan sebuah aktivitas
	<i>Decision</i>	Pilihan untuk mengambil keputusan
	<i>Fork</i>	Menunjukkan dua kegiatan paralel menjadi satu.
	<i>Flow Final</i>	Aliran akhir
	<i>Message</i>	Mengindikasikan komunikasi antara objek-objek.

Tabel 2.4 *Class Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property</i> /atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .
	<i>Assosiation</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i> ).
	<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.

	<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
	<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.
	<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi obyek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “ <i>parent</i> ”-nya/induknya.

## 2.5 Metode Pengujian *Black Box*

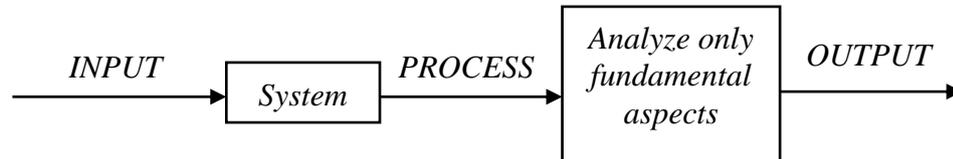
Menurut Sri Wahyuni (2011) pengujian *Black Box* merupakan pengujian program dengan cara mengamati keluaran (*output*) program tanpa harus mengetahui segala yang dilakukan oleh sistem dengan berbagaimacara masukan (*input*), menguji seluruh tombol dan fasilitas program. Jika keluaran program dan seluruh fasilitas program berjalan dan tidak terjadi kesalahan pada saat dioperasikan, data keluaran telah sesuai dengan yang diharapkan, maka program dianggap baik. Dalam hal ini pengujian program dilakukan oleh orang yang mengerti sistem secara keseluruhan.

*Black Box* dapat menemukan kesalahan dalam kategori berikut :

- 1) Fungsi-fungsi yang tidak benar atau hilang
- 2) Kesalahan antar muka
- 3) Kesalahan dalam strutur data atau akses basisdata eksternal
- 4) Inisialisasi dan kesalahan terminasi
- 5) validitas fungsional
- 6) Kesensitifan sistem terhadap nilai input tertentu

7) Batasan dari suatu data.

Sistem Kerja dari Teknik Pengujian *Black Box* digambarkan pada Gambar 2.2.



Gambar 2.2. Sistem Kerja dari Teknik Pengujian *Black Box*