

BAB II TINJAUAN PUSTAKA

1.1 Sistem

Menurut Krismiaji (2015) Sistem adalah kumpulan atau grup dari sub sistem/bagian/komponen atau apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan dapat bekerja sama untuk mencapai satu tujuan tertentu.

Menurut Nas (2018) Sistem adalah kumpulan elemen yang saling berinteraksi dalam kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

1.2 Informasi

Menurut Krismiaji (2015) Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Menurut Nas (2018) Informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

1.3 Sistem Informasi

Pengertian menurut Krismiaji (2015) Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Definisi menurut Nas (2018) Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegrasi dan saling melengkapi dengan menghasilkan output yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

1.4 Pengelolaan

Menurut (Rahayu *et al.*, 2018) pengelolaan merupakan proses khas yang terdiri atas tindakan-tindakan perencanaan, pengorganisasian, pergerakan dan pengendalian yang dilakukan untuk menentukan serta mencapai sasaran yang telah ditentukan.

Menurut (Fadly, *et al.*, 2020) pengelolaan adalah proses, cara, perbuatan pengelolaan yang membantu merumuskan kebijakan dan tujuan organisasi atau yang memberikan pengawasan suatu hal yang terlibat dalam pelaksanaan kebijakan dan pencapaian tujuan dengan menggunakan tenaga orang lain.

1.5 Asrama

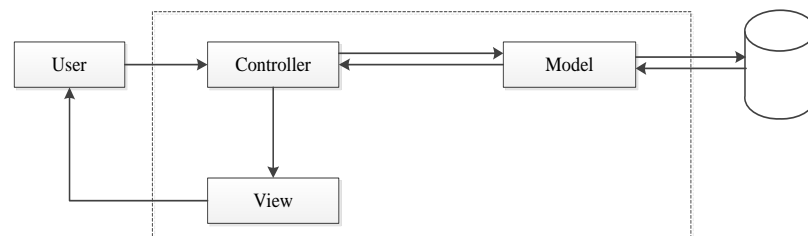
Menurut (Wulandari, 2017) Asrama adalah suatu bangunan seperti yang ada di perguruan tinggi, yang didalamnya terdapat sejumlah ruang privat atau semi privat untuk penghuninya, biasanya terdapat juga fasilitas kamar mandi bersama dan tempat untuk rekreasi.

Menurut (Jannah, *et al.*, 2015) Sebuah atau sekelompok bangunan tempat tinggal yang sedemikian untuk menampung sejumlah pelajar secara continue atau periodic dengan kepentingan yang sama yaitu menuntut ilmu, dengan tujuan dan harapan agar dapat belajar dan beraktifitas secara efisien dan efektif tanpa paksaan.

1.6 CodeIgniter

Menurut Raharjo (2018), *CodeIgniter* adalah *Framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat dikembangkan dalam perangkat *web*, dekstop maupun *mobile*".

CodeIgniter memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.



Gambar 1.1 Arsitektur MVC

Berdasarkan arsitektur tersebut diketahui konsep *Model-View-Controller* yaitu:

1. *Model*

Model merupakan suatu fungsi yang digunakan mengelola database pada sistem untuk menangani struktur data dari *controller*.

2. *View*

View merupakan bagian untuk mengelola tampilan dari *website* atau dapat disebut sebagai *user interface* yang diatur bagian *controller*.

3. *Controller*

Controller merupakan kunci dalam konsep MVC dengan fungsi untuk menghubungkan *model* dengan *view*.

CodeIgniter merupakan sebuah *toolkit* yang ditujukan untuk orang yang ingin membangun aplikasi web dalam bahasa pemrograman PHP. Beberapa keunggulan yang ditawarkan sebagai berikut:

1. CodeIgniter adalah *framework* yang bersifat *free* dan *open source*
2. Memiliki ukuran file yang relatif lebih kecil dibanding *framework* lain
3. Aplikasi yang dihasilkan dapat berjalan cepat.
4. Menggunakan pola desain MVC sehingga satu file tidak banyak berisi kode, hal tersebut menjadikan kode mudah dibaca dan dipahami.
5. Dapat diperluas sesuai kebutuhan
6. Dokumentasi penerapan CodeIgniter dapat dipahami dengan mudah.

1.6.1 Appserve

Menurut Fadly, *et al.*, (2018) Appserv adalah aplikasi yang memiliki fungsi untuk menginstal beberapa program yaitu Apache, PHP, MySQL. *Appserv* juga merupakan *software* yang digunakan untuk *database* secara grafis atau gambar, *software* ini memudahkan para *desain web* dalam membuat *database* daripada *cmd.exe* yang penulisannya manual dan berbasis text.

Berdasarkan pengertian diatas dapat ditarik kesimpulan *Appserv* adalah *software* basis data, MySQL, PHP, *phpMyAdmin* yang dilengkapi dengan *server apache*.

1.6.2 Web Based

Menurut Urbieta *et al*, (2019), *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya.

Menurut Purwati *et al*, (2018), Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

1.6.3 PHP

Menurut Subagja (2018), PHP adalah bahasa *server-side-scripting* yang menyatudengan HTML untuk membuat halaman *web* yang dinamis.

Menurut Aryani *et al.*, (2015), berpendapat bahwa *PHP Hypertext Preprocessor* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua script PHP dieksekusi pada *server* dimana script tersebut dijalankan. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

Sehingga PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

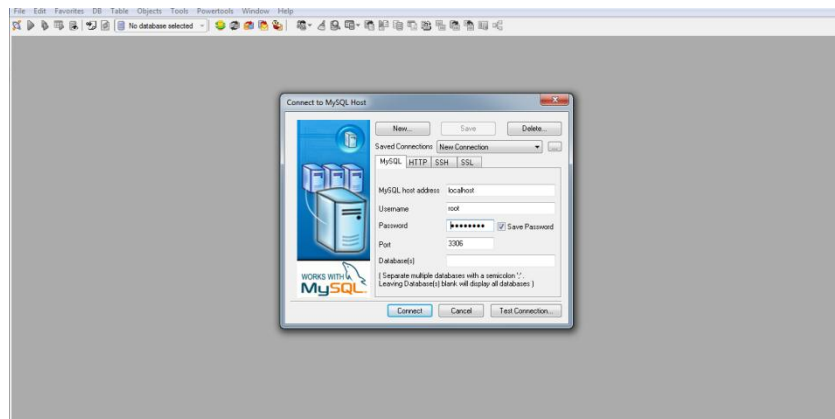
1.6.4 Database

Database merupakan kumpulan informasi yang terimpan secara sistematis di dalam komputer sehingga dapat diperiksa menggunakan sistem komputer.

Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan *structure* data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

Menurut Enterprise (2018) adalah server yang melayani database, untuk dapat mengolah database, diperlukan pemahaman bahasa SQL yaitu *structure query language*.

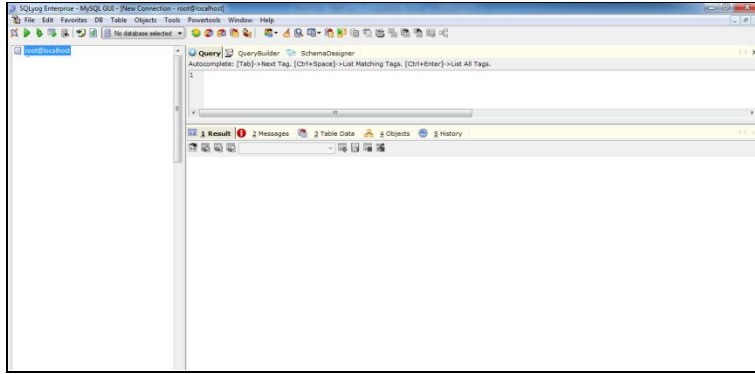
Sehingga Mysql sering digunakan karena dapat digunakan secara gratis dan dapat dikembangkan lebih luas sesuai kebutuhan. Untuk mempermudah membuka database dapat menggunakan *tools sql* yoq seperti berikut:



Gambar 1.2 SQLyoq
Sumber: (Setiadi, 2017)

Fungsi SQLyoq yaitu mempermudah dalam mengoperasikan database mysql dengan menggunakan tools, untuk dapat mengkoneksikan SQLyoq dengan mengisi informasi seperti :

1. *Mysql Host Adress* : *Localhost*
2. *Username* : *root*
3. *Password* : (Dapat diisi atau kosongkan)
4. *Port* :3306
5. Kemudian klik connect untuk membuka dan menampilkan database dalam mysql yang dapat dilihat seperti berikut :



Gambar 1.3 Tampilan Utama SQLyog
Sumber: (Setiadi, 2017)

1.6.5 MySQL

Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

Menurut Amin (2018), mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

1.6.6 Kamus Data

Kamus data adalah sistem yang terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi tersedia saat dibutuhkan. Kamus data berfungsi sebagai penyedia informasi bagi user dalam pengintegrasian data data agar saling terhubung antara satu dengan lainnya [11]. Beberapa istilah dalam basis data antara lain :

1. *Field*.

Merupakan bagian terkecil dari *record* yang memiliki nilai untuk dijadikan kunci sebagai perwakilan *record* lainnya.

2. *Cadidat Key*

Merupakan atribut yang mengidentifikasikan secara spesifik *field* kunci dari suatu *entity*.

3. *Primary Key*

Merupakan atribut yang mengidentifikasikan dan mewakili transaksi secara spesifik.

4. *Alternate Key*

Merupakan kunci kadidat dari *primary key* yang belum terpakai.

5. *Record*

Merupakan kumpulan dari elemen atau field yang saling terkait.

6. *Entity*

Merupakan tempat kejadian informasi direkam.

7. *Table*

Merupakan menu untuk memasukan data data pada database.

8. *Foreign Key*

Pelengkap hubungan antara suatu atribut dengan atribut induk.

Operasi yang dapat dilakukan oleh basis data adalah :

1. Pembuatan basis data baru (*create database*)
2. Pembuatan *file/table* (*create table*)
3. Penghapusan *file/table* (*drop table*)
4. Penambahan dan pengisian *table* (*insert*)
5. Pengubahan data (*update*)
6. Penghapusan data (*delete*)

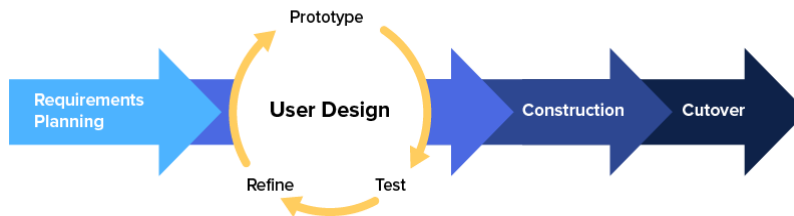
1.7 Metode Pengembang Sistem

Metode pengembang sistem merupakan metode yang digunakan sebagai alur proses dalam pengembangan, sehingga penelitian dapat di kembangkan sesuai tahapan dari metode pengembang sistem.

2.3.1. Rapid Application Development (RAD)

Menurut Rosa and Shalahuddin (2019), metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang. RAD diajukan oleh IBM pada tahun 1980 sampai 1990-an, ketika permintaan terhadap aplikasi semakin meningkat. Dengan banyaknya *demand*, orang-orang di dunia IT harus mencari solusi untuk memenuhi permintaan tersebut. Metode ini merupakan semacam cikal bakal *agile project management*, karena bisa mengikuti *pace* bisnis yang terus berkembang dan juga kebutuhan pasar yang terus meningkat. Pengembangan software pada umumnya seperti waterfall model membutuhkan perencanaan yang terbilang cukup kaku. Klien atau pelanggan seakan ‘dipaksa’ untuk menyetujui banyak hal di awal, tetapi mereka tidak bisa melihat proses pembuatannya.

Keuntungan utama menjalankan *rapid application development* adalah jangka waktu pengembangan lebih cepat. Hal ini dikarenakan *feedback* dari pelanggan cepat didapatkan dan semua perubahan yang dilakukan akan sesuai hasil tersebut. Akan tetapi, salah satu kekurangan RAD adalah kamu membutuhkan tim berisikan *developer* yang benar-benar memiliki *skill* tinggi dan juga metode ini hanya bisa digunakan untuk proyek yang bisa termodulasi.



Gambar 1.4 *Rapid Application Development (RAD)*

Sumber: (Rosa and Shalahuddin, 2019)

1. Kelebihan Model RAD

Kelebihan metodologi RAD menurut Marakas (2006):

- a. Penghematan waktu dalam keseluruhan fase proyek dapat dicapai.
- b. RAD mengurangi seluruh kebutuhan yang berkaitan dengan biaya proyek dan sumberdaya manusia.
- c. RAD sangat membantu pengembangan aplikasi yang berfokus pada waktu penyelesaian proyek.
- d. Perubahan desain sistem dapat lebih berpengaruh dengan cepat dibandingkan dengan pendekatan SDLC tradisional.
- e. Sudut pandang user disajikan dalam sistem akhir baik melalui fungsi-fungsi sistem atau antarmuka pengguna.
- f. RAD menciptakan rasa kepemilikan yang kuat di antara seluruh pemangku kebijakan proyek.

2. Kelemahan Model RAD

Kelemahan pada pengembangan tersebut dapat dilihat berdasarkan kesesuaian pengembangan yang dilakukan, berikut adalah kelemahan metode pengembang sistem RAD :

- a. Dengan metode RAD, penganalisis berusaha mempercepat proyek dengan terburu-buru.
- b. Kelemahan yang berkaitan dengan waktu dan perhatian terhadap detail. Aplikasi dapat diselesaikan secara lebih cepat, tetapi tidak mampu mengarahkan penekanan terhadap permasalahan-permasalahan perusahaan yang seharusnya diarahkan.
- c. RAD menyulitkan *programmer* yang tidak berpengalaman menggunakan prangkat ini di mana *programmer* dan *analyst* dituntut untuk menguasai kemampuan-kemampuan baru sementara pada saat yang sama mereka harus bekerja mengembangkan sistem

2.3.2 Tahapan Penelitian

Tahapan dalam penelitian sebagai langkah-langkah penelitian yang harus dikerjakan, berikut adalah tahapan penelitian Rapid Application Development.

1. Tahap *Requirements Project*

RAD dimulai dengan menentukan kebutuhan sebuah proyek (*project requirements*). Pada tahap ini, tim perlu menentukan kebutuhan yang ingin dipenuhi dari sebuah proyek. Kebutuhan ini tidak perlu spesifik. Tapi, sifatnya benar-benar umum dan jumlahnya bisa banyak. Baru dari situ, tim akan menentukan mana kebutuhan yang perlu diprioritaskan. Setelah mendapatkan kebutuhan yang jelas, barulah tim menentukan hal-hal yang lebih detail. Misalkan seperti tujuan, timeline, dan budget yang diperlukan.

2. Tahap Membuat *Prototype*

Hal yang selanjutnya dilakukan adalah membuat prototype. Developer secepat mungkin akan membuat prototype dari aplikasi yang diinginkan. Lengkap dengan fitur dan fungsi yang berbeda-beda. Tujuannya, sekadar untuk mengecek apakah *prototype* yang dibuat sudah sesuai dengan kebutuhan klien. Meski begitu, tahap ini bisa saja dilakukan berulang-ulang. Kadang juga melibatkan user untuk testing dan memberikan feedback. Proses ini memungkinkan tim mempelajari error yang mungkin muncul ke depannya. Ini berguna untuk mengurangi error dan debugging. Lewat tahapan ini, tim developer memiliki modal untuk membuat aplikasi yang mudah dipakai, stabil, tidak sering error, dan desainnya pun oke.

3. Proses Pengembangan dan Pengumpulan *Feedback*

Setelah tahu aplikasi seperti apa yang ingin dibuat, developer mengubah *prototype* ke bentuk aplikasi versi beta sampai dengan final. Jadi, bisa dibilang tahap RAD inilah yang cukup intens. Developer terus-menerus melakukan coding aplikasi, melakukan testing sistem, dan integrasi dengan bagian-bagian lainnya. Karena itulah, developer menggunakan tools dan *framework* yang mendukung RAD agar cepat. Apalagi proses ini terus diulang sambil terus mempertimbangkan feedback dari klien. Baik itu soal fitur, fungsi, interface, sampai keseluruhan aspek dari produk yang dibuat. Jika prosesnya berjalan lancar, developer akan melanjutkan ke langkah berikutnya. Yaitu, finalisasi produk atau implementasi. Kalau pun tidak, proses ini kemungkinan akan terus diulang.

4. Tahap *Implementation* (implementasi).

merupakan tahap pengujian terhadap aplikasi yang dikembangkan. Tahap ini programmer mengembangkan desain menjadi suatu program kemudian dilakukan proses pengujian untuk memeriksa kesalahan sebelum diaplikasikan.

1.8 Alat Pengembang Sistem

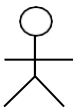
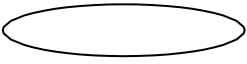
Menurut Rosa dan Salahuddin (2019), *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

1.8.1 *Use Case Diagram*


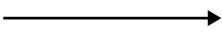
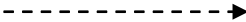
Menurut Rosa dan Salahuddin (2019), *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1 :

Tabel 1.1 Simbol-simbol Use Case Diagram

No	Simbol	Keterangan Fungsi
----	--------	-------------------

1.	Aktor 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	<i>Use Case</i> 	<i>Use Case</i> adalah deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Tabel 2.2 Simbol-simbol Use Case Diagram (Lanjutan)

No	Simbol	Keterangan Fungsi
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya atau sebaliknya dari bawah ke atas.
5.	<i>Defendancy</i> 	<i>Defendancy</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (<i>independen</i>).



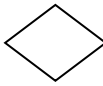

Sumber: (Rosa dan Salahuddin, 2019)

1.8.2 Activity Diagram

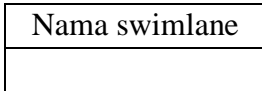

Menurut Rosa dan Salahuddin (2019), *activity* diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2:

Tabel 1.2 Simbol *Activity Diagram*

No.	Simbol	Keterangan
-----	--------	------------

1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Tabel 2.2 Simbol *Activity Diagram*

No.	Simbol	Keterangan
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber: (Rosa dan Salahuddin, 2019)

1.9 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

1.9.1 *Black Box Testing*

Black Box Testing merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data uji dan memeriksa fungsional yang terdapat pada perangkat lunak. Jadi dapat kita dianalogikan seperti halnya kita melihat ke dalam kotak hitam, sehingga kita hanya bisa melihat tampilan luarnya saja tanpa kita tau apa yang ada didalam kotak hitam tersebut.

Sehingga sama seperti halnya dengan *Black Box Testing* yang hanya dapat mengevaluasi dari tampilan luarnya dan fungsionalitasnya. Tanpa harus mengetahui apa sesungguhnya yang terjadi dalam proses detilnya. Pada pengetahuan khusus dari struktur kode internal dan

pengetahuan pada pemrograman dasar pada umumnya tidak diperlukan untuk *Black Box Testing*. Uji pada kasus yang dibangun disekitar spesifikasi dan persyaratan, yakni pada aplikasi yang seharusnya dilakukan.

1.9.2 Tahapan Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui proses pengujian yang akan dilakukan, berikut adalah tahapan pengujian *Black Box*:

1. *Decision Table*

Decision Table merupakan cara yang tepat untuk memodelkan logika yang cukup rumit, seperti diagram alur, *if-then-else* dan *switch* laporan kasus. Dalam kondisi ini mengaitkan dengan tindakan untuk melakukan, akan tetapi banyak kasus melakukannya dengan cara yang lebih elegan.

2. *All-Pairs Testing*

All-Pairs Testing atau disebut *pairwise testing* merupakan metode pengujian perangkat lunak kombinatorial yang digunakan untuk setiap pasangan parameter yang masuk kedalam sistem atau algoritma yang ada pada perangkat lunak.

3. *State Transition Table*

State Transition Table merupakan teori automata dan logika skuensial, pada table yang menunjukkan *state* dalam pengujian. Pada dasarnya sebuah table *state* merupakan table kebenaran yang digunakan untuk beberapa *input* dan *output* termasuk dengan *state* berikutnya dengan kondisi yang sebenarnya terjadi.

4. *Equivalence Partitioning*

Equivalence Partitioning merupakan teknik yang membagi data masukan dari beberapa unit perangkat lunak menjadi beberapa partisi data dari mana *test case* dapat diturunkan. Pada prinsipnya, uji kasus ini dirancang untuk menutupi setiap partisi minimal.

Teknik ini digunakan untuk mendefinisikan kasus pengujian yang mengungkap kelas kesalahan, sehingga mengurangi jumlah pengujian yang harus dilakukan.

5. *Boundry Values Analysis*

Boundary Value Analysis merupakan Pengujian yang dirancang untuk mencakup perwakilan dari batas Nilai-nilai batas. Pada nilai-nilai di sebuah partisi kesetaraan atau sebesar nilai terkecil di kedua sisi tepi.

