

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai sistem informasi pemesanan rumah sebelumnya pernah diteliti oleh Achmad Sidik Sekolah Tinggi Manajemen Informatika dan Komputer Bina Sarana Global pada tahun 2017 yang berjudul “Perancangan Sistem Informasi Penjualan Perumahan Citra Raya” menjelaskan bahwa dalam proses penjualan perumahan dan pendataan customer masih dilakukan secara manual oleh admin[3]. Dalam proses tersebut membutuhkan waktu atau proses yang lama sehingga efisien waktu dan kontrol yang tidak maksimal, sehingga informasi yang disampaikan lambat. Rancangan sistem ini menggunakan bahasa pemrograman PHP dan *database* menggunakan MySQL.

Penelitian lain juga dilakukan oleh Dawam Wahyu Wicaksono Universitas Islam Indonesia tahun 2018 yang berjudul “Sistem Informasi Pengelolaan Penjualan Perumahan pada PT. Anthurium Regency”. Menjelaskan bahwa pada PT. Arnthurium Regency untuk mengelola data-data transaksi penjualan masih menggunakan cara konvensional yaitu masih dengan pembukuan manual. Proses dalam memasukan data dan pelaporan yang masih manual menyebabkan pengelolaan data menjadi lambat sehingga memungkinkan terjadinya kesalahan dalam pencatatan. Penyimpanan data juga masih berupa dokumen astau arsip-arsip sehingga seringkali kesulitan dalam memperoleh infomasi calon pembeli dan laporan hasil transaksi penjuala perumahan. Sehingga memunculkan untuk mengembangkan sebuah desain sistem informasi pengelolaan penjualan perumahan menggunakan metode *user experience (UX)* yang sudah terkomputerisasi agar memudahkan dalam mengelola usaha dan pengambilan keputusan[7].

Penelitian sebelumnya yang berjudul “Aplikasi Sistem Informasi Real Estate PT. Greenhouse Berbasis asp.net”. penelitian ini dilaksanakan untuk mengetahui seberapa besar dampak positif dan negative yang akan ditimbulkan jika seseorang menggunakan jasa perumahan ini. Dimana disetiap keputusan yang diambil pasti akan memberikan dampak positif dan negatif terhadap kehidupan.

Dari penelitian yang disebutkan sebelumnya penulis bermaksud akan merancang Sistem Informasi Rumah Berbasis Website sebagai

media promosi penjualan rumah diperumahan OASE Residence Majenang sebagai sebuah sistem yang membantu seseorang yang mencari rumah. Yang ingin dicapai penulis bahwa hasil perancangannya dapat diaplikasikan diperumahan OASE Residence Majenang.

2.2 Landasan Teori

Dalam penelitian ini perlu adanya teori-teori yang mendasar untuk menunjang proses penelitian ini, teori-teori tersebut adalah :

2.2.1 Pengertian sistem

Sistem adalah kumpulan elemen-elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan[8]. Semua sistem tidak memiliki kombinasi elemen-elemen yang sama, tetapi memiliki susunan dasar. Sumber daya input diubah menjadi sumber daya *output*, sumber daya mengalir dari elemen input melalui elemen transformasi kepada elemen *output*. Suatu mekanisme kontrol memantau proses transformasi untuk menyakinkan bahwa proses tersebut memenuhi tujuannya.

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri atas komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk menggambarkan suatu set entitas yang berinteraksi, dimana suatu model matematika sering kali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggerak yaitu rakyat yang berada dinegara tersebut.

Kata “sistem” banyak digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dalam dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka. Dari penjabaran pengertian sistem diatas maka bisa

diambil kesimpulan bahwa sistem itu memang kompleks dan sangat terkait dengan hal yang ada didalamnya, karena sistem tidak akan jalan apabila salah satu elemen tersebut tidak jalan.

2.2.2 Pengertian Informasi

Informasi merupakan hasil pemrosesan data yang diperoleh dari setiap elemen sehingga mudah dipahami[8]. Istilah “informasi : yang bersal dari bahasa Perancis kuno, “*informacion*’ yang mengambil dari bahasa Latin “*informare*” yang artinya aktivitas dalam pengetahuan yang dikomunikasikan. Dalam beberapa pengetahuan tentang peristiwa tertentu yang telah dikumpulkan atau dari berita dapat juga dikatakan sebagai informasi. Dalam ilmu komputer, informasi adalah data yang disimpan, diproses atau ditransmisikan. Para ahli meneliti konsep informasi sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi.

Pengertian atau devinisi informasi lainnya yaitu data yang telah diberi makna. Sebagai contoh, dokumen berbentuk spreadsheet (*Ms. Excel*) sering digunakan untuk membuat informasi dari data yang ada didalamnya. Laporan laba rugi dan neraca merupakan salah satu bentuk informasi. Sedangkan angka yang terdapat didalamnya adalah data yang telah diproses sehingga menjadi berguna bagi siapa saja yang menggunakannya.

Jenis-jenis informasi :

1. Berdasarkan penyampaiannya :
 - a. Informasi yang disediakan secara berskala.
 - b. Informasi yang disediakan secara tiba-tiba.
 - c. Informasi yang disediakan setiap saat.
 - d. Informasi yang dikecualikan.
 - e. Informasi yang diperoleh berdasarkan permintaan.
2. Berdasarkan kegunaan :
 - a. Informasi yang menambah pengetahuan.
 - b. Informasi yang memberi edukaif bagi pembacanya.
 - c. Informasi berdasarkan format penyajian.

2.2.3 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan kombinasi dari manusia, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber data dimana mengumpulkan, mengubah, dan menyebarkan

informasi dari sebuah organisasi. Adapun pengertian lainnya yaitu sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu[9]. Komponen-komponen sistem informasi sebagai berikut :

1. Komponen Input

Input merupakan data yang masuk ke dalam sistem informasi.

2. Komponen Model

Kombinasi prosedur, logika, dan model matematik yang memproses data yang tersimpan dibasis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen Output

Output informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Komponen Teknologi

Teknologi merupakan alat dalam sistem informasi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan output, dan membantu pengendalian sistem.

5. Komponen Kontrol

Pengendalian yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

Karakter Sistem Informasi :

1. Sistem informasi memiliki komponen yang berupa subsistem yang merupakan elemen-elemen yang lebih kecil yang membentuk sistem informasi tersebut, misalnya bagian input, proses, output.

2. Ruang lingkup sistem informasi yaitu ruang lingkup yang ditentukan dari awal pembuatan yang merupakan garis batas lingkup kerja sistem tersebut sehingga sistem informasi tersebut tidak bersinggung dengan sistem informasi lainnya.

3. Tujuan sistem informasi adalah pokok yang harus ditentukan dan dicapai dengan menggunakan sistem informasi tersebut, sebuah informasi dianggap berhasil apabila dapat mencapai tujuan tersebut.

4. Lingkup sistem informasi yaitu sesuatu yang berada di luar ruang lingkup sistem informasi yang dapat mempengaruhi sistem informasi, hal ini dipertimbangkan pada saat perencanaan sistem informasi.

2.2.4 Pengertian Waterfall

Model *waterfall* ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial linear[6]. Menurut Ian Sommerville, yaitu :

1. *Requirement*

Merupakan tahapan fitur, melakukan konsultasi dengan pengguna sistem untuk mengetahui kendala dan tujuan sistem, kemudian dirinci menggunakan spesifikasi sistem.

2. *System and Software Design*

Merupakan tahapan identifikasi dan penggambaran mengenai abstrak dasar perangkat lunak beserta hubungannya.

3. *Implementation and Unit Testing*

Merupakan suatu tahapan yang diterapkan hasil dari desain untuk dijadikan suatu program.

4. *Integration and System Testing*

Merupakan suatu tahapan integrasi program dan diuji sebagai sistem yang utuh untuk memastikan bahwa sistem memenuhi persyaratan yang ada.

5. *Operation and Maintenance*

Merupakan suatu tahapan sistem mulai di-*install* atau dilakukan tes dan akan diperbaiki jika terjadi *error* yang tidak ditemukan pada pembuatan.

2.2.5 Pengertian Rumah

Rumah merupakan salahsatu bangunan yang dibuat seindah dan nyaman mungkin bagi penghuni tersebut. Adapun pengertian tentang perumahan yaitu sekelompok rumah yang berfungsi sebagai lingkungan tempat tinggal atau lingkungan hunian yang dilengkapi dengan sarana dan prasarana lingkungan.

2.2.6 Website

Website adalah kumpulan informasi/kumpulan page yang biasa diakses lewat jalur internet. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara online di jaringan internet[10]. Secara teknis, *website* adalah kumpulan dari page yang tergabung kedalam suatu domain atau subdomain tertentu. *Website-website* yang ada berada di dalam *World Wide Web*(WWW)

interet. *Website* sendiri pertama kali di buat leh Tim Berners-Lee pada akhir 1980an dan baru resmi online pada tahun 1991.

2.2.7 MySQL

MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya. SQL (*Structured Query Language*) adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan lebih mudah [11]. Seperti yang sudah disinggung, MySQL masuk kedalam jenis RDBMS (Relational Database Management System). Maka istilah semacam baris, kolom, tabel, yang dipakai pada MySQL. Contohnya didalam MySQL sebuah database terdapat satu atau beberapa tabel [12].

2.2.8 Pengertian PHP (*Personal Home Page*)

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. Disebut bahasa pemrograman *script server-side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*.

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya PHP menjelma menjadi bahasa pemrograman web yang *powerfull* dan tidak hanya digunakan untuk membuat web sederhana tetapi juga web populer yang digunakan oleh jutaan orang seperti *Wikipedia*, *Wordpress*, *Jomla*.

2.2.9 Pemrograman Berbasis Objek

Pemrograman berorientasi objek adalah sebuah paradigma pemrograman yang berorientasi kepada sebuah objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam suatu kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya secara terstruktur.

Berikut konsep dasar dalam mendalami Pemrograman Berbasis Objek :

1. Kelas

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.

2. Objek

Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berbasis objek.

3. Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk fokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

4. *Encapsulation* (Pembungkusan)

Encapsulation adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari interferensi dan penggunaan yang tidak perlu. Salah satunya dengan cara membentuk objek.

5. *Inheritance* (Pewaris)

Inheritance memungkinkan programmer meletakkan member yang sama dalam satu kelas dan kelas-kelas lain dapat mewarisi member tersebut. Kelas yang mengandung anggota yang sama dari beberapa kelas lain dinamakan *superclass* atau *parent class* sedangkan kelas yang mewarisi dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan *class hierarchy*.



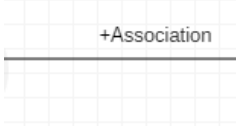

6. *Polymorphism* (Perbedaan Bentuk)

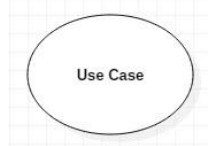

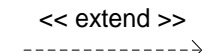
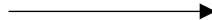
Polymorphism artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic*, bila objek-objek itu mempunyai antar muka yang *identical* namun mempunyai perilaku yang berbeda.

a) **Usecase Diagram**

Usecase diagram adalah suatu urutan interaksi yang saling berkaitan antara sistem dan aktor. Usecase dijalankan melalui cara menggambarkan tipe interaksi antara user suatu program dengan sistemnya sendiri. Usecase dipakai untuk membentuk perilaku (behavior) sistem yang akan dibuat. Sebuah use case menggambarkan sebuah interaksi antara pengguna dengan sistem yang sudah ada.

Tabel 2. 1 *Use Case Diagram*







| NO | Simbol | Nama | Keterangan |
|----|-----------------------------------------------------------------------------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 |  | <i>Actor</i> | Merupakan peran orang, sistem yang lain, atau alat etika berhubungan dengan use case. |
| 2 |  | <i>Include</i> | Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya. |
| 3 |  | <i>Association</i> | Abstraksi dari penghubung antara aktor dengan use case |
| 4 |  | <i>interaction</i> | Digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya <i>interaction</i> dilengkapi dengan <i>teks</i> bernama <i>operation signature</i> yang disusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan. |


| | | | |
|---|-----------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 |  | <i>Use Case</i> | Abstraksi dari penghubung antara aktor dengan use case |
| 6 |  | <i>Dependency</i> | Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah. |
| 7 |  | <i>Extend</i> | Menunjukkan bahwa suatu bagian dari elemen bisa disisipkan ke dalam elemen yang lain. |
| 8 |  | <i>generalization</i> | Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i> , <i>class</i> yang lebih spesifik (<i>subclass</i>) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum (<i>superclass</i>). |

b) Sequence Diagram

Sequence diagram adalah suatu diagram UML yang memodelkan logika dari suatu use case dengan menggambarkan interaksi berupa pengiriman pesan(*message*) antar obyek dalam urutan waktu.

Tabel 2. 2 *Sequence Diagram*

| No | Simbol | Nama | Keterangan |
|----|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. |  | <i>Object Lifeline</i> | Menggambarkan <i>object</i> apa saja yang terlibat. |
| 2. |  | <i>Actor</i> | Menggambarkan <i>actor</i> yang terlibat. |
| 3. |  | <i>Lifeline Control</i> | Digunakan untuk menggambarkan sebuah <i>form</i> . |
| 4. |  | <i>Lifeline Entity</i> | Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan. |
| 5. |  | <i>Lifeline Control</i> | Digunakan untuk menghubungkan <i>boundary</i> dengan tabel. |
| 6. |  | <i>Time Active</i> | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
| 7. |  | <i>Synchronous</i> | Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Synchronous</i> mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya. |

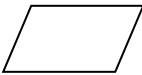

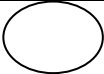

| No | Simbol | Nama | Keterangan |
|----|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8. |  | <i>Asynchr onous</i> | Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Asynchronous</i> memberikan kita fasilitas untuk menjalankan proses lain ketika proses sebelumnya belum selesai. |

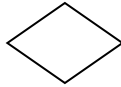


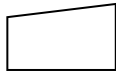

c) Flowchart

Flowchart atau sering disebut diagram air adalah prosedur atau program sitem yang diarahkan menggunakan bagan air. Digunakan untuk menjelaskan tahapan pemecahan masalah dengan cara mempresentasikan simbol-simbol tertentu yang mudah digunakan dan dipahami.

Berikut simbol-simbol flowchart:

Tabel 2. 3 *Flowchart*



| NO | Simbol | Nama | Keterangan |
|----|-------------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------------------------------|
| 1 |  | <i>Input/O utput</i> | Mempresentasikan <i>input</i> data atau <i>output</i> data yang diproses atau informasi. |
| 2 |  | Proses | Mempresentasikan operasi |
| 3 |  | Penghu bung | Keluar ke atau masuk dari bagian lain, khususnya halaman yang sama. |
| 4 |  | Anak Panah | Mempresentasikan alur kerja. |

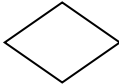

| | | | |
|---|-----------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------|
| 5 |  | Keputusan | Keputusan dalam program. |
| 6 |  | <i>Terminal Port</i> | Awal/akhir <i>Flowchart</i> |
| 7 |  | Dokumen | <i>Input/Output</i> dalam format yang dicetak. |
| 8 |  | <i>Manual Input</i> | <i>Input</i> yang dimasukkan secara manual dari <i>keyboard</i> . |
| 9 |  | <i>Manual Operation</i> | Operasi manual. |

d) **ERD (Entity Relationship Diagram)**

ERD atau *Entity Relationship Diagram* merupakan diagram yang memperlihatkan entitas-entitas yang telah diketahui atributnya yang terlibat dalam suatu sistem serta hubungan-hubungan (relasi) antar entitas tersebut. Berikut adalah simbol-simbol ERD ;

Tabel 2. 4 ERD

| No | Simbol | Nama | Keterangan |
|----|-------------------------------------------------------------------------------------|------------------------|--------------------------------------------------------------------------------------------------------------|
| 1 |  | Himpunan <i>Entity</i> | Sesuatu apa saja yang di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. |
| 2 |  | Atribut | Sifat atau karakteristik dari tiap-tiap entitas maupun tiap <i>relationship</i> . |

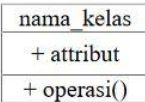
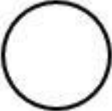



| | | | |
|---|-----------------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------|
| 3 |  | Himpunan Relasi | Kumpulan semua relasi diantara entitas-entitas yang terdapat dalam himpunan entitas. |
| 4 |  | <i>Link</i> | Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut. |



e) Class Diagram

Class Diagram merupakan sebuah pemodelan kelas yang berisikan variable-variabel dan fungsi-fungsi yang sesuai dengan kebutuhan sistem.

Pemodelan *class diagram* memerlukan symbol-simbol dimana memiliki deskripsi tertentu. Berikut merupakan symbol-simbol yang ada di *class diagram*.

Tabel 2. 5 *Class Diagram*

| No | Simbol | Nama | Keterangan |
|----|-------------------------------------------------------------------------------------|-------------------|---------------------------------------------------------------------------|
| 1 |  | Kelas | Kelas pada struktur sistem |
| 2 |  | Interface | Sama dengan konsep interface dalam pemrograman berorientasi objek |
| 3 |  | Asosiasi | Relasi antarclass dengan arti umum |
| 4 |  | Directed Asosiasi | Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas lain. |
| 5 |  | generalisasi | Relasi antarkelas dengan makna umum-khusus |

| | | | |
|---|-----------------------------------------------------------------------------------|-------------|-----------------------------------------------------------|
| 6 |  | Dependancy | Relasi antar kelas dengan makna kebergantungan antarkelas |
| 7 |  | Aggregation | Relasi antarkelas dengan makna semua bagian. |

2.3 Metode Pengembangan Sistem

Dalam merancang Sistem Informasi Pemesanan Rumah Berbasis *Website* ini menggunakan metode pengembangan *waterfall*, karena dalam metode ini merupakan metode yang menggunakan pendekatan sistematis secara urut dalam pengembangannya, yaitu dimulai dari :

1. *Requirement* (analisis kebutuhan)

Merupakan tahap pengembangan yang mengetahui seluruh informasi mengenai kebutuhan software seperti kegunaan software yang diinginkan pengguna dan batasan software.

2. *Design*

Berujuan untuk memberikan gambaran lengkap tentang apa yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem yang diinginkan.

3. *Coding* (Penulisan kode program)

Coding merupakan tahapan penerjemah kedalam bahasa pemrograman yang dilakukan oleh programmer. Programmer nantinya akan menuliskan bahasa tertentu yang dapat dimengerti oleh komputer.

4. *Testing* (Pengujian Program)

Setelah melakukan analisa, design dan pengkodean maka selanjutnya adalah tahap pengujian program. Pada tahap ini semua pernyataan dipastikan telah diuji dan memastikan setiap kesalahan program atau *error* sudah diuji.

5. *Maintenance* (Pemeliharaan)

Apabila semua sudah diuji maka tahap selanjutnya adalah pemeliharaan yaitu tahap terakhir dalam pembuatan sistem apabila

terjadi penambahan fitur dalam pembuatan maka akan langsung diperbaiki dalam sistem.

2.4 Metode Pengujian Sistem

Pengujian black-box adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja[13]. Pengetahuan khusus dari kode aplikasi/struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun disekitar spesifikasi dan persyaratan yakni aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi persyaratan dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional meskipun biasanya fungsional. Perancangan uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji. Pengujian pada Black-box berusaha menemukan kesalahan seperti :

1. Fungsi-fungsi yang tidak benar dan hilang.
2. Kesalahan interace.
3. Kesalahan dalam structure data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.