

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Pada bab ini dijelaskan tentang teori – teori yang berhubungan dengan pembuatan web tersebut dikutip.

A. Website

Menurut Ardhana (2013 : 3) *Word Wide Web* atau lebih sering dikenal sebagai Web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang penelusuran informasi melalui internet). Keistimewaan inilah yang telah menjadikan web sebagai *service* yang paling cepat pertumbuhannya.

a. Web Browser

Menurut Limantara (2009:1) menjelaskan bahwa “web *browser* merupakan aplikasi perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan teks, *image*, video, *games*, dan informasi lainnya yang berlokasi pada halaman web pada *Word Wide Web* (WWW) atau *Local Area Network* (LAN)”. Teks dan *image* pada halaman web dapat berisi *hyperlink* ke halaman web lain pada *website* yang sama maupun berbeda.

Dengan web *browser* seorang pengguna bisa mengakses informasi yang disediakan pada banyak *website* secara cepat dan mudah. Meskipun *browser* pada umumnya digunakan untuk mengakses *Word Wide Web*, *browser* juga bisa digunakan untuk mengakses informasi yang disediakan web *server* pada jaringan

privat atau pada *system file*. Contoh web *browser*: *Microsoft Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*, dan lain-lain.

Fungsi utama web *browser* adalah:

1. Memungkinkan untuk mengambil dan melihat informasi dari komputer server *www*, *ghoper* dan *FTP* di internet, atau media penyimpanan yang berisi dokumen *HTML*.
2. Berinteraksi dengan sistem berbasisan
3. Merupakan alat untuk melihat dokumen elektronik.
4. Untuk melakukan *downloading/upload* informasi
5. Untuk mengirim dan menerima *E-mail*.

b. Web Server

Menurut Supardi (2010:181) menjelaskan bahwa “*Web Server* merupakan perangkat lunak yang mengatur atau mengelola program berdasarkan permintaan *browser* dan dikirimkan kembali ke *browser*”. *Browser* sendiri merupakan perangkat lunak untuk menjalankan program atau *script* web. Setiap program web *server* bekerja dengan menerima permintaan *HTTP* dari klien, dan memberikan respon *HTTP* ke klien tersebut. Respon *HTTP* biasanya mengandung dokumen *HTML* tetapi dapat juga berupa berkas *raw*, gambar, dan berbagai jenis dokumen lainnya. Jika terjadi kesalahan permintaan dari klien atau terjadi masalah saat melayani klien maka web *server* akan mengirimkan respon kesalahan yang dapat berupa dokumen *HTML* atau teks yang memberi penjelasan penyebab terjadinya kesalahan.

Umumnya setiap web *server* mempunyai kemampuan untuk melakukan pencatatan/*logging* terhadap informasi detil mengenai permintaan klien dan respon dari web *server* dan disimpan dalam berkas log, dengan adanya berkas log ini maka akan memudahkan web *master* untuk mendapat statistik dengan menggunakan *tool log analyzer*. Pada penggunaan sehari-hari banyak web *server* mengimplementasikan fitur-fitur berikut:

1. Otentifikasi fitur untuk mengotorisasi suatu permintaan dari klien sebelum menggunakan sumber daya yang dimiliki oleh web *server* (biasanya *User* dan *Password*).
2. Dukungan HTTPS (dengan SSL, atau TLS) yang memungkinkan koneksi yang aman (dengan enkripsi) ke *server* pada port 433 berbeda dengan koneksi HTTP biasa di *port*
3. Pengatur *Bandwith* (*Bandwith throttling*) yang berfungsi untuk membatasi kecepatan respon dengan tujuan tidak membanjiri jaringan dan menghemat pita data (*bandwith*) agar dapat melayani klien lebih banyak.

Setiap *server* web telah menentukan batas beban yang dapat ditanggung, sehingga setiap *server* web mempunyai batasan jumlah klien yang terhubung pada satu waktu (umumnya antara 2 s.d. 60.000, secara bawaan (*default*) maksimal 500 atau 1000). Jika *server* web sudah mendekati batas limit yang mampu ditangani maka *server* menjadi kelebihan beban sehingga lambat dalam memberikan respon.

B. Bahasa pemrograman

a. PHP (*Hypertext Preprocessor*)

Menurut Raharjo, dkk (2014:47) “PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web.” Ketika dipanggil dari web *browser*, program yang tertulis dengan PHP akan diparsing didalam web *server* oleh *interpreter* PHP dan diterjemahkan kedalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web *browser*. Karena pemrosesan program PHP dilakukan dilingkungan web *server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah “*view source*” pada web *browser* yang mereka gunakan.

b. HTML (*Hypertext Markup Language*)

Menurut Raharjo, dkk (2014:343) “HTML merupakan file teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi web *browser*.” Setiap aplikasi di web selalu dibuat dengan menggunakan kode HTML. Oleh karena itu, dokumen HTML sering disebut juga sebagai web *page* (halaman web). Untuk membuat dokumen HTML, kita tidak tergantung kepada aplikasi tertentu, karena dokumen HTML dapat dibuat menggunakan Aplikasi *Texts Editor* apapun, bisa *Notepad* (untuk lingkungan *Ms Windows*), *Emacs* atau *Vi Editor* (untuk lingkungan *Linux*), dan sebagainya.

c. *Java Script*

Menurut Sigit (2011:1) *JavaScript* merupakan bahasa *Scripting* yang bekerja disisi *Client/Browser* sehingga *website* bisa lebih interaktif.

JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *JavaScript* dapat disisipkan dalam halaman web menggunakan *tag script*.

d. *Jquery*

Menurut sigit (2011:2) “*Jquery* adalah *library* atau kumpulan kode *Java Script* siap pakai .” keunggulan menggunakan *Jquery* dibandingkan dengan *java script* standar, yaitu menyederhanakan kode *java script* standar, yaitu menyederhanakan kode *java script* dengan cara memanggil fungsi-fungsi yang disediakan oleh *jquery*. *Java script* sendiri merupakan bahasa *scripting* yang bekerja di sisi klien/*browser* sehingga *website* bisa lebih interaktif.

e. *Twitter Bootstrap*

Menurut Widyantoro (2014:6),” *Bootstrap* adalah sebuah *framework* yang dapat menyelesaikan permasalahan dalam mendesain web. Slogan dari *framework* ini adalah “*Sleek, intuitive, and powerful front-end framework for faster and easier web development*”, yang berarti kita dapat mendesain sebuah *website* dengan lebih rapi, cepat dan mudah. Selain itu *Bootstrap* juga *responsive* terhadap banyak *platform*, artinya tampilan halaman *website* yang menggunakan *Bootstrap* ini akan tampak tetap rapi, baik versi *mobile* maupun *desktop*.”

C. Basis Data

Menurut Fathansyah (2012:2) basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep dan sebagainya. Basis data (*database*) dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan kedalam media penyimpanan elektronik.

Aplikasi Basis Data :

1. Database

Menurut MADCOMS (2016:145) *database* adalah “ sebuah sistem yang 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*, mengijinkan pembuatan, *maintenance* data dan pencarian akses yang lain.

Fungsi-fungsi dasar yang harus didukung DBMS adalah :

a. Pendefinisian data (*data definition*)

DBMS harus dapat menerima pendefinisian data (skema eksternal, skema konseptual dan semua asosiasi pemetaan) dari sumber dan mengkonversikan ke dalam bentuk objek yang sesuai.

b. Manipulasi data (*data manipulation*)

DBMS harus dapat menangani permintaan untuk mengambil, memperbaharui atau menghapus data yang sudah ada di basis data, maupun menambah data kedalam basis data.

c. Optimalisasi dan eksekusi (*optimization and execution*)

Permintaan Data *Manipulation Language* (DML) harus diproses dikomponen pengoptimalisi yang bertujuan untuk menentukan cara yang efisien untuk implementasi permintaan. Permintaan yang telah dioptimalisasi kemudian dieksekusi dibawah kendali *re-time manager*.

d. Keamanan dan integritas data (*data security and integrity*)

DBMS harus mengawasi permintaan pengguna dan menolak gangguan yang dapat membahayakan keamanan dan *integrity constraint* yang sudah ditentukan oleh *Database Administrator* (DBA).

e. Perbaikan data dan konkurensi (*data recovery and concurrency*)

DBMS dapat juga disebut *Transaction Processing Monitor* (TPM) harus melakukan kendali perbaikan dan konkurensi.

f. Kamus data (*data dictionary*)

Kamus data berisi “data mengenai data” adalah definisi dari objek lain disistem. Semua skema dan pemetaan, berbagai sistem keamanan, dan

integrity constraint akan disimpan, baik dari sumber maupun bentuk objek didalam kamus data.

g. Kinerja (*performance*)

DBMS harus dapat mengerjakan semua tugas seefektif mungkin.

2. MySQL (*My Structure Query Language*)

Menurut Arief (2011:152) “MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengolahan datanya”.

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan *Perl*). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP.

3. XAMPP

Menurut Wicaksono (2008:7) menjelaskan bahwa “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal”. XAMPP berperan sebagai *server* web pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

4. **PhpMyAdmin**

Menurut Sibero (2011:376) phpMyAdmin adalah aplikasi web yang dibuat oleh phpmyadmin.net phpMyAdmin digunakan untuk administrasi *database* MySQL.

D. **Pengembangan Perangkat Lunak**

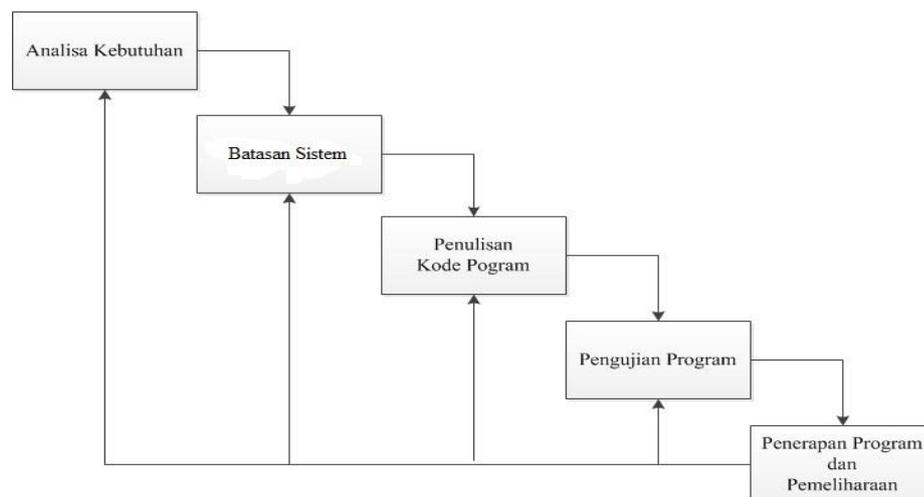
Menurut Rosa dan Shalahuddin (2013:28) “Model air terjun (*Waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisi, desain, pengkodean, pengujian, dan tahap pendukung (*support*)”.

Teknik pengembangan perangkat lunak menggunakan metode *incremental* yang sudah dikembangkan dari *waterfall model*, karena metode *incremental* ini terdiri dari tahap-tahap yang memberikan kemudahan, jika pada suatu tahap tidak sesuai atau mengalami kesalahan maka dapat kembali ketahap sebelumnya. Dengan berkembangnya perangkat lunak, pembuat program (*programmer*) secara umum langsung pembuatan pengkodean perangkat lunak, tanpa menggunakan prosedur perangkat lunak.

SDLC (*Software Development Life Cycle*) diperkenalkan pada tahun 1960-an, SDLC pertama digunakan oleh para pengusaha besar jaman itu. Dimana sistem-sistem yang dibangun untuk mengelolah informasi kegiatan dan aktivitas yang berpotensi memiliki data yang besar.

Tahapan-tahapan yang terdapat dalam metode waterfall adalah sebagai berikut:

1. Analisa Kebutuhan adalah kondisi yang harus dipenuhi dalam suatu produk baru atau perubahan produk, yang mempertimbangkan berbagai kebutuhan yang bersinggungan antar berbagai pemangku kebutuhan.
2. Batasan sistem adalah batasan yang diberikan kepada sistem, agar sistem tersebut tidak menyimpang dari prosedur yang diinginkan.
3. Penulisan Kode Program adalah desain program yang diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan.
4. Pengujian Program adalah penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).
5. Penerapan Program adalah mengoperasikan program dilingkungannya dan melakukan penyesuaian atau perubahan adaptasi dengan situasi sebenarnya.



Sumber: Rosa dan Shalahuddin (2013:28)

Gambar II.1.

Diagram metode waterfall

2.2. Teori Pendukung

Ada beberapa teori pendukung yang penulis gunakan guna melengkapi penulisan tugas akhir ini adalah sebagai berikut :

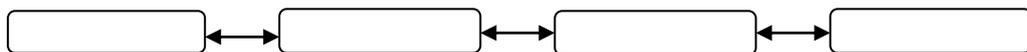
A. Struktur Navigasi

Menurut Kurniawan (2010:213) "Navigasi adalah fitur yang harus disediakan. Navigasi berfungsi untuk berpindah dari satu halaman ke halaman yang lain pada suatu aplikasi berbasis web, yang juga digunakan untuk memberikan informasi lokasi halaman yang sedang dibuka".

Ada empat macam bentuk dasar dari struktur navigasi yang biasa digunakan yaitu:

1. *Linear* (Satu Alur)

Linear (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurutan. Salah satu yang terpenting dari struktur ini adalah tidak diperkenankan terjadinya percabangan.



Sumber: (Kurniawan : 2010)

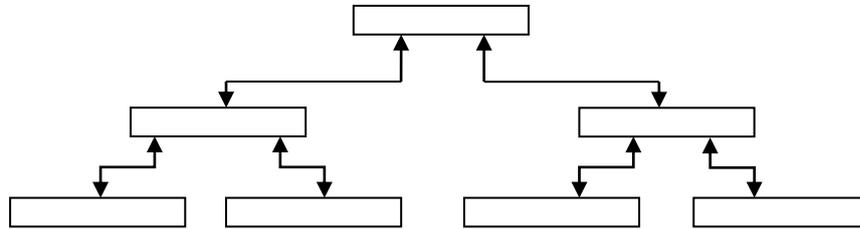
Gambar II.2.

Struktur Navigasi *Linear*

2. *Hierarchical* (Bercabang)

Struktur *Hierarchi* (bercabang) ini percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utaman kesatu), halaman utama ini akan mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan ,

maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya. Yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara *linear*.



Sumber: (Kurniawan : 2010)

Gambar II.3.

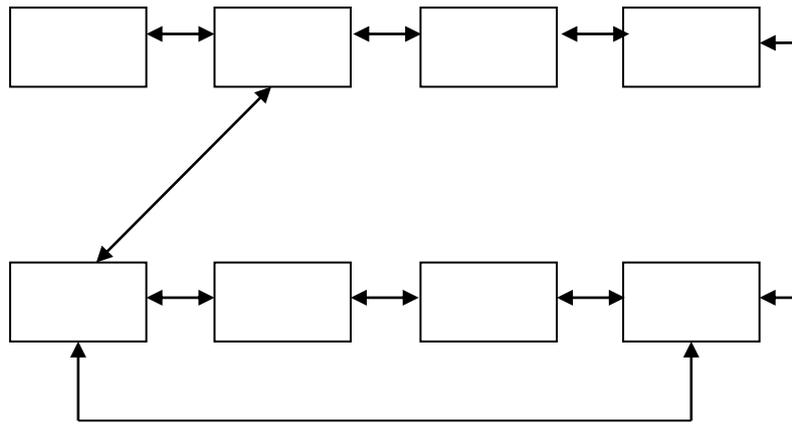
Struktur Navigasi *Hierarkis*

3. *Non Linear* (Tidak berurut)

Struktur penjejakan *Non Linear* (tidak berurut) merupakan pengembangan dari struktur penjejakan *Linear*. Pada struktur ini diperkenankan membuat penjejakan bercabang.

Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses ke semua halaman manapun.

Percabangan yang dibuat pada struktur *Non Linear* ini berbeda dengan percabangan yang dibuat pada struktur *Hierarchi*, Karena pada percabangannya *Non Linear* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master Page* dan *Slave Pag*.



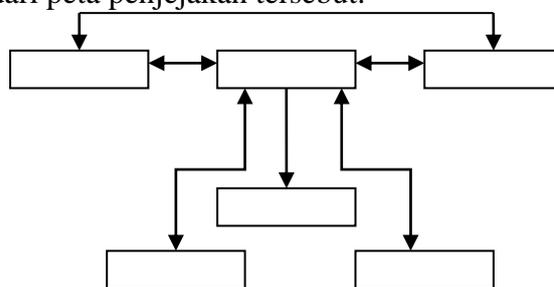
Sumber: (Kurniawan : 2010)

Gambar II.4.

Struktur Navigasi Non Linear

4. *Composite* (Campuran)

Composite (campuran) atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear*, dan *Hierarchi*. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut. Penggunaan peta penjejakan bergantung kepada kebutuhan dan tujuan dari *web* yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pembuatan *page* dari peta penjejakan tersebut.



Sumber: (Kurniawan : 2010)

Gambar II.5.

Struktur Navigasi Komposit

B. ERD (*Entity Relationship Diagram*)

Menurut Brady dan Loonam (2010), *Entity Relationship diagram (ERD)* adalah teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analysts* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

Menurut Brady dan Loonam (2010), *Entity Relationship diagram (ERD)* memiliki 3 komponen utama, yaitu :

a. Entitas

Entitas adalah objek yang menarik di bidang organisasi yang dimodelkan.

Entitas biasanya disebut sebagai kata benda tunggal dan direpresentasikan sebagai persegi panjang yang empuk dalam diagram entitas-hubungan.

b. Hubungan

Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

c. Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.

C. LRS (*Logical Record Structure*)

LRS adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas.

Menurut Hasugian dan Sidiq (2012 : 608) memberikan batasan bahwa LRS adalah “Sebuah model sistem yang digambarkan dengan sebuah diagram-ER dan mengikuti pola atau aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS”. Perubahan yang terjadi yaitu mengikuti aturan – aturan sebagai berikut menurut Hasugian dan Sidiq (2012 : 608) :

- a. Setiap entitas akan diubah kebentuk kotak.
- b. Setiap atribut relasi disatukan dalam kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi)
- c. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

D. Pengujian Web

Menurut Fatta (2007:170) Pengujian web adalah “proses eksekusi suatu program untuk menemukan kesalahan, sedangkan pengujian adalah tahap yang harus dilakukan”. Pengujian harus mencakup unit *testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen yang lain. Kemudian modul *testing* harus menyusul dilakukan untuk mengetahui apakah

penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Berikut ringkasan beberapa kategori *test* yang bisa dilakukan :

1. *Stub Testing*

Pengujian yang difokuskan pada pengujian struktur kendali sebelum semua modul dituliskan. Pengujian ini penting untuk mengecek apakah struktur kendali sudah memetakan kinerja keseluruhan modul secara tepat.

2. *Unit Testing*

Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 Metode untuk melakukan *unit testing*, yaitu:

- a. *Black Box Testing*, Terfokus pada apakah unit program memenuhi kebutuhan (requirement) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengensekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan. Jika ada yang unit yang tidak sesuai outputnya maka untuk menyelesaikannya, diteruskan pada pengujian yang kedua, yaitu *white box testing*.
- b. *White Box Testing*, cara pengujian dengan melihat kedalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program,

variabel, dan *parameter* yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang.

3. *Integration Testing*

Pengujian interaksi dari modul-modul yang menyusun sistem informasi untuk menjamin bahwa mereka bekerja dengan baik. *Integration test* terdiri dari serangkaian tes sebagai berikut:

- a. Ujicoba antarmuka
- b. Ujicoba skenario pengguna
- c. Ujicoba aliran data
- d. Ujicoba sistem antarmuka

4. Pengujian Sistem

Pengujian sistem yang terdiri dari sistem transmisi data perangkat keras (komputer, *magnetic reader tes*).