



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Asropudin (2013:19), komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.

Sujatmiko (2012:156), komputer adalah mesin yang dapat mengolah data *digital* dengan mengikuti serangkaian perintah atau program.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa komputer adalah alat bantu pemrosesan data secara elektronik dengan mengikuti serangkaian perintah atau program yang tersimpan dalam memori.

2.1.2 Pengertian Aplikasi Web

Raharjo (2016:37), aplikasi *web* adalah aplikasi yang disimpan dan dieksekusi di lingkungan *web server*.

Sujatmiko (2012:23), aplikasi *web* adalah suatu aplikasi yang diakses menggunakan penjelajah *web* melalui suatu jaringan seperti *internet* atau *intranet*.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa aplikasi *web* adalah aplikasi yang dieksekusi di lingkungan *web server* melalui suatu jaringan.

2.1.3 Pengertian Program

Sujatmiko (2012:223), program adalah serangkaian petunjuk berupa perintah-perintah yang disusun sedemikian rupa melaksanakan suatu tugas yang akan dikerjakan oleh komputer.

Sutarman (2012:3), program adalah barisan perintah/instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, di mana barisan perintah tersebut berhingga, berakhir, dan menghasilkan *output*.



Dari kedua pengertian di atas, penulis menyimpulkan bahwa program adalah barisan perintah/instruksi yang disusun sedemikian rupa melaksanakan suatu tugas yang akan dikerjakan oleh komputer dan menghasilkan *output*.

2.1.4 Pengertian Sistem

Indrajani (2015:69), sistem adalah sekelompok komponen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan bersama.

Lucas (dikutip Ladjamudin, 2013:3), sistem adalah suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain dan terpadu.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa sistem adalah suatu komponen atau variabel yang terorganisir dan saling berhubungan untuk suatu tujuan bersama.

2.1.5 Pengertian Data

Indrajani (2015:70), data adalah fakta-fakta mentah yang harus dikelola untuk menghasilkan suatu informasi yang memiliki arti bagi suatu organisasi atau perusahaan.

McLeod (dikutip Yakub, 2012:5), data adalah kenyataan yang menggambarkan adanya suatu kejadian (*event*), data terdiri dari fakta (*fact*) dan angka yang secara relatif tidak berarti bagi pemakai.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa data adalah kenyataan yang menggambarkan adanya suatu kejadian yang harus dikelola untuk menghasilkan suatu informasi yang berarti.

2.1.6 Pengertian Informasi

Kristanto (2011:7), informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.

Sutabri (2012:22), informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.



Dari kedua pengertian di atas, penulis menyimpulkan bahwa informasi adalah kumpulan data yang telah diolah menjadi bentuk yang lebih berguna dalam proses pengambilan keputusan.

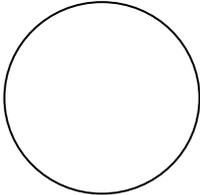
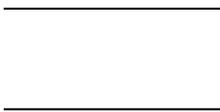
2.2 Teori Khusus

2.2.1 Pengertian *Data Flow Diagram (DFD)*

Indrajani (2015:27), *Data Flow Diagram (DFD)* adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut. Istilah dalam bahasa Indonesianya adalah Diagram Aliran Data.

Sukanto dan Shalahuddin (2013:71) menjelaskan notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut.

Tabel 2.1 Notasi-Notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasi dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>).</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda.</p>



Lanjutan Tabel 2.1 Notasi-Notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>

Sumber : Sukamto dan Shalahuddin (2013:71)

Sukamto dan Shalahuddin (2013:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan.

DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.



3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

Kristanto (2011:68) menjelaskan beberapa petunjuk yang dapat dipakai dalam proses pembuatan simbol DFD. Petunjuk-petunjuk itu adalah sebagai berikut.

1. Penamaan yang jelas
 - a. Sebaiknya menggunakan nama yang mengacu pada fungsi, yaitu gabungan antara kata kerja yang spesifik.
 - b. Jangan menggunakan nama terlalu umum.
 - c. Gunakan nama yang mudah dimengerti oleh pemakai.
2. Memberi nomor pada proses
 - a. Nomor yang diberikan pada proses tidak harus menjadi nomor urut.
 - b. Penomoran dimaksudkan sebagai identifikasi proses dan memudahkan penurunan (level yang lebih rendah) ke proses berikutnya.
3. Penggambaran kembali
 - a. Ukuran dan bentuk lingkaran tetap sama.
 - b. Panah yang melengkung dan lurus tidak menjadi masalah.
4. Hindarilah proses yang mempunyai masukan tetapi tidak mempunyai keluaran begitu pula sebaliknya, hindarilah proses yang mempunyai keluaran tetapi tidak mempunyai masukan.



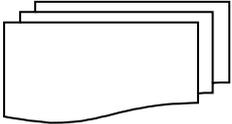
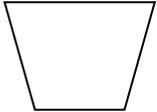
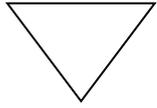
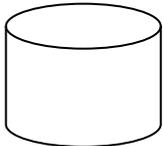
5. Hati-hati dengan aliran data dan proses yang tidak dinamai. Aliran proses yang tidak diberi nama dapat mengakibatkan elemen data yang saling tidak berhubungan akan menjadi satu.

2.2.2 Pengertian *Block Chart*

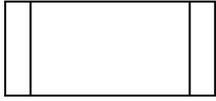
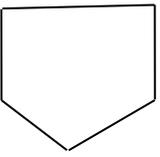
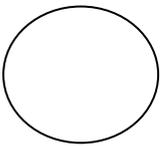
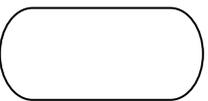
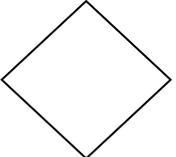
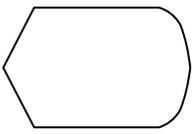
Kristanto (2011:65), *Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Block Chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Kristanto (2011:68) menjelaskan simbol-simbol yang sering digunakan dalam *Block Chart* dapat dilihat pada tabel berikut ini.

Tabel 2.2 Simbol-Simbol dalam *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas, atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual).
6.		Data penyimpanan (<i>data storage</i>).

Lanjutan Tabel 2.2 Simbol-Simbol dalam *Block Chart*

No.	Simbol	Keterangan
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukan data secara manual.

Sumber : Kristanto (2011:68)

2.2.3 Pengertian Flowchart

Indrajani (2015:36), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan suatu prosedur program.



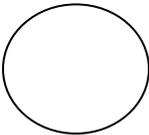
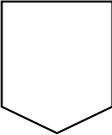
Yakub (2012:162), bagan alir (*flowchart*) adalah bagan yang menggambarkan urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu.

2.2.3.1 Simbol-Simbol *Flowchart*

Ladjamudin (2013:267) menjelaskan *flowchart* disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi tiga kelompok adalah sebagai berikut.

1. *Flow Direction Symbols* (Simbol Penghubung/Alur)
2. *Processing Symbols* (Simbol Proses)
3. *Input-Output Symbols* (Simbol *Input-Output*)

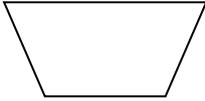
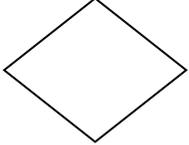
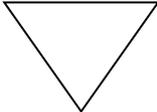
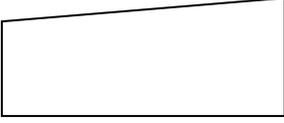
Tabel 2.3 Simbol-Simbol Penghubung/Alur (*Flow Direction Symbols*)

Simbol	Deskripsi
	<p>Simbol Arus/Flow Menyatakan jalannya arus suatu proses.</p>
	<p>Simbol <i>Communication Link</i> Menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.</p>
	<p>Simbol <i>Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.</p>
	<p>Simbol <i>Offline Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.</p>

Sumber : Ladjamudin (2013:263)



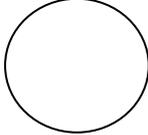
Tabel 2.4 Simbol-Simbol Proses (*Processing Symbols*)

Simbol	Deskripsi
	Simbol Proses Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	Simbol Manual Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).
	Simbol <i>Decision/Logika</i> Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.
	Simbol <i>Predifined Proses</i> Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	Simbol Terminal Meyatakan permulaan atau akhir suatu program.
	Simbol <i>Keying Operation</i> Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
	Simbol <i>Off-line Storage</i> Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
	Simbol Manual Input Memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

Sumber : Ladjamudin (2013:263)



Tabel 2.5 Simbol-Simbol Input-Output (*Input-Output Symbols*)

Simbol	Deskripsi
	Simbol <i>Input-Output</i> Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	Simbol <i>Punched Card</i> Menyatakan input berasal dari kartu atau output ditulis ke kartu.
	Simbol <i>Magnetic-tape unit</i> Menyatakan input berasal dari pita <i>magnetic</i> atau output disimpan ke pita <i>magnetic</i> .
	Simbol <i>Disk Storage</i> Menyatakan input berasal dari disk atau output disimpan ke disk.
	Simbol <i>Document</i> Untuk mencetak laporan ke printer.
	Simbol <i>Display</i> Menyatakan peralatan output yang digunakan berupa layar (video, komputer).

Sumber : Ladjamudin (2013:263)

2.2.4 Pengertian *Entity Relationship Diagram (ERD)*

Sukamto dan Shalahuddin (2013:50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan menggunakan *Entity Relationship Diagram (ERD)*. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

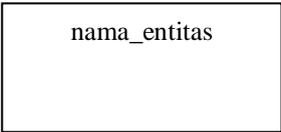
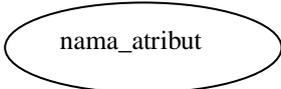
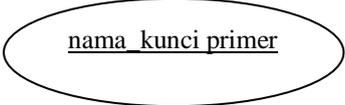
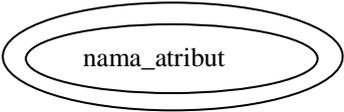
Yakub (2012:60), ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem yang abstrak. ERD juga



menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi.

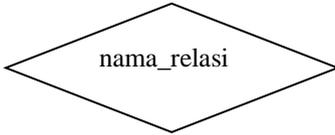
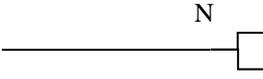
Sukanto dan Shalahuddin (2013:71) menjelaskan, berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen.

Tabel 2.6 Simbol-Simbol ERD dengan Notasi Chen

Simbol	Deskripsi
<p>Entitas / Entity</p> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
<p>Atribut</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
<p>Atribut Kunci Primer</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa <i>id</i> ; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
<p>Atribut Multi Nilai / Multi Value</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.



Lanjutan Tabel 2.6 Simbol-Simbol ERD dengan Notasi Chen

Simbol	Deskripsi
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>
<p>Asosiasi / Association</p> 	<p>Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p>Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.</p>

Sumber : Sukamto dan Shalahuddin (2013:50)

2.2.4.1 Kardinalitas

Ladjamudin (2013:147), kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain.

Kardinalitas relasi terdapat 3 macam, yaitu:

1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.



a. *One to Many* (satu ke banyak)

Yang berarti satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas B, berhubungan dengan paling banyak satu tupel pada entitas A.

b. *Many to One* (banyak ke satu)

Yang berarti setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.

3. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua.

Yang berarti setiap tupel ada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, di mana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

2.2.5 Pengertian Kamus Data (*Data Dictionary*)

Indrajani (2015:30), kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi.

Sukamto dan Shalahuddin (2013:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

1. Nama – nama dari data.
2. Digunakan pada – merupakan proses-proses yang terkait data.
3. Deskripsi – merupakan deskripsi data.
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.



Indrajani (2015:31) menjelaskan notasi-notasi yang digunakan dalam kamus data adalah sebagai berikut.

Tabel 2.7 Notasi Kamus Data

Simbol	Keterangan
=	<i>Is compared of</i> / Terdiri atas
+	<i>And</i> / Dan
()	<i>Optional (maybe present or absent)</i> / Opsional
{ }	<i>Iteration</i> / Iterasi
[]	<i>Select one of several alternative choices</i> / Memilih salah satu alternatif
* *	<i>Comment</i> / Komentar
@	<i>Identifier (key field) for a store</i> / Identifikasi atribut kunci
	<i>Separates alternative choices in the [] construct</i> / Pemisah alternatif

Sumber : Indrajani (2015:31)

2.2.6 Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*).

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

1. Analisis kebutuhan perangkat lunak

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.

2. Desain

Tahap desain adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan



sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan kode program

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain

4. Pengujian

Tahap pengujian fokus pada sistem dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah sistem mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem baru.

2.2.7 Metode Pengujian *Black Box Testing*

Fatta (2007:172), *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 mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.

Yakub (2012:150), *black box testing* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program.

Sukanto dan Shalahudin (2013:275), *black box testing* (pengujian kotak hitam), yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui



apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Asropudin (2013:6), *application* adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms-Word, Ms-Excel.

Sutabri (2012:147), aplikasi adalah alat terapan/alat bantu yang difungsikan secara khusus dan terpadu sesuai dengan kemampuan yang dimilikinya.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa aplikasi adalah *software* yang difungsikan secara khusus dan terpadu untuk mengerjakan tugas-tugas tertentu.

2.3.2 Pengertian Asesmen

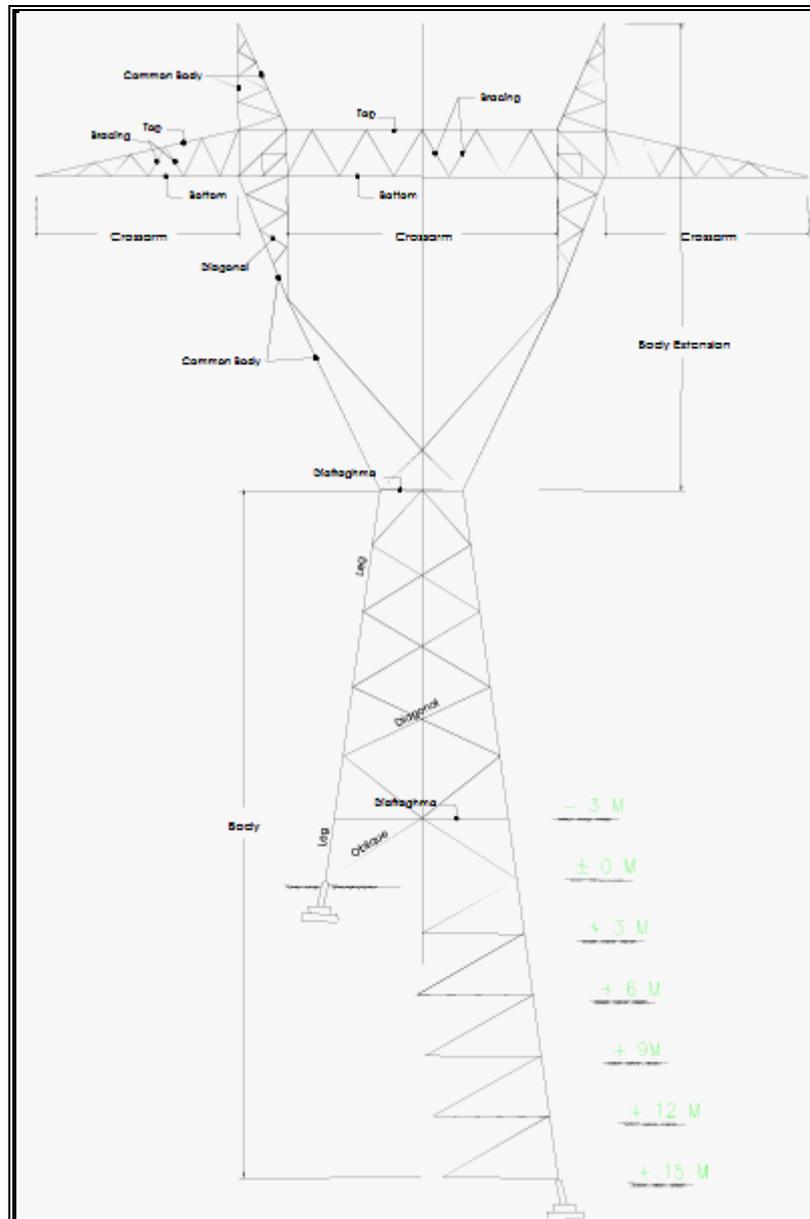
Berdasarkan Buku Pengawasan dan Asesmen SUTT dan SUTET (2014:76), asesmen adalah penilaian untuk penentuan kondisi peralatan.

2.3.3 Pengertian Kondisi

Menurut Kamus Besar Bahasa Indonesia Pusat Bahasa, kondisi adalah persyaratan; keadaan.



2.3.4 Pengertian Saluran Udara Tegangan Ekstra Tinggi (SUTET)

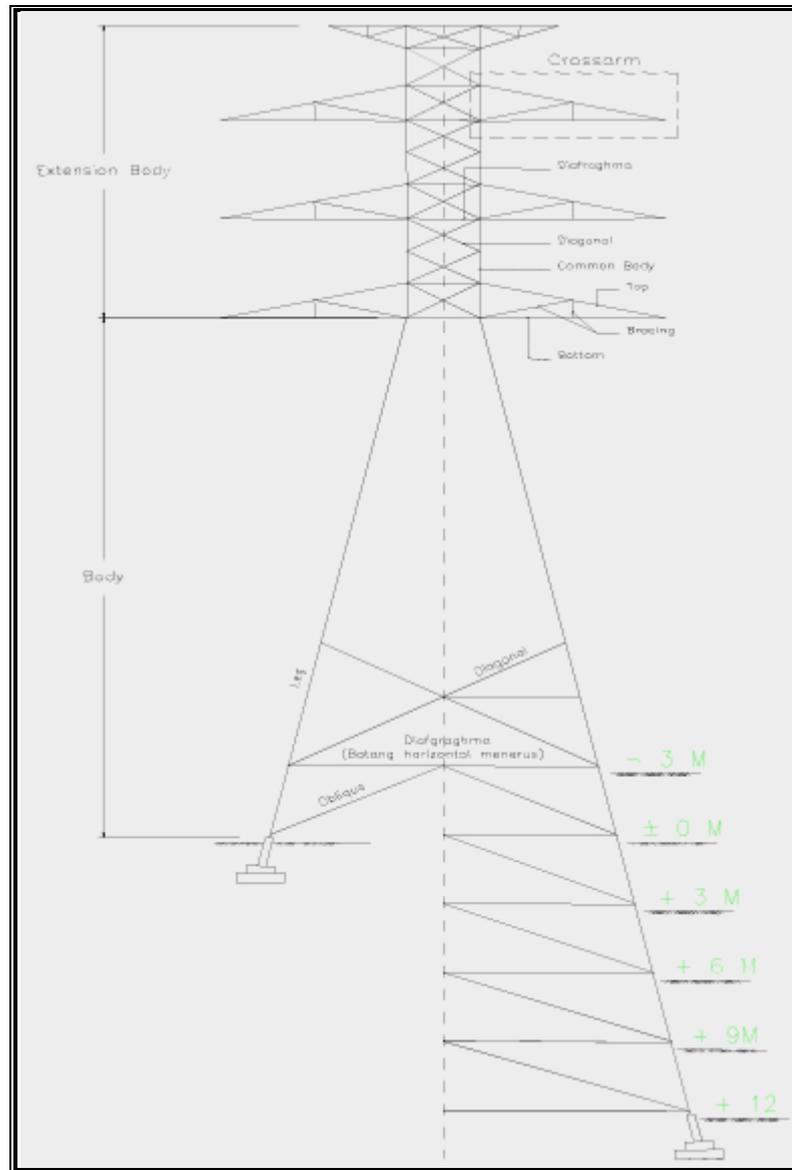


Gambar 2.1 Saluran Udara Tegangan Ekstra Tinggi

Berdasarkan Buku Pengawasan dan Asesmen SUTT dan SUTET (2014:78), SUTET adalah saluran tenaga listrik yang menggunakan kawat telanjang (penghantar) di udara bertegangan di atas 245 kV sesuai standar di bidang ketenagalistrikan.



2.3.5 Pengertian Saluran Udara Tegangan Tinggi (SUTT)



Gambar 2.2 Saluran Udara Tegangan Tinggi

Berdasarkan Buku Pengawasan dan Asesmen SUTT dan SUTET (2014:79), SUTT adalah saluran tenaga listrik yang menggunakan kawat telanjang (penghantar) di udara bertegangan di atas 35 sampai dengan 245 kV sesuai standar di bidang ketenagalistrikan.



2.3.6 Pengertian Website

Asropudin (2013:109), *website* merupakan sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan, serta program interaksi.

Sujatmiko (2012:317), *web* adalah salah satu aplikasi internet yang terdiri dari perangkat lunak, kumpulan *protocol* dan seperangkat aturan yang memungkinkan kita untuk mengakses informasi di internet.

Dari kedua pengertian di atas, penulis menyimpulkan bahwa *web* adalah salah satu aplikasi internet yang berisikan informasi, iklan, serta program interaksi.

2.3.7 Aplikasi Asesmen Kondisi Saluran Udara Tegangan Tinggi dan Saluran Udara Tegangan Ekstra Tinggi pada PT. PLN (Persero) Penyaluran dan Pusat Pengatur Beban Sumatera Unit Pelayanan Transmisi Palembang Berbasis Website

Aplikasi Asesmen Kondisi Saluran Udara Tegangan Tinggi dan Saluran Udara Tegangan Ekstra Tinggi pada PT. PLN (Persero) Penyaluran dan Pusat Pengatur Beban Sumatera Unit Pelayanan Transmisi Palembang Berbasis *Website* adalah suatu aplikasi yang memudahkan dalam melakukan kegiatan asesmen atau penilaian keadaan tower dengan terkoneksi *internet* dalam lingkungan PT. PLN (Persero) Penyaluran dan Pusat Pengatur Beban (P3B) Sumatera Unit Pelayanan Transmisi Palembang dan lingkungan seluruh tower Unit Pelayanan Transmisi Palembang.

2.4 Teori Program

2.4.1 Pengertian Basis Data (Database)

Indrajani (2015:70), basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.

Prasetio (2015:199), *database* adalah sebuah struktur yang umumnya dikategorikan dalam dua hal, sebuah *database flat* dan sebuah *database rasional*.



Winarno *et.al* (2014:102), *database* merupakan sebuah tempat untuk menyimpan data yang jenisnya beraneka ragam.

Berdasarkan beberapa pengertian tersebut, penulis menyimpulkan bahwa *database* adalah sebuah struktur untuk menyimpan data yang jenisnya beraneka ragam dan didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.

2.4.1.1 Pengertian MySQL (*My Structure Query Language*)

Raharjo (2016:524), SQL (biasa dibaca *sequel*) adalah kependekan dari *Structured Query Language*. SQL merupakan bahasa yang dirancang untuk berkomunikasi dengan *database*. Tujuan dirancangnya SQL adalah untuk mengefisienkan dan menyederhanakan dalam membaca atau menulis data dari/ke dalam *database*.

Faisal dan Irnawati (2015:5), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 (enam) juta instalasi di seluruh dunia.

Winarno *et.al* (2014:102), MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan. Berikut ini adalah keuntungan MySQL:

1. Gratis dan *open source*.
2. Ada versi komersialnya juga, digunakan jika ingin memberikan dukungan teknis.
3. Biaya harus dikeluarkan jauh lebih murah dibandingkan merek lainnya.
4. Tersedia di banyak *platform*.
5. Menggunakan standar penulisan SQL ANSI.

2.4.2 Pengertian HTML (*Hypertext Markup Language*)

Faisal dan Irnawati (2015:1), HTML adalah sebuah bahasa markah yang digunakan untuk membuat halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* internet dan pemformatan hiperteks sederhana yang



ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi.

Raharjo (2016:449), HTML adalah singkatan dari *HyperText Markup Language*. HTML merupakan file teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser*.

2.4.2.1 Struktur Dasar HTML

Raharjo (2016:450) mengemukakan bahwa dokumen HTML terbagi menjadi dua bagian, yaitu bagian *header* dan *body*. Berikut struktur dasar dari HTML.

```
<html>      <!-- Tag untuk memulai dokumen HTML-->
<!-- Bagian header-->
<head>
    <title> JUDUL HALAMAN </title>
</head>
<!-- Bagian body-->
<body>
    Informasi yang akan ditampilkan pada web browser harus
    ditulis dalam bagian ini.
</body>
</html>      <!-- Tag untuk mengakhiri dokumen HTML-->
```

2.4.3 PHP (*Hypertext Preprocessor*)

Prasetio (2015:130), PHP (PHP: *Hypertext Preprocessor*) adalah bahasa script yang ditanam di sisi *server*.

Winarno *et.al* (2014:49), PHP atau PHP *Hypertext Preprocessor*, adalah sebuah bahasa pemrograman web berbasis *server (server-side)* yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client (browser)*.



2.4.3.1 Sintaks PHP

Prasetio (2015:139), menjelaskan beberapa aturan-aturan dasar penulisan sintaks PHP, yaitu:

1. *File* PHP harus disimpan dengan ekstensi **.php**.
2. Untuk membuat komentar yang tidak dieksekusi oleh server, adalah dengan menambahkan “//” di awal baris. Apabila membuat komentar yang panjang, dilakukan dengan mengapit komentar dengan “/*” dan “*/”.
3. Blok kode PHP diawali dengan tag **<?php**.
4. Blok kode PHP ditutup dengan tag **?>**.
5. Setiap instruksi program diakhiri dengan tanda **;**.
6. Tanda kurung digunakan untuk memanggil fungsi.

Winarno *et.al* (2014:62) menjelaskan bahwa kode PHP dimasukkan ke dalam kode HTML dengan cara menyelipkannya di dalam kode HTML. Untuk membedakan kode PHP dengan kode HTML, di depan kode PHP tersebut diberi *tag* pembuka dan di akhir kode PHP diberi *tag* penutup.

Raharjo (2016:40), PHP adalah bahasa pemrograman yang dirancang untuk mudah diletakkan di dalam kode HTML. Berikut merupakan contoh penggunaan kode PHP.

```
<html>
<head>
  <title> Kode PHP dalam Kode HTML </title>
</head>
<body>
  <p>Paragraf 1: Teks dari kode HTML</p>

  <!--Menyisipkan kode PHP dalam kode HTML-->
  <?php
    echo "<p>Paragraf 2: Teks dari kode PHP</p>";
  ?>
  <!--Akhir kode PHP -->
</body>
</html>
```



2.4.3.2 PHP dan *Form* HTML

Raharjo (2016:173), *form web* merupakan suatu antarmuka (*user interface*) agar *user* dapat berkomunikasi dengan aplikasi. Setiap permintaan yang diminta oleh *user* (disebut *request*) akan direspon oleh aplikasi, kemudian hasilnya akan dikembalikan lagi ke hadapan *user* melalui *web browser*.

Berikut dua atribut penting dalam form HTML.

1. Atribut *action*

Atribut *action* digunakan untuk menunjuk *file* PHP yang akan digunakan untuk memproses data yang dikirim melalui *form*.

2. Atribut *method*

Atribut *method* digunakan untuk menentukan bagaimana cara informasi dari suatu *form* (komputer klien) dikirimkan ke komputer *server*.

- a. *Method GET*

Nilai *GET* pada atribut *method* digunakan untuk mengirimkan data ke *server* dengan memerintahkan *web browser* untuk menambah nilai yang dimasukkan oleh *user* ke dalam URL.

Informasi yang akan ditambahkan ke URL memiliki format berupa **pasangan nama/nilai**, yang ditulis dalam bentuk **nama=nilai**. **nama** yang dimaksud di sini adalah nama dari elemen kontrol yang menyimpan data, sedangkan **nilai** adalah data yang dimasukkan oleh *user*.

- b. *Method POST*

Pada *method* pengiriman ini, data yang dimasukan oleh *user* tidak akan ditambahkan/ditampilkan ke dalam URL. Dengan kata lain, *method POST* secara transparan mengirimkan semua informasi yang sudah dikumpulkan halaman *form*.

2.4.4 CSS (*Cascading Style Sheet*)

Bekti (2015:47), CSS (*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman *web* yang digunakan untuk mempercantik halaman *web* dan mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam.



Prasetio (2015:285), CSS adalah suatu teknologi yang digunakan untuk memperindah halaman *website* (situs).

2.4.4.1 Penulisan CSS

Bekti (2015:49) mengemukakan aturan penulisan pada CSS (Deklarasi), yaitu:

```
selector { property : value; }
```

selector yaitu *tag* HTML yang akan didefinisikan, sedangkan *property* merupakan atribut yang akan diubah, dan *value* berupa nilai dari atribut.

2.4.5 Javascript

Prasetio (2015:332), *javascript* adalah bahasa pemrograman yang digunakan untuk membuat *web* lebih dinamis dan interaktif.

Winarno *et.al* (2014:49), *javascript* adalah bahasa *scripting client side* yang sangat populer.

Cara menggunakan *JavaScript* adalah dengan dimasukkan di antara kode HTML menggunakan tag `<script>` dan `</script>`. *JavaScript* bisa diletakkan di tag `<body>` ataupun tag `<head>` dari kode HTML.