



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Kadir (2017: 2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan pekerjaan yang dilakukan oleh manusia.”

Menurut Paramytha (2016: 4), “komputer berasal dari bahasa latin yaitu *Computer* yang berarti menghitung (*to compute* atau *to reckon*). Kata komputer pada awalnya dipergunakan untuk menggambarkan pekerjaan orang yang melakukan perhitungan aritmatika dengan atau tanpa alat bantu”.

Menurut Irwansyah dan Moniaga (2014:2), “Komputer adalah perangkat elektronik yang beroperasi di bawah kendali instruksi yang memiliki memori, dapat menerima dan mengolah data sesuai dengan aturan yang ditetapkan, menghasilkan hasil atau informasi dan menyimpan hasil tersebut untuk digunakan di masa depan atau di kemudian hari”.

2.1.2 Pengertian Basis Data (*Database*)

Menurut Fathansyah (2015:2),”Basis data adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.”

Menurut Pamungkas (2017:2), “Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu”.

Menurut Sukamto dan Shalahudin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”



2.1.3 Pengertian Perangkat Lunak

Menurut Rifauddin (2016:172), “Software merupakan program-program komputer yang berguna untuk menjalankan suatu pekerjaan sesuai dengan yang dikehendaki”.

Sukanto dan Shalahuddin (2018:02) menyatakan “Perangkat Lunak (Software) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (user manual).”

Menurut Gary dan Misty (2012:15) menyatakan bahwa, “peranti lunak adalah seperangkat perintah yang terkait, disusun untuk tujuan yang umum, memerintahkan kepada komputer apa yang harus dilakukan dan bagaimana melakukannya”.

2.1.4 Pengertian Sistem

Menurut Fat yang dikutip dalam Hutahaeen (2014:1), “sistem adalah suatu himpunan suatu “benda” nyata atau abstrak (a set of thing) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan (Unity) untuk mencapai tujuan tertentu secara efisien dan efektif”.

Menurut Indrajit yang dikutip dalam Hutahaeen (2014:1), mengemukakan bahwa “Sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya”.

Menurut Jogianto yang dikutip dalam Hutahaeen (2014:1), “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

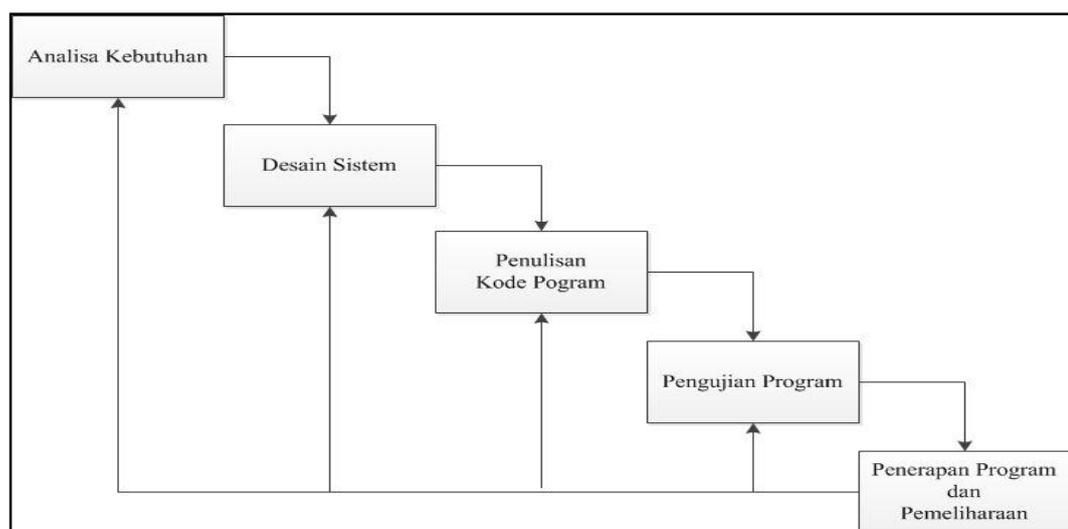
2.1.4.1 Elemen-elemen Sistem

Menurut Mcleod yang dikutip dalam Yakub (2012:3), tidak semua sistem memiliki elemen-elemen yang sama, tetapi susunan dasarnya sama. Ada beberapa elemen yang membentuk sebuah sistem yaitu:



- a. Tujuan, tujuan ini menjadi motivasi yang mengarahkan pada sistem, karena tanpa tujuan yang jelas sistem menjadi tak terarah dan tak terkendali.
- b. Masukan, masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud maupun tidak berwujud. Masukan berwujud adalah bahan mentah, sedangkan yang tidak berwujud adalah informasi.
- c. Proses, proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.
- d. Keluaran, keluaran (*output*) merupakan hasil dari pemrosesan sistem dan keluaran dapat menjadi masukan untuk subsistem lain.
- e. Batas, batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem. Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem.
- f. Mekanisme pengendalian dan umpan balik, mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), sedangkan umpan balik ini digunakan untuk mengendalikan masukan maupun proses. Tujuannya untuk mengatur agar sistem berjalan sesuai dengan tujuan.
- g. Lingkungan, lingkungan adalah segala sesuatu yang berada di luar sistem.

2.2.4.2 Metode Pengembangan Sistem



Gambar 2.1 Ilustrasi Model *Waterfall*



Sukanto dan Shalahudin (2018:28) 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 terurut mulai dari analisis, desain, pengkodean, pengujian dan pemeliharaan.

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean.

c. Pembuatan Kode Barang

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

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik 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. Dalam penelitian ini pengujian sistem akan menggunakan pengujian *Black-Box* Pengujian *Black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian metode ini memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

e. Pendukung (*support*) dan pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah mengirimkan ke *user*. Perubahan bisa terjadi adanya kesalahan



yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

2.1.5 Pengertian Aplikasi

Menurut Chan (2017:4), “Aplikasi adalah koleksi window dan objek-objek yang menyediakan fungsi untuk aktivitas user, seperti pemasukan data, proses, dan pelaporan”.

Menurut Hasugian yang dikutip dalam Sundawa (2018:7), “Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam bahasa Indonesia berarti pengolah”.

Menurut Indrajani (2018:3) menyatakan bahwa, “aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer”.

2.1.6 Pengertian Anggaran

Menurut Putra (2018:36), “Anggaran sendiri merupakan sebuah komitmen dari pihak manajemen terhadap pendapatan, biaya serta macam-macam transaksi keuangan dalam jangka waktu dimasa yang akan datang”.

Menurut Sodikin (2015:188), “Anggaran adalah rencana kegiatan perusahaan secara menyeluruh yang akan dilakukannya di tahun mendatang dan dinyatakan dalam satuan uang”.

Menurut Sujarweni (2016:120), “Pengertian anggaran merupakan suatu rencana kerja perusahaan yang terdiri dari pendapatan dan pengeluaran yang dinyatakan secara kuantitatif, diukur dalam satuan moneter dan dapat juga menggunakan satuan ukuran lain, yang mencakup jangka waktu tertentu, biasanya jangka waktu 1 tahun”.

2.1.7 Pengertian Pembayaran

Menurut Mulyadi yang dikutip dalam Ashari (2014:66), “pembayaran merupakan unsur penting bagi sebuah instansi maka di perlukan suatu sistem yang dapat mengelola sistem pembayaran secara baik.



Menurut Mukromah (2017:15), “Pembayaran adalah perpindahan hak atas nilai antara pihak pembeli dan pihak penjual yang secara bersamaan terjadi perpindahan hak atas barang atau jasa secara berlawanan”.

Menurut Rafika (2017:66), “Pembayaran adalah suatu tindakan menukarkan sesuatu (uang/barang) dengan maksud dan tujuan yang sama yang dilakukan oleh dua orang atau lebih”.

2.1.8 Pengertian Tunjangan

Menurut Firmandari (2014:3), “Tunjangan adalah unsur-unsur balas jasa yang diberikan dalam nilai rupiah secara langsung kepada karyawan individual dan dapat diketahui secara pasti.”.

Menurut Irma yang dikutip dalam Maulana (2016:19), “Tunjangan adalah program peningkatan kesejahteraan masyarakat yang pemberiannya tidak berdasarkan kinerja pegawai, akan tetapi berdasarkan keanggotannya sebagai bagian dari organisasi, serta pegawai yang memiliki banyak kebutuhan agar dapat menjalankan kehidupannya secara normal dan agar dapat bekerja dengan baik”.

Menurut Kadarisman yang dikutip dalam Firmandari (2014:27), “Tunjangan adalah tambahan penghasilan yang diberikan oleh perusahaan kepada para pegawainya”.

2.1.9 Pengertian Karyawan

Menurut Fahmi (2018:7), “Karyawan merupakan bagian penting, penentu, dan pencipta manajemen kinerja yang diharapkan”.

Menurut Safitri (2017:17), “Karyawan merupakan salah satu sumber daya yang di gunakan sebagai alat penggerak dalam memajukan suatu perusahaan. Kinerja karyawan cukup berpengaruh dalam keuntungan yang didapat oleh perusahaan tersebut”.

Menurut Safitri (2017:18), “Karyawan adalah tenaga kerja yang melakukan pekerjaan dan memberikan hasilkerjanya kepada pengusaha yang mengerjakan dimana hasil karyanya itu sesuai dengan profesi atau pekerjaan atas dasar keahlian sebagai matapencariannya”.



2.1.10 Pengertian AirNav Indonesia

Perum LPPNPI (Lembaga Penyelenggara Penyedia Navigasi Penerbangan Indonesia) atau yang dikenal dengan AirNav Indonesia adalah BUMN Indonesia yang bergerak di bidang usaha pelayanan navigasi udara. AirNav didirikan pada 13 September 2012 melalui PP No 77 tahun 2012. AirNav Indonesia terbagi menjadi dua ruang udara berdasarkan Flight Information Region (FIR) yaitu, FIR Jakarta dan FIR Ujung Pandang.

2.1.11 Pengertian Aplikasi Pengajuan Persetujuan Anggaran dan Pembayaran Tunjangan Karyawan pada AirNav Indonesia Cabang Palembang

Pengertian Aplikasi Pengajuan Persetujuan Anggaran dan Pembayaran Tunjangan Karyawan pada AirNav Indonesia Cabang Palembang adalah aplikasi yang dibuat untuk membantu Unit Keuangan dalam mengolah data pengajuan persetujuan anggaran beserta pembayaran tunjangan karyawan di AirNav Indonesia Cabang Palembang.

2.2 Teori Khusus

2.2.1 Data Flow Diagram (DFD)

2.2.1.1 Pengertian Data Flow Diagram (DFD)

Menurut Rosa dan Shalahuddin (2018:70) menyatakan bahwa, “*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

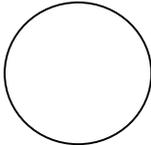
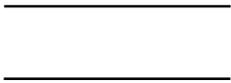
2.2.1.2 Simbol-simbol pada Data Flow Diagram (DFD)

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau



fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi.

Tabel 2.1 Simbol-Simbol Data Flow Diagram

No.	Notasi	Keterangan
1.		<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>
2.		<p><i>File</i> basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harusnya 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>
3.		<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>



Lanjutan Tabel 2.1 Simbol-Simbol Data Flow Diagram

No.	Notasi	Keterangan
4.		Aliran data: merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>).

(Sumber : Rosa dan Shalahuddin, 2018:71-72)

2.2.1.3 Tahapan-tahapan Perancangan *Data Flow Diagram* (DFD)

Rosa dan Shalahuddin (2018:72-73) menjelaskan, berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

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
DFD 2 merupakan *breakdown* dari Data Flow Diagram Level 1. Jumlah Data Flow Diagram Level 2 sama dengan modul pada Data Flow Diagram Level 1 dalam suatu sistem.
4. Membuat DFD Level 3 dan seterusnya
DFD Level 3 dan seterusnya merupakan *breakdown* dari data flow diagram sebelumnya, namun jarang digunakan karena dapat menyebabkan sistem yang dikembangkan menjadi rumit.



2.2.2 Entity Relationship Diagram (ERD)

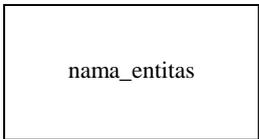
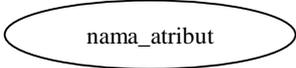
2.2.2.1 Pengertian Entity Relationship Diagram (ERD)

Rosa dan Shalahuddin (2018:50) menjelaskan, “Entity Relationship Diagram (ERD) digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain.”

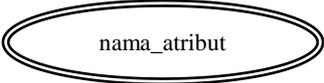
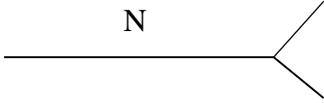
2.2.2.2 Simbol-simbol pada Entity Relationship Diagram (ERD)

Adapun simbol-simbol yang digunakan pada ERD dengan notasi Chen adalah sebagai berikut:

Tabel 2.2 Simbol-simbol pada Entity Relationship Diagram

No.	Simbol	Deskripsi
1.	Entitas/ <i>entity</i> 	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.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

Lanjutan Tabel 2.2 Simbol-simbol pada *Entity Relationship Diagram*

No.	Simbol	Deskripsi
3.	<p>Atribut Kunci Primer</p> 	<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 id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).</p>
4.	<p>Atribut Multinilai/ <i>Multivalued</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
5.	<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>
6.	<p>Asosiasi/<i>Association</i></p> 	<p>Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.</p>

(Sumber: Rosa dan Shalahuddin , 2018:50-51)



2.2.3 Flowchart

2.2.3.1 Pengertian Flowchart

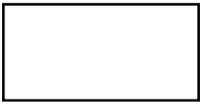
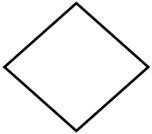
Menurut Indrajani (2015:36), “*flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah, khususnya yang perlu di pelajari dan dievaluasi lebih lanjut”.

Menurut Sitorus (2015:14), “*flowchart* merupakan sebuah alat bantu yang berbentuk diagram alir di gunakan untuk menggambarkan sebuah algoritma yang terstruktur yang mudah dipahami oleh orang lain”.

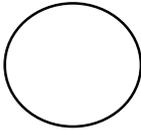
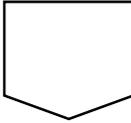
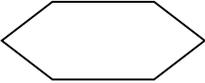
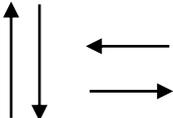
2.2.3.2 Simbol-simbol pada Flowchart

Sitorus (2015:14-15), menjelaskan tentang simbol-simbol yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol-simbol yang bersangkutan dapat dilihat pada tabel berikut ini:

Tabel 2.3 Simbol-simbol pada *Flowchart*

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program
2.		<i>Input / Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
2.		<i>Proses</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh computer
4.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya / tidak.

Lanjutan Tabel 2.3 Simbol-simbol pada *Flowchart*

No	Simbol	Nama	Keterangan
5.		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6.		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
7.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8.		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9.		<i>Punch Tape</i>	Menunjukkan input/output menggunakan pita kertas berlubang.
10.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer)
11.		<i>Flow</i>	Menyatakan jalannya arus suatu proses

(Sumber: Siturous, (2015:14-15))

2.2.4 Kamus Data (*Data Dictionary*)

Menurut Rosa dan Shalahuddin (2018:73) menyatakan bahwa, “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, merupakan nama dari data.
2. Digunakan pada, merupakan proses-proses yang terkait data.
3. Deskripsi, merupakan deskripsi data.
4. Informasi tambahan, merupakan tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2.4 Simbol-simbol pada Kamus Data

No.	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	Dan
3	[]	baik...atau...
4	{ } ⁿ	n kali diulang/bernilai banyak
5	()	data opsional
6	*..*	batas komentar

(Sumber: Rosa dan Shalahuddin, 2018:74)

2.2.5 Pengujian *Black Box Testing*

Menurut Rizky yang dikutip dalam Hanifah (2016:34), “Black Box Testing sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian black box ini menitik beratkan pada fungsi system”.

Berikut adalah 10 tipe pengujian dari metode Black Box menurut Julian yang dikutip dalam Hanifah (2016:34):



1. *Equivalence Partitioning*, Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing*, Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari *Equivalence Partitioning*.
3. *Comparison Testing*, Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing*, Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing*, Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
6. *Behavior Testing*, Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.
7. *Performance Testing*, Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi.
8. *Requirement Testing*, Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
9. *Endurance Testing*, Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.

Cause – Effect Relationship Testing, Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja

2.3 Teori Program

2.3.1 Pengertian PHP

Menurut Abdulloh (2015:3) menyatakan bahwa, “PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses disisi server”.



Menurut Sidik (2017:4) menyatakan bahwa, “PHP kependekan dari *PHP:HyperText Preprocessor* (rekursif, mengikuti gaya penamaan di *nix, merupakan bahasa utama script *server-side* yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop”.

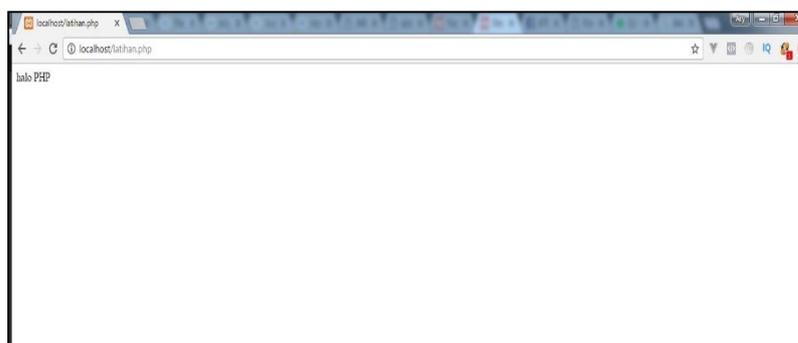
Menurut Zaki et al. (2014:49), “*PHP Hypertext Preprocessor* adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode *PHP* dari kode *web* dengan ekstensi *.PHP*, sehingga menghasilkan tampilan *website* yang dinamis disisi client (*browser*)”.

2.3.1.1 Script PHP

PHP yang merupakan sebuah bahasa scripting yang terpasang pada HTML, dan skrip HTML (*Hyper Text Markup Language*) adalah bahasa standar untuk membuat halaman *web*. Biasa *file* dituliskan dengan ekstensi *.htm* atau *.html*.

```
<html>
  <head>
    <title>Latihan PHP</title>
  </head>
  <body>
    halo PHP
  </body>
</html>
```

Bila dijalankan melalui *browser*, kode diatas akan menampilkan hasil seperti berikut:



Gambar 2.2 Tampilan program HTML pada *browser*



2.3.1.2 Teknik Penulisan Script *PHP*

Zaki et al. (2014:24), block kode *PHP* dimulai dengan sintak tag `<?PHP` dan di akhiri dengan `?>`, berikut adalah contoh blok kode *PHP*:

```
<?PHP
//kode tertentu diisikan di sini
?>
```

Selain itu ada beberapa tag lain yang bisa di pakai, seperti:

Tabel 2.5 Script Dasar *PHP*

Jenis Tag	Tag Pembuka	Tag Penutup
Tag standar	<code><?PHP</code>	<code>?></code>
Tag pendek	<code><?</code>	<code>?></code>
Tag ASP	<code><%</code>	<code>%></code>
Tag script	<code><script language="PHP"></code>	<code></script></code>

(Sumber: Zaki et al. 2014:24-25)

Echo sering digunakan di dalam *PHP*. Fungsinya untuk meng-ouput data. Berikut ini contoh penggunaan echo:

```
<?PHP
Echo '<p> selamat datang </p>';
?>
```

Di dalam *script PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah *script*. Beberapa cara untuk memberi keterangan di dalam *script PHP* adalah sebagai berikut:

- 1) Gunakan tag `/*` dan akhiri tag `*/` apabila jumlah keterangan lebih dari 1 baris.
- 2) Gunakan tag `//`, tag ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.



3) Gunakan *tag #*, *tag* ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah *script PHP* selalu diakhiri dengan menuliskan tanda titik koma (;). Satu baris *script PHP* tidak harus berada dalam satu baris, melainkan dapat menuliskan perintah *script PHP* lebih dari satu baris.

2.3.1.3 Tipe Data *PHP*

Tipe data merupakan jenis dari suatu data yang akan diproses oleh bahasa pemrograman. Yosef Murya (2014:26), menjelaskan beberapa tipe data dalam *PHP*, sebagai berikut :

1. **Integer** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. Range bilangan integer adalah antara -2.147.483.647 sampai dengan 2.147.483.647.
2. **Double Floating** adalah tipe data yang berguna untuk menyimpan bilangan desimal. Range bilangan floating point antara 1e308 sampai dengan 1e308.
3. **Boolean** adalah tipe data yang paling sederhana, hanya berupa **TRUE** dan **FALSE**.
4. **String** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan string harus diapit dengan tanda petik, baik berupa petik tunggal ('...') maupun petik ganda ("...").
5. **Objek** adalah tipe data dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data ini bias berupa bilangan.
6. **Array** merupakan **Tipe Compound Primitif**, terdapat pada bahasa pemrograman lain.
7. **Null** adalah tipe data yang tidak memuat apapun. Setiap variable yang diset menjadi tipe data Null, ini akan menjadikan variabel tersebut kosong.
8. **Resources** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.



2.3.2 Pengertian MySQL

Menurut Kristanto (2015:30) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia”.

Menurut Rosa dan Shalahuddin (2018:46) menyatakan bahwa, “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

Menurut Hendry (2015:7) menyatakan bahwa, “MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*)”.

2.3.2.1 Fungsi-Fungsi MySQL

Sejumlah fungsi yang berawalan **mysql_** yang digunakan untuk mengakses *database server MySQL* adalah sebagai berikut:

Tabel 2.6 Fungsi-fungsi MySQL

No.	Fungsi MySQL	Penggunaan
1	<i>Mysql_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu host
2	<i>Mysql_close()</i>	Menutup hubungan ke <i>database MySQL</i>
3	<i>Mysql_select_db()</i>	Memilih <i>database</i>
4	<i>Mysql_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel
5	<i>Mysql_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i>
6	<i>Mysql_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan yang menggunakan <i>SELECT</i>



Lanjutan Tabel 2.6 Fungsi-fungsi MySQL

No.	Fungsi MySQL	Penggunaan
7	<i>Mysql_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi <i>INSERT, DELETE, UPDATE</i>
8	<i>Mysql_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan
9	<i>Mysql_fetch_row()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil
10	<i>Mysql_fetch_array()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasilnya akan disimpan dua kali pada <i>array</i> hasil
11	<i>Mysql_fetch_field()</i>	Menghasilkan informasi suatu kolom
12	<i>Mysql_data_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu
13	<i>Mysql_field_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu
14	<i>Mysql_create_db()</i>	Membuat <i>database MySQL</i>
15	<i>Mysql_drop_db()</i>	Menghapus <i>database MySQL</i>
16	<i>Mysql_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i>
17	<i>Mysql_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i>
18	<i>Mysql_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i>

(Sumber: Abdul Kadir (2008:360-379))



2.3.2.2. Alasan Menggunakan MySQL

Berikut ini beberapa alasan mengapa mereka memilih *MySQL* sebagai *server database* :

1. **Fleksibel.** *MySQL* dapat digunakan untuk mengembangkan aplikasi *desktop* maupun aplikasi *web* dengan menggunakan teknologi yang bervariasi. Ini berarti bahwa *MySQL* memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi, apakah itu *PHP*, *JSP*, *Java*, *Delphi*, *C++*, maupun yang lainnya dengan cara menyediakan *plug-in* dan *driver* yang spesifik untuk masing-masing teknologi tersebut.
2. **Performa Tinggi.** *MySQL* memiliki *main query* performa tinggi dengan demikian proses transaksional dapat dilakukan dengan sangat cepat.
3. **Lintas Platform.** *MySQL* dapat digunakan pada *platform* atau lingkungan sistem operasi yang beragam.
4. **Gratis.** *MySQL* dapat digunakan secara gratis.
5. **Proteksi Data yang Handal.** Perlindungan terhadap keamanan data merupakan hal nomor satu yang dilakukan oleh para profesional di bidang *database*. *MySQL* menyediakan mekanisme yang powerful untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.
6. **Komunitas Luas.** Karena penggunaannya banyak maka *MySQL* memiliki komunitas yang luas. Hal ini berguna jika kita menemui suatu permasalahan dalam proses pengolahan data menggunakan *MySQL*.

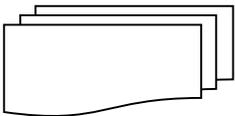
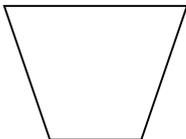
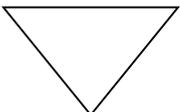
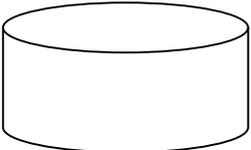
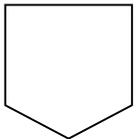
2.3.3. Block Chart

Kristanto (2018:75) menjelaskan, “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol–simbol tertentu”. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem.

Adapun simbol–simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

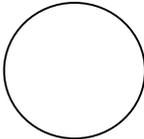
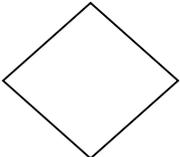
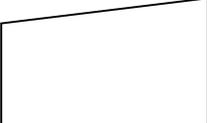


Tabel 2.7 Simbol-simbol pada *Block chart*

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



Lanjutan Tabel 2.7 Simbol-simbol pada *Block chart*

No.	Simbol	Arti
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.		Pemasukkan data secara manual.

(Sumber : Kristanto, 2018:75-77)