



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1. Sistem

Sistem adalah kumpulan dari obyek-obyek seperti orang, *resources*, konsep dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Kemudian sistem juga merupakan kumpulan dari komponen yang berinteraksi bersama-sama secara kolektif untuk melaksanakan tugas (Arif, 2017:41).

Sistem adalah sekumpulan elemen yang saling berhubungan dan berinteraksi untuk mencapai suatu tujuan (Arif, 2017:42).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk mencapai sasaran tertentu (Hutahaean, 2015:2).

Berdasarkan pendapat di atas, dapat disimpulkan bahwa sistem merupakan kumpulan dari beberapa elemen yang saling berhubungan untuk mencapai suatu tujuan tertentu.

2.1.2. Prioritas

Menurut kamus besar Bahasa Indonesia, “Prioritas adalah suatu hal yang didahulukan dan diutamakan daripada yang lain.”(www.kbbi.web.id)

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian Prioritas adalah suatu hal yang didahulukan, tersusun dari hal yang paling penting sampai dengan hal yang bersifat bisa ditunda pemenuhannya atas pertimbangan-pertimbangan dari banyak hal.

2.1.3. Jalan

Simonds (dalam Jurnal Ruliyansyah, 2017) menyatakan bahwa jalan merupakan satu kesatuan yang harus lengkap, aman, efisien, menarik, memiliki



sirkulasi, dan interaksi yang baik serta mampu memberikan pengalaman yang menarik bagi pengguna jalan.

Pengertian jalan di dalam Undang-Undang Republik Indonesia No. 38 Tahun 2004 adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang digunakan untuk lalu lintas, yang berada pada permukaan tanah, di bawah permukaan tanah, dan atau di bawah permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel.

Menurut Pasal 8 Undang Undang Republik Indonesia No. 38 Tahun 2004, jalan menurut fungsinya dikelompokkan ke dalam jalan arteri, jalan kolektor, jalan lokal, dan jalan lingkungan, dengan perincian sebagai berikut:

1. Jalan arteri merupakan jalan umum yang berfungsi melayani angkutan utama dengan ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara berdaya guna.
2. Jalan kolektor merupakan jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi.
3. Jalan lokal merupakan jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.
4. Jalan lingkungan merupakan jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan jarak dekat dan kecepatan rata-rata rendah.

2.2. Teori Yang Berhubungan dengan Penelitian

2.2.1. Keputusan

Keputusan merupakan hasil pemikiran berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapi (Pratiwi, 2016:2).

Jenis-jenis keputusan dibedakan menjadi tiga macam yaitu keputusan terstruktur, keputusan tidak terstruktur, dan keputusan semi terstruktur (Pratiwi, 2016:5-6).



1. Keputusan terstruktur

Keputusan-keputusan yang berkaitan dengan persoalan yang telah diketahui sebelumnya. Proses pengambilan keputusan seperti ini biasanya didasarkan atas teknik-teknik tertentu dan sudah dibuat standarnya. Kategori keputusan ini juga dapat dikatakan suatu proses jawaban secara otomatis pada kebijakan yang sudah ditentukan sebelumnya. Secara alamiah hampir semua masalah rutin dan berulang memiliki parameter-parameter persoalan yang telah diketahui dan terdefinisi dengan baik, sehingga jawaban atau proses pengambilan keputusan pun bersifat rutin dan terjadwal.

2. Keputusan tak terstruktur

Keputusan-keputusan yang berkaitan dengan berbagai persoalan baru. Keputusan tidak terstruktur biasanya juga berkaitan dengan persoalan yang cukup pelik, karena banyak parameter yang tidak diketahui atau belum diketahui. Oleh karena itu, untuk mengambil keputusan ini biasanya intuisi serta pengalaman seorang pelaku organisasi akan sangat membantu.

Keputusan tak terstruktur, adalah “*fuzzy*”, permasalahan kompleks dimana tak ada solusi yang mengikutinya. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur. Keputusan tidak terstruktur (*unstructured decision*) bukan merupakan keputusan yang berulang dan rutin. Contohnya adalah memilih sampul depan sebuah majalah, mengontrak manajemen tingkat senior, dan memilih proyek penelitian awal yang akan dilakukan. Tidak ada kerangka atau model yang dapat memecahkan masalah sejenis ini. Bahkan, dibutuhkan banyak sekali pertimbangan dan intuisi. Walaupun demikian, keputusan tidak terstruktur dapat didukung oleh bantuan dari keputusan yang diambil berdasar hasil komputer, yang berfungsi untuk memfasilitasi pengumpulan informasi dari berbagai sumber. Contohnya adalah keputusan untuk pengembangan teknologi baru, pengembangan jenis usaha baru, keputusan untuk bergabung dengan perusahaan lain, perekrutan eksekutif.

3. Keputusan semi terstruktur

Terdapat beberapa keputusan terstruktur, tetapi tak semua dari fase-fase yang ada. Keputusan semi terstruktur (*semistructured decision*) ditandai dengan



dengan peraturan-peraturan yang tidak lengkap untuk mengambil keputusan, dan adanya kebutuhan untuk membuat penilaian serta pertimbangan subjektif sebagai pelengkap analisis data yang formal. Menetapkan anggaran pemasaran untuk suatu produk baru adalah contoh dari keputusan semi terstruktur. Walaupun keputusan seperti ini biasanya tidak dapat secara penuh diotomatisasikan, namun sering didukung dari komputer (*computer-based decision*). Contoh keputusan jenis adalah investasi keuangan, pengevaluasian kredit, penjadwalan produksi, pemberian dana rehabilitasi sekolah, dan pengendalian persediaan.

Berdasarkan pendapat di atas, dapat disimpulkan bahwa keputusan merupakan hasil dari pemikiran seseorang atau kelompok yang dapat digunakan untuk memecahkan suatu masalah.

2.2.2. Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision System. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditunjukkan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Dewanto, 2015).

Sistem Pendukung Keputusan adalah sistem penghasil informasi spesifik yang ditujukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manajer pada berbagai tingkatan. Sistem pendukung keputusan juga merupakan suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model (Pratiwi, 2016:4).

Susanto (2018), "Sistem pendukung keputusan atau *Decission Support System* (DSS) adalah sistem penghasil informasi yang ditujukan pada suatu masalah tertentu yang harus dipecahkan oleh manager dan dapat membantu manager dalam pengambilan keputusan."



Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian Sistem Pendukung Keputusan merupakan suatu sistem keputusan yang dapat digunakan sebagai pendukung suatu organisasi atau perusahaan dalam melakukan pengambilan keputusan agar lebih terarah dan mendapatkan hasil yang terbaik.

2.2.3. Metode *Additive Rasio Assessment* (ARAS)

Metode ARAS, sebuah utilitas nilai fungsi yang menentukan efisiensi relatif kompleks dari alternatif yang layak adalah langsung sebanding dengan efek relatif dari nilai dan bobot kriteria utama yang dipertimbangkan dalam proyek-proyek. Dalam melakukan proses perankingan, metode ARAS memiliki lima tahapan yang harus dilakukan untuk menghitung metode aras yaitu: (Edmundas Kazimieras Zavadskas, 2010).

1. Pembentukan *Decision Making Matriks*

$$x = \begin{bmatrix} x_{0i} & x_{0j} & \dots & x_{0n} \\ x_{i1} & x_{ij} & \dots & x_{in} \\ \vdots & \vdots & \ddots & \vdots \\ x_{ni} & x_{mj} & \dots & x_{mn} \end{bmatrix} \quad (i = 0, m; \dots j = 1, n)$$

Dimana :

m = Jumlah Alternatif

n = Jumlah Kriteria

x_{ij} = Nilai performa dari alternatif ;

terhadap kriteria J_{x0j} = nilai optimum dari kriteria J

Jika nilai optimum kriteria J (x_{0j}) tidak diketahui, maka:

$$x_{0j} = \text{Max} \frac{\text{Max}}{i} = x_{ij} \cdot \text{if} \frac{\text{Max}}{i} \cdot x_{ij} \text{ is Preference}$$

$$x_{0j} = \text{Max} \frac{\text{Min}}{i} = x_{ij} \cdot \text{if} \frac{\text{Min}}{i} \cdot x_{ij} \text{ is Prefeerable}$$

2. Penormalisasian matriks keputusan untuk semua kriteria

a. Jika kriteria *beneficial* (*max*) maka dilakukan normalisasi mengikuti:



$$x_{ij}^* = \frac{x_{ij}}{\sum_{i=0}^m x_{ij}}$$

Dimana :

x_{ij}^* adalah nilai normalisasi

- b. Jika kriteria *non beneficial* maka dilakukan normalisasi:

$$\text{Tahap 1} = x_{ij} = \frac{1}{x_{ij}}$$

$$\text{Tahap 2} = R = \frac{x_{ij}}{\sum_{i=0}^m x_{ij}}$$

3. Menentukan bobot matriks yang sudah dinormalisasikan

$$\sum_{j=1}^n W_j = 1$$

Dimana :

W_j = bobot kriteria

4. Menentukan nilai fungsi optimalisasi (S_i)

$$S_i = \sum_{i=1}^n d_{ij} : (i = 1, 2, \dots, m : j = 1, 2, \dots, n)$$

Dimana S_i adalah nilai fungsi optimalisasi alternatif i . Nilai terbesar adalah nilai yang terbaik, dan nilai yang paling sedikit adalah yang terburuk. Dengan memperhitungkan proses, hubungan proporsional dengan nilai dan bobot kriteria yang diteliti berpengaruh pada hasil akhir.

5. Menentukan tingkat peringkat tertinggi dari alternatif

$$K_i = \frac{S_i}{S_0}$$

Dimana S_i dan S_0 merupakan nilai kriteria optimalitas, diperoleh dari persamaan sudah jelas. Itu dihitung nilai U_i berada pada interval dan merupakan pesanan yang diinginkan didahulu efisiensi relatif kompleks dari alternatif yang layak bias ditemukan sesuai dengan nilai fungsi utilitas.



2.3. Teori Khusus

2.3.1. Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan obyek yang berisi data dan operasi yang diberlakukan terhadapnya (Sukamto dan Shalahuddin, 2016:99).

Beberapa karakter yang menjadi ciri-ciri pendekatan berorientasi objek adalah sebagai berikut (Nugroho, 2019) :

1. Pendekatan lebih pada data dan bukannya pada prosedur atau fungsi
2. Program besar dibagi pada apa yang dinamakan objek-objek
3. Struktur data dirancang dan menjadi karakteristik dari objek-objek.
4. Fungsi- fungsi yang mengoperasikan data tergabung dalam suatu objek yang sama
5. Data tersembunyi dan terlindung dari fungsi atau prosedur yang ada diluar.
6. Objek-objek dapat saling berkomunikasi dengan saling mengirim *message* satu sama lain.
7. Pendekatan adalah dari bawah ke atas.

2.3.2. Metode Pengembangan Sistem

Metode pengembangan yang penulis pakai adalah metodologi *Rapid Application Development* (RAD). *Rapid Application Development* (RAD) adalah salah satu metodologi yang pertama kali muncul untuk menjawab masalah kelambatan penyelesaian pengembangan sistem informasi pada *Systems Development Life Cycle* (SDLC) konvensional.

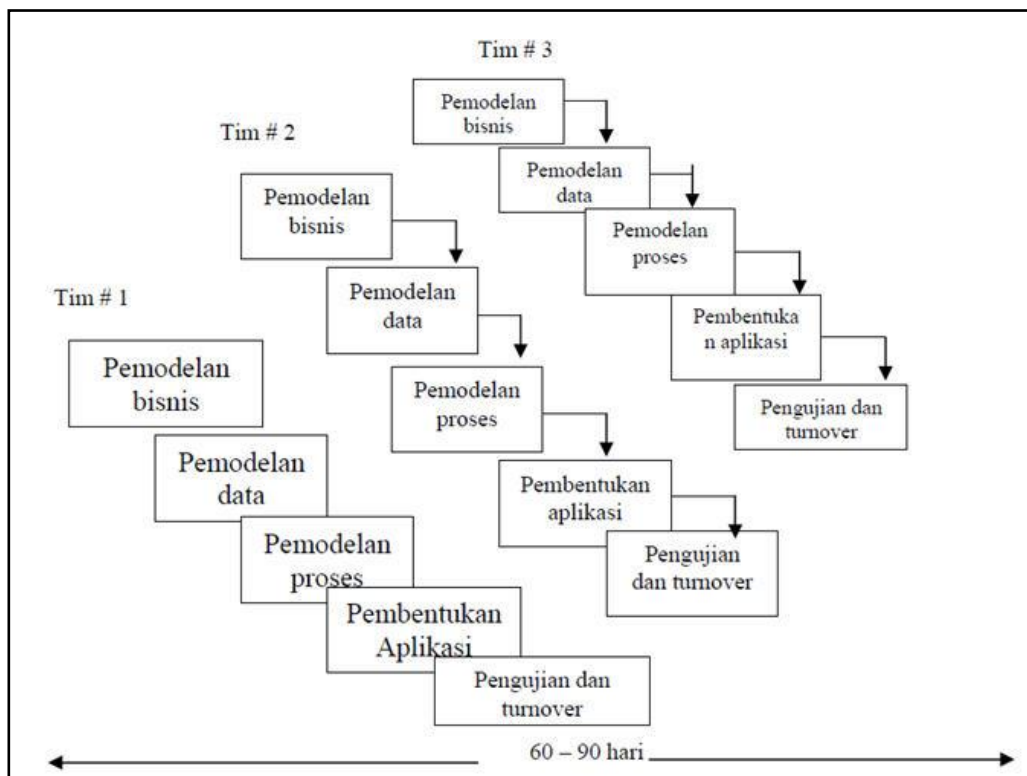
Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak (Sukamto dan Shalahuddin, 2018:73)

Dari defenisi di atas, dapat penulis simpulkan bahwa metode *Rapid Application Development* (RAD) merupakan suatu metode pengembangan sistem



yang dapat mempersingkat waktu pengerjaan dari sebuah sistem dengan tahapan-tahapannya yang sangat terstruktur.

Metode RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel. Berikut adalah gambar dari metode RAD (Sukamto dan Shalahuddin 2018:34) :



Gambar 2.1. Ilustrasi model RAD

- **Pemodelan Bisnis**
Pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa saja yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkait informasi itu.
- **Pemodelan Data**
Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.



- **Pemodelan Proses**
Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.
- **Pembuatan Aplikasi**
Mengimplementasikan pemodelan proses dan data menjadi program. Model RAD sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.
- **Pengujian dan pergantian**
Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat beranjak untuk mengembangkan komponen berikutnya.

2.3.3. Jenis-jenis Diagram

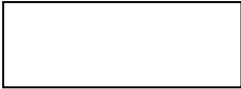
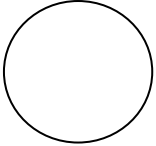
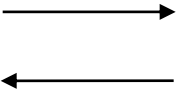
2.3.3.1. Data Flow Diagram (DFD)

Menurut Sukanto dan Shahaluddin (2018:244) menyatakan, “DFD sering juga disebut DAD (Diagram Aliran Data) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”, dan Kristanto (2018:55) menyatakan, “DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem”.



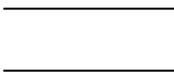
Notasi – notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :

Tabel 2.1. Simbol – simbol *Data Flow Diagram* (DFD)

No	Simbol	Keterangan
1.		<p>Entitas luar atau masukan atau keluaran atau orang yang akan memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Note: nama yang digunakan pada masukan atau keluaran biasanya berupa kata benda.</p>
2.		<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 didalam kode program.</p> <p>Note: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
3.		<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses atau dari proses ke masukan atau keluaran.</p> <p>Note: nama yang digunakan biasanya berupa kata benda, dapat diawali dengan kata data atau tanpa kata data.</p>



Lanjutan Tabel 2.1. Simbol – simbol *Data Flow Diagram* (DFD)

4.		<p><i>File</i> atau basis data 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 harus sesuai dengan perancangan tabel-tabel pada basis data (ERD).</p> <p>Note: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
----	---	---


Sumber : Rosa S dan Shalahuddin (2018:71)

2.3.3.2. Block Chart

Kristanto (2018:68) mengemukakan bahwa, “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”.

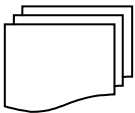


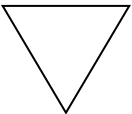

Adapun simbol-simbol *block chart* yang sering digunakan adalah sebagai berikut:

Tabel 2.2. Simbol-Simbol *Block Chart*


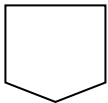
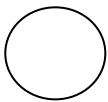
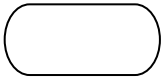
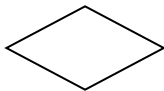

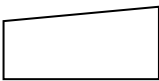
No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan



Lanjutan Tabel 2.2. Simbol-Simbol *Block Chart*

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>)

Lanjutan Tabel 2.2. Simbol-Simbol *Block Chart*

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 (2018:68-70)

2.3.3.3. Flowchart

Rosa S (2018:843) menyatakan, “*Flowchart* adalah sebuah bagan aliran dari sesuatu, dan sesuatu itu dapat juga berupa aliran proses. *Flowchart* juga sebagai sebuah perangkat (tool) untuk membantu membuat rancangan algoritma, karena merupakan perangkat maka yang menggunakannya adalah yang merasa

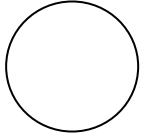
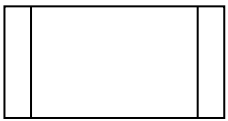
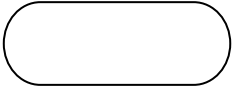
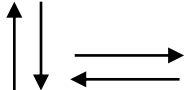



sesuai dengan perangkat ini, tidak harus digunakan untuk semua karena ada banyak perangkat untuk membantu cara berpikir manusia”.

Tabel 2.3. Tabel Simbol-simbol *Flowchart*



No.	Simbol	Nama	Keterangan
1.		Proses	Proses yang dilakukan secara internal di dalam computer atau memori.
2.		Data	Digunakan untuk beberapa operasi masukan/keluaran (<i>input/output(I/O)</i>) dengan berbagai tipe data dimaksudkan bahwa computer memperoleh masukan atau menghasilkan keluaran.
3.		Keputusan(<i>decision</i>)	Digunakan untuk pemilihan dalam bentuk dua jawaban seperti iya/tidak, benar/salah.
4.		Komentar	Digunakan untuk menuliskan komentarpada diagram <i>flowchart</i> .
5.		Inisialisasi (<i>preparation</i>)	Digunakan untuk menggambarkan proses inisialisasi untuk blok pengulangan (<i>for</i>).

Lanjutan Tabel 2.3. Tabel Simbol-simbol *Flowchart*

6.		Konektor/penghubung	Mengizinkan <i>flowchart</i> digambar tanpa irisan garis atau tanpa aliran balik, atau bisa juga untuk menyambungkan dua buah garis.
7.		Proses yang telah didefinisikan sebelumnya (<i>predefined process</i>)	Digunakan untuk memanggil sebuah rutin program atau bagian dari rutin program (<i>subroutines</i>) (fungsi atau prosedur selain yang sedang dirancang), proses, atau program yang menginterupsi (program lain dari yang dirancang).
8.		Pemberhentian (<i>terminal</i>)	Digunakan untuk memulai atau mengakhiri sebuah program, proses, atau program yang menginterupsi.
9.		Garis aliran (<i>flow line</i>)	Digunakan untuk menunjukkan arah aliran.
10.		Menampilkan sesuatu ke layar (<i>display</i>)	Digunakan jika ada yang ditampilkan ke layar.



Lanjutan Tabel 2.3. Tabel Simbol-simbol *Flowchart*

11.		Masukan manual (<i>manual input</i>)	Digunakan jika ada masukan manual dari <i>user</i> .
12.		Operasi manual (<i>manual operasi</i>)	Biasanya digunakan jika ada blok pengulangan yang diberhentikan secara manual dari masukan <i>user</i> .

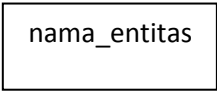
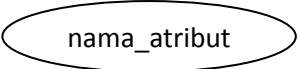
(Sumber: Rosa A.S, 2018:844)

2.3.3.4. Entity Relational Diagram (ERD)

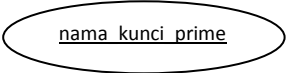
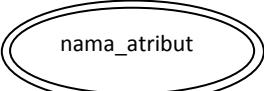
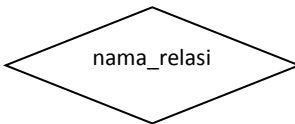
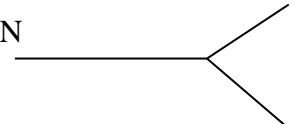
Sukamto dan Shalahuddin. (2018:289) menyatakan, “*Entity Relational Diagram* (ERD) adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional.”

Sukamto dan Shalahuddin (2018:50) menyatakan, beberapa simbol-simbol pada *Entity Relational Diagram* (EDR) yaitu sebagai berikut:

Tabel 2.4. Tabel Simbol-simbol *Entity Relational Diagram* (ERD)

No.	Simbol	Keterangan
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.

**Tabel 2.4** Tabel Simbol-simbol *Entity Relational Diagram* (ERD)

3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record 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).
4.	Atribut multivalued/multivalued 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / association 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.

(*Simbol: Sukamto, Shalahuddin, 2018:50*)

2.3.3.5. Kamus Data

Sukamto dan Shalahuddin (2018:73) menyatakan, “Kamus data adalah daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan



(*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Pada umumnya kamus data berisi:

- a. Nama-nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data.

Sukamto dan Shalahuddin (2018:74) menyatakan, simbol-simbol yang ada dalam kamus data adalah sebagai berikut:

Tabel 2.5. Tabel Simbol-simbol Kamus Data

No.	Simbol	Keterangan
1.	=	Disusun atau terdiri atas
2.	+	Dan
3.	[]	Baik ... atau ...
4.	{ }n	n kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

(*Sumber: Sukamto, Shalahuddin, 2018:74*)

2.4 Teori Program

2.4.1 Hypertext Preprocessor (PHP)



Gambar 2.2. Logo PHP

Sumber : <https://www.php.net/images/logos/new-php-logo.png>

Menurut Hidayatullah dan Kawistara (2014:231), PHP (*Hypertext Preprocessor*) atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk web *development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*.



2.4.2. Hypertext Markup Language (HTML)



Gambar 2.3. Logo HTML

Sumber : <https://images.app.goo.gl/i9VGAKTEDmaDoUnb9>

Menurut Hidayatullah dan Kawistara (2014:13), *Hypertext Markup Language* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halamn *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*.

2.4.3. Mysql

Menurut Hidayatullah dan Kawistara (2014: 180), *MySQL* adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah.



2.4.4. Database

Rusdiana dan Irfan (2014:302) menyatakan bahwa, “*Database* adalah kumpulan informasi yang disimpan dalam komputer secara sistematis untuk memperoleh informasi dari basis data”.

Berdasarkan dari pendapat di atas, dapat disimpulkan bahwa *Database* merupakan kumpulan data yang disusun secara terstruktur dan saling terhubung dan kemudian disimpan di dalam sebuah media penyimpanan di dalam komputer.

2.4.5. XAMPP



Gambar 2.4. Logo XAMPP

Sumber : <https://www.apachefriends.org/about.html>

Haqi dan Setiawan (2019:8), menyatakan bahwa “XAMPP adalah perangkat lunak bebas (*free software*) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari beberapa program, antara lain: Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl”.

2.4.6. *phpMyAdmin*

Pengertian *phpMyAdmin* sebagaimana dikemukakan oleh Hikmah, dkk (2015:2), “*phpMyAdmin* merupakan aplikasi yang dapat digunakan untuk membuat database, pengguna (*user*), memodifikasi tabel, maupun mengirim database secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL”.



2.5. **Black-Box Testing**

Menurut Jaya (2018), *Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black-box Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Menurut Roger S. Pressman (2012:597), Pengujian kotak hitam (*black box*) berupaya untuk menemukan kesalahan dalam kategori berikut: (1) fungsi yang salah atau hilang, (2) kesalahan dalam struktur data atau akses basis eksternal, (3) kesalahan perilaku atau kinerja, dan (4) kesalahan inisialisasi dan penghentian.