



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

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

Astropudin dalam Ismarina (2018:23), “Komputer adalah sebuah alat bantu pemrosesan data secara elektronik dan cara datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer”.

2.1.2 Pengertian Perangkat Lunak

Abdul Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi *Microsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak”.

Sukanto dan Shalahuddin (2018:2), “Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Kristanto (2018:1), “Perangkat Lunak (*software*) adalah instruksi (program komputer) yang ketika dijalankan menyediakan fungsi dan tampilan yang diinginkan, struktur data yang memberi kesempatan program untuk memanipulasi informasi dan dokumen yang mendeskripsikan operasi dan penggunaan program”.

2.1.3 Pengertian Basis Data

Yudho dan Helmi (2019:16), “Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.”



Rosa dan Shalahuddin dalam Pradana dkk (2016:25), “ Sistem Basis Data adalah sistem yang terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi teredia saat dibutuhkan”.

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

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Jogiyanto HM (dalam Suhartini (2017)), “Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru.

Pengertian aplikasi secara umum adalah terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

2.2.2 Pengertian Website

Abdulloh (2016:01) dalam Delia (2019:5) “*Website* adalah sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa gambar, teks, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.

Sidik (2017:1) dalam Delia (2019:5) “*Website* merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di internet) untuk mendapatkan informasi, dengan cukup mengklik suatu link berupa teks atau gambar, maka informasi dari teks atau gambar akan ditampilkan secara lebih rinci (detail).”



2.2.3 Pengertian Sistem Pendukung Keputusan

Little, (2017) dalam (Delia, 2019 : 6) mengungkapkan bahwa SPK adalah suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternative keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data dan model.

Pada dasarnya Sistem Pendukung Keputusan atau dikenal juga dengan istilah Decision Support System (DSS) ini merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedekimian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif ini dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan yang bersifat fleksible.

Berikut adalah keterbatasan sistem pendukung keputusan :

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya.
3. Proses-proses yang dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
4. SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Karena waktu bagaimanapun canggihnya suatu SPK hanyalah suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir.

2.2.4 Pengertian Metode *Simple Additive Weighting* (SAW)

Latif (2018:21) dalam (Delia, 2019 : 7) SAW merupakan metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep metode ini adalah dengan mencari rating kinerja (skala prioritas) pada setiap alternative di semua atribut.



Adapun algoritma penyelesaian metode ini adalah sebagai berikut :

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolak ukur penyelesaian masalah.
2. Langkah 2 : Menormalisasi setiap nilai alternative pada setiap atribut dengan cara menghitung rating kinerja.
3. Langkah 3 : Menghitung nilai bobot preferensi pada setiap alternative.
4. Langkah 4 : Melakukan perangkaian.

Adapun rumus yang digunakan pada metode *Simple Additive Weighting* yaitu :

Menormalisasikan setiap alternatif (menghitung nilai rating kinerja)

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\max_i X_{ij}} & \rightarrow \text{Jika } j \text{ adalah attribute keuntungan (benefit)} \\ \frac{\min_i X_{ij}}{X_{ij}} & \rightarrow \text{Jika } j \text{ adalah attribute biaya (cost)} \end{cases}$$

Menghitung nilai bobot preferensi pada setiap alternative

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Keterangan :

V_i = Nilai Bobot Preferensi dari setiap alternative

w_j = Nilai Bobot Kriteria

r_{ij} = Nilai Rating Kinerja



2.2.5 Pengertian Kelayakan

Kamus Besar Bahasa Indonesia (2016), “Kelayakan adalah perihal layak (patut, pantas); kepastian; kepatutan; atau perihal yang dapat (pantas, patut) dikerjakan.”

Berdasarkan pengertian diatas, dapat disimpulkan bahwa kelayakan merupakan kriteria penentuan apakah suatu subyek layak atau tidak. Subyek yang ditentukan juga bermacam-macam.

Dapat disimpulkan juga kelayakan adalah penelitian yang bertujuan untuk mengetahui dapat atau tidaknya suatu proyek dilaksanakan atau dikerjakan dengan berhasil.

2.2.6 Pengertian Aplikasi Manajemen Aset Desa Nantal Kecamatan Lahat Selatan Berbasis *Website* menggunakan Metode *Simple Additive Weighting*

Aplikasi Manajemen Aset Desa Nantal Kecamatan Lahat Selatan Berbasis *Website* menggunakan Metode *Simple Additive Weighting* merupakan sebuah aplikasi sistem pendukung keputusan yang bertujuan untuk membantu perangkat Desa dalam menentukan kelayakan manajemen aset desa yang layak, untuk digunakan aplikasi ini dapat memberikan referensi atau saran untuk memilih aset yang layak digunakan dengan kriteria penilaian aset berdasarkan kriteria dan subkriteria masing-masing.

2.3 Teori Khusus

2.3.1 Model *Rapid Application Development* (RAD)

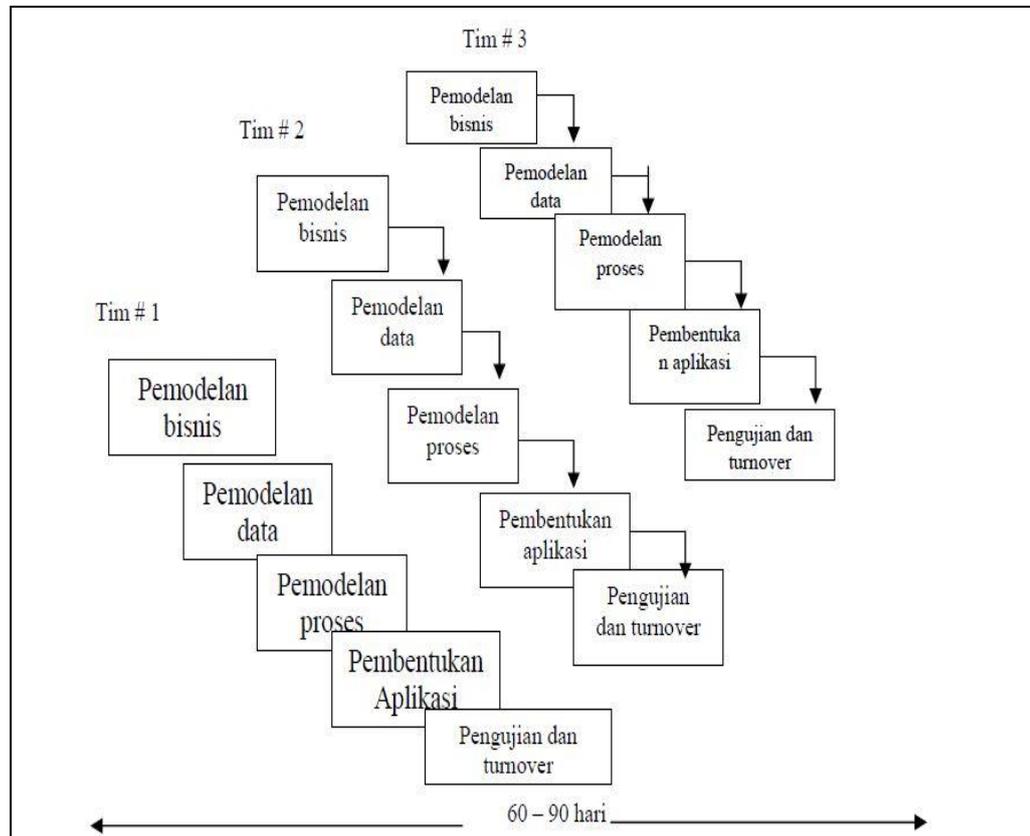
Metode pengembangan sistem yang dipakai yakni *Rapid Application Development*, menurut Sukamto dan Shalahuddin (2018:34), RAD adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

Jika kebutuhan perangkat lunak dipahami dengan baik dan lingkup perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan



pembuatan perangkat lunak dengan waktu yang pendek. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel.

Berikut adalah gambar model RAD:



(Sumber: Sukamto dan Shalahuddin, (2016:35))

Gambar 3.4 Model RAD

1. Pemodelan Bisnis

Pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa 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.

2. Pemodelan Data

Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.



3. Pemodelan Proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.

4. Pembentukan Aplikasi

Mengimplementasikan pemodelan proses dan data menjadi program. Model *RAD* sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.

5. Pengujian dan Pergantian

Menguji komponen-komponen yang dibuat. Pada sistem ini menggunakan *blackbox testing*, yaitu pengujian perilaku, yang berfokus pada persyaratan fungsional perangkat lunak (Pressman dalam Febiharsa dkk, 2018: 118). Jika sudah teruji maka tim pengembang komponen dapat beranjak untuk mengembangkan komponen berikutnya.

2.3.2 Pengertian *Data Flow Diagram* (DFD)

Agus Saputra (2018:5), “DFD merupakan suatu diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas”.

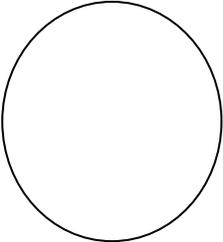
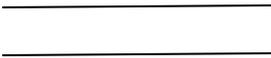
Feri Sulianta (2017:148), “*Data Flow Diagram* (DFD) atau disebut juga dengan Diagram Aliran Data (DAD) merupakan diagram yang menggambarkan suatu aliran data suatu sistem.

Sukamto dan Shalahuddin (2016:70) dalam Delia (2019:9), “ *Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

Adapun notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :



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> <p>Catatan :</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>
2		<p>File basis 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</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)</p> <p>Catatan :</p> <p>Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</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> <p>Catatan :</p> <p>Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>



Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*

No	Notasi	Keterangan
4		<p>Aliran data : merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan :</p> <p>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, Shalahuddin (2016:71-72)*)

Sukamto dan Shalahuddin (2016:72-73) 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 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 tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-breakwon



lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul ada DFD Level 1 yang di-breakdown.

4. Membuat DFD Level 3 dan seterusnya

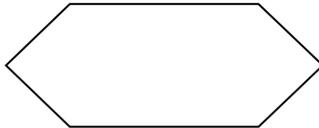
DFD Level 3,4,5 dan seterusnya merupakan breakdown dari modul pada DFD di atasnya. Breakdown pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.3.3 Pengertian *Flow Chart*

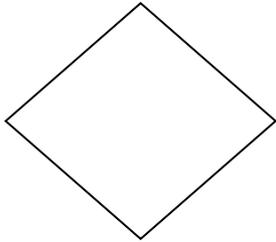
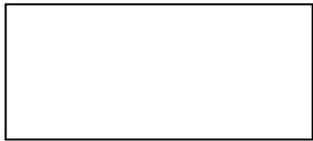
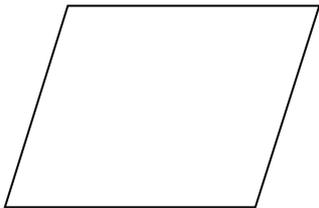
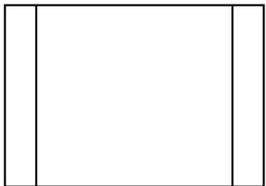
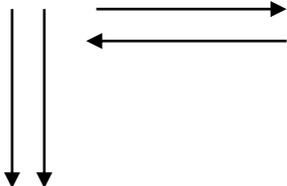
Rosa A. S. (2018:844), "*Flow Chart* adalah sebagai sebuah perangkat (tool) untuk membantu membuat rancangan algoritma, karena merupakan perangkat maka yang menggunakannya adalah yang merasa sesuai dengan perangkat ini, tidak harus digunakan untuk semua karena ada banyak perangkat untuk membantu cara berpikir manusia."

Rosa A. S. (2018:843), "*Flow Chart* adalah sebuah bagan aliran dari sesuatu, dan sesuatu itu dapat juga berupa aliran proses.

Tabel 2.2 Simbol-simbol *Flow Chart*

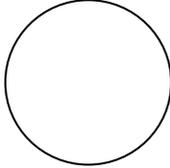
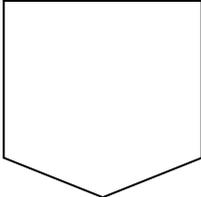
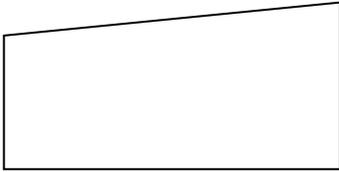
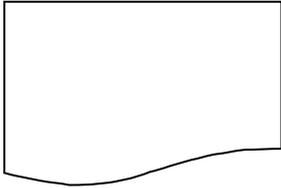
No	Simbol	Nama	Keterangan
1		Terminal	Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses atau kegiatan.
2		<i>Preparation</i>	Symbol ini digunakan untuk mempersiapkan harga awal atau nilai awal suatu variabel yang akan diproses dan digunakan untuk proses <i>loop</i> .

Lanjutan Tabel 2.2 Simbol-simbol *Flow Chart*

No	Simbol	Nama	Keterangan
3		Keputusan (<i>decision</i>)	Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.
4		Proses	Simbol ini digunakan untuk menggambarkan suatu proses yang sedang dieksekusi.
5		<i>Input/Output</i>	Simbol ini digunakan untuk menggambarkan proses <i>input</i> (<i>read</i>) maupun proses <i>output</i> (<i>print</i>).
6		Subroutine	Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari <i>main</i> program.
7		<i>Flow Line</i>	Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

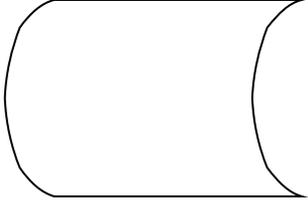
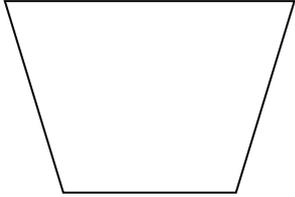
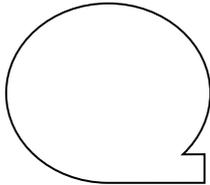


Lanjutan Tabel 2.2 Simbol-simbol *Flow Chart*

No	Simbol	Nama	Keterangan
8		<i>Conector</i>	Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman.
9		<i>Page Conector</i>	Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.
10		<i>Manual Operation</i>	Simbol ini digunakan untuk menggambarkan suatu kegiatan alur proses yang bersifat manualisasi.
11		<i>Printer</i>	Simbol ini digunakan untuk menggambarkan suatu dokumen atau suatu kegiatan mencetak suatu informasi dengan mesin printer.



Lanjutan Tabel 2.2 Simbol-simbol *Flow Chart*

No	Simbol	Nama	Keterangan
12		<i>Concole</i>	Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT (<i>Cathode Ray Tube</i>).
13		<i>Disk</i>	Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media magnetic disk.
14		<i>Manual Input</i>	Simbol ini digunakan untuk menggambarkan proses pemasukan data melalui media keyboard.
15		<i>Tape</i>	Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media magnetic tape.

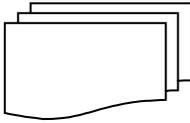
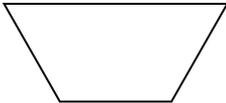
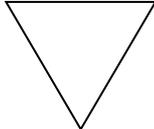
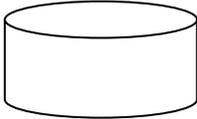
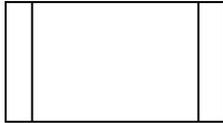
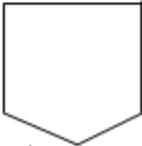
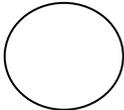
(*Sumber : Rosa A. S. (2018:843)*)

2.3.4 Pengertian *Blockchart*

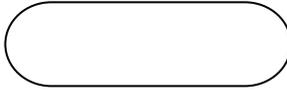
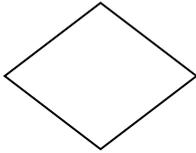
Kristanto (2018:68) “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Kristanto (2018:65) menjelaskan, “Simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini :

Tabel 2.3 Simbol-simbol *Block chart*

No	Simbol	Arti
1		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/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>)
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

Lanjutan Tabel 2.3 Simbol-simbol *Block chart*

No	Simbol	Arti
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 (2008:68-70))

2.3.5 Pengertian *Entity Relationship Diagram* (ERD)

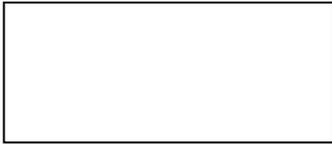
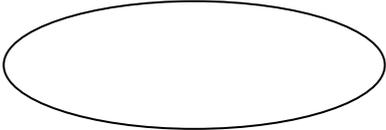
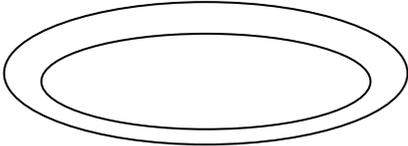
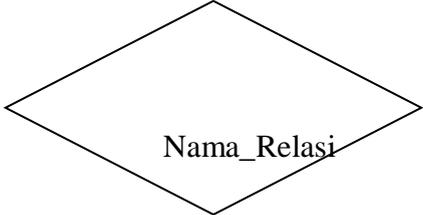
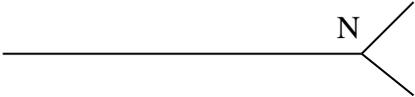
Feri Sulianta (2019:215), “*Entity Relationship Diagram* (ERD) merupakan diagram yang digunakan untuk merancang tabel-tabel yang nantinya akan diimplementasikan pada basis data”.

Sukamto dan Shalahuddin (2018:18-19), “*Entity Relationship Diagram* (ERD) adalah Pemodelan awal basis data yang paling banyak digunakan. ERD sendiri digunakan untuk pemodelan basis data relasional”.

ERD memiliki beberapa aliran notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Adapun simbol-simbol dari *entity Relationship Diagram* (ERD) adalah sebagai berikut :


Tabel 2.4 Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1	Entitas/Entity 	<i>Entitas</i> , merupakan data inti yang akan disimpan; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke data benda dan belum merupakan nama table.
2	Atribut 	<i>Atribut</i> , Field atau kolom data yang butuh disimpan dalam suatu entitas.
3	Atribut multinilai/multivalue 	Field atau kolom data yang dibutuhkan disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
4	 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
5	Asosiasi / Association 	<i>Asosiasi</i> , adalah penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.

(Sumber: Sukamto, Shalahuddin (2018:75))



2.3.6 Kamus Data

Andri Kristanto (2008:72), “Kamus Data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file di dalam sistem.

Uus (2019:36), “Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Sukamto dan Shalahuddin (2014:19), “Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum memiliki *standard* cara penulisan.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel 3.5 Simbol-Simbol Kamus Data (*Data Dictionary*)

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: Sukamto, Shalahuddin (2014:19)*)



2.3.7 Pengertian Metode *Rapid Application Development*

Sukamto dan Shalahuddin (2018:34), “*Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek.”

2.4 Teori Program

2.4.1 Pengertian PHP

Rohi Abdullah (2018:127), “PHP merupakan kependekan dari *PHP Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip *HTML* dan bekerja di sisi *server*. Tujuan dari bahasa ini adalah membantu para pengembangan *web* untuk membuat *web* dinamis dengan cepat”.

Miftahul Jannah dan Surwandi (2019:1), “Hypertext Preprocessor atau lebih akrab dengan sapaan PHP merupakan bahasa pemrograman script server-side yang didesain untuk pengembangan web”.

Mundzir (2018:3), “PHP berasal dari kata “*HypertextPreprocessor*”, yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan *HTML*”.

2.4.2 Pengertian MySQL

Yudho dan Helmi (2019:16), “*MySQL* yang merupakan singkatan dari “*My Structured Query Language*” adalah database yang paling *favorite* saat ini. Program ini berjalan sebagai server yang menyediakan *multi-user*, mengakses ke sejumlah database baik *multithread* maupun *multi-user*, dan telah diinstal oleh sekitar 6 juta kali di seluruh dunia”.

Bunafit Nugroho (2019:133), “*MySQL* merupakan *database* yang paling digemari dikalangan *Programmer Web*, dengan alasan bahwa program ini merupakan *database* yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpan data”.

Mundzir (2018:217), “*MySQL* adalah sistem manajemen *database* yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem



database MySQL mampu mendukung beberapa fitur seperti *multithreaded*, *multiuser*, dan *SQLdatabase management system (DBMS)*”.

2.4.3 Pengertian XAMPP

Yudho dan Helmi (2019:17), “XAMPP adalah kompilasi program aplikasi gratis terfavorit di kalangan developer/programmer yang berguna untuk pengembangan *website* berbasis *PHP* dan *MySQL*”.

Jubile Enterprise (2019:3), “XAMPP merupakan *server* yang paling banyak digunakan untuk para *programmerPHP*, khususnya level pemula”.

Wahana dalam Prayitno (2015:2), “ XAMPP adalah salah satu paket instalasi *apache*, *PHP*, dan *MySQL* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”.

2.5 Referensi Penelitian Sebelumnya

Penelitian ini menggunakan beberapa jurnal dari penelitian sebelumnya yang dilakukan oleh beberapa orang yang menggunakan Metode *Simple Additive Weighting* (SAW), yaitu :

Erlina, Nofita, dan Acep (2019), menjelaskan bahwa sistem aplikasi manajemen aset sering terjadi kesalahan dalam pencatatan terkait manajemen aset yang masih dilakukan secara no-komputerisasi sehingga penelusuran data aset akan lebih sulit dan kurang efisien dalam hal waktu agak lama.

Jerry Ariska dan M.Jazman (2016) menjelaskan sekolah pada MAN 2 di Pekanbaru memiliki aset yang banyak. Namun dalam pengelolaan aset yang ada, sekolah ini masih menggunakan cara pembukuan. Oleh karena itu aset yang telah berpindah tempat atau dipinjam sangat sulit untuk dilacak keberadaannya. Sehingga resiko kehilangan aset pun akan meningkat, kemudian sulitnya bagian sarana dan prasarana dalam proses pembuatan laporan manajemen aset. Pembuatan laporan membutuhkan waktu yang cukup lama karena petugas harus mengecek kembali data barang tersedia dan data kondisi barang yang sudah rusak maupun membutuhkan maintenance dengan membuka kembali berkas-berkas yang ada. Resiko kehilangan data pun cukup tinggi hal ini dikarenakan



setelah melakukan pencatatan data barang di buku, petugas biasanya tidak langsung menginputkannya ke dalam komputer. Akibatnya jika terjadi kehilangan buku maka hilang pula semua data aset yang ada.

Suryani, Linda dan Yuwan (2018) menjelaskan sistem informasi yang dibutuhkan dalam memonitoring aset dan barang inventaris di perusahaan tersebut akan mengalami perubahan yang berlaku dengan cepat sehingga akan membuat petugas menjadi kesulitan dalam memonitoring aset dan barang inventaris yang ada. Informasi yang kurang akurat dan terupdate mengenai barang inventaris yang tersedia, menjadi kendala dalam proses peminjaman barang inventaris karena karyawan yang ingin meminjam barang inventaris harus menemui pihak-pihak yang berkaitan.

Riana, Wina, dan Irma (2020) menjelaskan permasalahan yang sering terjadinya keterlambatan dalam melakukan pengelolaan dan pemeliharaan aset yang dimiliki, sehingga kejadian tersebut dapat menyebabkan kerusakan pada aset. Penelitian manajemen aset dilakukan pada tahap perencanaan, pengadaan, penerimaan, penggunaan, penyusunan, penghapusan dan penatausahaan. Pemeliharaan aset di perusahaan juga tidak memiliki jadwal khusus untuk melakukan pemeriksaan aset dikarenakan pemeliharaan aset masih menggunakan corrective maintenance sehingga kerusakan pada aset tidak dapat dikontrol dengan baik.

Ridwan, Muhammad, dan Siti (2017) menjelaskan masalah dalam manajemen aset dari proses perawatan aset karena masih menggunakan ingatan perorangan saja sedangkan asetnya banyak. Jika perorangan tersebut terlupa akan menjadi masalah. Penelitian pada sistem informasi manajemen aset di PT.Sentral Tukang Indonesia ini untuk memudahkan sentral tukang untuk melakukan pencatatan dan perawatan aset pada perusahaan tersebut.