

BAB II TINJAUAN PUSTAKA

Penelitian Terdahulu merupakan salah satu acuan bagi penulis agar penulis mendapatkan hasil penelitian yang optimal dan memperoleh perbandingan dari penelitian sebelumnya. Adapun hasil-hasil penelitian yang dijadikan ialah tidak terlepas dari topik penelitian mengenai aplikasi sistem pendukung keputusan dengan metode *fuzzy simple additive weighting* (FSAW). Berikut adalah penelitian terdahulu beberapa jurnal terkait dengan penelitian yang dilakukan penulis:

Penelitian oleh Rahman Abdillah dengan judul **“Implementasi *fuzzy simple additive weighting* (SAW) sebagai pendukung keputusan pada beasiswa penelitian.”** membahas tentang Lembaga Pengelola Dana Pendidikan (LPDP) seperti program beasiswa dan bantuan dana penelitian. LPDP memiliki satu misi mengembangkan program RISPRO (Riset Inovatif dan Produktif). Adapun tujuan dari penelitian ini adalah untuk mengimplementasikan *Fuzzy Simple Additive Weighted* (*Fuzzy SAW*) ke dalam bagian dari sistem pendukung keputusan (DSS) dalam penentuan beasiswa penelitian RISPRO. Nilai akhir yang digunakan sebagai bahan acuan pengambilan keputusan akan diurutkan dan kemudian pengambil keputusan dapat mengambil data beberapa peserta dengan nilai yang tertinggi sesuai dengan kebijakan lembaga. Dari hasil pengolahan data dan penghitungan yang didapat, penggunaan metode *Fuzzy SAW* membantu pengambil keputusan dalam menentukan calon penerima beasiswa dengan lebih cepat dan akurat.

Dalam penelitian Kurniawan, Denni, and Catur Nugroho yang berjudul **“Sistem Pendukung Keputusan Dalam Penilaian Prestasi Kerja Menggunakan Fuzzy-Ahp Dan Saw”**, peneliti ini membahas tentang upaya perusahaan mengevaluasi kinerja dan produktifitas karyawan. Sebagai hasilnya, perusahaan juga dapat memberikan penghargaan kepada karyawan yang dianggap memberikan kontribusi bagi perkembangan perusahaan. Penelitian ini bertujuan melakukan penilaian prestasi kerja karyawan dengan menggunakan gabungan metode logika Fuzzy-Analytic Hierarchy Process (AHP) dan Simple Additive Weighting (SAW). Metode AHP menggunakan matrik perbandingan berpasangan dalam melakukan perhitungan pembobotan. Namun metode ini dianggap masih memiliki subjektifitas

dalam memberikan penilaian kriteria, sehingga digunakan logika Fuzzy untuk mengatasi permasalahan ini. Hasil perhitungan menggunakan metode Fuzzy-AHP tersebut akan dilanjutkan kepada proses perankingan dengan menggunakan metode SAW. Karyawan yang memiliki nilai bobot paling besar akan dianggap sebagai karyawan yang paling produktif dan memiliki kinerja yang paling baik. Adapun hasil dari penelitian ini adalah Karyawan dengan nomor A027 merupakan peringkat tertinggi yang memiliki nilai bobot 0,907.

Penelitian **“Penerapan Sistem Pendukung Keputusan Berbasis Komputer Peminjaman Buku Menggunakan Metode Simple Additive Weighting (SAW) Di Politeknik Negeri Sriwijaya Palembang”** oleh A. Zaky Kurniawan. Mengangkat permasalahan tentang pembuatan laporan tahunan mengenai data buku yang sering dipinjam, agar dapat menyediakan lebih banyak buku yang dibutuhkan dan sesuai dengan karakteristik buku yang sering dipinjam mahasiswa dibutuhkan sistem pendukung keputusan. Adapun tujuan dari penelitian ini adalah dengan menggunakan Metode *Simple Additive Weighting* (SAW) memudahkan untuk mencari skala prioritas pada setiap *alternatif* di semua atribut dan dapat melihat data peminjaman buku dan hasil perhitungan dari metode perankingan *Simple Additive Weighting* (SAW) pada tampilan hasil.

Penelitian yang berjudul **“Fuzzy Multi-Attribute Decision Making (Fuzzy MADM) Dengan Metode SAW Untuk Pemilihan Mahasiswa Berprestasi”** oleh Christioko, Bernadus Very, Henny Indriyawati, dan Nurtriana Hidayati. Membahas tentang menentukan mahasiswa berprestasi terbaik. Dikarenakan mahasiswa berprestasi terkadang mengalami kendala saat penilaian kriteria tertentu pada kandidat kurang memenuhi namun dikriteria lain penilaian yang diperoleh melebihi ketentuan. Seperti nilai akademik yang tinggi tetapi tidak aktif dalam kegiatan kokurikuler dan ekstrakurikuler begitu sebaliknya. Maka fakultas mengatasi dengan metode pengambilan keputusan untuk menentukan mahasiswa berprestasi terbaik tingkat fakultas. Adapun tujuan dari penelitian ini adalah menerapkan metode *Fuzzy* untuk mengatasi ketidakpastian data atau informasi dan *Multi-Attribute Decision Making* (FMADM). Dengan metode *Simple Additive Weighting* (SAW) untuk metode pengambilan keputusan dalam pemilihan Mahasiswa berprestasi. Hasil dari penelitian ini adalah penerapan metode FMADM untuk

mendukung pengambilan keputusan dalam pemilihan mahasiswa berprestasi tingkat fakultas dengan kandidat yang berasal dari setiap program studi.

Pada penelitian yang berjudul **“Perancangan Sistem Pendukung Pengambilan Keputusan Untuk Penerimaan Beasiswa Di SMA PGRI 1 Talang Padang Dengan Model Fuzzy Multiple Attribute Menggunakan Metode Simple Additive Weighting (SAW).”** oleh Muslihudin, Muhamad, and Dewi Hartini. Mengangkat masalah tentang penerimaan beasiswa yang diberikan oleh pemerintah. Dibutuhkan sistem pendukung keputusan untuk memecahkan masalah dan memberikan solusi terbaik. Karena banyaknya kriteria-kriteria yang harus dimiliki individu sebagai syarat mendapatkan beasiswa. Maka, pencarian *alternatif* terbaik berdasarkan kriteria-kriteria yang telah ditentukan dengan menggunakan metode SAW (*Simple Additive Weighting*). Penelitian dilakukan dengan mencari nilai bobot untuk setiap atribut, kemudian dilakukan proses perangkingan yang akan menentukan alternatif yang optimal, yaitu siswa terbaik. Adapun tujuan dari penelitian ini adalah Untuk mempercepat dan mempermudah dalam menentukan dan menetapkan siswa yang berhak mendapat beasiswa sesuai dengan bobot perhitungan penerimaan beasiswa.

2.1 Sistem Pendukung Keputusan

Menurut Nofriansyah dan Sarjon di dalam jurnal (Nanda, 2019), “Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur”.

Sedangkan menurut Little dalam buku *Multi Criteria Decision Making* (MCDM) pada Sistem Pendukung Keputusan yang dikutip Nofriansyah dan Sarjon di dalam jurnal (Nanda, 2019), mendefinisikan sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan sehingga membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model.

Dari berbagai definisi diatas dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi yang dirancang untuk membantu

seseorang maupun sekelompok orang dalam mengambil keputusan terhadap permasalahan yang bersifat terstruktur maupun tidak terstruktur melalui beberapa alternatif dan kriteria yang telah ditentukan sehingga dapat menghasilkan keputusan yang lebih cepat dan tepat. Komponen-komponen Sistem Pendukung Keputusan menurut Nofriansyah dan Sarjon di dalam jurnal (Nanda, 2019), terdiri dari:

- a. Data Management. Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*.
- b. Model Management. Melibatkan model *finansial*, *statistical*, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analisis, dan manajemen *software* yang diperlukan.
- c. *Communication (dialog subsystem)*. *User* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antaramuka.
- d. *Knowledge Management*. Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.2 Logika Fuzzy

Menurut (Abdillah, 2017), pada tahun 1962, Prof. Lotfi Astor Zadeh memperkenalkan konsep tentang logika *Fuzzy*. Logika *Fuzzy* adalah salah satu metodologi pemecahan masalah yang dapat diimplementasikan pada sistem yang sederhana, *embedded system*, PC dan jaringan, *multi-channel*, *workstation* berbasis akuisisi data, dan *sistem control*. Metodologi ini dapat diterapkan pada perangkat keras dan perangkat lunak. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner yang berarti hanya memiliki dua kemungkinan, Ya atau Tidak, Benar atau Salah, Baik atau Buruk, dan lain lain. Oleh karena itu, semua dapat mempunyai nilai keanggotaan 0 atau 1. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara nilai 0 dan 1, sehingga hasil dari proses yang menggunakan logika *fuzzy* akan lebih presisi jika dibandingkan dengan logika biner.

2.3 *Multiple Atribut Decision Making (MADM)*

Menurut (Abdillah, 2017). Pada dasarnya, proses dari MADM dalam tiga fase, yakni : tahap persiapan komponen yang sesuai dengan situasi, analisis dan mensintesis informasi . Ada beberapa metodologi yang digunakan dalam penyelesaian masalah FMADM:

- a. *Simple Additive Weighting Method (SAW)*
- b. *Weighted Product (WP)*
- c. *ELECTRE*
- d. *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*
- e. *Analytic Hierarchy Process (AHP)*

Sebagai bagian dari pendukung keputusan, model MADM dapat digambarkan ke dalam matriks sebagai berikut :

$$\begin{array}{c}
 C_1 \quad C_2 \quad C_3 \quad C_k \\
 \begin{array}{l}
 A_1 \\
 A_2 \\
 \vdots \\
 A_k
 \end{array}
 \left[\begin{array}{cccc}
 d_{11} & d_{12} & d_{13} & d_{14} \\
 d_{21} & d_{22} & d_{23} & d_{24} \\
 \vdots & \vdots & \vdots & \vdots \\
 d_{m1} & d_{m2} & d_{m3} & d_{mk}
 \end{array} \right]
 \end{array}$$

Dalam matriks tersebut $A_1, A_2, A_3, \dots, A_m$ sebagai alternatif dan memiliki nilai maksimal m , sedangkan C_1, C_2, C_3, C_k adalah atribut-atribut dalam komponen pendukung keputusan yang digunakan untuk mengukur setiap alternatif. Menurut Memariani yang terdapat di dalam jurnal (Abdillah, 2017). Hal yang paling penting dalam penggunaan model MADM adalah data yang digunakan tidak stabil dan bisa berubah, sehingga sensitifitas analisis setelah penyelesaian masalah dapat sangat efektif berkontribusi dalam proses pengambilan keputusan. Walaupun hasil penghitungan yang diperoleh cukup sensitive, bisa berubah dan tidak stabil, namun hasil akhir tetap diserahkan kepada seseorang untuk dapat membuat keputusan dalam proses penyelesaian masalah.

2.4 *Simple Additive Weighting (SAW)*

Metode SAW lebih banyak dikenal dan digunakan dalam menyelesaikan permasalahan pada *Multiple Atribut Decision Making (MADM)*. Metode ini merupakan salah satu metode yang biasanya diterapkan pada sistem pendukung keputusan atau biasa digunakan dalam pemecahan masalah yang melibatkan

banyak alternatif pilihan sehingga dapat membantu proses pengambilan keputusan dengan lebih cepat dan tepat.

Metode SAW adalah salah satu yang paling sering digunakan dikarenakan caranya yang sederhana. Metode SAW sering juga dikenal dengan istilah metode penjumlahan terbobot. Menurut (Nofriansyah dan Defit, 2017), metode *Simple Additive Weighting* (SAW) merupakan metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep dalam metode ini adalah dengan mencari peringkat kinerja (skala prioritas) pada setiap *alternatif* di semua atribut.

2.5 Fuzzy Simple Additive Weighted (Fuzzy SAW)

Fuzzy SAW termasuk dalam salah satu metode penyelesaian MADM. Konsep dasar metode *Fuzzy SAW* adalah mencari penjumlahan terbobot dari *rating* kinerja pada setiap *alternatif* yang ada pada semua atribut. Metode *Fuzzy SAW* membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua *rating alternatif* yang ada (Abdillah, 2017).

Adapun algoritma penyelesaian metode ini (Nofriansyah dan Defit, 2017) diantaranya sebagai berikut:

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

Adapun rumus yang digunakan pada metode *simple additive weighting* (SAW), yaitu:

- a. Menormalisasikan setiap *alternating* (Menghitung nilai rating kinerja)

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i X_{ij}} ; \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i X_{ij}}{x_{ij}} ; \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana r_{ij} adalah rating kinerja *ternormalisasi* dari *alternatif* A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

r_{ij}	= nilai rating kinerja ternormalisasi
X_{ij}	= nilai atribut yang dimiliki dari setiap kriteria
Max X_{ij}	= nilai terbesar dari setiap kriteria ij
Min X_{ij}	= nilai terkecil dari setiap kriteria ij
Benefit	= jika nilai terbesar adalah terbaik
Cost	= jika nilai terkecil adalah terbaik

- b. Menghitung nilai bobot *preferensi* pada setiap *alternatif*

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

Keterangan :

V_i = ranking untuk setiap alternatif

W_j = nilai bobot dari setiap kriteria

r_{ij} = nilai *rating* kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.6 Aplikasi dan Website

Menurut Jogiyanto di dalam jurnal (Hikmah, 2022), Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. Menurut Kadir di dalam jurnal (Astuti dan Budiarti, 2022), Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus.

Website adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink yang memudahkan user untuk menggunakannya (Angelina, 2019). *Website* adalah sebuah komponen yang terdiri dari teks, gambar, suara, animasi sehingga menjadi media informasi yang menarik dikunjungi oleh orang lain (Zahir, 2020).

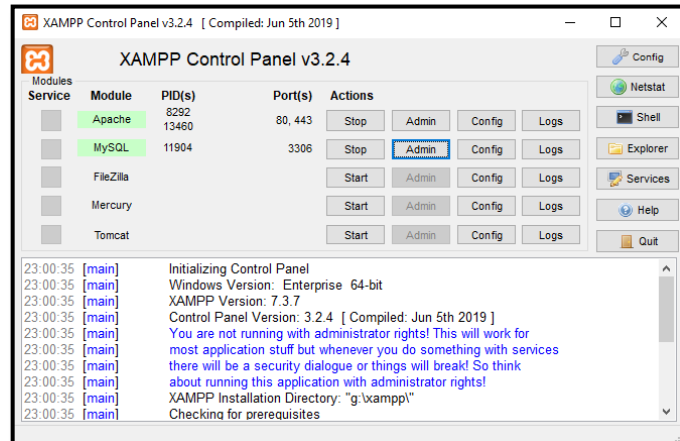
Menurut Nugroho di dalam jurnal (Suheri, 2022), Aplikasi Web merupakan aplikasi yang berjalan diberbagai *browser* seperti *internet Explorer*, *Mozilla*, *Opera* dan sebagainya. Menurut Simarmata di dalam jurnal (Hikmah, 2022), Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web.

Dari pengertian diatas dapat disimpulkan bahwa Aplikasi *website* adalah sistem informasi yang dapat memproses *input* menjadi *output* dan bisa diakses oleh *user* dari berbagai browser seperti *Explorer*, *Chrome*, *Mozilla* dan sebagainya.

2.7 XAMPP

Menurut Jaelani dalam jurnal (Angelina, 2019), XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kumpulan dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri atas program *Apache HTTP server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari:

- X yang berarti *cross platform* karena XAMPP bisa dijalankan di windows, Linux, Mac, dsb.
- A yang artinya *Apache* sebagai *web server*-nya.
- M yang berarti *MySQL* sebagai *Database Management System (DBMS)*- nya
- PP yang berarti PHP dan Perl sebagai bahasa yang didukung.



Gambar 2. 1 Tampilan XAMPP

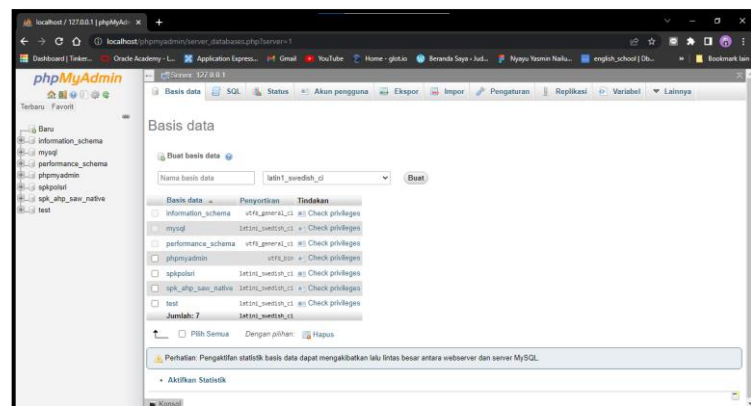
2.8 PHPMysqlAdmin

Menurut MADCOMS dalam jurnal (Hidayat dkk., 2017), *PhpMyAdmin* adalah sebuah aplikasi *Open Source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *PhpMyAdmin*, dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan mengupdate data tanpa perlu mengetikkan perintah SQL secara manual. Menurut (Firliana dkk., 2020),

phpMyAdmin adalah salah satu aplikasi yang digunakan untuk memudahkan dalam melakukan pengelolaan *database* MySQL. *PhpMyAdmin* merupakan aplikasi web yang bersifat *open source*.

Dari pengertian diatas dapat disimpulkan bahwa *PhpMyadmin* adalah aplikasi *open source* yang digunakan untuk mempermudah pengelolaan *database* baik membuat tabel, menambahkan, menghapus dan mengedit data. Adapun beberapa menu yang ada di *PhpMyAdmin*, yaitu:

- *Structure*. Menu ini berguna untuk mengorganisir *schema*, tabel, kolom, baris, dan lain lain.
- *SQL*. Menu ini digunakan untuk menjalankan *query* atau kueri pada *database*.
- *Search*. Untuk mencari kata-kata atau *value* di dalam tabel *database*.
- *Query*. *SQL* mendefinisikan seperangkat perintah, seperti *SELECT*, *INSERT*, *UPDATE*, *DELETE*, *CREATE TABLE* dan lain-lain.
- *Export*. Anda bisa meng-*export database* dengan format berbeda-beda seperti CSV, PDF, SQL, XML, Text, dan masih banyak lagi.
- *Import*. Anda bisa meng-*import database* dengan format berbeda-beda seperti *OpenDocument Spreadsheet*, CSV, SQL, ESRI *Shape* file, *Mediawiki Table*, XML.
- *Operations*. Ada beberapa jenis operations yang bisa Anda eksekusi di seluruh database dan tabel terpisah.
- *Triggers*. Sebuah *trigger* dikenal sebagai objek basis data yang terhubung dengan tabel, dan akan diaktifkan kapan pun terjadi suatu *event*. Misalnya *insert*, *update* atau *delete* terjadi pada tabel.



Gambar 2. 2 Tampilan phpMyAdmin

2.9 Database atau Basis Data

Database atau basis data adalah sekumpulan informasi yang disimpan dalam komputer secara sistematis dan merupakan sumber informasi yang dapat diperiksa menggunakan suatu program komputer dan berfungsi untuk menyimpan informasi atau data (Andaru, 2018). *Database* adalah kumpulan data yang tersimpan dalam tabel-tabel. Tabel-tabel tersebut itu disusun berdasarkan baris dan kolom (Putri dan Putra, 2020).

Dari pengertian diatas dapat disimpulkan bahwa *database* adalah kumpulan data atau informasi yang disimpan dari suatu program komputer dan tersimpan ke *database* dalam bentuk tabel berupa baris dan kolom.

Model *database* adalah jenis model data yang menentukan struktur logis dari suatu *database* agar data dapat disimpan, diatur dan dimanipulasi. Adapun jenis model *database* yang paling populer adalah *database relasional*. Dimana model *database* yang saling ada hubungan antara sejumlah data tabel yang masing-masing tabel terdiri dari beberapa kolom dan baris yang namanya unik (Efitra, 2021). Salah satu jenis model yang mengilustrasikan konsep model *database* salah satunya menggunakan *Entity Relationship Diagram* (ERD).

Relasi adalah hubungan antara beberapa entitas. Beberapa nama-nama simbol yang digunakan dalam pembuatan *Entity Relationship Diagram* (ERD) ditunjukkan pada Tabel berikut :

Tabel 2. 1 Nama simbol yang ada pada ERD

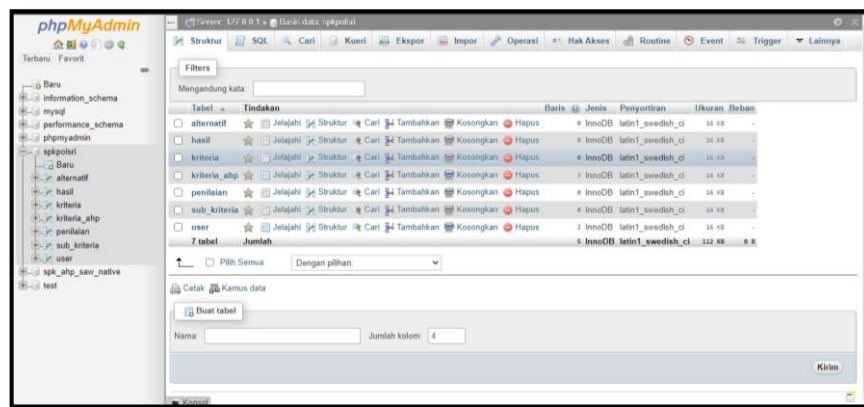
Entitas	Entitas dimana menggambarkan suatu objek yang dapat diidentifikasi dalam sebuah lingkungan
Atribut	Atribut memiliki fungsi mendeskripsikan karakter entitas
Relasi	Relasi untuk menunjukan sebuah hubungan atau relasi antara entitas
Garis Penghubung	Garis penghubung adalah sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

2.10 MySQL (Structured Query Language)

MySQL adalah sebuah perangkat lunak *system* manajemen *database* SQL (DBMS) yang *multithread*, dan *multi-user*. *MySQL* dapat didefinisikan sebagai

system manajemen *database*. *Database* sendiri merupakan struktur Penyimpanan Data. Jadi *Database MySQL* merupakan *system* manajemen basis data SQL yang sangat terkenal dan bersifat *Open Source* (Anthonie, 2016). kepopuleran *MySQL* antara lain karena *MySQL* menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. *MySQL* juga bersifat *open source* dan *free* pada berbagai *platform*.

Dari pengertian diatas dapat disimpulkan bahwa *MySQL* adalah aplikasi yang digunakan untuk mengelola dan memajemen *database* SQL (DBMS) yang banyak digunakan untuk membangun aplikasi yang menggunakan *database* untuk menyimpan data.



Gambar 2. 3 Tampilan MySQL jenis *database* server

2.11 PHP (*Personal Hypertext Preprocessor*)

PHP merupakan kependekan dari *PHP Hypertext Preprocessor* yaitu bahasa pemrograman *web* yang dapat di sisipkan dalam *script* HTML dan bekerja di sisi *server*. Tujuan dari bahasa ini adalah membantu para pengembang *web* untuk membuat *web* dinamis dengan cepat.

Agar dapat menjalankan PHP harus menyediakan perangkat lunak sebagai berikut:

1. *Web Server* (*Apache, IIS, Personal Web Server/PWS*).
2. *PHP Server*.
3. *Database server* (*MySQL, Interbase, dan lain-lain*)

Aplikasi diatas tidak perlu di instal satu per satu karena biasanya telah tersedia paket aplikasi yang telah menyediakan ketiga ini dalam satu paket aplikasi seperti XAMPP atau *Appserv*. Jika menggunakan XAMPP, maka *script* PHP ditulis di dalam *folder htdocs*. Namun jika menggunakan *Appserv*, maka *script* PHP ditulis di dalam *folder www*. Fungsi PHP dalam pemrograman *web* diantaranya sebagai berikut :

1. Menghasilkan halaman dinamis pada *website*.
2. Membuat, membuka, menulis, membaca, menghapus dan menutup file pada *server*.
3. Memproses data yang dikirim dari *form* dan mengirim serta mengakses *cookie*.
4. Modifikasi data pada *database* dan mengontrol akses *user*.
5. Mengenkripsi data.

Penulisan Menurut(Hasyim, 2022), *script* PHP dituliskan di antaraa tanda yang memisahkan *script* PHP dengan *script* lainnya. Satu file PHP dapat berisi full *script* PHP atau dapat di sisipkan diantara *script* lain seperti HTML, CSS maupun *JavaScript*. Setiap baris *script* PHP harus diakhiri dengan tanda *semicolon* (;). Jika tidak, maka akan menampilkan pesan eror.

```

<body>
<?php
echo “ ” ;
?>
</body>
```

2.12 HTML (*HyperText Markup Language*)

Menurut (Khozaimi, 2020), *HyperText Markup Language* (HTML) merupakan standard bahasa yang diaplikasikan dalam penampilan halaman web dan terdiri dari beberapa elemen yang diwakili oleh beberapa tag. HTML dapat digunakan sebagai berikut:

1. Mempublikasikan halaman web secara *online*
2. Pembuatan tabel pada halaman *website*
3. Pembuatan form untuk transaksi dan registrasi *online*

4. Menambahkan beberapa objek pada halaman *website* seperti: animasi, audio, video dan gambar.

Tabel 2. 2 Tabel Tag Dasar HTML

Tag	Keterangan
!DOCTYPE html	Deklarasi yang mendefinisikan bahwa dokumen ini sebagai HTML 5.
<html>...</html>	Elemen dasar pada sebuah html.
<head>...</head>	Berisi kode-kode untuk menuliskan keterangan atau informasi tentang dokumen HTML tersebut, salah satu contoh bisa digunakan untuk <i>insert</i> file seperti CSS.
<title>...</title>	Digunakan untuk membuat judul pada dokumen html atau halaman web.
<body>...</body>	Berisi tag-tag dan elemen untuk isi konten dan <i>layout</i> tampilan halaman atau dokumen HTML yang biasa terlihat.
<header>...</header>	Digunakan untuk menampilkan bagian kepala (kop) atau bagian <i>header</i> (atas), tag ini hanya bisa digunakan pada HTML 5.
<footer>...</footer>	Digunakan untuk memberikan bagian kaki pada sebuah dokumen HTML, biasanya digunakan untuk informasi atau hak cipta

```

<!DOCTYPE html>
<html>
  <head>
    <title>Judul Halaman Web</title>
  </head>
  <body> <!-- isi konten dan layout tampilan -->
  </body>
</html>

```

2.13 CSS (*Cascading Style Sheet*)

CSS adalah singkatan dari *cascading style sheets*, berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web. Perancangan desain text dapat dilakukan dengan mendefinisikan *fonts* (huruf), *colors* (warna), *margins* (ukuran), *background* (latar belakang), *font size* (ukuran font) dan lain-lain. Elemen-elemen seperti *colors* (warna), *fonts* (huruf), *sizes* (ukuran) dan *spacing* (jarak) disebut juga *styles* (Muslim dan Dayana, 2016).

Dari pengertian diatas dapat disimpulkan bahwa *cascading style sheet* (CSS) adalah pemrograman yang berisi instruksisi untuk mempercantik tampilan dari suatu halaman website. Menurut (Lewenusa, 2020) cara penulisan *script* css dibagi menjadi 3 bagian, yaitu:

3.1 *Inline Style Sheet*

Penulisan *script* css di dalam elemen HTML. Untuk metode ini fungsinya hampir sama dengan menuliskan *property* pada tag HTML, penggunaannya untuk hal-hal tertentu saja. Misalkan memberi warna hijau pada tag “<p>” maka cara penulisannya sebagai berikut:

```
<body>
  <p style="color : green"> Warna tulisan hijau </p>
</body>
```

Penulisan *script* css dilakukan pada tag pembuka suatu elemen html. *Property style* akan mensupport semua *property* yang di support oleh css.

3.2 *Embedded Style Sheet*

Penulisan *script* css di dalam dokumen HTML. Metode ini menggunakan tag awal <style> dan tag penutup </style>. Contoh:

```
<head>
  <style type="text/css">
    P { color : red;
      } </style>
</head>
```

Biasanya penulisan *script* css ini dilakukan pada bagian elemen head

3.3 *Linked Style Sheet / External Style Sheet*

Penulisan *script* css di halaman yang berbeda atau terpisah dari HTML. Jadi kita tinggal melakukan link ke file css yang telah dibuat dengan menggunakan tag <link rel> yang ditempatkan pada bagian tag <head>. Beberapa keuntungan menggunakan aturan *style* ini diantaranya:

- a. Menghemat pengulangan aturan *style* yang sama di setiap halaman.
- b. Dapat mengubah tampilan beberapa halaman dengan mengubah hanya pada *style sheet* nya saja.

- c. *Style sheet* dapat bertindak sebagai *template style* untuk membantu penulis mencapai *style sheet* dokumen yang sama.

```
<head>
  <title>Judul Halaman Web</title>
  <link rel="stylesheet" href="style.css">
</head>
```

2.14 Bootstrap

Menurut Spurlock dalam jurnal (Pahlevi dkk., 2018), *Bootstrap* adalah sebuah *framework* untuk CSS dan berupa produk *open source* yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya *bootstrap* ini dibuat untuk membuat standarisasi *front end* untuk semua *programmer* di perusahaannya. *Bootstrap* telah berubah dari yang sebelumnya *CSS-driven* proyek ke sebuah host dari *JavaScript plugins* dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol. *Bootstrap* merupakan *framework* ataupun *tools* untuk membuat aplikasi web ataupun situs web responsive secara cepat, mudah dan gratis. *Bootstrap* terdiri dari CSS dan HTML untuk menghasilkan *grid, layout, typography, table, form, navigation*, dan lain-lain. Di dalam *Bootstrap* juga sudah terdapat *Jquery plugins* untuk menghasilkan komponen *user interface* yang cantik seperti *transitions, modal, dropdown, scrollspy, tooltip, tap, popover, alert, button, carousel* dan lain-lain. Dengan bantuan *bootstrap*, bisa membuat *responsive website* dengan cepat dan mudah dan dapat berjalan sempurna pada *browser-browser* populer seperti *chrome, firefox, safari, opera* dan *internet explorer* (Masuara dkk., 2015).

2.15 JavaScript

Javascript merupakan bahasa *script* yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti *internet Explorer (IE), Mozilla Firefox* dan lainnya. *Javascript* digunakan untuk membuat aplikasi *web*, sifatnya *client-side* sehingga dapat diolah langsung di *browser* tanpa harus terhubung ke *server* terlebih dahulu. Menurut Sibero dalam jurnal (Hidayat dkk., 2017), *Javascript* adalah bahasa skrip (*Scripting Language*), yaitu kumpulan instruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari

sistem operasi. *Javascript* merupakan modifikasi dari bahasa c++ dengan pola penulisan yang lebih sederhana (Hariadi, 2012).

Dari pengertian diatas dapat disimpulkan bahwa *javascript* adalah sekumpulan perintah modifikasi yang mengendalikan beberapa bagian sistem operasi seperti bahasa c++ dengan penulisan pola yang sederhana. Berikut adalah contoh *program javascript* untuk menampilkan notifikasi atau *alert*. operasi seperti.

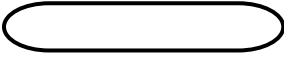

```
<script language="javascript">
  window.alert("ERROR! Kembali kehalaman awal");
</script>
```


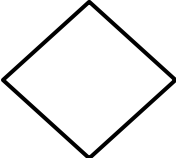

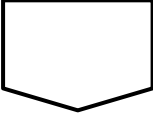

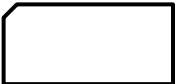

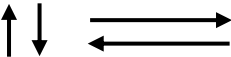

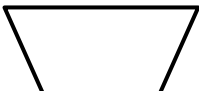

2.16 Flowchart




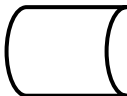

Flowchart adalah diagram alur yang dapat menggambarkan langkah, urutan dan keputusan untuk melakukan sebuah proses dalam suatu program. Langkah tersebut digambarkan dengan simbol-simbol tertentu dan saling dihubungkan menggunakan garis atau tanda panah. *Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antar suatu proses (intruksi) dengan proses lainnya dalam suatu program (Wibawanto, 2017). Diagram alur dapat menunjukkan secara jelas, arus pengendalian suatu algoritma yakni bagaimana melaksanakan suatu rangkaian kegiatan secara logis dan sistematis.

Dari pengertian diatas dapat disimpulkan bahwa *flowchart* adalah suatu diagram atau bagan yang menggambarkan urutan dari proses atau instruksi langkah-langkah secara mendetail dan sistematis.

Tabel 2. 3 Simbol-Simbol Flowchart

No.	Simbol	Nama	Fungsi
1.		Terminal	Menyatakan permulaan (start) atau akhir (end).
2.		<i>Input/Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya

No.	Simbol	Nama	Fungsi
3.		<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
4.		<i>Decision</i>	Menunjukkan suatu kondisi atau perbandingan tertentu yang akan menghasilkan dua kemungkinan jawaban: ya dan tidak.
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>Preparation</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal atau proses inisialisasi
8.		<i>Fatched Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu
9.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui Printer)
10.		<i>Flow</i>	Menyatakan jalannya arus suatu proses
11.		<i>Database</i>	Simbol penyimpanan ke <i>database</i> atau storage
12.		<i>Manual Operation</i>	Untuk menyatakan suatu tindakan yang tidak dilakukan oleh komputer
13.		<i>Keying Operation</i>	untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu

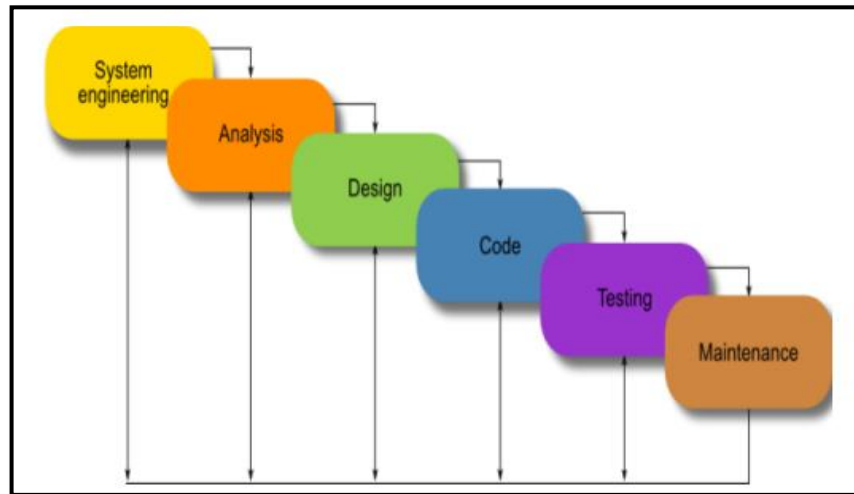
No.	Simbol	Nama	Fungsi
			mesin yang mempunyai <i>keyboard</i> .
14.		<i>Manual Input</i>	untuk memasukkan data secara manual dengan menggunakan online <i>keyboard</i> .
15.		<i>Multiple Document</i>	Sama seperti simbol document, hanya saja dalam simbol ini dokumen yang digunakan lebih dari satu
16.		<i>Predefined</i>	Untuk pelaksanaan suatu bagian (sub program) atau prosedur
17.		<i>On-Line Storage</i>	Menyatakan input yang berasal dari disk atau disimpan ke disk
18.		<i>Display</i>	Menyatakan perangkat output yang digunakan

2.17 Metode Pengembangan Sistem

Menurut (Nugraha dkk., 2018), Metode SDLC Waterfal Merupakan salah satu metode yang mempunyai ciri khas bahwa pengerjaan setiap fase harus dikerjakan terlebih dahulu sebelum melanjutkan ke fase berikutnya. Dengan demikian hasilnya akan fokus terhadap masing-masing fase sehingga pengerjaan dilakukan secara maksimal karena tidak adanya pengerjaan secara paralel. Adapun kelebihan menggunakan metode waterfall diantaranya:

1. Urutan proses pengerjaan menggunakan metode ini menjadi lebih teratur dari satu tahap ke tahap yang selanjutnya.
2. Dari sisi pengguna juga lebih menguntungkan karena dapat merencanakan dan menyiapkan seluruh kebutuhan data dan proses yang akan diperlukan
3. Jadwal menjadi lebih menentu karena jadwal setiap proses dapat ditentukan secara pasti. Sehingga dapat dilihat jelas target penyelesaian pengembangan

program. Dengan adanya urutan yang pasti, dapat dilihat pula progress untuk setiap tahap secara pasti.



Gambar 2. 4 Metode *Waterfall*

Berikut adalah tahapan-tahapan SDLC menurut (Dwanoko, 2016) sebagai berikut:

1. Tahap Analisis Kebutuhan Sistem

a. Analisis sistem (analyze Requirements)

Dengan memperoleh pengertian dari permasalahan, efisiensi, dan pertimbangan yang mengarah ke Rancang Bangun Sistem Pendukung Keputusan, serta mencari kendala yang dihadapi dalam sistem sehingga dapat menentukan solusi alternatif pendahuluan. Mempelajari konsep sistem dan permasalahan yang hendak diselesaikan.

b. Kegiatan wawancara dan Survei sistem (survey Requirement)

Kegiatan pada tahap survei pengumpulan data ini adalah mengumpulkan informasi dan data selengkap-lengkapunya. Selain melaksanakan survei pengumpulan data dan informasi, dilakukan pula proses wawancara. Proses wawancara diperlukan sebagai dasar untuk mengetahui kebutuhan aktor yang terlibat dalam sistem, kebutuhan fungsional mulai dari input, proses dan output pada sistem yang dibutuhkan.

c. Analisis terstruktur

Pada tahap ini tim pengembang akan menganalisis bahan masukan yang telah diperoleh dari hasil survei. Kemudian diolah menjadi sebuah dokumen

analisa terstruktur sebagai gambaran umum dari pembangunan aplikasi sistem pendukung keputusan yang akan buat atau dikembangkan.

d. Identifikasi masalah

Identifikasi masalah merupakan kegiatan mengidentifikasi masalah yang terjadi dalam sistem yang sedang berjalan. Dari hasil wawancara dan observasi akan menemukan beberapa identifikasi masalah yang ada pada pihak customer sehingga hasil dari indentifikasi masalah terbentuk sebuah dokumen daftar identifikasi masalah pada sistem yang akan dibuat atau dikembangkan.

2. Tahap Desain (Design Analize)

Tahap Desain, penetapan pembenahan, dan pengembangan sistem tahap ini meliputi presentasi awal, desain konseptual, desain basis data dan sistem, desain detail input/output Sistem Pendukung Keputusan. Dalam sistem desain fungsi desain dan operasi dijelaskan secara rinci, termasuk tata letak layar, aturan bisnis, diagram proses dan dokumentasi lainnya. Output dari tahap ini akan menjelaskan sistem yang baru sebagai koleksi modul atau subsistem. Tahap desain diperlukan sebagai masukan awal persyaratan identifikasi dalam dokumen persyaratan disetujui. Untuk setiap persyaratan, satu set satu atau lebih elemen desain akan diproduksi sebagai hasil dari wawancara, lokakarya, dan / atau upaya prototipe.

3. Tahap konstruksi (Design and Coding Program)

Tahap ini yaitu melakukan coding sistem yaitu menyusun bahasa pemrogram yang dipilih misalnya yang berbasis web dengan menggunakan HTML dan PHP, membuat dan mengkoneksi database dengan sistem, membuat form sistem dan pembuatan manual book. Disamping itu juga desain Grapical User Interface (GUI). Desain GUI ini dibutuhkan untuk mepresentasikan kebutuhan sistem yang dibuat dalam bentuk form-form mulai dari input, proses dan output, bentuk grafik statistik, bentuk simulasi atau bentuk visual lainnya. Dari GUI ini nantinya akan kelihatan sebuah sistem yang dibuat.

4. Tahap implementasi sistem (Implementation program)

Tahap ini meliputi menjalankan / implementasi program dan uji coba /evaluasi sistem, bertujuan untuk mengetahui hasil dari kebutuhan fungsional sistem yang

disusun dicocokkan dengan hasil implementasi dalam bentuk Rancang Bangun Sistem Pendukung Keputusan. Tahap Implementasi Sistem secara keseluruhan sebagai berikut :

- a. Perancangan Implementasi terdiri dari tugas implementasi, jadwal implementasi, perkiraan biaya, penanggung jawab.
 - b. Pengembangan dan uji coba software terdiri dari : Menentukan kebutuhan pemakai, Rencana pengembangan, Menulis program, Uji coba program, Pendokumentasian program, Meneliti pemakai, Installasi sistem
 - c. Penyiapan lokasi meliputi berbagai aktifitas dan penyiapan peralatan sampai dengan lokasi dimana sistem siap digunakan.
 - d. Memilih dan melatih pemakai yaitu pemakai sistem perlu dilatih agar dapat menggunakan dengan benar.
 - e. Dokumentasi
5. Tahap Tes Program (test unit, integrate and test system)
- Tahap ini melakukan uji coba sistem berdasarkan pengolahan data yang real/ nyata agar diperoleh hasil tes. Selain itu melakukan uji coba sistem dari segi konektifitas, fungsional sistem untuk mengetahui program bisa diaplikasikan dan mendapatkan evaluasi uji coba sistem.
6. Perawatan perangkat lunak (Maintenance)
- Tahap ini adalah tahapan terakhir dari Rancang Bangun Sistem Pendukung Keputusan yaitu mengevaluasi seluruh program dengan mengetahui data-data fungsional sistem berjalan sesuai rencana dan fungsional sistem yang belum jalan sebagaimana mestinya, tahap ini juga mengkaji untuk pengembangan berikutnya tentang Sistem Pendukung Keputusan yang dibuat. Maintenance bertujuan untuk memastikan bahwa sistem yang digunakan oleh pihak pengguna benar-benar telah stabil dan terbebas dari error dan bug. Sistem perlu dirawat karena beberapa hal yaitu:
- a. Sistem mengandung kesalahan yang dulunya belum terdeteksi, sehingga kesalahan sistem perlu di perbaiki.
 - b. Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.
 - c. Sistem perlu di tingkatkan.