

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 sistem pendukung keputusan dengan metode *Fuzzy Analytical Hierarchy Process* (AHP). Berikut adalah penelitian terdahulu beberapa jurnal terkait dengan penelitian yang dilakukan penulis:

Pada penelitian dengan judul **“PENERAPAN METODE FUZZY AHP (ANALITICAL HIERARCHY PROCESS) SEBAGAI SISTEM PENDUKUNG KEPUTUSAN DOSEN TERBAIK (STUDI KASUS : STMIK PRINGSEWU)”** oleh Afrizal Martin, Fahlul Rizki, Widiyanto, dan Alfina. Menjelaskan bahwa pengambilan keputusan perguruan tinggi masih banyak yang masih menggunakan cara-cara yang belum memiliki prinsip yang adil, sebagai contoh adalah dalam penerimaan Dosen yang tidak menggunakan seleksi terlebih dahulu dan tidak sesuai dengan kompetensinya, serta dalam kenaikan pangkat atau jabatan masih sering menggunakan cara solusi dan *nepotisme*. Permasalahan di atas dapat digolongkan dalam permasalahan MCDM (*Multi Criteria Decision Making*) karena melibatkan beberapa kriteria-kriteria dalam menentukan Dosen yang terbaik diantara sejumlah Dosen yang ada. AHP (*Analytical Hierarchy Process*) merupakan salah satu metode *Multi Criteria Decision Making* (MCDM) yang sangat baik dalam memodelkan pendapat para ahli dalam sistem pendukung keputusan. Sehingga hasil penelitian ini menggunakan metodologi *Fuzzy AHP* (*Analytical Hierarchy Process*) untuk pemilihan pemasok masalah, tetapi juga merilis tinjauan literatur yang komprehensif tentang masalah pengambilan keputusan multi kriteria. (Martin dkk., 2022).

Permasalahan dari penelitian **“PENERAPAN METODE ANALYTICAL HIERARCHY PROCESS DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN MAHASISWA BERPRESTASI”** oleh Agnia Eva Munthafa dan Husni Mubarak (2017) yaitu Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian Riset, Teknologi dan Pendidikan Tinggi selalu mengadakan kegiatan

Pemilihan Mahasiswa Berprestasi Tingkat Nasional setiap tahunnya. Sehingga setiap perguruan tinggi tentunya harus objektif, transparan, dan akurat yang disertai dengan bukti yang *valid* dalam memilih mahasiswa yang akan disertakan pada tingkat nasional. Namun, pelaksanaan ini dapat menimbulkan banyaknya pendaftar yang dapat menyebabkan sulitnya mengelola data dan nilai dalam menentukan mahasiswa berprestasi jika tanpa dibantu dengan sistem yang mumpuni. Perguruan tinggi dapat menggunakan konsep sistem pendukung keputusan sebagai alat bantu dalam menentukan mahasiswa berprestasi. Salah satu metode yang memiliki penghitungan nilai konsistensi dalam menentukan tingkat prioritas kriteria adalah metode *Analytical Hierarchy Process*.(Munthafa dan Mubarak, 2017).

Penelitian oleh Resti Lia Andharsaputri yang berjudul “**SISTEM PENDUKUNG KEPUTUSAN DENGAN PENERAPAN METODE AHP (ANALYTICAL HIERARCHY PROCESS) DALAM PEMILIHAN THE BEST TELESALLES**”. Membahas masalah mengenai *Telemarketing* dimaksudkan untuk memberikan motivasi yang baik kepada seluruh karyawan *Telesales* agar mereka bersemangat untuk bekerja sehingga kinerja mereka semakin baik demi kemajuan perusahaan. Permasalahannya adalah karena banyak calon kandidat yang pantas dalam sebuah tim untuk dijadikan *The Best* maka cukup membingungkan bagi manajemen untuk memilih orang yang tepat. Banyak sisi penilaian yang menjadi bahan pertimbangan. Dan semua hampir mendekati sama untuk *presentase* nilainya. Untuk pengambilan keputusan bagi permasalahan yang melibatkan multi pilihan dengan multi kriteria maka algoritma AHP dapat digunakan sebagai algoritma pengambilan keputusan untuk permasalahan multikriteria (*Multi Criteria Decision Making*). Metode AHP dan *Expert Choice* dipilih berdasarkan pertimbangan penggunaannya yang relatif mudah, serta untuk *Expert Choice* tersedia versi trial yang dapat digunakan secara bebas untuk jumlah kriteria yang telah ditentukan.(Andharsaputri, 2017).

Dalam penelitian oleh Erna dkk dengan judul “**PENERAPAN METODE ANALYTICAL HIERARCHY PROCESS (AHP) PADA SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN MAHASISWA BERPRESTASI MENGGUNAKAN FRAMEWORK LARAVEL (STUDI KASUS : INSTITUT SAINS & TEKNOLOGI AKPRIND YOGYAKARTA)**”. Terdapat

permasalahan dari pemilihan Mawapres yang dilakukan di Institut Sains & Teknologi AKPRIND Yogyakarta telah berjalan dengan baik setiap tahunnya, namun belum ada sistem yang mendukung baik untuk menampung maupun mengolah data pemilihan Mawapres yang ada. Metode yang diterapkan Dalam sistem pendukung keputusan ini adalah Metode *Analytical Hierarchy Process* (AHP). Dengan adanya sistem pendukung keputusan ini diharapkan kualitas keputusan meningkat, serta proses pengambilan keputusan menjadi lebih hemat, mudah dan cepat.(Nurnawati, 2015).

Permasalahan yang ada pada penelitian “**SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN MAHASISWA BERPRESTASI DENGAN METODE FUZZY MULTI-ATTRIBUTE DECISION MAKING SAW**” oleh Galih Arizza Candra Aditya Dimas Fatoni Perdana, Rawansyah dan Elly Setyo Astuti. Mengenai pengambilan keputusan juga sering dialami beberapa universitas yang mengadakan pemilihan mahasiswa berprestasi. Karena dalam pemilihan mahasiswa berprestasi memerlukan waktu *processing* data yang tidak sebentar. Disebabkan karena jumlah mahasiswa yang ada dalam suatu universitas serta kriteria harus benar – benar sesuai dengan ketentuan yang ditetapkan. Seperti IPK, peran organisasi, peran membuat karya ilmiah (PKM), sertifikasi, dan kemampuan berbahasa inggris dengan ketentuan dan bobot yang sudah ada. Dengan teknologi yang dapat diakses dengan cepat oleh juri melalui perangkat keras yang terkoneksi internet, perangkat *mobile* atau laptop maka akan mempercepat proses penilaian Dalam melakukan pemilihan mahasiswa berprestasi serta mempercepat hasil keputusan mahasiswa berprestasi secara cepat dan akurat.(Candra dkk., 2015).

2.1 Logika Fuzzy

Dalam jurnal (Abdillah, 2017), menurut saleh 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 kontrol. 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.

2.2 *Multiple Atribut Decision Making (MADM)*

Pada dasarnya, proses dari MADM dalam 3 *fase*, yakni : tahap persiapan komponen yang sesuai dengan situasi, analisis dan men-sintesis informasi (Abdillah, 2017). Ada beberapa metodologi yang digunakan dalam penyelesaian masalah FMADM :

1. *Simple Additive Weighting Method (SAW)*
2. *Weighted Product (WP)*
3. *ELECTRE*
4. *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*
5. *Analytic Hierarchy Process (AHP)*

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

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

Menurut memariani, dalam jurnal (Abdillah, 2017) alam 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. 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.

2.3 Analytical Hierarchy Process (AHP)

Menurut Syaifullah AHP (*Analytical Hierarchy Process*) merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hierarki.(Nurnawati, 2015).

Menurut Saaty dari jurnal (Nurnawati, 2015), langkah dalam pengambilan keputusan dengan metode AHP adalah sebagai berikut:

1. Membuat hierarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki dan menggabungkannya atau mensintesisnya.

2. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty, untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan tabel sebagai berikut.

Tabel 2. 1 Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan - pertimbangan yang berdekatan
Kebalikan	Jika aktivitas I mendapat satu angka dibandingkan dengan akitivitas j, maka j memiliki nilai kebalikannya dibandingkan dengan i

3. *Synthesis of priority* (Menentukan Prioritas)

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*Pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi *matriks* atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

Prosedur dalam metode AHP menurut Saaty dalam jurnal (Nurnawati, 2015)

meliputi :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan prioritas elemen
Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.
3. Sintesis
Pertimbangan-pertimbangan terhadap perbandingan berpasangan di sintesis untuk memperoleh keseluruhan prioritas.
4. Mengukur Konsistensi
Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah.

5. Hitung *Consistency Index* (CI) dengan rumus :

$$CI = (\lambda_{\max} - n) / n$$

Dimana n = banyaknya elemen.

6. Hitung Rasio Konsistensi/*Consistency Ratio* (CR) dengan rumus:

$$CR = CI/IR$$

Dimana CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Index Random Consistency*

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data *judgment* harus diperbaiki. Namun jika rasio konsistensi (CI/CR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar. Berikut daftar *Indeks Random Konsistensi* (IR).

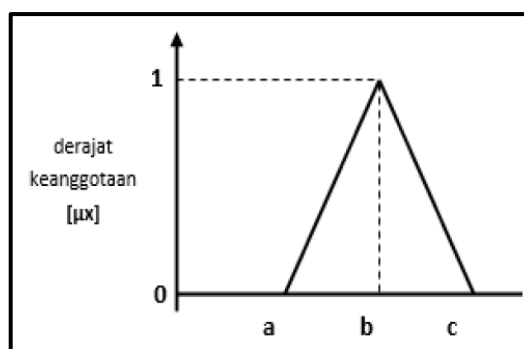
Tabel 2. 2 Daftar Indeks Random Konsistensi (IR)

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

2.4 Fuzzy Analytical Hierarchy Process

Menurut jurnal (Alwi, 2015) di dalam penerapan *Analytical Hierarchy Process* (AHP) untuk pengambilan keputusan dengan banyak kriteria yang bersifat subjektif, seringkali seorang pengambil keputusan dihadapkan pada suatu permasalahan yang sulit dalam penentuan bobot setiap kriteria. Untuk menangani kelemahan AHP ini, diperlukan suatu metode yang lebih memperhatikan keberadaan kriteria-kriteria yang bersifat subjektif tersebut. Model *Fuzzy-AHP* telah dikembangkan oleh beberapa peneliti. *Fuzzy-AHP* merupakan representasi dan eksistensi dari AHP Saaty dengan mengkombinasikannya dengan teori himpunan *fuzzy*. Di dalam *Fuzzy-AHP*, skala rasio *fuzzy* digunakan untuk mengindikasikan kekuatan relatif dari faktor-faktor pada kriteria atau subkriteria yang bersangkutan, sehingga sebuah matrik keputusan *fuzzy* dapat dibentuk. Nilai akhir dari alternatif-alternatif dapat juga disajikan dalam angka-angka *fuzzy*. Alternatif optimal diperoleh dengan meranking angka *fuzzy* menggunakan operator aljabar khusus. Konsep *fuzzy* yang dipakai dalam pengembangan *Fuzzy-AHP* ini adalah *Triangular Fuzzy Numbers* (TFN).

Secara umum prosedur perhitungan *Fuzzy-AHP* terdiri dari empat langkah, yaitu: (1) penilaian alternatif terhadap setiap kriteria, (2) pembobotan kriteria, (3) perhitungan nilai akhir, dan (4) ranking dan keputusan akhir. Pada langkah (1), yaitu penilaian alternatif, pengambil keputusan diminta memberikan suatu rangkaian penilaian terhadap alternatif x yang ada dalam bentuk bilangan *fuzzy triangular* (*triangular fuzzy number* (TFN)) yang disusun berdasarkan variabel linguistik. Selanjutnya nilai *fuzzy* didefinisikan bagi setiap alternatif pada setiap kriteria. Dalam TFN diberikan tiga kondisi untuk nilai fungsi keanggotaan.



Gambar 2. 1 Fungsi keanggotaan triangular (segitiga)

Dalam langkah (2), yaitu pembobotan kriteria, Zeleny (1983) membaginya menjadi dua tipe, yaitu: (1) bobot prior w_i , yang sifatnya relatif stabil, menggambarkan keadaan psikologis dan sosial dari pengambil keputusan, (2) bobot informasi λ_i yang sifatnya tidak stabil.

Pada model AHP orisinal, *pairwise comparison* menggunakan skala 1 — 9. Dengan mentransformasi *Triangular Fuzzy Number* terhadap skala AHP, maka skala yang digunakan adalah seperti pada tabel 2.3.

Tabel 2. 3 Tabel Fuzzifikasi skala AHP ke TFN

Skala AHP	Skala Fuzzy	Invers Skala Fuzzy
1	$\tilde{1} = (1,1,1)$ = jika diagonal $\tilde{1} = (1,1,3)$ = selainnya	$(1/3, 1/1, 1/1)$
3	$\tilde{3} = (1,3,5)$	$(1/5, 1/3, 1/1)$
5	$\tilde{5} = (3,5,7)$	$(1/7, 1/5, 1/3)$
7	$\tilde{7} = (5,7,9)$	$(1/9, 1/7, 1/5)$
9	$\tilde{9} = (7,9,9)$	$(1/9, 1/9, 1/7)$
2	$\tilde{2} = (1,2,4)$	$(1/4, 1/2, 1/1)$
4	$\tilde{4} = (2,4,6)$	$(1/6, 1/4, 1/2)$
6	$\tilde{6} = (4,6,8)$	$(1/8, 1/6, 1/4)$
8	$\tilde{8} = (6,8,9)$	$(1/9, 1/8, 1/6)$

Langkah penyelesaian F-AHP menurut Chang 1996 dalam jurnal (Alwi, 2015) adalah sebagai berikut :

1. Membuat struktur hirarki masalah yang akan diselesaikan dan menentukan perbandingan matriks berpasangan antar kriteria dengan skala TFN
2. Menentukan nilai situsis fuzzy (S_i) prioritas dengan rumus :

$$S_i = \sum_{j=1}^m M_i^j \times \frac{1}{\sum_{i=1}^n \sum_{j=1}^m M_i^j} \dots\dots\dots(1)$$

Dimana :

$$\sum_{j=1}^m M_i^j = \sum_{j=1}^m l_j, \sum_{j=1}^m m_j, \sum_{j=1}^m u_j \dots\dots\dots(2)$$

Sedangkan

$$\frac{1}{\sum_{i=1}^n \sum_{j=1}^m M_i^j} = \frac{1}{\sum_{i=1}^n u_i, \sum_{i=1}^n m_i, M_{i=1}^n l_i} \dots\dots\dots(3)$$

3. Menentukan nilai vector (v) dan nilai ordinat Defuzifikasi (d'). Jika hasil yang diperoleh pada setiap matriks *fuzzy*, $M_2 \geq M_1$ ($M_2 = (l_2, m_2, u_2)$ dan $M_1 = (l_1, m_1, u_1)$) maka nilai *vector* dapat dirumuskan sebagai berikut :

$$V(M_2 \geq M_1) = \sup [\mu M_1(x), \min (\mu M_2(y))]$$

Atau sama dengan grafik pada persamaan berikut :

$$V (M_2 \geq M_1) = \begin{cases} 1, & \text{if } M_2 \geq M_1 \\ 0, & \text{if } l_1 \geq \mu_2 \\ \frac{l_1 - \mu_2}{(m_2 - \mu_2) - (m_1 - l_2)}, & \text{lainnya} \end{cases}$$

Jika hasil nilai *fuzzy* lebih besar dari k, M_i ($i = 1, 2, \dots, k$) maka nilai *vector* dapat didefinisikan sebagai berikut :

$$V (M \geq M_1, M_2, \dots, M_k) = V (M \geq M_1) \text{ dan } V (M \geq M_2, M_2) \text{ dan}$$

$$V (M \geq M_k) = \min V (M \geq M_1) \dots\dots\dots(5)$$

Asumsikan bahwa :

$$d'(A_i) = \min V (S_i \geq S_k) \dots\dots\dots(6)$$

Untuk $k = 1, 2, \dots, n$; $k \neq i$, maka diperoleh nilai bobot *vector*

$$W' = (d'(A_1), d'(A_2), \dots, d'(A_n))^T \dots\dots\dots(7)$$

Dimana A_i ($i = 1, 2, \dots, n$) adalah n elemen keputusan

4. Normalisasi nilai bobot vector *fuzzy* (w)

Setelah dilakukan normalisasi dari persamaan (7) maka nilai bobot *vector* yang ternormalisasi adalah seperti rumus berikut :

$$W = (d(A_1), d(A_2), \dots, d(A_n))^T \dots\dots\dots(8)$$

Dimana W adalah bilangan non *fuzzy*.

2.5 Sistem Pendukung Keputusan

Menurut Abdul Khadir didalam jurnal (Martin dkk., 2022) Sistem Pendukung Keputusan atau *Decision Support System* (DSS) adalah pengambilan keputusan dilakukan pemimpin untuk menyelesaikan masalah yang dihadapinya dengan memulai satu alternatif pemecahan masalah terbaik dengan berdasarkan pertimbangan-pertimbangan (kriteria) tertentu. Pengambilan keputusan harus

dilakukan secara sistematis , kemudian mengumpulkan fakta-fakta, kemudian ada penentuan yang matang dari alternatif yang dihadapi, dan selanjutnya mengambil tindakan yang menurut perhitungan merupakan tindakan yang paling.

2.6 Aplikasi

Menurut Jogiyanto HM, 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 alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*.(Suhartini, 2017).

2.7 Website

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).

Halaman *website* biasanya berupa dokumen yang ditulis dalam format *HyperText Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para *user* atau pemakai melalui *web browser*.(Ibrahim dan Ambarita, 2018).

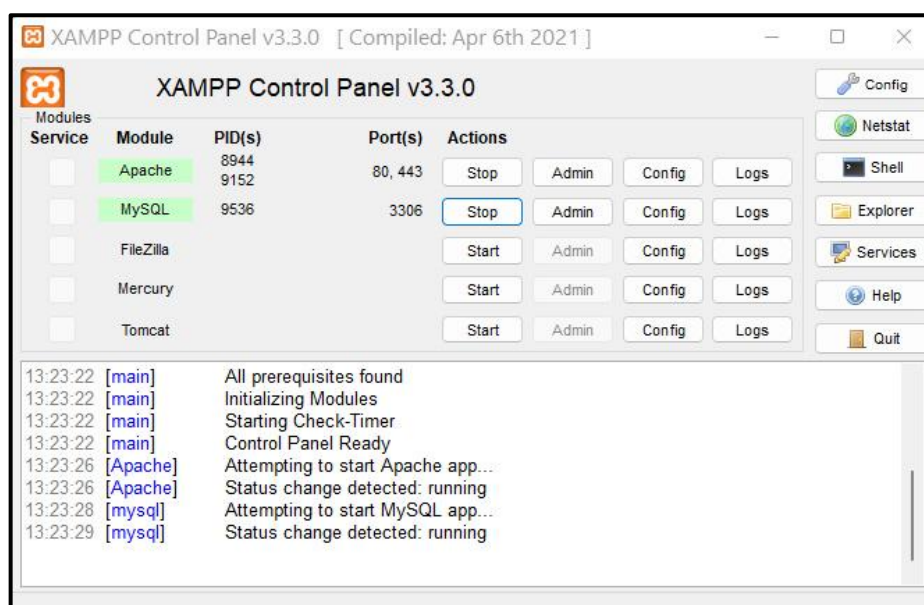
2.8 XAMPP

Menurut Jaelani dalam jurnal (Angelina, 2019), XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi 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*.

Menurut (Buana dan Setia, 2014) “XAMPP adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi

seperti *windows*, *linux*, *solaris*, dan *mac*". Bagian penting yang biasa digunakan dalam XAMPP pada umumnya :

1. XAMPP *Control Panel Application* berfungsi mengelola layanan (*service*) XAMPP. Seperti mengaktifkan layanan (*start*) dan menghentikan (*stop*) layanan. Dimana dalam hal ini untuk mengaktifkan program tersebut *control panel* akan menjadi langkah awalnya.
2. *htdocs* adalah folder tempat meletakkan berkas-berkas yang akan dijalankan dan diproses. Di *Windows*, folder ini akan berada di folder *xampp* dengan direktori dimana letak pengistalannya.
3. *phpMyAdmin* merupakan bagian untuk mengelola *database* yang akan digunakan.



Gambar 2. 2 Tampilan XAMPP

2.9 PHP MyAdmin

Perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL. PHPMyAdmin mendukung berbagai operasi MySQL diantaranya mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), *indeks*, pengguna (*users*), perijinan (*permissions*). (Barri dkk., 2015). Adapun beberapa menu yang ada di *php my admin*, yaitu :

1. *Structure*, untuk mengorganisir skema, tabel, kolom, baris.
2. *SQL*. Untuk menjalankan *query* atau kueri pada *database*.

3. *Search*. Untuk mencari kata-kata atau *value* didalam tabel *database*.
4. *Query*. SQL mendefinisikan seperangkat perintah, seperti *SELECT*, *INSERT*, *UPDATE*, *DELETE*, dan *CREATE TABLE*.
5. *Export*. Dapat meng-*export database* dengan format berbeda-beda seperti CSV, PDF, SQL, XML, dan *Text*.
6. *Import*. Dapat meng-*import database* dengan format berbeda-beda seperti *Open Document Spreadsheet*, CSV, SQL, *Mediawiki Table*, dan XML.
7. *Operations*. Ada beberapa jenis *operations* yang bisa eksekusi di seluruh *database* dan tabel terpisah.
8. *Triggers*. Sebagai objek basis data yang terhubung dengan tabel, dan akan diaktifkan. Misalnya *insert*, *update* atau *delete* terjadi pada tabel.

2.10 Database

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. (Fauzi dkk., 2021).

Model basis data 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 antar 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 basis data 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. 4 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 menunjukkan sebuah hubungan atau relasi antar entitas
Garis Penghubung	Garis penghubung adalah sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

Menurut jurnal (Wibowo dkk., 2019), *Database* didefinisikan sebagai kumpulan informasi yang terintegrasi, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Tujuan dari desain *database* adalah untuk menentukan data-data yang dibutuhkan dalam sistem sehingga informasi yang dihasilkan dapat terpenuhi dengan baik. Desain *database* perlu dilakukan untuk menghindari pengulangan data. Adapun hirarki *database* adalah sebagai berikut :

1. *Database* yaitu kumpulan dari beberapa *file*/tabel yang saling berhubungan antara *file* yang satu dengan *file* yang lain.
2. *File* yaitu kumpulan dari *record* yang saling berkaitan dan memiliki format *field* yang sama dan sejenis.
3. *Record* yaitu kumpulan dari *field* yang menggambarkan suatu unit data individu tertentu.
4. *Field* yaitu suatu atribut dari *record* yang menunjukkan suatu *item* dari data nilai *record* sebuah *field*.
5. *Byte* yaitu atribut dari *field* yang berupa karakter yang membentuk nilai dari sebuah *field*.

2.11 MySQL

Menurut jurnal (Puspitasari, 2016), MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet. MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *web* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya Menggunakan bahasa pemrograman *script* PHP.

Menurut Suryadi dan Zulaikhah dalam jurnal yang ditulis oleh (Catriwati, 2020), menjelaskan bahwa, “MySQL merupakan perangkat lunak sistem manajemen berbasis data SQL (*database management system*) atau DBMS yang *multi thread* dan *multi user*, dengan sekitar 6 juta instalasi di seluruh dunia”.

2.12 HTML (*Hyper Text Markup Language*)

Menurut Suryatiningsih dalam jurnal yang ditulis oleh (Fauzi dkk., 2021) “*HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk membuat suatu situs *web* atau *homepage*. Setiap dokumen dalam *web* ditulis dalam format HTML. Semua format dokumen, *hyperlink* yang dapat diklik, gambar, dokumen multimedia *form* yang dapat diisi dan sebagainya didasarkan atas HTML.

Menurut (Khozaimi, 2020), *HyperText Markup Language* (HTML) merupakan standar 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*

Tabel 2. 5 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 <i>web</i> .

<code><body>...</body></code>	Berisi <i>tag-tag</i> dan elemen untuk isi konten dan <i>layout</i> tampilan halaman atau dokumen HTML yang biasa terlihat.
<code><header>...</header></code>	Digunakan untuk menampilkan bagian kepala (kop) atau bagian <i>header</i> (atas), tag ini hanya bisa digunakan pada HTML 5.
<code><footer>...</footer></code>	Digunakan untuk memberikan bagian kaki pada sebuah dokumen HTML, biasanya digunakan untuk informasi atau hak cipta

2.13 PHP (*Personal Hypertext Preprocessor*)

Merupakan suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, *JavaScript*, *Jquery*, *Ajax*. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML. Dengan menggunakan PHP anda bisa membuat *website* yang dinamis dengan disertai manajemen *database*-nya. (Priyanto dan Kawistara, 2017).

Penulisan sintak PHP dituliskan di antara tanda `<?php` dan `?>` yang memisahkan *script* PHP dengan *script* lainnya. Satu *file* PHP dapat berisi *full script* PHP atau dapat disisipkan antara *script* lainnya seperti HTML, CSS maupun *JavaScript*. Namun, *file* yang berisi *script* PHP wajib disimpan dengan ekstensi `*.php` dan disimpan di *server* (folder *htdocs* atau *www*). Jika disimpan dengan ekstensi HTML atau disimpan disembarang tempat maka *script* PHP tidak diproses sebagaimana mestinya. Berikut contoh penulisa *script* PHP di dalam *script* HTML. (Abdullah, 2018).

```

<!DOCTYPE html>
<html>
  <head> <title></title> </head>
  <body>
    <?php
      echo "";
    ?>
  </body>
</html>

```


2.14 CSS (*Cascading Style Sheets*)

Cascading Style Sheets (CSS) adalah suatu bahasa *style sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh *World Wide Web Consortium* (W3C). (Puspitasari, 2016).

Menurut (Lewenusa, 2020) cara penulisan skrip css dibagi menjadi 3 bagian, yaitu:

1. *Inline Style Sheet*

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

```
<body>
  <!--isi konten dan layout tampilan -->
  <p style="color : green"> Warna tulisan hijau </p>
</body>
```

2. *Embedded Style Sheet*

Penulisan *script* css didalam dokumen HTML. Metode ini menggunakan tag awal <style> dan tag penutup </style>.

Contoh :

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

3. *Linked Style Sheet/External Style Sheet*

Penulisan *script* css dihalaman yang berbeda atau terpisah dari HTML. Jadi dapat 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 disetiap halaman.

- b. Dapat mengubah tampilan beberapa halaman dengan mengubah hanya pada *style sheet*.
- 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>
```

2.15 *Bootstrap*

Menurut (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 *icon* 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, dan navigation*. Di dalam *Bootstrap* 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*, dapat 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).

2.16 *Javascript*

Menurut (Sahi, 2020) *Javascript* adalah bahasa yang berupa kumpulan *script*, yang tugasnya dijalankan dalam dokumen HTML. Bahasa ini adalah bahasa *script* pertama di *web*. Bahasa ini merupakan bahasa pemrograman yang memberikan fungsionalitas tambahan pada HTML dengan memungkinkan eksekusi perintah di sisi pengguna, yaitu di sisi *browser*, bukan *server web*. *JavaScript* adalah bahasa untuk membuat program yang digunakan untuk membuat dokumen HTML yang digunakan, yang ditampilkan di *browser* lebih interaktif.

JavaScript menyediakan beberapa fungsionalitas ke halaman *web*, sehingga bisa menjadi program yang disajikan melalui antarmuka *web*. *JavaScript* adalah bahasa yang tidak memerlukan kompiler, cukup dengan interpreter.(Hermiati dkk., 2021).

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

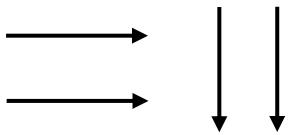
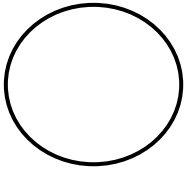
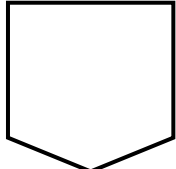


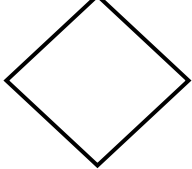

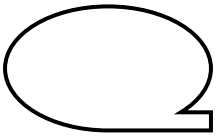
2.17 *Visual Studio Code*

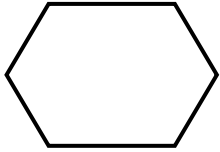

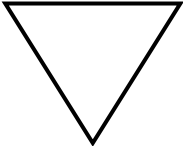


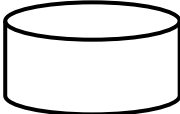



Visual Studio Code adalah kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *macOS*. Ini termasuk dukungan untuk *debugging*, *control git* yang tertanam dan *GitHub*, penyorotan *syntax*, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan. (Agustini dan Kurniawan, 2020).

2.18 *Flowchart*

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau Langkah-langkah instruksi yang berurutan dalam sistem. *Flowchart* digambarkan dengan menggunakan simbol-simbol setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. *Flowchart* dibuat dengan tujuan untuk menunjukkan setiap proses yang harus dilalui dalam suatu sistem.(Rosaly dan Prasetyo, 2019). *Flowchart* atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. *Flowchart* dibuat dengan tujuan untuk menunjukkan setiap proses yang harus dilalui dalam suatu sistem, *flowchart* memiliki peran penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus, seperti yang ditunjukkan pada tabel berikut.

Tabel 2. 6 Simbol-Simbol Flowchart

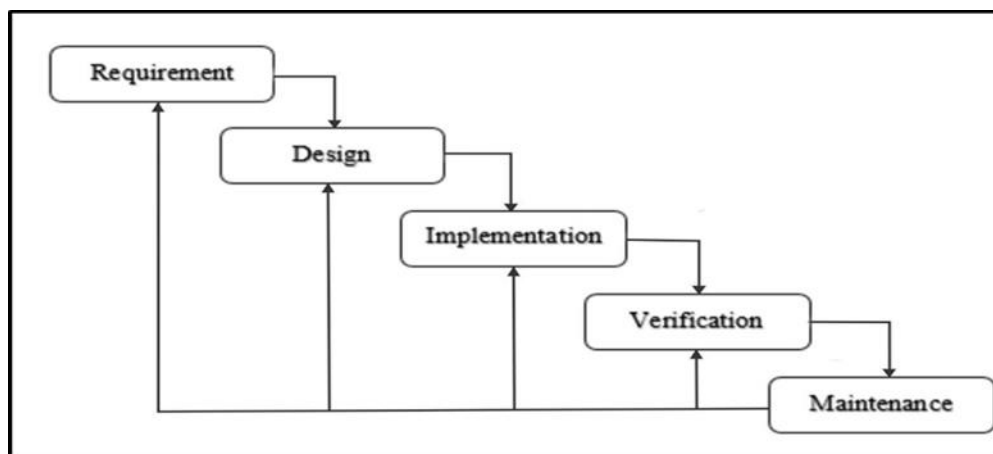
NO	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses.
2		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
3		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
4		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
5		Simbol manual, berfungsi untuk menyatakan suatu tindakan yang tidak dilakukan oleh komputer.
6		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak.
7		Simbol <i>terminal</i> , berfungsi untuk menyatakan permulaan atau akhir suatu program.
8		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis

9		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
10		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
11		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
12		Simbol manual <i>input</i> , berfungsi untuk memasukkan data secara manual dengan menggunakan online <i>keyboard</i> .
13		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
14		Simbol <i>magnetic disk</i> , berfungsi untuk untuk <i>input</i> atau <i>output</i> yang menggunakan <i>disk magnetic</i> .
15		Simbol <i>disk storage</i> , berfungsi untuk menyatakan input berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
16		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i>).
17		Simbol <i>punched card</i> , berfungsi untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.

2.19 Metode Pengembangan Sistem

Metode *Waterfall* adalah salah satu model SDLC (*System Development Life Cycle*) yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model *Waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *Waterfall* antara lain *requirement*, *design*, *implementation*, *verification*, dan *maintenance* (Wahid, 2020). Kelebihan menggunakan metode *Waterfall* adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang diperlukan juga mahal. 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 user 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. Dengan adanya urutan yang pasti, dapat dilihat pula progress untuk setiap tahap secara pasti.



Gambar 2. 3 Metode *Waterfall*

2.19.1 Requirement

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei

langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2.19.2 Design

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

2.19.3 Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

2.19.4 Verification

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian.

2.19.5 Maintenance

Ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

2.20 Blackbox Testing

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