

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini memiliki beberapa penelitian yang berkaitan dengan penerapan sistem pendukung keputusan pemilihan siswa berprestasi metode *Fuzzy Tsukamoto* dapat dilihat pada Tabel 2.1:

Tabel 2. 1 Tahapan Penelitian

No.	Nama (Tahun)	Topik	Metode	Hasil
1	Zulfahmi, (2019)	Sistem Pendukung Keputusan Pemilihan Siswa Berprestasi	Menggunakan Metode <i>Profile Matching</i>	menyimpulkan bahwa aplikasi ini efektif dalam meningkatkan mutu pendidikan. Dengan menggunakan parameter pengetahuan, keterampilan, dan sikap, aplikasi ini mampu memberikan nilai akhir siswa dengan tingkat keberhasilan tertinggi mencapai 8.893.
2	Fitriana dkk. (2018)	Sistem Pendukung Keputusan Pemilihan Mahasiswa berprestasi	Menggunakan Metode <i>Profile Matching</i>	Hasilnya menunjukkan bahwa sistem merespons dengan benar terhadap input data yang valid dan menangani kesalahan input dengan baik. Saat diuji oleh responden R01, R02, dan R03, sistem berhasil menghasilkan

No.	Nama (Tahun)	Topik	Metode	Hasil
				output yang sesuai dengan variabel indeks prestasi kumulatif, karya tulis ilmiah, prestasi yang diunggulkan, bahasa Inggris, dan kepribadian, dengan nilai tertinggi 5,75.
3	Syafik dkk (2015)	Sistem Pendukung Keputusan Pemilihan Siswa Berprestasi	Menggunakan Metode <i>Fuzzy Multi Attribute Decision Making</i>	Penelitian ini mengembangkan sistem untuk membantu pengguna dalam pengolahan data siswa, termasuk pemilihan siswa berprestasi. Metode FMADM digunakan untuk perhitungan pemilihan. Tahapan pengembangan meliputi perumusan masalah, analisis sistem, perancangan, pengujian, dan implementasi. Hasil perhitungan menunjukkan nilai tertinggi 1,00 dengan variabel rata-rata nilai per semester, penghargaan, keaktifan organisasi, dan poin tingkah laku.

No.	Nama (Tahun)	Topik	Metode	Hasil
4	Adib dkk (2020)	Sistem Pendukung Keputusan untuk Menentukan Jumlah Produksi Opak Gambir.	Menggunakan Metode <i>Fuzzy Tsukamoto</i>	Penggunaan algoritma Fuzzy Tsukamoto untuk menentukan jumlah produksi opak gambir lebih efektif daripada sistem manual yang digunakan saat ini. Metode ini menggunakan 3 variabel input: persediaan barang, permintaan, dan output, serta mengikuti tahapan fuzzifikasi, aturan, dan defuzzifikasi. Dengan menerapkan Metode Fuzzy Tsukamoto, estimasi jumlah produksi opak gambir berdasarkan permintaan dan persediaan dianggap lebih efektif dalam mengurangi kerugian, dengan nilai persediaan barang mencapai 4104.
5	Satria & Sibarani, (2020)	Pemilihan Karyawan Terbaik Berbasis <i>Java Desktop</i> .	Menggunakan Metode <i>Fuzzy Tsukamoto</i>	Penilaian dilakukan dengan mengambil data dari mesin absensi dan pengamatan secara visual. Untuk kriteria kepribadian dan problem solving

No.	Nama (Tahun)	Topik	Metode	Hasil
				<p>sering diabaikan karena nilai yang bersifat subjektif. Seluruh perhitungan dilakukan secara manual. Dan hasil pengujian, seluruh fungsi berjalan dengan baik dan diperoleh hasil karyawan yang memperoleh nilai tertinggi dengan status “TERBAIK” memperoleh nilai probabilitas 90,85586.</p>

Penelitian yang dilakukan oleh Zulfahmi (2019), menerapkan metode *Profile Matching* pada sistem pendukung keputusan pemilihan siswa berprestasi. Penelitian ini merupakan penelitian yang menggunakan metode pengembangan sistem *System Development Life Cycle (SDLC) Waterfall*. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pendukung keputusan dengan metode *profile matching*. Parameter dalam penelitian ini menggunakan aspek nilai pengetahuan, aspek nilai keterampilan, aspek nilai sikap untuk pemilihan siswa berprestasi. Berdasarkan dari penelitian yang dilakukan, dapat disimpulkan bahwa sebuah aplikasi sistem pendukung keputusan menggunakan metode *profile matching* untuk seleksi siswa berprestasi dapat dianggap sebagai salah satu metode untuk mengukur tingkat keberhasilan proses belajar mengajar di sekolah dengan nilai akhir paling tinggi 8.893. Hal ini memungkinkan pihak sekolah untuk melakukan evaluasi dan menentukan langkah-langkah perbaikan yang diperlukan guna meningkatkan mutu pendidikan.

Penelitian dari Fitriana dkk. (2018), menerapkan metode *Profile Matching* pada sistem pendukung keputusan pemilihan mahasiswa berprestasi pada Fakultas

Teknik UNTAN . Penelitian ini menggunakan pengujian *black box* untuk menguji respons sistem terhadap input data, dengan hasil yang menunjukkan bahwa sistem merespons dengan benar terhadap input data yang valid dan menangani kesalahan input dengan menampilkan pesan *error*. Saat diuji oleh responden R01, R02, dan R03, sistem berhasil menghasilkan output yang sesuai dengan variabel indeks prestasi kumulatif, karya tulis ilmiah, prestasi yang diunggulkan, bahasa Inggris, dan kepribadian dengan nilai tertinggi yang didapat 5,75. Selain itu, pengujian UAT juga dilakukan dan menunjukkan bahwa sistem Pendukung Keputusan (SPK) Pemilihan mahasiswa berprestasi memberikan keputusan yang membantu, meskipun masih memerlukan peningkatan untuk meningkatkan kinerjanya. Penggunaan *dummy* data digunakan untuk melakukan simulasi perhitungan pada *profile matching*. Hasil pengujian *black box* dan UAT menunjukkan bahwa sistem mampu memberikan ranking yang sesuai dengan proses pemilihan mahasiswa berprestasi. Dari penelitian ini, dapat disimpulkan bahwa SPK dapat memberikan kontribusi yang signifikan dalam proses pengambilan keputusan pemilihan mahasiswa berprestasi dan meningkatkan efektivitas serta fleksibilitas dalam penggunaannya.

Penelitian yang dikembangkan oleh Syafik dkk., (2015), menerapkan metode *Fuzzy Multi Attribute Decision Making* pada sistem pendukung keputusan pemilihan siswa berprestasi. Tujuan sistem adalah untuk membantu pengguna mengolah data siswa, termasuk pemilihan siswa berprestasi, hasil seleksi, dan laporan-laporan. Metode *Fuzzy Multiple Attribute Decision Making* (FMADM) digunakan untuk perhitungan sistem untuk melakukan pemilihan siswa berprestasi. Tahap-tahap proses pengembangan sistem dalam penelitian ini adalah perumusan masalah, analisis sistem, perancangan, pengujian, dan implementasi. Hasil perhitungan sistem dengan variabel rata-rata nilai persemester, penghargaan, keaktifan organisasi dan ekstrakurikuler, poin tingkah laku dan diperoleh hasil dengan nilai tertinggi 1,00.

Penelitian dari Adib dkk (2020), dengan menerapkan metode *Fuzzy Tsukamoto* dalam Sistem Pendukung Keputusan untuk Menentukan Jumlah Produksi Opak Gambir. Peneliti memanfaatkan metode sintesis dalam pengumpulan data, serta melakukan wawancara dengan salah satu pemilik usaha

UKM yang bergerak di bidang produksi opak gambar. Untuk teknik pengolahan data, peneliti menggunakan aplikasi *Microsoft Excel*. Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa penggunaan algoritma *Fuzzy Tsukamoto* untuk menentukan jumlah produksi opak gambar lebih efektif daripada sistem manual yang digunakan saat ini. Metode ini menggunakan 3 variabel input, yaitu persediaan barang, permintaan, dan output, serta mengikuti tahapan *fuzzifikasi*, aturan, dan *defuzzifikasi* untuk menghasilkan hasil yang spesifik. Dengan menerapkan Metode *Fuzzy Tsukamoto*, estimasi jumlah produksi opak gambar berdasarkan permintaan dan persediaan dianggap lebih efektif dalam mengurangi kerugian dengan nilai persediaan barang 4104.

Penelitian yang dilakukan oleh Satria & Sibarani (2020), dengan menerapkan Metode *Fuzzy Tsukamoto* untuk pemilihan karyawan terbaik berbasis java desktop. hasil analisis dan desain sistem pemilihan karyawan terbaik menggunakan metode *Fuzzy Tsukamoto*, maka dapat diambil kesimpulan bahwa *fuzzy* dapat digunakan untuk memilih karyawan terbaik berdasarkan data karyawan berdasarkan tiga kriteria, yaitu: absensi, kepribadian, dan penyelesaian masalah. Hasil perhitungan bergantung pada absensi, kepribadian, dan penyelesaian masalah, serta rule yang digunakan dalam *fuzzy*. Hasil perhitungan dapat digunakan sebagai rekomendasi untuk membantu perusahaan dalam menentukan karyawan berkinerja terbaik dengan nilai 90,85586.

2.2 Dasar Teori

Dasar teori merupakan kumpulan definisi, konsep, dan proposisi yang tersusun secara terstruktur mengenai variabel-variabel yang relevan dalam suatu penelitian. Oleh karena itu, pembuatan dasar teori secara baik dan benar dalam sebuah penelitian menjadi salah satu hal yang sangat penting, karena dasar teori akan menjadi sebuah fondasi dan landasan dalam sebuah penelitian itu sendiri.

2.2.1 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditunjukkan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat (Daihani., 2001) :

1. Terstruktur, yaitu berhubungan dengan persoalan yang telah diketahui sebelumnya dengan penyelesaian standar aturan yang telah ditentukan.
2. Semi terstruktur, yaitu berhubungan dengan persoalan yang belum diketahui sebelumnya, dengan parameter yang sudah ada.
3. Tidak terstruktur, yaitu berhubungan dengan persoalan baru yang cukup pelik, karena banyaknya data yang belum diketahui.

2.2.1.1 Proses Pengambilan Keputusan

Menurut Simon (1960), proses pengambilan keputusan memiliki tiga fase, yaitu inteligensi, perancangan, dan pemilihan. Namun kemudian ditambahkan fase keempat yaitu implementasi (Zulita, 2013).

1. Fase Penelusuran (*Intelligence*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. Fase Perancangan (*Design*)

Pada tahap ini dilakukan dengan melakukan perancangan seperti: perancangan fitur, menu aplikasi, perancangan data, perancangan arsitektur, perancangan interface dan perancangan prosedur.

3. Fase Pemilihan (*Choice*)

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

4. Fase Implementasi (*Implementation*)

Setelah itu menentukan modelnya, tahap ini selanjutnya adalah proses mengimplementasikannya dalam aplikasi pengambilan keputusan.

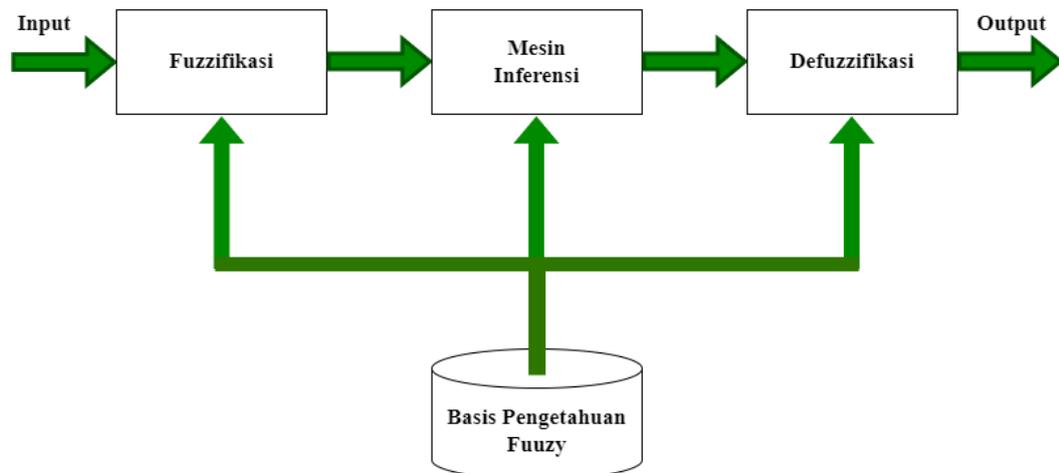
2.2.2 Logika Fuzzy

Prof. Lotfi A. Zadeh merupakan peneliti yang memperkenalkan Logika *fuzzy* pada tahun 1965. Dasar dari logika *fuzzy* diambil teori himpunan *fuzzy* (Khoirun dkk 2020). Pada teori ini, derajat keanggotaan merupakan faktor penting dalam penentuan keberadaan elemen pada suatu himpunan. Nilai pada keanggotaan (*membership function*) disebut-sebut menjadi ciri utama dari penalaran dengan logika *fuzzy*.

2.2.2.1 Cara Kerja Logika Fuzzy

Berikut merupakan cara kerja Logika Fuzzy dan dapat dilihat pada Gambar

2.2.



Gambar 2. 1 Cara Kerja Logika Fuzzy

Sumber : Nisa Khoirun dkk, (2020)

a. *Fuzzifikasi*

Menggambarkan langkah/tahapan pengubahan suatu inputan dari bentuk bukan *fuzzy (crisp)* menjadi variabel *linguistic* (kata-kata alami) yang dipetakan kebentuk himpunan *fuzzy* sesuai fungsi keanggotaannya masing-masing dan disimpan dalam basis pengetahuan *fuzzy*.

b. Pembentukan Basis Aturan *Fuzzy*

Kumpulan perintah-perintah (rules base) yang ditampilkan dengan pernyataan **IF...THEN...**

c. *Inference System*

Menggambarkan langkah dari *fuzzy* untuk pengambilan keputusan yang didasarkan aturan pada rule base guna menghubungkan antara perubah masukan *fuzzy* dan perubah keluaran *fuzzy*.

d. *Defuzzifikasi*

Tahapan pengubahan nilai *fuzzy* yang dipetakan kedalam bentuk himpunan keluaran (*output*) serta fungsi keanggotaannya guna mendapatkan bentuk tegas (*crisp*). Perihal ini dilakukan karena plant hanya mengetahui nilai tegas sebagai nilai sebenarnya untuk regulasi prosesnya. langkah ini juga berguna untuk menentukan nilai tegas dari data keluaran.

Berikut adalah istilah umum yang digunakan pada Sistem *Fuzzy* :

1. *Variable Fuzzy*

Variable fuzzy merupakan variabel yang akan diperhitungkan pada suatu sistem *fuzzy*. Contoh: permintaan, temperature, kecepatan, dsb.

2. Himpunan *Fuzzy*

Menggambarkan kumpulan kelompok yang mengindikasikan suatu kondisi atau keadaan tertentu pada suatu variabel *fuzzy*. Contoh : variabel kehadiran terdiri dari 3 himpunan, yaitu tidak aktif, aktif, dan sangat aktif.

3. Semesta Pembicaraan

Merupakan kelompok keseluruhan nilai yang digunakan dalam pengoperasian pada variabel *fuzzy*. Contoh: semesta pembicaraan untuk kehadiran $[0,10]$.

4. Domain (*Scope*)

Merupakan kumpulan nilai-nilai yang digunakan dalam semesta pembicaraan dan dioperasikan dalam suatu himpunan *fuzzy*. Contoh : tidak aktif = $[0, 3]$, aktif = $[4, 7]$, sangat aktif = $[8, 10]$.

5. Derajat Keanggotaan

Peranan dari derajat keanggotaan adalah memberikan bobot pada suatu input sehingga dapat dinyatakan dengan nilai. Misalnya kehadiran adalah tidak aktif, dengan menggunakan derajat keanggotaan maka kehadiran tidak aktif dapat mempunyai suatu nilai misal 3, batas dari derajat keanggotaan dari 0 – 5.

6. *Crisp Input*

Nilai tegas dari suatu masukan yang kita berikan untuk mencari derajat keanggotaan.

2.2.2.2 Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu group yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Menurut T.Sutojo,dkk pada tahun 2011, Untuk memahami logika fuzzy, sebelumnya perhatikan dahulu tentang konsep himpunan *fuzzy* (Nasir & Suprianto, 2017). Himpunan fuzzy memiliki 2 atribut yaitu :

1. Linguistik, yaitu penamaan suatu grup menggunakan bahasa sehari-hari, misal : kuat, lemah, sangat tinggi, panas.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti 40, 25, 89 dsb.

2.2.2.3 Fungsi Keanggotaan Fuzzy

Fungsi Keanggotaan *Fuzzy* Biasa disebut juga derajat keanggotaan merupakan sekumpulan nilai yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaan dengan rentang nilai 0 hingga 1 (Mujilahwati & Wardhani, 2021). Nilai suatu keanggotaan dapat diperoleh menggunakan pendekatan fungsi. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Salah satu pendekatan fungsi keanggotaan dalam *fuzzy* yang akan dipakai adalah representasi linear. Pada representasi linear, ini memetakan masukan kedalam derajat keanggotaan yang digambarkan dengan garis lurus/liner.

Terdapat 2 kondisi himpunan *fuzzy* yang linear. Pertama, linear naik digambarkan dari domain nilai 0 bergerak kekanan menuju domain nilai 1. Berikut merupakan fungsi keanggotaan representasi linear naik, ditunjukkan pada persamaan berikut ini :

$$\mu(x) = \begin{cases} 0 & ; x \leq a \\ \frac{x-a}{b-a} & ; a < x < b \\ 1 & ; x \geq b \end{cases}$$

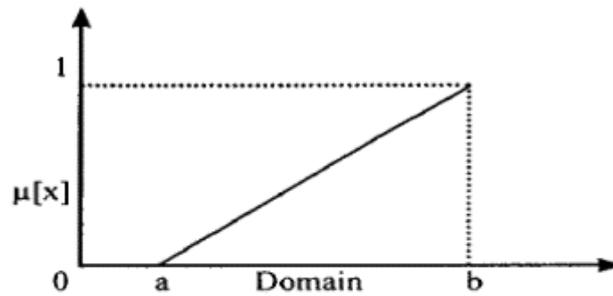
Dimana :

μ : Nilai Crips Keanggotaan *Fuzzy*

x : variabel data

a : Nilai Minimum

b : Nilai Maksimum



Gambar 2.2 Representasi Linear Naik
Sumber : Mujilahwati & Wardhani, (2021)

Kedua, linear turun garis digambarkan dari domain nilai 1 bergerak kekiri menuju domain nilai 0. Fungsi keanggotaan linear turun ditunjukkan pada Gambar berikut ini :

$$\mu(x) = \begin{cases} 1 & ; x \leq \alpha \\ \frac{x-\alpha}{b-\alpha} & ; \alpha < x < b \\ 0 & ; x \geq b \end{cases}$$

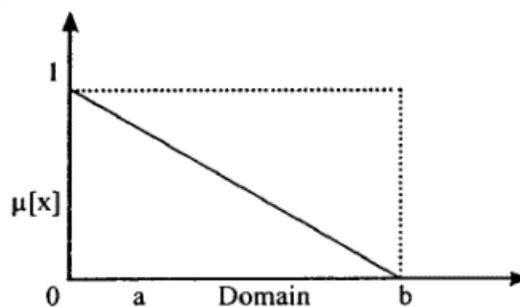
Dimana :

μ : Nilai Crips Keanggotaan *Fuzzy*

x : variabel data

a : Nilai Minimum

b : Nilai Maksimum



Gambar 2.3 Reprantasi Linear Turun
Sumber : Mujilahwati & Wardhani, (2021)

Selain dua grafik linear diatas, himpunan *fuzzy* juga bisa berbentuk kurva segitiga. Kurva ini pada dasarnya merupakan gabungan antara 2 garis linear seperti terlihat pada gambar di bawah ini:

$$\mu(x) = \begin{cases} 0 & ; x \leq \alpha \text{ atau } x \geq c \\ (x-\alpha)/(b-\alpha) & ; \alpha < x < b \\ (b-x)/(c-b) & ; x \geq b \end{cases}$$

Dimana :

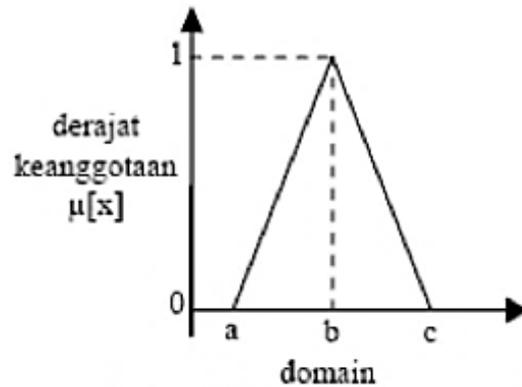
μ : Nilai Crips Keanggotaan *Fuzzy*

x : variabel data

a : Nilai Minimum

b : Nilai Maksimum

c : Nilai Domain Terbesar



Gambar 2. 4 Reprntasi Linear Segitiga
Sumber : Mujilawati & Wardhani, (2021)

2.2.3 *Fuzzy Tsukamoto*

Menurut Widaningsih (2017) dalam ini pengambilan keputusan dapat menggunakan konsep *fuzzy* (samar). Konsep logika *fuzzy* ini dipilih karena mudah dimengerti. Dan konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana. Logika *fuzzy* sangat fleksibel, *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat. Logika *fuzzy* dapat mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks. Dan metode yang akan digunakan dalam pengambilan keputusan untuk menentukan Siswa Berprestasi adalah metode *Tsukamoto*.

Pada Metode *Tsukamoto* menggunakan himpunan *fuzzy* untuk menunjukkan setiap konsekuensi dari aturan **IF-Then**. dengan fungsi keanggotaan yang sama sekali tidak berubah. Oleh karena itu, hasil inferensi dari tiap aturan diberikan secara tegas (*crisp*) berdasarkan α - predikat (kekuatan ledakan). Hasil terakhir didapat dengan menggunakan rata-rata terbobot.

Pada Metode *Tsukamoto* menggunakan himpunan *fuzzy* untuk menunjukkan setiap konsekuensi dari aturan **IF-Then**. dengan fungsi keanggotaan yang sama

sekali tidak berubah. Oleh karena itu, hasil inferensi dari tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (kekuatan ledakan). Hasil terakhir didapat dengan menggunakan rata-rata terbobot.

IF x is A_1 **and** y is B_2 **THEN** z is C_1

IF x is A_2 **and** y is B_2 **THEN** z is C_1

@predikat untuk aturan pertama dan kedua, masing-masing adalah a_1 dan a_2 , diperoleh nilai Z_1 pada aturan pertama dan Z_2 pada aturan kedua dengan menggunakan penalaran monoton. Terakhir, menggunakan aturan terbobot, diperoleh hasil akhir dengan formula berikut:

$$z = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}$$

Menurut Ilmiyah & Resti, (2022) langkah-langkah penyelesaian dalam inferensi Fuzzy Tsukamoto yaitu:

- 1) Mengidentifikasi kasus.
- 2) Mengidentifikasi jenis variabel input dan variabel output beserta himpunan-himpunan *Fuzzy* yang terkait
- 3) Menyusun grafik dan fungsi keanggotaan untuk masing-masing variabel input dan variabel output
- 4) Mencari derajat keanggotaan untuk setiap variabel input dalam himpunan-himpunan *Fuzzy* yang terkait
- 5) Mengkronstruksikan aturan *Fuzzy* (*Fuzzy Rules*)
- 6) Menentukan Fire Strength (α – predikat) dan nilai *crisp* hasil inferensi (z_n) untuk setiap *Fuzzy Rules*
- 7) Menghitung rata-rata nilai *crisp* hasil inferensi (z)

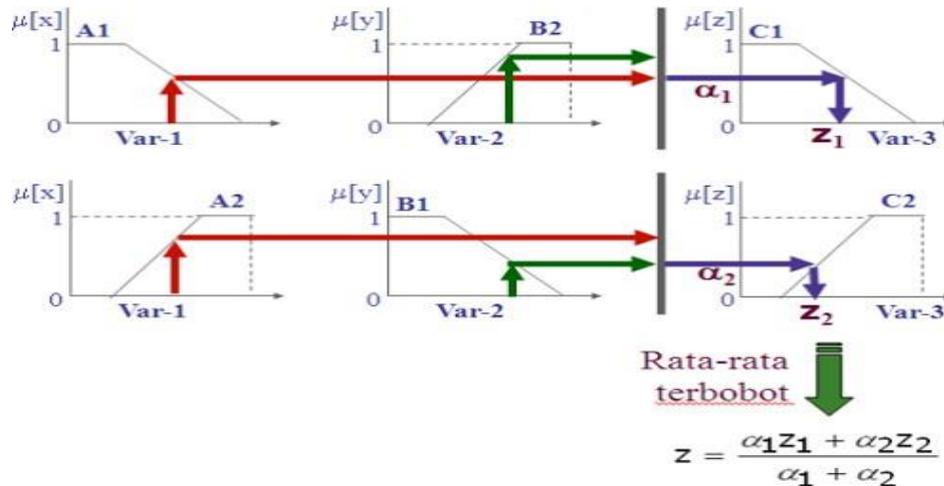
Nilai z sebagai hasil akhir dihitung menggunakan rata-rata terbobot dengan rumus:

$$z = \frac{(\alpha_{predikat_1} * z_1) + (\alpha_{predikat_2} * z_2) + (\alpha_{predikat_3} * z_3) + \dots + \alpha_{predikat_n} * z_n}{\alpha_{predikat_1} + \alpha_{predikat_2} + \alpha_{predikat_3} + \dots + \alpha_{predikat_n}}$$

dimana :

apredikat : nilai min pada setiap rule

z_1-z_n : nilai z pada setiap rule



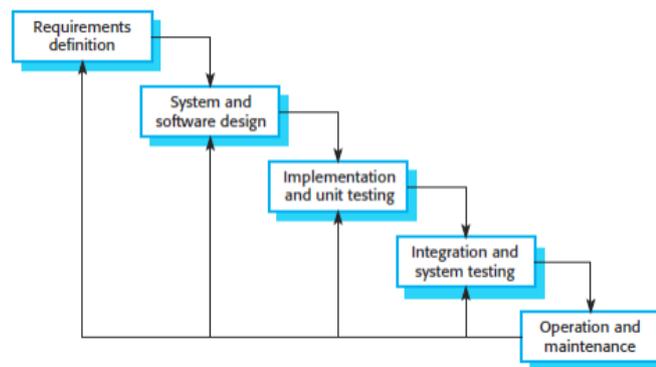
Gambar 2.5 Inferensi Fuzzy Tsukamoto

Sumber : Widaningsih, (2017)

2.2.4 Software Development Life Cycle (SDLC) Waterfall

System Development Life Cycle (SDLC) adalah metodologi umum yang digunakan untuk mengembangkan sistem informasi. SDLC terdiri dari beberapa fase, mulai dari perencanaan, analisis, perancangan, implementasi, hingga pemeliharaan sistem. Konsep SDLC menjadi dasar untuk pengembangan sistem informasi, memberikan kerangka kerja untuk perencanaan dan pengendalian sistem. Salah satu model SDLC yang sering digunakan adalah *Waterfall* (Badrul, 2021).

Waterfall Model merupakan model yang menyediakan pendekatan alur hidup perangkat lunak secara sistematis atau terurut mulai dari melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi) dan pengujian (Badrul, 2021). Tahapan dalam Waterfall Model dapat dilihat sebagai berikut.



Gambar 2.6 Model *Waterfall*

Sumber : Badrul, (2021)

Gambar tersebut merupakan tahapan secara umum dari *model Waterfall*. Menurut Badrul, (2021) *model Waterfall* diuraikan dengan tahap-tahap sebagai berikut.

1. Analisis Kebutuhan (*Requirement Analysis*)

Proses ini dimulai dengan pemahaman terhadap kebutuhan dan tujuan dari perangkat lunak yang akan dibuat. Tim pengembang akan mempelajari kebutuhan dan persyaratan dari pengguna, serta menentukan fitur-fitur dan fungsi yang diperlukan.

2. Perancangan (*Design*)

Setelah memahami kebutuhan, peneliti yang menggunakan Metode Waterfall akan melakukan perancangan arsitektur, desain, dan spesifikasi teknis perangkat lunak. Proses perancangan juga mencakup pembuatan diagram alir dan desain antarmuka pengguna.

3. Implementasi (*Implementation*)

Implementasi melibatkan pembuatan kode program serta pengujian untuk memverifikasi kualitas dari perangkat lunak yang dibuat.

4. Pengujian (*Testing*)

Setelah penyelesaian kode program, tahap pengujian dilakukan untuk memeriksa kinerja perangkat lunak dan memastikan bahwa ia memenuhi kebutuhan pengguna. Tujuannya adalah menghasilkan perangkat lunak yang sesuai dengan persyaratan yang telah ditetapkan.

5. Pemeliharaan (*Maintenance*)

Proses pemeliharaan dimulai setelah produk dirilis kepada konsumen. Tim pengembang akan terus memperbaiki, memperbarui, dan memperluas perangkat lunak sesuai kebutuhan pengguna. Ini tidak hanya mempertahankan kinerja perangkat, tetapi juga melakukan upgrade berkala, meningkatkan kepuasan pengguna. Metode *Waterfall* membutuhkan tahapan yang berurutan dan cocok untuk proyek-proyek dengan kebutuhan yang jelas, meskipun kurang fleksibel.

2.2.5 Unified Modeling Language (UML)

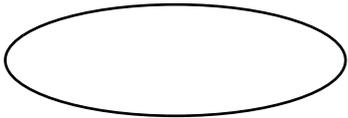
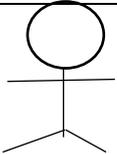
Bahasa Pemodelan Pengembangan Sistem atau bisa disebut *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (MulKhoir et al., 2018).

2.2.5.1 Use Case Diagram

Use case diagram atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat (Sopriani & Purwanto, 2023).

Use case mendefinisikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat, menjelaskan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.3 di bawah ini:

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Di awal frasa nama <i>Use Case</i>, kata kerja biasanya digunakan untuk menggambarkan fungsionalitas sistem sebagai unit-unit yang dapat bertukar pesan satu sama lain dan antar <i>actor</i> atau unit.</p>
<p><i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor</p>

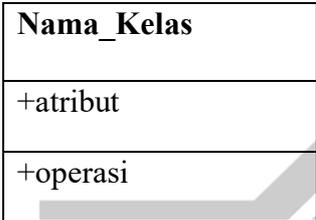
Simbol	Deskripsi
	adalah gambar orang, <i>actor</i> belum tentu merupakan orang, biasanya dinyatakan dengan kata benda di awal frasa nama <i>actor</i> .
<i>Asosiasi</i> 	Komunikasi antara <i>actor</i> dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan <i>actor</i>
<i>Ekstensi/extend</i> <<extend>> 	Hubungan/Relasi <i>Use Case</i> tambahan dengan <i>Use Case</i> , dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>Use Case</i> tambahan tersebut, mirip dengan prinsip inheritance/pewarisan pemrograman berorientasi objek, biasanya <i>Use Case</i> tambahan memiliki nama depan..
Generalisasi 	Ini adalah hubungan antara generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> di mana fungsi yang satu lebih umum daripada fungsi lainnya.
Menggunakan/ <i>Include/uses</i> <<include>> 	hubungan atau relasi antara use case tambahan dan use case yang ditambahkan yang membutuhkan use case ini untuk menjalankan fungsinya atau sebagai syarat untuk menjalankan fungsinya

Sumber : (Sopriani & Purwanto, 2023)

2.2.5.2 Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur suatu sistem dengan mendefinisikan kelas-kelas yang dibuat untuk membentuk sistem. Kelas mempunyai apa yang disebut atribut dan metode, atau operasi (Sopriani & Purwanto, 2023), yang menjelaskan simbol-simbol pada Tabel 2.4 diagram kelas.

Tabel 2. 3 Simbol *Clas Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Interface/antar muka  <i>Nama_Interface</i>	Sama seperti konsep interface atau antarmuka pada pemrograman objek
Asosiasi/hubungan  Asosisasi Berarah	Hubungan atau relasi antar kelas yang mempunyai arti umum, hubungan tersebut biasanya juga mencakup multiplicity.
	hubungan antara kelas dengan arti kelas yang digunakan oleh kelas lain, biasanya disertai dengan multiplicity
Generalisasi 	hubungan antara kelas dan arti generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi kelas terkait dengan kebergantungan kelas
Agregasi/ <i>agregation</i> 	Relasi antara kelas dengan arti setiap bagian (<i>whole-part</i>)

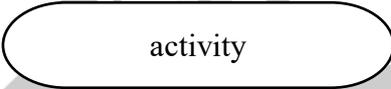
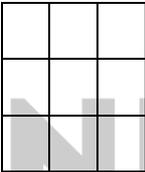
Sumber : Sopriani & Purwanto, (2023)

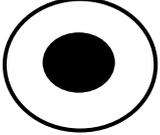
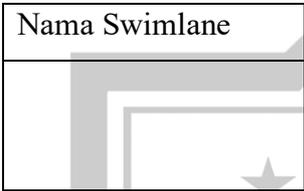
2.2.5.3 Activity Diagram

Activity Diagram menggambarkan sebuah *workflow* (aliran kerja) atau juga aktivitas dari sebuah sistem atau juga proses bisnis (Sopriani & Purwanto, 2023).

Berikut menjelaskan Simbol-simbol yang ada pada *Activity Diagram* dapat dilihat pada tabel 2.5:

Tabel 2.4 Simbol *Activity Diagram*

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem dapat dilihat dalam diagram aktivitas.
Aktivitas 	Semua aktivitas yang dilakukan sistem biasanya dimulai dengan kata kerja.
<i>Decision</i> /Percabangan 	Asosiasi percabangan di mana jika ada banyak pilihan aktivitas
<i>Join</i> /Penggabungan 	penggabungan dimana lebih dari satu aktivitas lalu digabungkan menjadi satu
Tabel 	Suatu file komputer yang dapat dibaca atau direkam data selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan

Simbol	Deskripsi
Status Akhir 	Sebuah status akhir ditampilkan dalam diagram aktivitas, bersama dengan status akhir yang dilakukan sistem.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Sopriani & Purwanto, (2023)

2.2.6 Perangkat Lunak

2.2.6.1 XAMPP

XAMPP adalah sebuah program yang berfungsi sebagai server web pada komputer Anda dan memungkinkan Anda menjalankan website berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP juga dapat disebut sebagai server virtual CPanel, yang membantu Anda melakukan preview dan memungkinkan Anda mengubah tampilan website tanpa harus terhubung ke internet atau online (Leni Natalia Zulita, 2013).

2.2.6.2 PHP

PHP (*Hypertext Preprocessor*) adalah kode atau script yang dieksekusi di server. Ini memungkinkan aplikasi untuk diintegrasikan kedalam HTML, yang membuat halaman web menjadi dinamis daripada statis (Sutaji, 2012). Sifat *server side* berarti bahwa proses dilakukan di server dan kemudian hasil dikirim ke browser.

Menurut Sutaji, (2012), Bahwa aturan penulisan *script* PHP adalah sebagai berikut:

- a. *Embedded script*, yaitu cara meletakkan tag PHP diantara tag-tag HTML. Contohnya:


```
<html>
<body>
<?php echo "Belajar"; ?>
</body>
</html>
```

- b. *Non embedded script*, yaitu cara semua script HTML diletakkan dalam script PHP. Contohnya:

```
<?php
echo "<html>";
echo "<body>";
echo "Belajar PHP";
echo "</body>";
echo "</html>";
```

PHP memiliki variabel yang digunakan untuk menyimpan data sementara. Setelah program selesai dieksekusi, data tersebut akan hilang. Sebagai aturan umum, nama variabel harus digunakan menurut (Sutaji, 2012) sebagai berikut:

- 1) Diawali dengan \$
- 2) bersifat case sensitive, sehingga \$Var tidak sama dengan \$var atau \$VAR.
- 3) Karakter pertama harus berupa huruf atau garis bawah (_).
- 4) Karakter berikutnya boleh huruf, angka atau garis bawah (_).

Berikut adalah beberapa contoh penulisan dan pendeklarasian variabel:

```
$namadepan = "Ali Shodikin";
```

```
$harga_barang = 3500;
```

```
$nilai3 = 3.45;
```

2.2.6.3 MySQL

Database Management System (DBMS) adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data, DBMS adalah alat atau alat yang berperan dalam membangun struktur tersebut. DBMS juga terdiri dari kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses database. Banyak jenis program DBMS saat ini, termasuk MySQL, Oracle, Interbase/Firebird, dan IBM DB2.

Menurut Silberschatz et al., (2006) *Database Management System* (DBMS) adalah suatu kumpulan data yang saling terkait dan kumpulan program untuk mengakses data tersebut. Database adalah kumpulan data yang berisi informasi yang penting untuk dipublikasikan. Tujuan utama DBMS adalah menyediakan sebuah cara untuk menyimpan dan menerima kembali informasi database secara mudah dan efisien.

MySQL adalah program *database management system* (DBMS) yang memiliki kemampuan untuk mengelola database dengan cepat dan memiliki

kapasitas untuk menampung data yang sangat besar yang dapat diakses oleh banyak pengguna. Selain itu, data dapat melakukan proses berbarengan atau secara sinkron.

Menurut Sutaji, (2012) *My SQL* dibagi menjadi 3 bentuk Query, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah metode query SQL yang membantu mendefinisikan data pada sebuah database. Berikut ini adalah query yang termasuk dalam DDL:

- a. *CREATE* melakukan pembuatan tabel dan database;
- b. *DROP* melakukan penghapusan kedua tabel dan database;
- c. *ALTER* melakukan perubahan struktur tabel yang telah dibuat, termasuk menambah *Field (add)*, mengganti nama *Field (change)*, menamakannya kembali (*rename*), dan menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML adalah metode query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsinya adalah untuk melakukan manipulasi database yang telah ada atau yang telah dibuat sebelumnya. *Query-query* yang termasuk dalam DML adalah sebagai berikut:

- a. *SELECT* menampilkan data pada tabel.
- b. *INSERT* memasukkan atau memasukkan data ke dalam tabel database.
- c. *UPDATE* mengubah atau memperbarui data yang ada pada tabel.
- d. *DELETE* untuk melakukan penghapusan data pada tabel.

3. DCL (*Data Control Language*)

DCL adalah teknik pertanyaan yang dapat digunakan untuk mengontrol eksekusi perintah. Seringkali berkaitan dengan pengaturan hak akses. DCL termasuk query berikut:

- a. *GRANT*: memberikan hak akses kepada user tertentu.
- b. *REVOKE*: menghapus hak akses dari user tertentu.

2.2.7 Pengujian Black Box

Pengujian sistem adalah tahap dalam pengembangan perangkat lunak yang bertujuan untuk memverifikasi apakah perangkat lunak yang telah dibuat memenuhi standar yang ditetapkan. Salah satu metode yang digunakan dalam pengujian sistem adalah *Black Box Testing*, yang merupakan pendekatan yang melingkupi teknik *White Box Testing*. *Black Box Testing* memiliki kemampuan

untuk menemukan berbagai kesalahan karena tidak bergantung pada pengetahuan internal tentang kode program. Pendekatan ini terutama berfokus pada pengujian persyaratan fungsional perangkat lunak dengan mencari serangkaian kondisi input yang memenuhi persyaratan tersebut (Wijaya & Astuti, 2021).

Jadi, pengujian sistem dapat dilakukan melalui evaluasi input, proses, dan output, yang terdiri dari:

- 1) Evaluasi input, yang mencakup pengecekan keseluruhan item input, kemudahan penggunaan, manipulasi data, dan kontrol kesalahan.
- 2) Evaluasi proses, yang dilakukan dengan memeriksa hasil output dari program.
- 3) Evaluasi output, yang mencakup verifikasi format dan struktur laporan yang dihasilkan.

2.2.7.1 Kelebihan Black Box Testing

Keunggulan dari pengujian tersebut adalah:

- 1) Tidak memerlukan pemeriksaan yang mendalam terhadap kode sumber.
- 2) Mampu menemukan kesalahan pengetikan (*typo*).
- 3) Mampu mengidentifikasi kesalahan dalam desain atau antarmuka pengguna dari perangkat lunak atau situs web.
- 4) Mampu menampilkan asumsi yang tidak akurat, untuk kemudian dianalisis dan diperbaiki.
- 5) Tidak memerlukan keahlian pemrograman bagi seorang pengujian.

2.2.7.2 Kekurangan Black Box Testing

Kekurangan dari pengujian tersebut adalah bergantung pada dokumen dan desain perangkat lunak, namun tidak mencapai tingkat detail kode sehingga tester tidak dapat menilai tingkat keamanan perangkat lunak tersebut.

2.2.8 Penilaian Siswa Berprestasi

“Evaluation could be a sistematic prepare deciding the degree ot which intrictional objectivves are accomplished by pupils”. Kalimat tersebut menjelaskan bahwa penilaian adalah suatu proses dalam mengumpulkan informasi dan membuat keputusan berdasarkan informasi - informasi tersebut (Purwanto, 2010). Dalam proses mengumpulkan informasi, tentunya tidak semua informasi biasa digunakan untuk membuat sebuah keputusan. Informasi – informasi yang relevan

dengan apa yang dinilai akan mempermudah dalam melakukan sebuah penilaian dalam kegiatan pembelajaran.

Menurut Arifin (2009), penilaian merupakan suatu proses atau kegiatan yang sistematis dan berkesinambungan untuk mengumpulkan informasi tentang proses dan hasil belajar peserta didik dalam rangka membuat keputusan-keputusan berdasarkan kriteria dan pertimbangan tertentu.

Berdasarkan pendapat diatas, maka dapat disimpulkan bahwa penilaian adalah suatu proses pengumpulan informasi secara menyeluruh yang dilakukan secara terus menerus untuk mengetahui kemampuan atau keberhasilan siswa dalam pembelajaran dengan menilai kinerja mereka, baik kinerja secara individu. Penilaian tersebut harus mendapatkan perhatian yang lebih dari Kepala Sekolah. Dengan demikian, penilaian tersebut harus dilaksanakan dengan baik, karena penilaian siswa merupakan komponen crucial dari pengembangan diri yang sehat.

2.2.8.1 Kriteria Siswa Berprestasi

Pada penelitian ini dalam melakukan penilaian perengkingan siswa menggunakan kriteria dan bobot nilai yang ditentukan. Berikut adalah nilai kriteria yang digunakan, dapat dilihat pada tabel 2.4 dan Tabel 2.5:

Tabel 2. 4 Kriteria yang Digunakan

Kriteria
Karya Tulis
Rata Rata Rapor
Kehadiran
Sikap/Tingkah Laku
Bahasa Asing

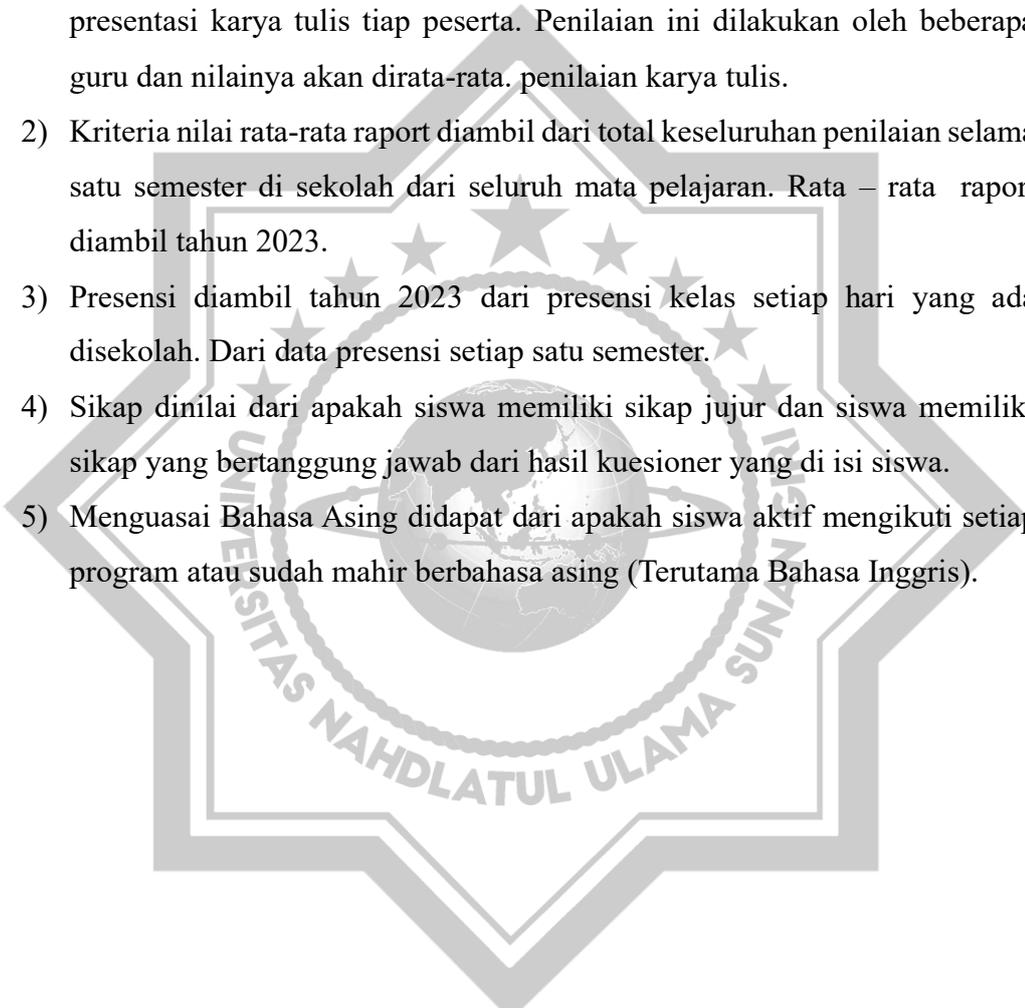
Tabel 2. 5 Nilai yang Digunakan

Keterangan	Nilai
Sangat Baik	90
Baik	80
Cukup Baik	70
Kurang Baik	60

Buruk	50
-------	----

Kriteria dalam penelitian ini didapat berdasarkan yaitu :

- 1) Karya Tulis merupakan semua guru yang menilai naskah karya tulis dan presentasi karya tulis tiap peserta. Penilaian ini dilakukan oleh beberapa guru dan nilainya akan dirata-rata. penilaian karya tulis.
- 2) Kriteria nilai rata-rata raport diambil dari total keseluruhan penilaian selama satu semester di sekolah dari seluruh mata pelajaran. Rata – rata raport diambil tahun 2023.
- 3) Presensi diambil tahun 2023 dari presensi kelas setiap hari yang ada disekolah. Dari data presensi setiap satu semester.
- 4) Sikap dinilai dari apakah siswa memiliki sikap jujur dan siswa memiliki sikap yang bertanggung jawab dari hasil kuesioner yang di isi siswa.
- 5) Menguasai Bahasa Asing didapat dari apakah siswa aktif mengikuti setiap program atau sudah mahir berbahasa asing (Terutama Bahasa Inggris).



UNUGIRI