

BAB II

TINJAUAN PUSTAKA

2.1 Peneliti Terdahulu

Pada penelitian kali ini peneliti telah melakukan studi literatur terhadap 5 jurnal yang berkaitan dengan penelitian ini. Tabel 2.1 menunjukkan penelitian terkait metode SAW dan TOPSIS.

Penelitian yang dilakukan oleh Heny Novita Sari dan Azizah Fatmawati yang berjudul “Sistem Pendukung Keputusan Rekomendasi Penentu Beras Miskin Menggunakan Metode SAW dan TOPSIS (Studi Kasus: Desa Semagar Girimarto Wonogiri)”. Penelitian ini bertujuan untuk mengetahui kesalahan dalam pencarian data dengan berbasis web. Penelitian ini menghasilkan 86,67% dari pengujian kuesioner yang dilakukan oleh responden. Kekurangan dari penelitian ini hanya memakai 10 responden untuk melakukan pengujian [3].

Penelitian yang dilakukan oleh Rahardhiyan Wahyu Putra dan Ulla Delfana Rosiani yang berjudul “SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI PEKERJAAN BAGI FRESH GRADUATE DENGAN PENGGABUNGAN SAW DAN TOPSIS”. Penelitian ini bertujuan untuk memperoleh tiga terbaik rekomendasi pekerjaan dalam sistem. Kekurangan dari penelitian ini tidak memberikan hasil akurasi dari perhitungan penggabungan dari metode SAW dan TOPSIS [5].

Penelitian yang dilakukan oleh Ade Davy Wiranata, Irwansyah, Agus Budiyantra dan Asrul Sani yang berjudul “Pemilihan Karyawan Teladan Menggunakan Metode SAW dan TOPSIS”. Penelitian ini bertujuan untuk mencari karyawan teladan pada PT. AeroTRANS Service Indonesia dan mendapatkan hasil pengujian sistem dengan rata-rata sebesar 78,02% untuk 4 karakteristik yaitu *functionality*, *reability*, *usability* dan *efficiency*. Kekurangan dari penelitian ini adalah hanya menggunakan 5 orang untuk perhitungan pemilihan karyawan teladan [2].

Penelitian yang dilakukan oleh Mustika Mentari dan Rahardhiyan Wahyu Putra yang berjudul “Sistem Pendukung Keputusan Guru Berprestasi Berbasis Java Desktop dengan Penggabungan Metode SAW dan TOPSIS”. Penelitian ini

bertujuan untuk menentukan guru paling berprestasi dengan perhitungan SAW dan TOPSIS. Hasil yang diperoleh dari perhitungan yaitu ada pada A11 dengan total skor 0.717. Kekurangan dari penelitian ini adalah tidak dapat dipastikan akurasi dari perhitungan SAW dan TOPSIS untuk penentuan guru berprestasi [6].

Penelitian yang dilakukan oleh I Gede Teguh Heriawan dan I Gede Bendesa Subawa yang berjudul “Sistem Pendukung Keputusan Pemberian Beasiswa Bidikmisi Menggunakan Metode SAW dan TOPSIS di STAH Mpu Kuturan Singaraja”. Penelitian ini bertujuan untuk menentukan pemberian beasiswa bidikmis kepada mahasiswa di Sekolah Tinggi Agama Hindu Negeri Mpu Kuturan Singaraja. Hasil yang diperoleh dari metode SAW dan TOPSIS dengan perhitungan manual mendapatkan hasil 9 yang sama, sehingga jika dihitung mendapatkan akurasi sebesar 90%. Kekurangan dari penelitian ini adalah pengujian terhadap beasiswa bidikmisi hanya digunakan 10 data dari 30 data yang tersedia sehingga data yang dihitungpun tidak lebih dari 50% untuk pengujiannya.

No	Judul Penelitian dan Tahun Penelitian	Nama Peneliti	Metode Penelitian	Hasil Penelitian	Batasan Penelitian
1	Sistem Pendukung Keputusan Rekomendasi Penentu Beras Miskin Menggunakan Metode SAW dan TOPSIS (2019) [3]	Heny Novita Sari dan Azizah Fatmawati	Metode SAW (<i>Simple Additive Weighting</i>) dan TOPSIS (<i>Technique for Order Preference by Similiarity to Ideal Solution</i>)	Hasil dari aplikasi sistem ini dengan melakukan perhitungan dari penilaian responden mendapat rata-rata 86,67%.	Kekurangan dari penelitian ini hanya memakai 10 responden untuk melakukan pengujian.
2	SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI PEKERJAAN BAGI FRESH GRADUATE DENGAN PENGGABUNGAN SAW DAN TOPSIS (2021) [5]	Rahardhiyan Wahyu Putra dan Ulla Delfana Rosiani	Metode SAW (<i>Simple Additive Weighting</i>) dan TOPSIS (<i>Technique for Order Preference by Similiarity to Ideal Solution</i>)	Hasil yang diperoleh adalah menampilkan pilihan tiga terbaik dari perhitungan SAW dan TOPSIS.	Kekurangan dari penelitian ini tidak memberikan hasil akurasi dari perhitungan penggabungan dari metode SAW dan TOPSIS.
3	Pemilihan Karyawan Teladan Menggunakan	Ade Davy Wiranata,	Metode SAW (<i>Simple Additive Weighting</i>)	Hasil dari uji sistem yang dilakukan oleh pengguna	kekurangan dari penelitian ini adalah

	Metode SAW dan TOPSIS (2020) [2]	Irwansyah, Agus Budiyantara dan Asrul Sani	dan TOPSIS (<i>Technique for Order Preference by Similiarity to Ideal Solution</i>)	yang terdiri dari 4 karakteristik yaitu <i>functionality, reability, usability</i> dan <i>efficiency</i> memperoleh hasil rata-rata nilai sebesar 78,02%.	hanya menggunakan 5 orang untuk perhitungan pemilihan karyawan teladan.
4	Sistem Pendukung Keputusan Guru Berprestasi Berbasis Java Desktop dengan Penggabungan Metode SAW dan TOPSIS (2020) [6]	Mustika Mentari dan Rahardhiyan Wahyu Putra	Metode SAW (<i>Simple Additive Weighting</i>) dan TOPSIS (<i>Technique for Order Preference by Similiarity to Ideal Solution</i>)	Hasil dari penelitian ini adalah menentukan predikat guru paling berprestasi dan didapat nilai terbesar ada pada A11 dengan skor yaitu sebesar 0.717.	Kekurangan dari penelitian ini adalah tidak dapat dipastikan akurasi dari perhitungan SAW dan TOPSIS untuk penentuan guru berprestasi.
5	Sistem Pendukung Keputusan Pemberian Beasiswa Bidikmisi Menggunakan Metode SAW dan TOPSIS di	I Gede Teguh Heriawan dan I Gede Bendesa Subawa	Metode SAW (<i>Simple Additive Weighting</i>) dan TOPSIS (<i>Technique for Order Preference by</i>	Sistem pengambilan keputusan yang diperoleh dari nilai TOPSIS dari 10 data yang diuji coba mendapatkan hasil 90%.	Kekurangan dari penelitian ini adalah pengujian terhadap beasiswa bidikmisi hanya digunakan 10

	STAH Mpu Kuturan Singaraja (2019)		<i>Similiarity to Ideal Solution)</i>		data dari 30 data yang tersedia sehingga data yang dihitungpun tidak lebih dari 50% untuk pengujiannya
--	--------------------------------------	--	---	--	--

Tabel 2.1 Ringkasan Penelitian Terdahulu

Perbedaan antara penelitian ini dengan penelitian terdahulu adalah penelitian ini menggunakan metode SAW untuk mencari nilai bobot dari kriteria yang telah diinputkan oleh pengguna hingga mendapatkan perhitungan matrik dan dilanjutkan dengan metode TOPSIS untuk mencari nilai solusi positif dan solusi negatif hingga penentuan preferensi untuk menentukan hasil rekomendasi resep masakan dari hasil perhitungan metode SAW.

2.2 Dasar Teori

2.2.1 Kuliner Nusantara

Kuliner Nusantara atau yang sering disebut masakan Indonesia merupakan suatu bentuk pencerminan terhadap tradisi dan beragam budaya dari masing-masing pulau yang berada dari Sabang sampai Merauke. Masakan Indonesia memegang peranan penting dalam budaya nasional Indonesia, masakan Indonesia memiliki cita rasa yang khas serta kaya akan bumbu yang berasal dari rempah-rempah yang berkualitas tinggi [15]. Masakan Indonesia pada masing-masing daerah memiliki ciri khas kuliner yang menarik, unik dan penuh cita rasa. Indonesia juga dikenal sebagai negeri yang kaya, bukan karena hanya hasil alam yang melimpah, namun juga dari beragamnya suku sehingga menciptakan banyak kuliner [16].

2.2.2 MCDM (*Multiple Criteria Decision Making*)

MCDM merupakan seperangkat metode dan prosedur beberapa standar yang saling bertentangan yang dapat dimasukkan ke dalam proses pengambilan keputusan untuk memilih alternatif terbaik dengan serangkaian kriteria yang banyak [12]. MCDM memiliki dua kategori yakni *MODM (Multiple Objective Decision Making)* dan *MADM (Multiple Attribute Decision Making)*. *MADM (Multiple Attribute Decision Making)* dan *MODM (Multiple Object Decision Making)* merupakan dua kategori yang dimiliki oleh MCDM. Sebuah metode yang mengambil keputusan dan penilaian yang subjektif merupakan karakteristik dari MADM. Sebuah metode yang mencakup masalah perancangan, matematik untuk optimasi dalam penggunaan yang alternatif yang sangat besar merupakan ciri MODM, dimana MODM mengambil banyak kriteria untuk sebuah pengambilan keputusan [13]. Terdapat beberapa metode MCDM yang telah banyak digunakan seperti berikut [14]:

1. *AHP (Analytical Hierarchy Process)*
2. *ANP (Analytical Network Process)*
3. *Promethee (Preference Ranking Organization Method for Enrichment of Evaluations)*
4. *VIKOR (Visekriterijumska Optimizacija I Kompromisno Resenje)*

5. *ELECTRE (Elimination Et Choix Traduisant la Realite)*
6. *BWM (Best Worst Method)*
7. *SAW (Simple Additive Weighting)*
8. *DEMATEL (Decision Making Trial and Evaluation Laboratory)*
9. *TOPSIS (Technique for Order of Preference by Similiarity to Ideal Solution)*
10. *WP (Weighting Product)*

2.2.3 SAW (Simple Additive Weighting)

SAW merupakan salah satu metode pengambilan keputusan dimana mencari jumlah bobot kinerja peringkat pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses menormalkan matriks keputusan (X) ke skala yang sebanding dengan semua alternatif yang ada peringkat [7]. Terdapat langkah penyelesaian dalam metode SAW, yaitu:

1. Menentukan Alternatif, yaitu A_i .
2. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan yaitu C_j .
3. Memberikan nilai bobot untuk masing-masing kriteria yaitu W .
4. Memberikan *rating* kecocokan setiap alternatif pada setiap alternatif.
5. Membuat matriks keputusan yang dibentuk dari tabel *rating* kecocokan dari setiap alternatif pada setiap kriteria. Nilai setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan.

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2n} \\ X_{31} & X_{32} & X_{33} & \dots & X_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & X_{m3} & \dots & X_{mn} \end{bmatrix} \quad (2.1)$$

6. Melakukan normalisasi matrik keputusan dengan cara menghitung nilai *rating* kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada kriteria C_j .

$$r_{ij} = \frac{X_{ij}}{\text{Max}_i X_{ij}}, \text{ Jika } j \text{ adalah atribut } \textit{benefit} \quad (2.2)$$

$$r_{ij} = \frac{\text{Min}_i X_{ij}}{X_{ij}}, \text{ Jika } j \text{ adalah atribut } \textit{cost} \quad (2.3)$$

7. Hasil dari *rating* kriteria ternormalisasi (r_{ij}) membentuk matrik ternormalisasi (R).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \dots & r_{2n} \\ r_{31} & r_{32} & r_{33} & \dots & r_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ r_{m1} & r_{m2} & r_{m3} & \dots & r_{mn} \end{bmatrix} \quad (2.4)$$

8. Nilai preferensi (V_i) merupakan hasil akhir yang diperoleh dari penjumlahan dari perkalian elemen baris matrik ternormalisasi (R) dengan bobot preferensi (W) yang bersesuaian elemen kolom matrik (W).

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad (2.5)$$

9. Hasil perhitungan nilai V_i yang paling besar merupakan alternatif yang terbaik.

2.2.4 TOPSIS (*Technique for Order Preference by Similiarity to Ideal Solution*)

TOPSIS merupakan salah satu metode pengambilan keputusan dengan kriteria yang dikenalkan oleh Yoon dan Hwang. TOPSIS menggunakan prinsip bahwa alternatif yang dipilih harus memiliki yang terpendek dari solusi ideal positif dan terjauh dari solusi ideal negatif. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan dikarenakan memiliki konsep yang sederhana, komputasi yang efisien dan memiliki kemampuan untuk mengukur secara relative untuk kinerja alternatif keputusan [8]. TOPSIS memiliki tahapan-tahapan untuk perhitungannya, yaitu:

1. Membuat matriks keputusan ternormalisasi

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.6)$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Dimana :

r_{ij} adalah matriks ternormalisasi $[i][j]$

x_{ij} adalah matriks keputusan $[i][j]$

m adalah jumlah baris matriks

n adalah jumlah kolom baris matriks

2. Membuat matriks keputusan ternormalisasi terbobot

$$y_{ij} = w_i r_{ij} \quad (2.7)$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Dimana :

y_{ij} adalah matriks ternormalisasi $[i][j]$

w_{ij} adalah vector bobot $[i]$

3. Menentukan matriks solusi ideal positif dan negatif

$$A^+ = \left\{ (\max v_{ij} | j \in J), (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m = \{V_1^+, V_2^+, V_3^+, \dots, V_n^+\} \right\} \quad (2.8)$$

$$A^- = \left\{ (\max v_{ij} | j \in J), (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m = \{V_1^-, V_2^-, V_3^-, \dots, V_n^-\} \right\} \quad (2.9)$$

4. Menghitung pemisahan

$$S_i^+ = \sqrt{\sum_j^n (v_{ij} - v_j^+)^2} \quad (2.10)$$

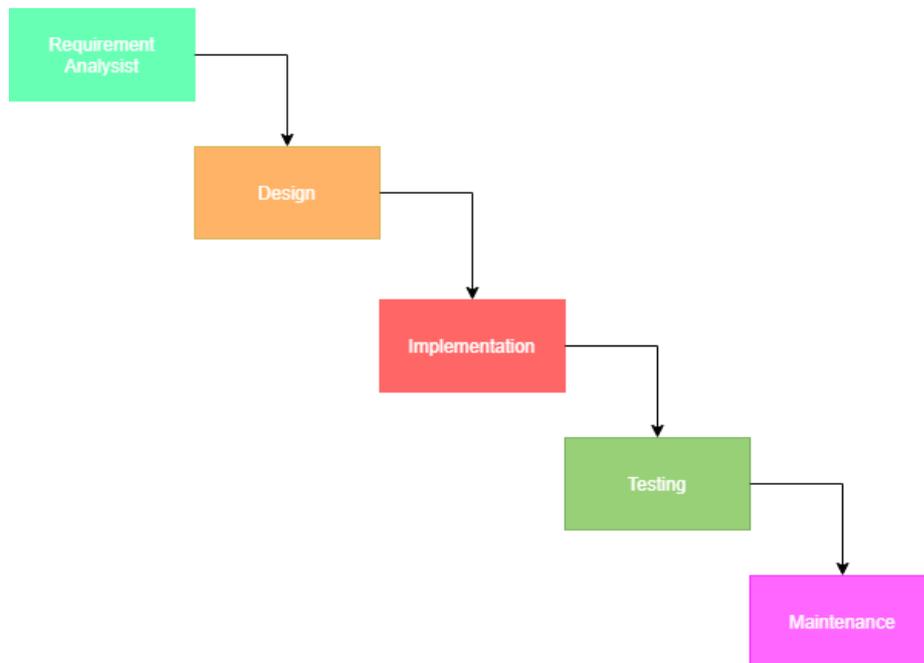
$$S_i^- = \sqrt{\sum_j^n (v_{ij} - v_j^-)^2} \quad (2.11)$$

5. Perhitungan nilai kedekatan masing-masing alternatif terhadap solusi ideal

$$Ci = \frac{S_i^-}{S_i^- + S_i^+} \quad (2.12)$$

2.2.5 Waterfall Model

Model *Waterfall* merupakan metode pengembangan sistem, model ini memiliki pendekatan sistematis dan urutan mulai dari sistem kebutuhan kemudian menuju ke tahap analisis, desain, pengkodean, pengujian dan pemeliharaan. Metode *waterfall* disebut air terjun karena terdiri dari urutan tahap demi tahap, tahapan harus diselesaikan sebelum memulai tahap berikutnya [9]. Tahapan dalam model *waterfall* dapat dilihat pada gambar 2.1.



Gambar 2.1 Metode *Waterfall* [9]

Tahapan model *Waterfall* memiliki tahapan yang berurutan untuk pengerjaannya, berikut gambaran singkat tentang metode *waterfall*.

1. Kebutuhan Analisis

Tahapan kebutuhan analisis merupakan informasi yang diberikan untuk menetapkan kesepakatan yang akan di analisis dan kemudian dokumentasi disiapkan untuk membantu lebih lanjut dalam proses pengembangan.

2. Perancangan

Tahapan perancangan merupakan proses dari perencanaan dan pemecahan masalah untuk solusi perangkat lunak, desain berkaitan dengan pemilihan algoritma yang sesuai desain, desain arsitektur perangkat lunak dan desain untuk basis data yang akan digunakan.

3. Implementasi

Tahapan implementasi merupakan proses pembuatan perangkat lunak yang di lanjutkan dari tahapan perancangan.

4. Pengujian

Tahapan pengujian merupakan bentuk pengujian perangkat lunak untuk memeriksa perangkat lunak terdapat bug atau gangguan sistem.

5. Pemeliharaan

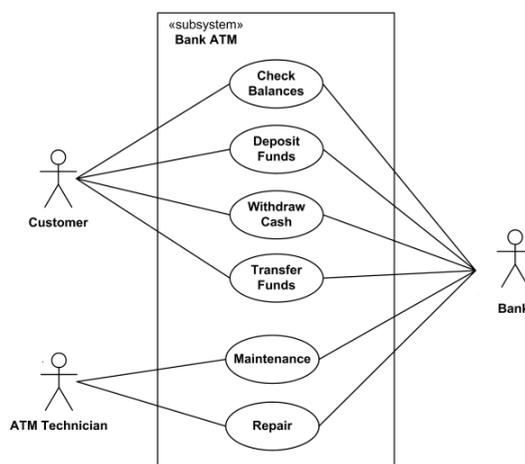
Tahapan pemeliharaan merupakan tahapan setelah pengujian perangkat lunak, tahapan ini memungkinkan beberapa modifikasi, perbaikan dan penyempurnaan yang sesuai.

2.2.6 UML (*Unified Model Language*)

UML atau *Unified Model Language* merupakan metode dalam pemodelan secara visual yang digunakan sebagai saran perancangan sistem berorientasi objek. UML dapat didefinisikan sebagai suatu bahasa standar visualisasi, pendokumentasian dan perancangan sistem. Salah satu tujuan dan fungsi perlu adanya UML yaitu mempermudah pengguna membaca suatu sistem. Adapun contoh dari UML yaitu sebagai berikut [21].

1. *Use Case Diagram*

Use Case Diagram merupakan satu jenis dari diagram UML (*Unified Model Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya [21]. Contoh *Use Case Diagram* dapat dilihat pada gambar 2.2.

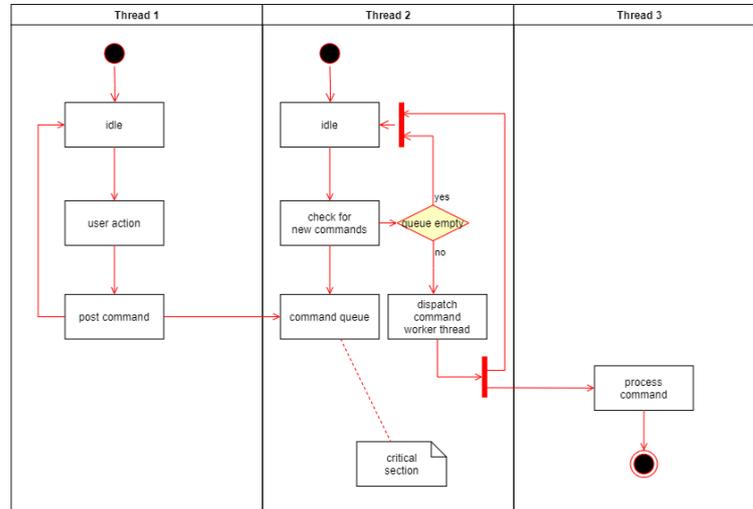


Gambar 2.2 Contoh *Use Case Diagram*

2. *Activity Diagram*

Activity Diagram merupakan satu jenis dari diagram UML (*Unified Model Language*) yang dapat memodelkan berbagai proses yang terjadi pada sistem. *Activity Diagram* seperti layaknya urutan proses berjalannya suatu sistem dan

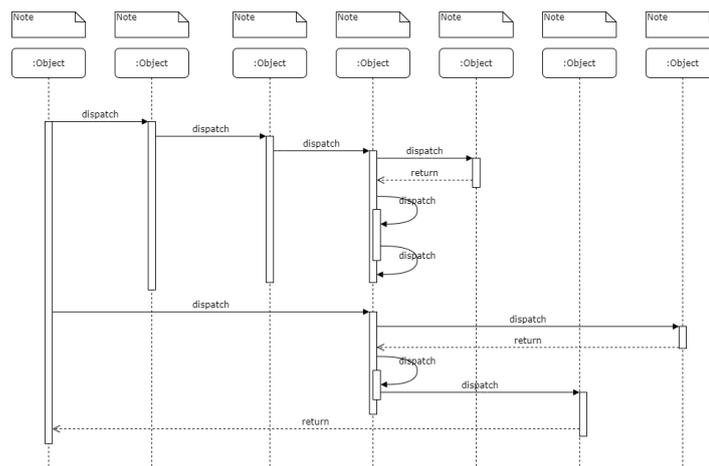
digambarkan secara vertikal [21]. Contoh *Activity Diagram* dapat dilihat pada gambar 2.3.



Gambar 2.3 Contoh *Activity Diagram*

3. *Sequence Diagram*

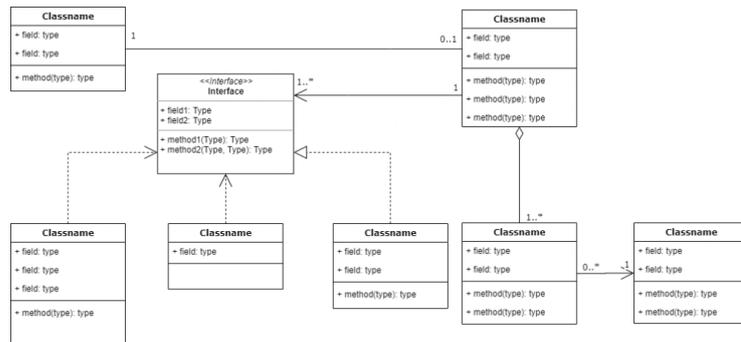
Sequence Diagram merupakan satu jenis diagram UML (*Unified Model Language*) yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu [21]. Contoh *Sequence Diagram* dapat dilihat pada gambar 2.4.



Gambar 4.4 Contoh *Use Case Diagram*

4. *Class Diagram*

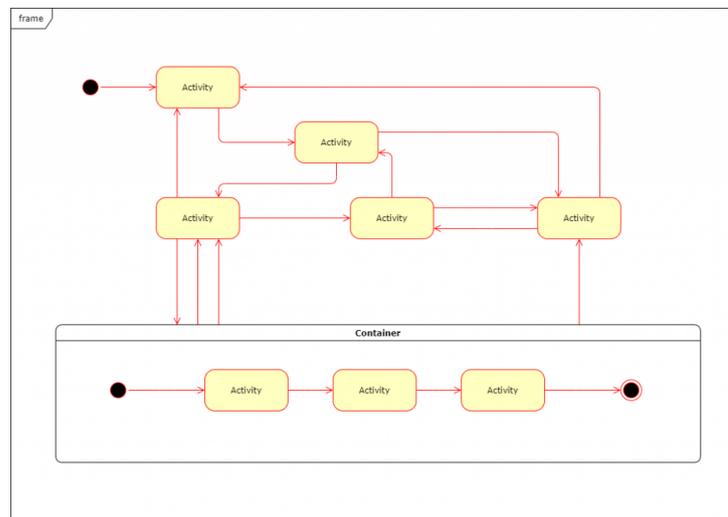
Class Diagram merupakan satu jenis diagram UML (*Unified Model Language*) yang digunakan untuk menampilkan kelas-kelas berupa paket-paket untuk memenuhi salah satu kebutuhan [21]. Contoh *Class Diagram* dapat dilihat pada gambar 2.5.



Gambar 5.5 Contoh *Class Diagram*

5. *Statemachine Diagram*

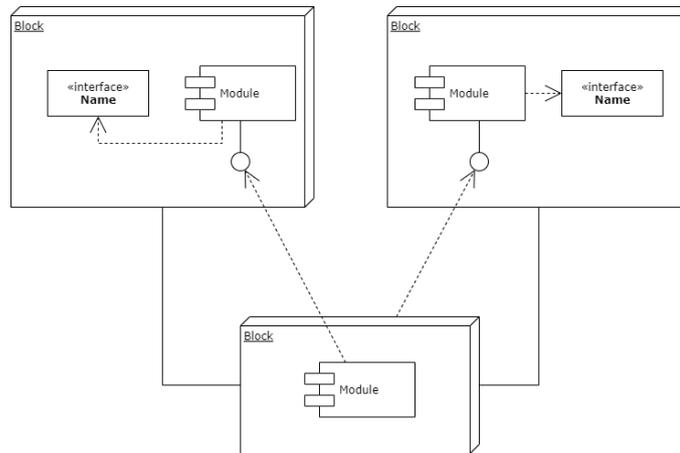
Statemachine Diagram merupakan satu jenis dari UML (*Unified Model Language*) yang berfungsi untuk menggambarkan transisi serta perubahan pada suatu objek pada sistem [21]. Contoh *Statemachine Diagram* dapat dilihat pada gambar 2.6.



Gambar 6.6 Contoh *Statemachine Diagram*

6. *Component Diagram*

Component Diagram merupakan satu jenis UML (*Unified Model Language*) yang berfungsi untuk menggambarkan perangkat lunak pada suatu sistem [21]. Contoh *Component Diagram* dapat dilihat pada gambar 2.7.



Gambar 7.7 Contoh *Component Diagram*

2.2.7 Wireframe

Wireframe merupakan kerangka dasar untuk menata suatu *item* di laman *website* atau aplikasi. *Wireframe* menampilkan lembaran yang terdiri dari kotak-kotak dan garis-garis untuk mengatur tata letak berbagai elemen pada *website* atau aplikasi. *Wireframe* dibagi menjadi 3 jenis, yaitu [22] [23]:

1. *Low-Fidelity Wireframe*

Low-Fidelity Wireframe merupakan desain paling sederhana dengan bentuk yang masih kasar, tanpa menggunakan skala, kisi dan akurasi piksel [22].

2. *Mid-Fidelity Wireframe*

Mid-Fidelity Wireframe merupakan desain yang menampilkan representasi tampilan yang lebih akurat, meskipun masih belum menggunakan gambar [22].

3. *High-Fidelity Wireframe*

High-Fidelity Wireframe merupakan kerangka desain yang sudah menggunakan gambar dan tulisan konten yang sebenarnya, *wireframe high-fidelity* juga sudah dilengkapi menu interaktif dalam desainnya [22].

2.2.8 White Box Testing

White Box testing adalah pengujian yang dilakukan untuk menguji dan menganalisis kode program apabila terjadi kesalahan [24]. *White Box testing* memiliki teknik yang dapat digunakan untuk melakukan pengujian pada perangkat lunak, berikut teknik pada *white box testing*.

1. *Basis Path Testing*

Basis Path Testing merupakan teknik yang bertujuan untuk mengukur kompleksitas kode program dan mendefinisikan alur yang dieksekusi.

2. *Branch Coverage*

Branch Coverage merupakan teknik yang dirancang agar setiap *branch code* diuji setidaknya satu kali.

3. *Condition Coverage*

Condition Coverage merupakan teknik yang bertujuan untuk menguji seluruh kode agar menghasilkan nilai *TRUE* atau *FALSE*.

4. *Loop Testing*

Loop Testing merupakan teknik yang wajib dilakukan untuk menguji berbagai perulangan yang ada dalam program, seperti *do-while*, *for* dan *while*.

5. *Multiple Condition Coverage*

Multiple Condition Coverage merupakan teknik untuk menguji seluruh kombinasi dari kode yang mungkin digunakan dalam berbagai kondisi. Seluruh kombinasi harus diuji minimal satu kali, tujuannya untuk memastikan perangkat lunak agar berjalan dengan baik.

6. *Statement Coverage*

Statement Coverage merupakan teknik yang dilakukan minimal satu kali untuk menguji setiap *statement* dalam perangkat lunak. Teknik ini untuk mengetahui kode-kode yang error sehingga dapat segera memperbaikinya.

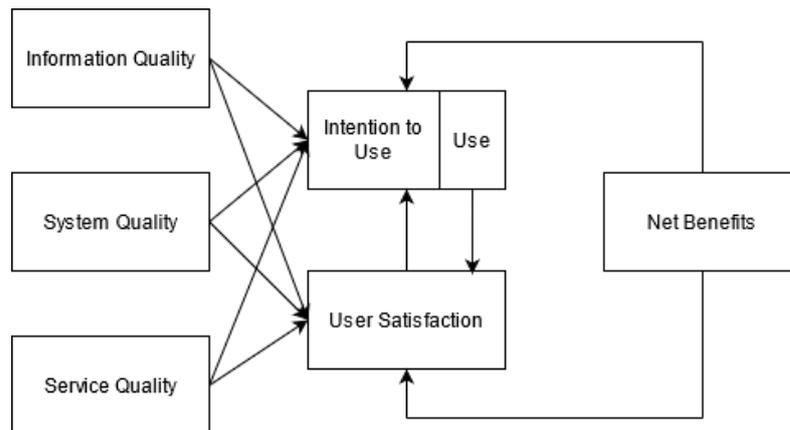
2.2.9 Black Box Testing

Black Box testing adalah pengujian yang dilakukan untuk mendeteksi kesalahan atau cacat pada sistem aplikasi. Pengujian berfungsi untuk mengevaluasi kerangka kerja aplikasi yang memenuhi tujuan desain sistem aplikasi [26].

2.2.10 Model DeLone dan McLean

Model DeLone dan McLean merupakan suatu model penelitian yang dikembangkan untuk mengukur kesuksesan suatu sistem informasi yang berdasarkan penelitian yang dilakukan oleh DeLone dan McLean pada tahun 1992.

Model ini lebih dikenal dengan *D&M IS Success*, pada model ini dimensi-dimensi kesuksesannya yang diukur semuanya saling berkaitan [10]. Terdapat pengukuran faktor kesuksesannya yaitu : *Information Quality*, *Service Quality*, *System Quality*, *Actual Use*, *User Satisfaction* dan *Net Benefit (Individual Impact)* [11]. Pengukuran kesuksesan dapat dilihat pada gambar 2.2.



Gambar 8.8 Model DeLone dan McLean 2003 [11]

Berikut adalah uraian dari masing-masing variabel model DeLone dan McLean [11]:

1. Kualitas Informasi (*Information Quality*)

Kualitas informasi merupakan informasi yang diharapkan pengguna saat menggunakan aplikasi. Indikator pengukuran kualitas informasi meliputi ketersediaan/kelengkapan informasi (*completeness*), penyajian informasi (*format*), keakuratan informasi (*accurate*), kemudahan pemahaman (*ease of understanding*), relevansi permintaan (*relevance*).

2. Kualitas Sistem (*System Quality*)

Kualitas sistem mencakup kinerja sistem ketika pengguna menggunakan aplikasi, baik dari segi *software* maupun *hardware*. Pengukuran indikator kualitas sistem antara lain adalah fleksibilitas (*flexibility*), kemudahan penggunaan (*ease of use*), keandalan sistem (*reliability*), kecepatan akses (*response time*), dan keamanan sistem (*security system*).

3. Kualitas Layanan (*Service Quality*)

Variabel kualitas layanan ini menjamin layanan yang disediakan oleh aplikasi, seperti pembaruan dan tanggapan terhadap umpan balik yang digunakan oleh pengguna. Indikator pengukur kualitas layanan meliputi jaminan sistem (*assurance*), empati (*empathy*), dan waktu tanggap layanan (*responsiveness*).

4. Pemakaian (*Use*)

Variabel ini mengacu pada seberapa sering pengguna menggunakan aplikasi. Indikator pengukuran penggunaan aplikasi meliputi semua hal tentang kebiasaan penggunaan sistem, seperti frekuensi penggunaan (*frequency of use*) dan sifat penggunaan (*nature of use*).

5. Kepuasan Pengguna (*User Satisfaction*)

Variabel ini merupakan respon yang disampaikan oleh pengguna setelah menggunakan aplikasi. Indikator untuk mengukur kepuasan pengguna adalah efisiensi (*efficiency*), efektivitas (*effectiveness*), dan kepuasan secara keseluruhan (*overall satisfaction*).

6. Keuntungan Bersih (*Net Benefits*)

Variabel ini merupakan dampak keberadaan serta penggunaan aplikasi terhadap kualitas kinerja pengguna. Indikator pengukur keuntungan bersih adalah peningkatan kinerja, efisiensi dan efektivitas, dan produktivitas.

2.2.11 Skala Likert

Metode *Skala likert* merupakan metode untuk mengukur tingkat kepuasan pengguna yang dikembangkan oleh Liker pada tahun 1932. *Skala likert* merupakan skala yang digunakan dalam pengukuran sikap, persepsi dan opini individu mengenai sebuah fenomena tertentu [25]. *Skala likert* memiliki empat atau lebih pertanyaan yang dikombinasikan sehingga membentuk sebuah nilai yang mempresentasikan individu [17]. Skala likert dapat dilihat pada tabel 2.2.

Tabel 2.2 Skala Likert [1]

Skala	Kategori	Skor	Interval
1	Sangat Setuju	5	Skor \geq 80%
2	Setuju	4	80% > Skor \geq 60%
3	Cukup	3	60% > Skor \geq 40%

4	Kurang Setuju	2	40% > Skor >= 20%
5	Tidak Setuju	1	Skor < 20%

Jenis skala Likert yang akan digunakan pada penelitian ini adalah skala Liker lima tingkat. Skala Likert dihitung menggunakan rumus seperti berikut.

$$Presentase\ Skor = \frac{\left(\begin{array}{c} ((Sangat\ Setuju\ x\ 5)+(Setuju\ x\ 4) \\ +(Cukup\ x\ 3)+ \\ (Kurang\ Setuju\ x\ 2)+(Tidak\ Setuju\ x\ 1) \end{array} \right)}{(5\ x\ jumlah\ responden))\ x\ 100\%} \quad (2.13)$$

2.2.12 SPSS (*Statistical Package for the Social Sciences*)

SPSS merupakan perangkat lunak pengolah data statistik yang paling populer dan banyak digunakan di dunia. SPSS digunakan untuk pengolahan data dan analisis data kuantitatif karena saling terkait dan juga termasuk dalam ruang lingkup statistik [28].

2.2.13 *Cronbach's Alpha*

Cronbach's Alpha merupakan pengukuran batas bawah nilai reliabilitas suatu konstruk. *Cronbach's Alpha* digunakan untuk mengukur keandalan indikator yang digunakan untuk kuesioner penelitian. *Cronbach's Alpha* memiliki nilai sebesar 0 sampai 1 [28].