

BAB II TINJAUAN PUSTAKA

Bab ini membahas metode yang digunakan penulis untuk merancang dan membuat aplikasi web sistem pendukung keputusan yang memungkinkan evaluasi kinerja guru melalui penerapan metode *Analytical Hierarchy Process* (AHP).

2.1 Pencapaian Terdahulu

Berbagai referensi dari penelitian sebelumnya digunakan sebagai dasar dan rujukan untuk penelitian ini.

Tabel 2.1 Referensi Pencapaian Terdahulu

No	Peneliti (Tahun)	Judul	Hasil	Publikasi
1	Rohmat, M. A., & Kusrini. (2021)	Penerapan Metode Analytical Hierarchy Process (AHP) Dalam Sistem Pendukung Keputusan Penilaian Kinerja Guru	Berdasarkan penelitian ini, metode Analytical Hierarchy Process (AHP) digunakan untuk mengidentifikasi empat kriteria utama dan delapan subkriteria sebagai dasar perhitungan dalam penilaian kinerja guru. Hasil perankingan menunjukkan bahwa setiap kriteria dan subkriteria diberikan bobot sesuai tingkat kepentingannya terhadap kinerja guru. Dengan metode ini, penilaian menjadi lebih objektif dan terukur, sehingga hasil perankingan dapat digunakan sebagai dasar untuk rekomendasi sertifikasi, kenaikan pangkat atau golongan.	Media Teknologi Informasi dan Komputer (METIK) Jurnal Vol. 5, No. 1, 2021
2	Ramadhan, I., & Buani,	Sistem Pendukung Keputusan Pemilihan	Berdasarkan penelitian, bahwa metode Analytical Hierarchy Process (AHP)	Evolusi: Jurnal Sains dan Manajemen Vol.

No	Peneliti (Tahun)	Judul	Hasil	Publikasi
	D. C. P. (2023)	Karyawan Terbaik Berdasarkan Kinerja Dengan Metode Analytical Hierarchy Process (AHP)	efektif dalam memudahkan pemilihan calon karyawan terbaik di PT. Bytel Sarana Telkomindo. Dengan AHP, pemilihan calon karyawan menjadi lebih tepat sasaran sesuai dengan kriteria yang telah ditentukan. Metode ini memungkinkan perbandingan alternatif data secara efisien, sehingga memudahkan dan mempercepat pemilihan karyawan terbaik.	11, No. 1, Maret 2023
3	Sutrisno, S., Prasetyo, H. A., & Faot, A. I. (2022)	The Measurement of Human Resources Employees by using Human Resources Score Card Method and Analytical Hierarchy Process Method	Human Resource Score Card digunakan sebagai alat strategis untuk mengelola sumber daya manusia dengan empat perspektif utama: pribadi, kompensasi, keselarasan, dan kinerja tinggi. Melalui metode Analytical Hierarchy Process (AHP), penelitian ini menunjukkan bahwa perspektif kompensasi memiliki peranan sentral. Bonus, insentif, dan penghargaan finansial terbukti efektif dalam meningkatkan sumber daya manusia perusahaan.	Budapest International Research and Critics Institute-Journal (BIRCI-Journal) Vol. 5, No. 2, May 2022
4	Awaludin, M. (2020)	Application Of Analytical Hierarchy Process Method for	Kehadiran (0.3794), pencapaian target kerja (0.2699), keterampilan (0.1742), kerja sama	Jurnal Sistem Informasi (JSI) Universitas

No	Peneliti (Tahun)	Judul	Hasil	Publikasi
		Employee Performance Evaluation at PT XYZ	(0.1113), dan sikap (0.0652) adalah lima faktor utama dalam penilaian kinerja karyawan PT XYZ, menurut penelitian ini. Sebagai hasil dari penilaian awal, Muryan Awaludin memenuhi syarat untuk kenaikan posisi dengan skor 0.3370.	Suryadarma Vol. 7, No. 1, 2020
5	Marpaung, E. Z. (2023)	Sistem Pendukung Keputusan Pemilihan Karyawan Berprestasi Berdasarkan Kinerja Menggunakan Metode Analytical Hierarchy Process (AHP)	Penelitian ini menemukan bahwa Analytical Hierarchy Process (AHP) memungkinkan penggunaan metode input nilai untuk mempercepat penilaian kinerja karyawan sambil mempertimbangkan kinerja mereka. Hasilnya menunjukkan bahwa AHP dapat menunjukkan bahwa salah satu pilihan harus diberi prioritas yang lebih tinggi saat membuat keputusan.	Bulletin Information Systems Vol. 1, No. 1, July 2023
6	Ikhwani, M. D. I., Diana, A., Usino, W., & Hasugian, H. (2023).	Implementation of Analytical Hierarchy Process (AHP) and Simple Additive Weighting (SAW) Methods in Decision Support Systems for Employee Performance Assessment at the Komite Olahraga	Penelitian menggunakan metode Analytic Hierarchy Process (AHP) dan Simple Additive Weighting (SAW) pada KONI DKI Jakarta untuk memilih karyawan terbaik. Hasil penelitian menunjukkan bahwa sistem pendukung keputusan (SPK) ini efektif dalam menentukan karyawan terbaik berdasarkan evaluasi kriteria	Scientific Journal of Information Systems and Informatics Vol. 5, No. 1, April 2023

No	Peneliti (Tahun)	Judul	Hasil	Publikasi
		Nasional Indonesia (KONI) DKI Jakarta	dengan nilai yang lebih aman dan akurat.	

2.2 Tinjauan Teoritis

Penelitian ini mencakup sejumlah analisis dan pemahaman terkait dengan teori-teori dan konsep yang mendukung resolusi dari permasalahan yang akan diteliti. Tinjauan teoritis bertujuan untuk memahami, mengevaluasi, mengintegrasikan konsep, teori, dan literatur yang relevan mengenai penelitian ini.

2.2.1 Evaluasi Kinerja

Menurut Aini (2019), evaluasi kinerja adalah proses penilaian yang dilakukan secara sistematis untuk mengukur hasil pekerjaan dan bertujuan untuk memberikan tanggung jawab yang sesuai agar karyawan atau pegawai dapat meningkatkan kinerjanya di masa mendatang. Evaluasi kinerja dapat dilakukan dengan mengukur capaian kinerja secara kuantitatif dan kualitatif.

Selain itu, kinerja karyawan dapat dinilai berdasarkan pencapaian target dan hasil kerja yang dihasilkan, serta kemampuan mereka dalam menghadapi tantangan dan mengambil inisiatif untuk meningkatkan efisiensi dan efektivitas kerja. Evaluasi kinerja karyawan memberikan manfaat bagi perusahaan untuk mengidentifikasi kekuatan dan kelemahan karyawan, serta sebagai dasar untuk memberikan umpan balik dan pengembangan karir sesuai potensi masing-masing. Dengan demikian, kinerja karyawan memiliki peran penting dalam memastikan kesuksesan dan pertumbuhan jangka panjang organisasi (Ekhsan, 2019).

2.2.2 Sistem Pendukung Keputusan

Dengan menggunakan model, data, dan teknik analisis tertentu, sistem informasi yang disebut Sistem Pendukung Keputusan (SPK) bertujuan untuk membantu proses pengambilan keputusan. SPK memecahkan masalah yang kompleks dengan memberikan pengguna informasi yang akurat dan relevan. SPK

dapat digunakan dalam banyak bidang, seperti manajemen, bisnis, dan ilmu kesehatan, antara lain.

Sistem pendukung keputusan memiliki tujuan utama untuk meningkatkan efektivitas dan efisiensi pengambilan keputusan. Dengan bantuan sistem ini, pengambil keputusan dapat menganalisis data dengan lebih baik dan lebih cepat, yang memungkinkan mereka untuk membuat keputusan yang tepat dalam waktu yang lebih singkat. SPK juga membantu mengurangi kemungkinan kesalahan pengambilan keputusan, yang membuat hasil yang diperoleh lebih dapat dipercaya (Sarwandi et al., 2023).

Menurut Lubis et al. (2022), berikut adalah komponen sistem pendukung keputusan:

1. Manajemen Data

Sistem manajemen basis data, perangkat lunak yang digunakan dalam manajemen data, mengelola basis data yang berisi informasi yang relevan untuk situasi tertentu.

2. Manajemen Model

Dalam manajemen model, paket perangkat lunak termasuk model kuantitatif seperti manajemen, statistik, finansial, atau lainnya yang memungkinkan analisis sistem serta perangkat lunak manajemen yang terkait.

3. Antarmuka Pengguna

Antarmuka pengguna memungkinkan pengguna berinteraksi dengan sistem dan memberikan input yang kemudian diproses untuk menghasilkan keputusan.

4. Subsistem Berbasis Pengetahuan

Subsistem berbasis pengetahuan terdiri dari komponen yang memiliki kemampuan untuk berdiri sendiri atau membantu subsistem lain.

2.2.3 Analytical Hierarchy Process

Analytical Hierarchy Process (AHP) pertama kali diciptakan oleh Profesor Thomas L. Saaty sebagai model untuk mendukung keputusan yang menguraikan masalah yang kompleks dengan banyak kriteria menjadi hierarki. *Analytical*

Hierarchy Process (AHP) membagi situasi yang kompleks dan tidak terstruktur menjadi beberapa bagian dalam susunan hierarki. AHP melibatkan penilaian subjektif terhadap pentingnya setiap variabel secara relatif dan menentukan variabel mana yang paling mempengaruhi hasil (Parhusip, 2019).

Hierarki adalah representasi dari masalah kompleks dalam struktur multi-level, di mana tujuan utama berada di level pertama, diikuti oleh level faktor, kriteria, sub-kriteria, dan seterusnya hingga level terakhir, yang memiliki alternatif.

Menurut Marsono (2020), penggunaan AHP terdapat tiga prinsip dasar, yaitu:

1. Dekomposisi (*Decomposition*)

Pada tahap ini, masalah yang rumit diuraikan menjadi bagian-bagian hierarkis. Tujuannya adalah untuk merinci masalah dari hal-hal yang umum menjadi hal-hal yang lebih khusus. Struktur ini digunakan untuk melakukan perbandingan antara tujuan, kriteria, dan alternatif pada berbagai tingkatan. Setiap kelompok alternatif dapat dibagi menjadi tingkatan yang lebih rinci yang mencakup lebih banyak kriteria. Tingkat tertinggi dalam hierarki ini mewakili tujuan utama dalam pemecahan masalah dan terdiri dari hanya satu elemen. Tingkat berikutnya mungkin terdiri dari beberapa elemen sebagai kriteria, di mana setiap elemen dapat dibandingkan satu sama lain. Tapi jika ada perbedaan yang signifikan, tingkat yang lebih rendah harus dibuat di bawahnya, yang disebut sub kriteria.

2. Perbandingan penilaian/pertimbangan (*Comparative Judgments*)

Pada tahap ini, setiap elemen dalam hierarki dibandingkan satu sama lain untuk mengetahui tingkat kepentingan relatif dari masing-masing elemen. Perbandingan ini menghasilkan angka untuk skala penilaian. Ketika digabungkan, perbandingan ini akan menghasilkan prioritas yang didasarkan pada perhitungan vektor eigen.

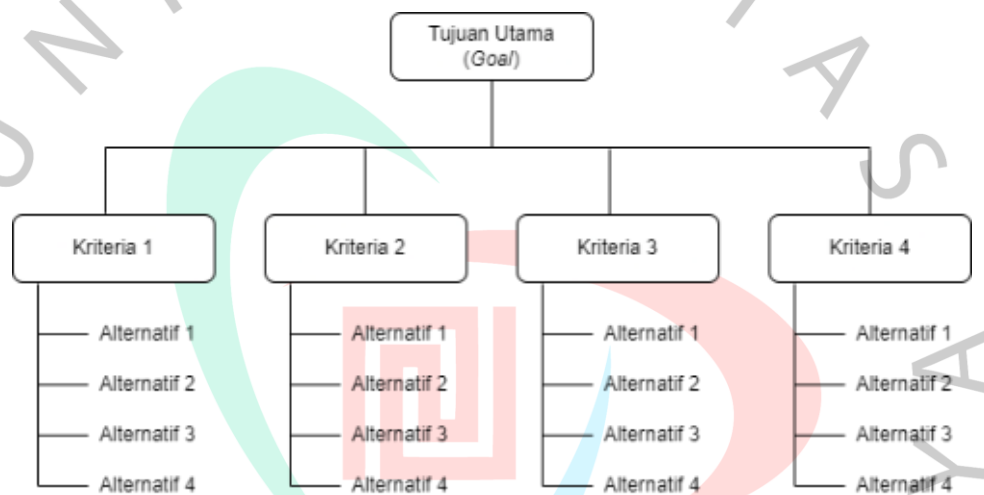
3. Sintesa Prioritas (*Priority Synthesis*)

Nilai prioritas lokal dihitung dengan nilai prioritas dari kriteria pada tingkat tertinggi, kemudian dijumlahkan untuk setiap elemen yang dipengaruhi oleh

kriteria tersebut. Nilai prioritas global diciptakan sebagai hasil dari proses sintesis prioritas, yang digunakan untuk menilai nilai prioritas lokal dari elemen pada tingkat terendah sesuai dengan kriterianya.

Menurut Putra & Diana (2021), terdapat langkah-langkah penyelesaian yang dilakukan menggunakan metode AHP:

1. Menjelaskan situasi masalah dan memberikan solusi, kemudian menyusun struktur hierarki.



Gambar 2.1 Struktur Hierarki AHP

2. Menampilkan matriks dalam angka desimal dan memberikan nilai prioritas pada elemen melalui perbandingan berpasangan yang didasarkan pada kriteria pada skala 1 hingga 9.

Tabel 2.2 Skala Penilaian AHP

Intensitas Kepentingan	Keterangan
1	Keduanya sangat penting.
3	Elemen satu sedikit lebih penting daripada yang lainnya.
5	Elemen satu lebih penting dari elemen lainnya.
7	Elemen satu sangat penting dari elemen lainnya.
9	Elemen satu mutlak sangat penting dari elemen lainnya.
2,4,6,8	Nilai-nilai dari dua nilai pertimbangan yang dekat satu sama lain.

Kebalikan	Jika aktivitas j menerima nilai yang berbeda dari aktivitas i, maka aktivitas j memiliki nilai yang berlawanan dengan nilai i.
-----------	--

3. Lakukan sintesis perbandingan untuk mendapatkan masing-masing prioritas. Setelah itu, nilai-nilai tersebut dimasukkan ke dalam matriks.

Bobot metode AHP dihitung dengan langkah berikut:

- a. Menampilkan matrik dalam angka desimal.
- b. Kalikan matrik tersebut dengan matriksnya sendiri.
- c. Menjumlahkan hasil proses perkalian matriks.
- d. Vektor eigen dibuat dengan menjumlahkan normalisasi baris dalam matrik, kemudian membagi jumlah baris dengan nilai akhir baris. Nilai yang dihasilkan secara rata-rata disebut vektor eigen.
- e. Merangkum nilai vektor eigen.

4. Mengevaluasi konsistensi perhitungan Index Konsistensi (CI) dengan menggunakan rumus berikut:

$$CI = (\lambda \text{ maks} - n) / (n - 1)$$

Dimana,

CI: Konsistensi index

λ Maks: Nilai eigen

N: Banyaknya kriteria

5. Kemudian menghitung *Consistency Ratio* (CR) dengan rumus:

$$CR = CI/RI$$

Dimana,

CR: Rasio konsistensi

CI: Indeks konsistensi

RI: Indeks acak

Berikut tabel indeks acak (RI):

Tabel 2. 3 Indeks Acak (RI)

Matriks Ordo	Indeks Acak (RI)
1	0,00

2	0,00
3	0,58
4	0,9
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49

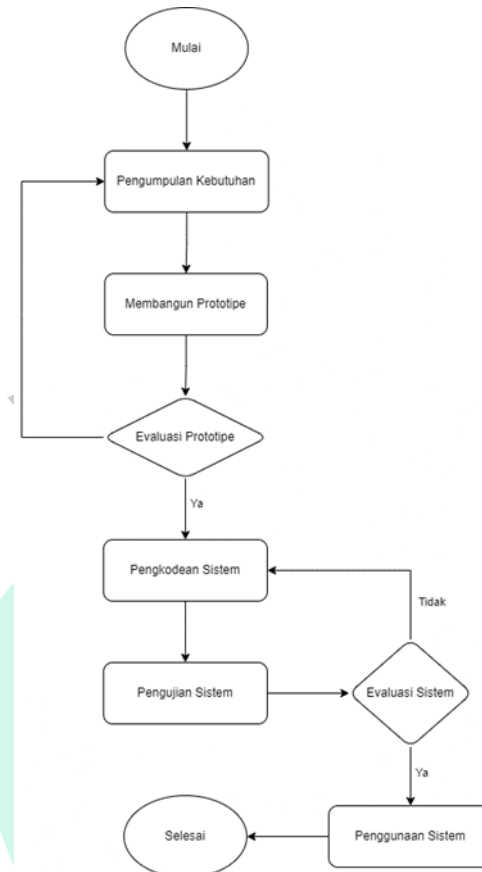
Apabila hasil rasio konsistensi (CR) lebih besar dari 10% atau 0,1, maka survei harus diulang. Jika hasil rasio konsistensi (CR) kurang dari 0,1, maka hasil perhitungan tersebut dianggap valid dan dapat diterima.

2.2.4 Metodologi Pengembangan Sistem (*Prototype*)

Metode pengembangan sistem prototipe, juga dikenal sebagai metode prototipe, digunakan untuk mengumpulkan data tentang kebutuhan pengguna secara cepat. Fokusnya adalah menyediakan elemen perangkat lunak yang dapat dilihat pengguna. Pengguna kemudian memeriksa protokol ini untuk mengidentifikasi kebutuhan pengembangan perangkat lunak tambahan (Pricillia, 2021).

Tujuan utama metode prototipe adalah menggambarkan fitur dan fungsionalitas dasar dari sistem yang diinginkan. Prototipe ini tidak harus sempurna atau lengkap, tetapi harus cukup representatif untuk memberikan gambaran yang jelas tentang bagaimana sistem akhir akan berfungsi.

Menurut Islami et al. (2023), berikut adalah tahapan-tahapan pada metode pengembangan prototipe.



Gambar 2.2 Tahapan Metode Pengembangan Prototipe

Berdasarkan Gambar 2.2 di atas, berikut adalah penjelasan mengenai tahapan-tahapan metode pengembangan prototipe:

1. Pengumpulan Kebutuhan

Pada tahap ini, analisis kebutuhan dilakukan secara menyeluruh untuk sistem yang akan dibangun, dengan mempertimbangkan kebutuhan pengguna dari perspektif user dan admin.

2. Membangun Prototipe

Pada tahap ini, membuat prototipe yang mencakup desain tampilan setiap halaman serta fungsi sistem yang akan dibangun.

3. Evaluasi Prototipe

Prototipe yang telah dibuat dievaluasi untuk memastikan kesesuaiannya dengan kebutuhan sistem yang akan dibangun.

4. Pengkodean Sistem

Pada tahap ini, kode program ditulis menggunakan bahasa pemrograman PHP dengan basis data MySQL.

5. Pengujian Sistem

Sistem yang telah dikodekan diuji secara menyeluruh untuk mengidentifikasi masalah atau *bug* yang mungkin ada. Pengujian ini mencakup skenario pengguna yang berbeda-beda dari segi fungsional dan non-fungsional untuk memastikan sistem berperilaku sesuai dengan harapan.

6. Evaluasi Sistem

Setelah pengujian sistem selesai, Tahap evaluasi ulang dilaksanakan dengan tujuan untuk memverifikasi bahwa sistem beroperasi sesuai dengan kebutuhan yang telah ditetapkan. Jika dalam tahapan evaluasi tersebut teridentifikasi adanya kekurangan atau kesalahan, langkah perbaikan akan dilakukan berulang sampai sistem berhasil mencapai tingkat kinerja yang optimal.

7. Penggunaan Sistem

Pada tahap akhir ini, sistem dianggap siap untuk digunakan.

2.2.5 Metode Pengembangan Sistem

Unified Modelling Language (UML) adalah sebuah alat yang digunakan untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain, dengan menggunakan sintaksis untuk memodelkan sistem secara visual. UML juga merupakan sekumpulan konvensi pemodelan yang digunakan untuk mendefinisikan atau menggambarkan sistem perangkat lunak yang terkait dengan objek (Ronal, Yunita, & Yuliana, 2022). Pada proses pengembangan perangkat lunak, UML membantu dalam merencanakan dan merancang solusi yang efektif dan efisien. Hal ini memungkinkan pengembang untuk mengidentifikasi kebutuhan, merancang struktur, dan mengantisipasi potensi masalah atau konflik sebelum implementasi sebenarnya dimulai.

UML terdiri dari berbagai jenis diagram yang masing-masing menggambarkan aspek khusus dari suatu sistem. Beberapa diagram UML yang umum digunakan antara lain diagram *use case*, diagram aktivitas, diagram kelas, dan diagram urutan.

Use Case Diagram

Kegunaan kasus diagram menunjukkan bagaimana satu atau lebih aktor berinteraksi dengan sistem informasi. Mereka digunakan untuk menentukan fungsi sistem informasi dan siapa yang memiliki hak untuk menggunakannya (Simatupang & Sianturi, 2019).

Berikut fungsi dari *use case diagram*:

- a) Menunjukkan berbagai aksi atau interaksi yang mungkin terjadi antar aktor dan sistem.
- b) Memberikan representasi visual tentang fungsionalitas utama yang akan disediakan oleh sistem.
- c) Menggambarkan bagaimana aktor berinteraksi dengan berbagai *use case* di dalam sistem.
- d) Membantu dalam berkomunikasi antara pengembang, pemangku kepentingan, dan tim proyek tentang kebutuhan dan fungsionalitas sistem.

Komponen-komponen pada *use case diagram*:

- a) Sistem

Komponen ini menetapkan batasan sistem dengan melibatkan aktor yang menggunakan sistem tersebut. Sistem diberi label yang sesuai, namun, umumnya tidak diwakili secara visual karena fokus lebih pada desain diagram.

- b) Aktor

Dalam *use case diagram*, entitas atau pengguna yang berinteraksi dengan sistem disebut sebagai aktor. Tergantung pada konteks sistem yang sedang dikembangkan, aktor dapat diwakili sebagai manusia atau objek.

- c) *Use Case*

Aktivitas atau tindakan yang dapat dilakukan oleh aktor terhadap sistem disebut sebagai *use case*. Biasanya, elemen ini digambarkan dalam bentuk oval.

Activity Diagram

Alur kegiatan sistem yang sedang direncanakan digambarkan dalam diagram aktivitas. Diagram ini menunjukkan bagaimana setiap alur dimulai, keputusan yang mungkin terjadi, dan proses paralel yang dapat terjadi dalam beberapa eksekusi (Sandfreni, Ulum & Azizah, 2021). Aktivitas diagram dirancang untuk merekam dinamika perilaku sistem dengan menggambarkan alur pesan yang mengalir dari satu aktivitas ke aktivitas lainnya.

Berikut fungsi dari *activity diagram*:

- a) Memberikan gambaran yang komprehensif, jelas, dan terstruktur tentang serangkaian aktivitas.
- b) Menampilkan urutan proses dalam sistem yang akan dilaksanakan.
- c) Merupakan diagram khusus yang memodelkan tahapan proses berdasarkan satu atau beberapa kasus.
- d) Mendukung penguraian algoritma sekuensial yang sangat kompleks.
- e) Menganalisis penggunaan sistem termasuk langkah-langkah atau keputusan yang harus diambil beserta momentum nya.
- f) Membantu dalam pemodelan aplikasi dengan sistem pemrosesan paralel.
- g) Berfungsi sebagai medium untuk menelusuri kebutuhan bisnis pada tingkat yang lebih mendalam.

Komponen-komponen pada *activity diagram*:

- a) *Initial State* (Keadaan Awal)

Initial state merujuk pada titik awal suatu alur kerja dalam *activity diagram* dan dalam satu diagram hanya terdapat satu keadaan awal.

- b) *Final State* (Keadaan Akhir)

Final state merupakan titik akhir dari alur kerja dalam suatu *activity diagram* dan satu *activity diagram* dapat memiliki lebih dari satu keadaan akhir.

- c) *Activity* (Aktivitas)

Activity merujuk pada pekerjaan atau tindakan yang dilakukan dalam alur kerja suatu sistem.

d) *Decision* (Keputusan)

Decision digunakan untuk menggambarkan kondisi yang memungkinkan terjadinya percabangan atau transisi yang berbeda, memastikan bahwa alur kerja dapat beralih ke jalur yang berbeda.

e) *Merge* (Menggabungkan)

Merge berfungsi untuk menggabungkan kembali jalur alur kerja yang sebelumnya dipisahkan oleh keputusan.

f) *Transition* (Transisi)

Transition mengindikasikan perpindahan dari satu aktivitas ke aktivitas berikutnya dalam alur kerja.

g) *Synchronization* (Sinkronisasi)

Sinkronisasi menunjukkan lokasi di diagram di mana aktivitas paralel dapat dijalankan bersama-sama. Sinkronisasi terdiri dari dua bagian: *fork* dan *join*. *Fork* memisahkan perilaku menjadi aktivitas paralel, sementara *join* menggabungkan kembali aktivitas tersebut.

Class Diagram

Dengan menggunakan definisi kelas yang akan dibentuk untuk membangun sistem, diagram kelas menunjukkan struktur sistem. Selain itu, diagram ini menunjukkan aturan dan tanggung jawab entitas yang memengaruhi perilaku sistem. Diagram kelas berfungsi sebagai berikut:

- a) Memberikan representasi visual tentang struktur dan komposisi kelas-kelas dalam sistem perangkat lunak. Setiap kelas diilustrasikan dengan nama kelas dan atribut-atributnya.
- b) Menggambarkan hubungan dan asosiasi antar kelas, baik satu-ke-satu, satu-ke-banyak, atau banyak-ke-banyak, membantu memahami bagaimana objek berinteraksi satu sama lain.
- c) Menunjukkan hierarki warisan (*inheritance*) antar kelas dan hubungan polimorfisme. Ini membantu menggambarkan bagaimana kelas-kelas dapat mewarisi sifat-sifat atau perilaku dari kelas lain.
- d) Memberikan atribut dan metode setiap kelas, membantu pemahaman struktur data dan fungsionalitas kelas.

- e) Memberikan pemodelan objek yang memberikan gambaran konseptual tentang komponen yang akan dibuat dalam sistem. Diagram kelas membantu dalam pemikiran dan desain struktur kelas secara keseluruhan.
- f) Digunakan selama fase analisis dan perancangan sistem untuk mengidentifikasi dan menyusun komponen-komponen utama serta hubungan antar mereka.
- g) Membantu dalam menjaga konsistensi antara tim pengembang dengan memberikan pandangan visual yang dapat dipahami bersama tentang struktur kelas dan interaksi antar mereka.
- h) Menyediakan dokumentasi yang berguna untuk memahami arsitektur sistem dan membantu dalam pengembangan, pemeliharaan, dan perubahan sistem pada tahap-tahap selanjutnya.

Komponen-komponen pada *class diagram*:

- a) *Class* (Kelas)
Class adalah elemen dasar dalam *class diagram* yang mewakili suatu tipe objek atau entitas dalam sistem.
- b) *Attribute* (Atribut)
Atribut karakteristik yang dimiliki oleh suatu kelas yang menunjukkan data yang dimiliki oleh objek-objek yang termasuk dalam kelas tersebut.
- c) *Method* (Metode)
Metode adalah fungsi atau operasi yang dapat dilakukan oleh suatu kelas. Metode juga mendefinisikan perilaku atau tindakan yang dapat dilakukan oleh objek.
- d) *Association* (Asosiasi)
Association menggambarkan hubungan antar dua atau lebih kelas dalam sistem. Ini menunjukkan bagaimana objek dari satu kelas berinteraksi atau terkait dengan objek dari kelas lain.
- e) *Composition* (Komposisi)

Komposisi adalah jenis asosiasi khusus yang menunjukkan bahwa objek dari satu kelas adalah bagian penting dari objek dari kelas lain jika objek induk dihapus.

f) *Inheritance* (Warisan)

Inheritance menunjukkan hubungan hierarki antar kelas, di mana suatu kelas dapat mewarisi atribut dan metode dari kelas lain.

g) *Aggregation* (Agregasi)

Aggregation adalah tipe asosiasi yang menunjukkan hubungan keseluruhan dan bagian antara dua kelas. Objek dalam hubungan *aggregation* tetap dapat eksis ketika objek keseluruhan dihapus.

h) *Abstract Class* (Kelas Abstrak)

Kelas abstrak adalah kelas yang tidak dapat diinstansiasi dan biasanya mengandung teknik abstrak yang harus digunakan oleh kelas turunannya.

Sequence Diagram

Urutan Diagram menggunakan rentang waktu hidup objek dan pesan yang dikirimkan dan diterima untuk menunjukkan perilaku objek dalam suatu kegunaan kasus (Simatupang & Sianturi, 2019). Tujuan utama pembuatan *sequence diagram* adalah untuk mendapatkan pemahaman tentang urutan kejadian yang menghasilkan keluaran yang diinginkan. Tujuan pembuatan *sequence diagram* ini juga sebanding dengan tujuan pembuatan diagram aktivitas, yang mencakup penggambaran alur kerja dari suatu aktivitas dan kemampuan untuk merinci aliran data dengan lebih rinci, termasuk data atau perilaku yang diterima atau dikirimkan.

Berikut fungsi dari *sequence diagram*:

- a) Memberikan representasi visual tentang urutan kejadian atau interaksi antar objek dalam suatu skenario atau *use case*.
- b) Mendeskripsikan rentang waktu hidup objek dalam sistem, mulai dari penciptaan objek hingga penghancuran atau selesai digunakan.
- c) Menggambarkan pesan yang dikirim dan diterima antara objek dalam urutan waktu tertentu, menunjukkan bagaimana objek berinteraksi satu sama lain.

- d) Menunjukkan koordinasi dan sinkronisasi antar objek dalam suatu skenario atau *use case*, membantu dalam pemahaman alur logika dan pemrosesan sistem.
- e) Membantu mengidentifikasi potensi kesalahan atau ketidakjelasan dalam desain atau logika sistem dan memfasilitasi pemecahan masalah.
- f) Digunakan untuk memodelkan logika bisnis atau proses dalam suatu skenario, membantu pengembang dan pemangku kepentingan memahami bagaimana sistem berperilaku.
- g) Menyediakan dokumentasi visual yang dapat dipahami dengan mudah untuk tim pengembang dan pemangku kepentingan, memfasilitasi komunikasi yang efektif dalam pengembangan perangkat lunak.

Komponen-komponen pada *sequence diagram*:

- a) *Actor* (Aktor)
Pengguna yang tidak termasuk dalam lingkup sistem dan berinteraksi dengannya disebut aktor. Simbol *stick figure* biasanya digunakan untuk mewakili aktor dalam *sequence diagram*.
- b) *Activation Box* (Kotak Aktivasi)
Kotak aktivasi terletak di atas lifeline dan membantu melihat durasi aktivitas objek. Ini menunjukkan periode waktu ketika objek sedang aktif atau menjalankan suatu operasi.
- c) *Lifeline* (Waktu Hidup Objek)
Lifeline menggambarkan rentang waktu hidup objek dalam suatu skenario. Ini adalah garis vertikal yang mengindikasikan waktu objek aktif dalam interaksi.
- d) Objek
Objek adalah entitas yang berpartisipasi dalam interaksi. Objek direpresentasikan oleh nama kelas atau instance dari kelas tertentu, tergantung pada apakah objek tersebut merupakan kelas atau instance.
- e) *Message* (Pesan)

Message menggambarkan komunikasi atau interaksi antar objek. *Message* diberi label dengan notasi panah dan dapat mencakup informasi tambahan seperti parameter atau nilai kembalian.

2.2.6 Metode Pengembangan Sistem Berbasis *Website*

Web juga dikenal sebagai situs web adalah kumpulan halaman media yang saling terhubung melalui hyperlink. Informasi dapat dikirim melalui situs web dalam berbagai bentuk, seperti teks, gambar, video, suara, animasi, atau kombinasi dari semua ini. Adanya halaman web yang saling terkait, adanya domain sebagai alamat URL, dan adanya hosting yang berfungsi sebagai penyimpanan data adalah karakteristik utama dari situs web. Peramban seperti Chrome, Firefox, Internet Explorer, dan Opera adalah beberapa contoh peramban yang dapat digunakan untuk mengakses situs web melalui jaringan internet.

- Bahasa pemrograman seperti *Hypertext Preprocessor* (PHP) biasanya digunakan untuk membuat situs web atau aplikasi berbasis web. Bahasa pemrograman ini biasanya digunakan bersama dengan JavaScript, *Cascading Style Sheet* (CSS), dan *Hypertext Markup Language* (HTML).

***Hypertext Markup Language* (HTML)**

Hypertext Markup Language (HTML) adalah bahasa markah yang digunakan untuk membuat dan merancang struktur halaman web. Dokumen HTML adalah dokumen teks yang disimpan dengan ekstensi file.html dan terdiri dari berbagai elemen yang dikelilingi oleh tag, yang dimulai dengan simbol "<" dan diakhiri dengan simbol ">" (Sari & Abdilah, 2019).

HTML terdiri dari berbagai elemen yang membangun struktur dasar halaman web, seperti elemen paragraf, heading, daftar, tautan, gambar, tabel, dan formulir. Elemen-elemen ini memungkinkan desainer web untuk mengatur konten secara terstruktur dan mudah diakses melalui browser. Selain itu, HTML sering digunakan bersama dengan CSS (*Cascading Style Sheets*) dan JavaScript untuk memperbaiki tampilan dan meningkatkan interaktivitas halaman web. CSS berfungsi mengatur gaya dan tata letak elemen HTML, sementara JavaScript

menambahkan fungsi dinamis ke halaman web, seperti respons terhadap interaksi pengguna dan manipulasi konten secara real-time.

Hypertext Preprocessor (PHP)

Hypertext Preprocessor adalah bahasa pemrograman sisi server yang dibuat untuk pengembangan web. PHP dapat dimasukkan ke dalam kode HTML dan dijalankan di server web, sehingga menghasilkan konten dinamis yang ditampilkan di browser pengguna. Penggunaan paling umum PHP adalah untuk membuat halaman web yang dapat berinteraksi dengan basis data, mengelola formulir, menghasilkan konten dinamis, dan menangani berbagai tugas pemrograman sisi server lainnya (Al Hadi, 2024).

Selain itu, bahasa pemrograman PHP juga mendukung berbagai jenis basis data, seperti MySQL, yang memungkinkan pengguna mengakses dan mengelola data dengan mudah. Berikut beberapa kelebihan bahasa pemrograman PHP:

1. PHP dijalankan di server, bukan di klien. Keluaran PHP adalah HTML, yang kemudian dikirim ke browser pengguna. Keuntungan ini memberikan kontrol yang lebih baik atas sumber daya server dan memungkinkan pengembang untuk menyembunyikan implementasi detail pengguna.
2. PHP relatif mudah dipelajari, terutama bagi mereka yang sudah memiliki dasar pemrograman web. Syntax PHP mirip dengan C dan Java, yang membuatnya mudah dipahami oleh banyak pengembang.
3. PHP bersifat open source dan gratis. Hal ini memudahkan pengembang untuk mengakses kode sumber, melakukan modifikasi, dan berbagi dengan komunitas.
4. PHP memiliki kemampuan untuk terintegrasi dengan sejumlah sistem basis data populer seperti MySQL, PostgreSQL, dan Oracle, dan kemampuan ini sangat penting untuk mengembangkan aplikasi web yang membutuhkan pengelolaan dan penyimpanan data.

MySQL

MySQL adalah akronim untuk sistem manajemen basis data relasional yang dimaksudkan untuk menyimpan, mengelola, dan mengatur data dalam bentuk tabel yang terstruktur. Untuk berinteraksi dengan basis data, MySQL menggunakan bahasa SQL (Bahasa Pertanyaan Struktural) yang memudahkan pengguna untuk membuat, mengakses, memperbarui, dan menghapus data. MySQL memiliki setiap tabel dengan baris yang menunjukkan entitas dan kolom yang menunjukkan atribut atau karakteristik entitas tersebut. Selain itu, MySQL mendukung transaksi, yang merupakan bagian penting dari pengembangan sistem informasi (Maulana, 2023).

Berikut adalah fitur utama dari MySQL:

1. MySQL mendukung prinsip ACID (*Atomicity, Consistency, Isolation, Durability*), yang menjamin keandalan dan konsistensi transaksi.
2. MySQL mendukung berbagai jenis penyimpanan tabel seperti MyISAM, InnoDB, MEMORY, dan lainnya.
3. MySQL memungkinkan penggunaan indeks untuk meningkatkan kecepatan pencarian dan pengurangan waktu eksekusi kueri.
4. MySQL menyediakan alat-alat administrasi yang efisien seperti MySQL Workbench untuk memudahkan pengelolaan basis data, membuat dan mengelola skema, dan menjalankan kueri.
5. MySQL mendukung pembatasan data dan trigger yang memungkinkan pengembang untuk menentukan aturan perilaku tertentu pada tingkat basis data.

Apache

Apache yang dikembangkan dan dikelola oleh Apache Software Foundation adalah salah satu server web yang paling populer dan paling banyak digunakan di seluruh dunia. Apache menyediakan lingkungan server yang stabil dan aman untuk meng-host situs web dan aplikasi web. Ini adalah fungsi-fungsi utama Apache:

1. Apache memiliki kemampuan untuk menghost berbagai macam situs web, mulai dari situs web statis sederhana hingga aplikasi web dinamis yang kompleks.

2. Apache memiliki struktur modular yang memungkinkan administrator untuk menambahkan dan konfigurasi modul sesuai kebutuhan, seperti modul untuk keamanan, autentikasi, *caching*, dan lain-lain.
3. Apache menyediakan berbagai fitur keamanan, termasuk kontrol akses, autentikasi, serta SSL/TLS untuk memastikan komunikasi terenkripsi.

Windows Server

Windows server adalah sistem operasi server yang dikembangkan oleh Microsoft untuk digunakan pada server, yaitu perangkat yang menyediakan layanan dan dukungan untuk jaringan komputer. Windows server memiliki banyak fitur dan fungsi yang membantu organisasi mengelola aplikasi, layanan, dan jaringan dengan lebih baik. Ini adalah fitur utama Windows Server:

1. Windows server menyediakan layanan *Active Directory* (AD) untuk mengelola pengguna dan sumber daya jaringan. AD mempermudah administrator dalam mengatur pengguna, komputer, dan grup di lingkungan jaringan.
2. Teknologi virtualisasi *Hyper-V* memungkinkan pengguna untuk menjalankan berbagai sistem operasi pada satu server fisik, yang meningkatkan efisiensi dan pemanfaatan sumber daya.
3. *Internet Information Service* adalah server web yang termasuk dalam windows server, digunakan untuk meng-*host* situs web dan aplikasi web.
4. Windows server menyediakan *remote desktop service* yang memungkinkan akses jarak jauh ke desktop dan aplikasi yang berjalan di server.

2.2.7 Metode Pengujian Sistem

Proses pengujian diperlukan untuk memastikan kualitas dan kesesuaian sistem perangkat lunak yang dikembangkan sesuai dengan spesifikasi yang telah ditetapkan. Alat pengujian yang digunakan adalah Kotak Hitam dan Kotak Putih.

Pengujian Kotak Hitam

Pengujian kotak hitam adalah teknik pengujian perangkat lunak yang tidak membutuhkan informasi internal atau rincian implementasi sistem yang diuji. Pengujian kotak hitam fokus pada masukan dan keluaran program tanpa mempertimbangkan rincian proses aplikasi (Nurfauziah & Jamaliyah, 2022). Perangkat lunak ini dianggap sebagai kotak hitam pengujian, di mana masukan dan keluaran dinilai. Pengujian metode *Black Box* dilakukan untuk memastikan apakah perangkat lunak atau aplikasi memenuhi standar kelayakan, sesuai dengan spesifikasi, dan memenuhi kebutuhan pengguna. Tujuan dari proses ini adalah untuk mencegah masalah yang dapat muncul saat menggunakan perangkat lunak atau aplikasi, sehingga masalah tersebut dapat dihindari di masa mendatang.

Pengujian Kotak Putih

Pengujian kotak putih adalah teknik yang memperhatikan komponen internal sistem, terutama kode sumber program. Pengujian kotak putih sangat penting bagi pengembang perangkat lunak untuk menilai seberapa kompleks suatu kode (Dhaifullah, Salsabila, & Yaqin, 2022). Pengujian kotak putih juga dapat digunakan untuk memastikan sumber kode sesuai dengan desain, memenuhi persyaratan fungsional, dan bebas dari kerentanan.

Menurut Nurfauziah & Jamaliyah (2022), berikut terdapat beberapa kelebihan dari *white box testing*:

1. Meningkatkan akurasi dalam implementasi perangkat lunak.
2. Mempermudah identifikasi kesalahan atau bug dalam perangkat lunak yang sebelumnya tidak terdeteksi.
3. Memudahkan proses pengujian secara menyeluruh, mengurangi kemungkinan terjadinya kesalahan pada kode.
4. Mengurangi potensi kesalahan atau bug karena pengujian dapat dilakukan sebelum perangkat lunak diluncurkan.