

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Kata sistem berasal dari bahasa Yunani sistem yang artinya kesatuan. Faktor lingkungan mempunyai pengaruh yang besar terhadap sistem. Sistem dan lingkungan mempunyai kaitan yang erat. Sistem merupakan sebuah kumpulan unsur, komponen, maupun variabel yang terorganisir, saling bergantung, serta saling berinteraksi satu dengan lainnya (Ahdiyenti, 2022).

2.1.1 Pengertian Sistem

Sistem merupakan kumpulan elemen-elemen yang saling terkait antara satu dengan yang lain yang tak dapat dipisahkan, untuk mencapai satu tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berintegrasi saling tergantung satu sama lain, dan terpadu. Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk satu tujuan (Hendrastuty, N, 2022).

2.1.2 Karakteristik Sistem

Karakteristik yang mendefinisikan sebuah sistem yang memiliki tujuan atau sasaran yang ingin dicapai sebagai panduan utama, sistem terdiri atas elemen-elemen yang saling berinteraksi dan bekerja sama untuk mendukung

keberlangsungan sistem. beberapa karakteristik sistem adalah sebagai berikut (Afira, 2023):

1. Komponen atau elemen (*Components*)

Suatu sistem tidak mungkin ada dalam lingkungan yang kosong, tetapi suatu sistem ada dan memiliki fungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem juga terdiri dari beberapa bagian yang saling berinteraksi satu sama lain dan melakukan kerja sama dalam membentuk satu kesatuan.

2. Batas Sistem (*Boundary*)

Batas sistem adalah daerah yang memisahkan antara sistem dengan lingkungan luarnya atau dengan sistem lain. Batas ini penting untuk menentukan ruang lingkup kerja suatu sistem. Batas sistem juga membantu membedakan komponen internal yang termasuk ke dalam sistem dan elemen eksternal yang berada di luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan adalah apapun di luar batas dari sebuah sistem yang dapat mempengaruhi operasi dari sistem tersebut, baik pengaruh yang merugikan ataupun yang menguntungkan. Pengaruh yang merugikan ini tentunya harus ditahan dan dikendalikan sehingga tidak mengganggu keberlangsungan sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan suatu media antara satu subsistem dengan subsistem lainnya yang membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem lainnya.

5. Masukkan (*Input*)

Input adalah energi atau sesuatu yang dimasukkan kedalam suatu sistem yang dapat berupa masukkan yaitu energi yang dimasukkan supaya sistem dapat beroperasi atau masukkan sinyal yang merupakan energi yang diproses untuk menghasilkan suatu luaran.

6. Luaran (*Output*)

Luaran adalah hasil dari proses pengolahan dalam sistem, baik berupa produk, informasi, maupun layanan. Output ini merupakan tujuan akhir dari keberadaan suatu sistem. Kualitas output sangat bergantung pada kualitas input dan proses yang terjadi di dalam sistem.

7. Pengolah (*Process*)

Pengolah adalah bagian dari sistem yang berfungsi untuk mengubah input menjadi output. Proses ini mencakup serangkaian langkah atau operasi yang dirancang untuk mencapai hasil yang diinginkan. Tanpa proses, input tidak dapat diubah menjadi sesuatu yang bermanfaat.

8. Sasaran (*Objective*)

Sebuah sistem pasti mempunyai sasaran (objective) atau tujuan (goal). Jika suatu sistem tidak mempunyai tujuan, maka operasi dari sistem tersebut tidak akan ada gunanya. Tujuan inilah yang mengarahkan kemana suatu sistem tersebut berjalan.

2.1.3 Pengertian Sistem Informasi

Informasi merupakan salah satu sumber daya yang sangat diperlukan dalam suatu organisasi. Informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Nst, 2023).

2.1.4 Komponen Sistem Informasi

Dalam sistem informasi, komponen-komponennya sering disebut sebagai blok bangunan terdiri dari bagian-bagian yang diucap dengan blok gedung (*building block*), selaku sesuatu sistem blok itu saling berhubungan antara yang satu dengan yang lain serta membuat satu kesatuan untuk menggapai sasaran yang dituju, sebagai berikut (Dewi Suci Rahmadhani, 2022):

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan *input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan *toolbox* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, serta menghasilkan dan mengirimkan keluaran. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya yang tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi seperti bencana alam, api, temperature, air, debu, kecurangan, kegagalan, kesalahan, dan sabotase. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun jika terlanjur terjadi kesalahan dapat langsung cepat diatasi.

2.2 Kecerdasan Buatan

Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia (Hanila & Alghaffaru, 2023).

Kecerdasan buatan yang ditunjukkan oleh suatu entitas. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia (Na'am & Rahmawati, 2023).

Artificial Intelligence (Inteligensi/Kecerdasan Buatan) merupakan salah satu bidang dari ilmu komputer yang membahas tentang kemungkinan komputer untuk dapat berlaku secara intelligen seperti halnya manusia.

Artificial Intelligence (AI) merupakan sub bidang pengetahuan computer yang khusus ditujukan untuk membuat *software* dan *hardware* yang sepenuhnya bisamenirukan beberapa fungsi otak manusia (Renhoran, 2022).

2.2.1 Bidang Bidang Kecerdasan Buatan

Kecerdasan diciptakan dan ditanamkan ke dalam mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain (Na'am & Rahmawati, 2023):

1. Sistem pakar

Sistem pakar adalah sistem komputer yang meniru kemampuan berpikir dan mengambil keputusan seorang ahli dalam suatu bidang tertentu. Sistem ini menggunakan basis pengetahuan (knowledge base) dan mesin inferensi (inference engine) untuk memberikan solusi atau rekomendasi dalam menyelesaikan masalah.

2. Permainan komputer (*games*)

Permainan komputer adalah aplikasi interaktif yang memungkinkan pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu, seperti menyelesaikan teka-teki, bertarung, atau membangun dunia virtual. Game sering menggunakan grafis, animasi, dan kecerdasan buatan untuk menciptakan pengalaman yang menarik dan realistis.

3. Logika *fuzzy*

Logika fuzzy adalah bentuk logika yang memungkinkan penanganan konsep ketidakpastian dan ambiguitas dengan menggunakan nilai kebenaran yang kontinu antara 0 dan 1, alih-alih hanya dua nilai (benar atau salah).

4. Jaringan syaraf tiruan

Jaringan syaraf tiruan adalah model komputasi yang meniru cara otak manusia bekerja, digunakan untuk memecahkan masalah seperti pengenalan pola, klasifikasi data, dan prediksi. Jaringan ini terdiri dari lapisan neuron yang terhubung, dengan masing-masing neuron mengolah informasi dan menghasilkan output berdasarkan bobot yang telah dilatih.

5. Robotika

Robotika adalah bidang ilmu yang berfokus pada desain, konstruksi, pengoperasian, dan aplikasi robot. Robot digunakan untuk menggantikan atau melengkapi tugas manusia dalam berbagai industri, mulai dari manufaktur hingga layanan kesehatan.

Kecerdasan buatan adalah model statistik yang digunakan untuk mengambil keputusan dengan menggeneralisir karakteristik dari suatu objek berbasis data yang kemudian dipasang di berbagai perangkat elektronik.

2.3 Sistem Pakar

Sistem pakar adalah suatu program komputer atau sistem informasi yang mengandung beberapa pengetahuan dari satu atau lebih pakar manusia terkait suatu bidang yang cenderung spesifik (Zamroni & Wahyudi, 2022).

Pakar yang dimaksudkan merupakan seseorang yang memiliki keahlian khusus di bidangnya masing – masing, contohnya dokter, psikolog, mekanik, dan lain sebagainya. Perangkat lunak ini pertama kali dikembangkan oleh periset program kecerdasan buatan (AI) sekitar tahun 1960-an dan 1970-an, serta baru diterapkan pada tahun 1980-an (Zamri, dkk., 2022).

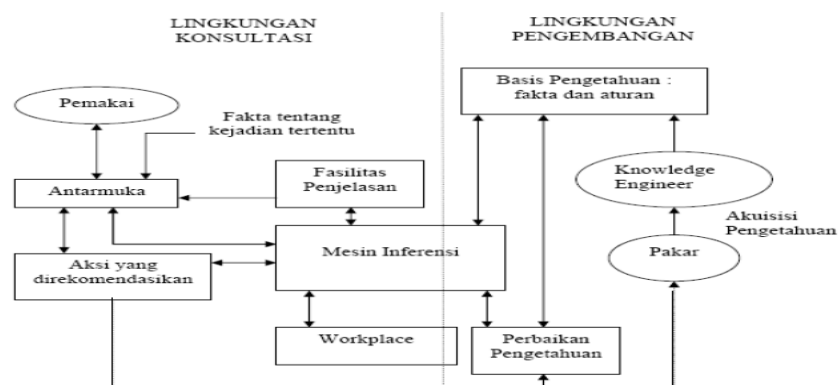
2.3.1 Konsep Dasar Sistem Pakar

Sistem pakar adalah program komputer cerdas yang menggunakan pengetahuan dan prosedur-prosedur inferensi untuk menyelesaikan masalah masalah yang membutuhkan seorang pakar dalam menemukan solusinya.

sistem pakar merupakan sebuah sistem yang mengemulasikan kemampuan membuat keputusan dari seorang pakar (Josua, 2022).

2.3.2 Komponen Sistem Pakar

Pengembangan *expert system*, tersusun atas beberapa komponen atau struktur pembentuk sebuah sistem informasi yang komprehensif. Berikut ini merupakan beberapa bagian penyusun arsitektur dari sistem ini (Muhammad Robith Adani, 2021):



Gambar 2.3.2 Komponen Sistem Pakar

1. *User Interface* (Antarmuka Pengguna)

Antarmuka atau *interface* merupakan mekanisme yang digunakan sebagai sarana untuk berkomunikasi dan berinteraksi dengan pengguna (*user*). Antarmuka akan menerima informasi dari pengguna, dan akan mengubahnya ke dalam instruksi yang dapat diterima oleh sistem.

2. Basis Pengetahuan

Basis pengetahuan mengandung pemahaman mengenai formulasi dan skema penyelesaian masalah.

3. *Knowledge Acquisition* (Akuisisi Pengetahuan)

Knowledge acquisition adalah proses akumulasi, transformasi, dan transfer tiap keahlian untuk dapat menyelesaikan permasalahan dari sumber pengetahuan, ke dalam suatu sistem komputer. Pada tahap ini, seorang *engineer* bertugas untuk menyerap segala pengetahuan untuk dikirim ke dalam basis pengetahuan (*insight*).

4. *Inference Engine* (Mesin atau Motor Inferensi)

Pada komponen ini mengandung mekanisme penalaran dan pola pikir yang dimanfaatkan oleh para pakar untuk dapat memecahkan suatu masalah dengan baik. Mesin inferensi sendiri merupakan program komputer untuk memberikan metodologi yang ada dalam *workplace*, dan nantinya akan diolah menjadi sebuah kesimpulan.

5. *Workplace* / Blackboard

Workplace merupakan area dari kumpulan memori kerja yang digunakan untuk merekam setiap kejadian yang ada, termasuk pembuatan keputusan sementara.

6. Fasilitas Penjelasan

Fasilitas penjelasan termasuk ke dalam komponen tambahan untuk meningkatkan penggunaan sistem pakar, serta melacak respon dan hasil penjelasan mengenai tingkah laku pada *expert system* secara interaktif.

7. Perbaikan Pengetahuan

Pakar juga mempunyai kemampuan analisis yang baik untuk dapat meningkatkan kinerjanya sedemikian rupa. Kemampuan tersebut terdiri atas, keahlian dalam pembelajaran yang terkomputerisasi. Sehingga, program dapat membedakan antara kesuksesan dengan kegagalan yang dialami, berdasarkan pengetahuan yang masih relevan untuk diaplikasikan di masa mendatang.

2.4 Unified Modelling Language (UML)

UML (*Unified Modelling Language*) merupakan Bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Suryanto & Baydhowi, 2022).

2.4.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Use case

digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Alam, dkk., 2022).


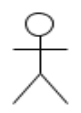

Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case (Syaputra & Budiman, 2021).

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Simbol-simbol yang digunakan pada diagram use case dapat dilihat pada

Tabel 2.1 Berikut ini :

Tabel 2.1 Simbol Use Case Diagram

No	Simbol	Keterangan
1.	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	Aktor/ <i>actor</i> 	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
3.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan usecase yang berpartisipasi pada usecase atau

No	Simbol	Keterangan
		usecase memiliki interaksi dengan aktor.
4.	Ekstensi/ <i>extend</i> ----->	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
5.	Generalisasi/ <i>generalization</i> —————>	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum darinya.
6.	Menggunakan/ include / uses <<include>> —————> <<uses>> —————>	Relasi use case tambahan ke sebuah usecase dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case.

Sumber : Kusumo, 2021



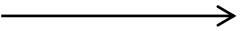
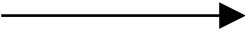
2.4.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan dibuat untuk merancang sistem. Kelas memiliki atribut dan metode atau proses. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Class Diagram dibuat agar programmer membuat kelas-kelas sesuai rancangan di

dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron (Saputro, 2021).

Simbol-simbol yang digunakan pada diagram kelas dapat dilihat pada Tabel 2.2 berikut ini :

Tabel 2.2 Simbol Class Diagram

No	Simbol	Keterangan
1.	<div> <div>Kelas</div> <div> <div>nama_kelas</div> <div>+atribut</div> <div>+operasi()</div> </div> </div>	Kelas pada struktur sistem.
2.	<div>Antarmuka / <i>interface</i></div> <div>  </div>	Sama dengan konsep interfacedalam pemrograman berorientasi objek.
3.	<div>Asosiasi / <i>association</i></div> <div>  </div>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai multiplicity.
4.	<div>Asosiasi berarah/ <i>directed association</i></div> <div>  </div>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5.	<div>Generalisasi</div> <div>  </div>	Relasi antar kelas dengan makna generalisasi - spesifikasi (umum - khusus).

Sumber : Kusumo et al., 2021

2.4.3 Activity Diagram



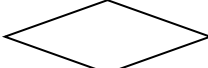


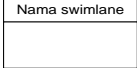

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa activity diagram menggambarkan

aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Oktavia, dkk., 2021).

Simbol-simbol yang digunakan pada activity diagram dapat dilihat pada

Tabel 2.3

Tabel 2.3 Simbol Activity Diagram

No	Simbol	Keterangan
1.	Status awal 	Status awal aktifitas sistem, sebuah diagram aktifitas memiliki sebuah status awal.
2.	aktivitas 	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Assosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Assosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Farhan & Leman, 2023



2.4.4 Sequence Diagram

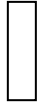


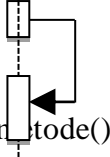


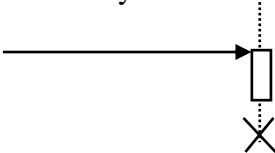
Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan sequence diagram maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat sequence diagram juga dibutuhkan untuk melihatkan skenario yang ada pada use case (Puspitasari & Budiman, 2021).

Simbol-simbol yang digunakan pada diagram sequence dapat dilihat pada

Tabel 2.4

Tabel 2.4 Simbol Sequence Diagram

No	Simbol	Keterangan
1.	Aktor  nama aktor atau <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama</div>	Orang, Proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda di awal fase nama aktor.
2.	Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	Objek <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama objek : nama kelas</div>	Menyatakan objek yang berinteraksi pesan.

No	Simbol	Keterangan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.	Pesan tipe <i>create</i> <code><<create>></code> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.
6.	Pesan tipe <i>call</i> <code>1 : nama_metode()</code>  	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.	Pesan tipe <i>send</i> <code>1: masukan</code> 	Menyatakan suatu objek bahwa mengirim data, masukan dan informasi ke objek lain, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang dikirim.
9.	Pesan tipe <i>destroy</i> <code><<destroy>></code> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.




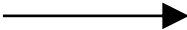
Sumber : Augustiningrum & Puspaningrum, 2021

2.4.5 State Mechine Diagram

State Machine Diagram adalah sebuah teknik untuk menggambarkan perilaku dinamis sebuah sistem, dalam pendekatan OO, sebuah state machine diagram memodelkan perilaku dari sebuah objek tunggal, menunjukkan urutan kejadian yang terjadi pada sebuah objek selama hidupnya dalam meresponse suatu event, Elemen dasar adalah state dan transisi dari satu state ke state lainnya (Wiyono, 2022).

Simbol-simbol yang digunakan pada state machine diagram dapat dilihat pada Tabel 2.5

Tabel 2.5 Simbol State Machine Diagram

No	Simbol	Keterangan
1.	State 	State, digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisi saat itu
2.	Titik Awal 	Awal (start), digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram statechart
3.	Titik Akhir 	Titik akhir (end), digunakan untuk menggambarkan akhir dari kejadian dalam suatu diagram statechart
4.	Guard [guard]	Guard, yang merupakan syarat terjadinya transisi yang bersangkutan
5.	Point 	Point, digunakan untuk menggambarkan apakah akan masuk (entry point) ke dalam state atau akan keluar (exit point).

No	Simbol	Keterangan
6.	Event	Event, digunakan untuk mendiskripsikan kondisi yang menyebabkan sesuatu pada state

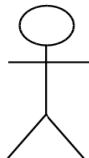
Sumber : Wiyono, 2022



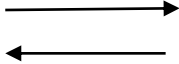
2.4.6 Collaboration Diagram

Diagram kolaborasi digunakan untuk menunjukkan bagaimana objek berinteraksi untuk melakukan perilaku kasus penggunaan tertentu, atau bagian dari kasus penggunaan. Bersama dengan diagram sekuens, kolaborasi digunakan oleh desainer untuk mendefinisikan dan mengklarifikasi peran objek yang melakukan aliran peristiwa tertentu dari kasus penggunaan. Diagram ini merupakan sumber informasi utama yang digunakan untuk menentukan tanggung jawab dan antarmuka kelas (Fikri M, 2022).

Simbol-simbol yang digunakan pada collaboration diagram dapat dilihat pada Tabel 2.6

Tabel 2.6 Simbol Collaboration Diagram

No	Simbol	Keterangan
1.	Aktor 	Aktor : sebuah instansi aktor muncul dalam diagram kolaborasi, sebagai pemanggil interaksi. Jika Anda memiliki beberapa instansi aktor dalam diagram yang sama, cobalah untuk menyimpannya di pinggiran diagram.

No	Simbol	Keterangan
2.	Object instance 	Object instance: Obyek yang dibuat, melakukan tindakan, dan / atau dimusnahkan selama lifeline
3.	Link 	Interaksi link: Merupakan indikasi bahwa obyek kejadian dan berkolaborasi aktor dan pertukaran pesan.
4.	Sinkronis Pesan 	Sinkronis pesan: Seketika sebuah komunikasi antara objek-objek yang menyampaikan informasi, dengan harapan bahwa tindakan akan dimulai sebagai hasil.

Sumber : Apriadi, 2022

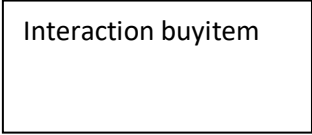
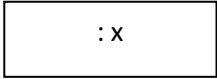
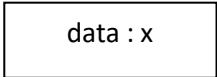
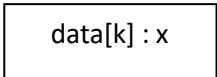
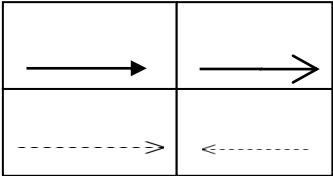
2.4.7 Communication Diagram

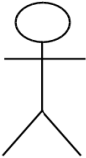

Communication diagram merupakan merupakan model komunikasi diagram interaksi antara benda atau bagian untuk mengurutkan suatu proses. Fokusnya pada komunikasi yang berhubungan dengan struktur dari objek yang terlibat dalam tugas.

Communication diagram menggunakan pengaturan bebas berupa objek dan link seperti yang digunakan dalam diagram objek. Dalam rangka mempertahankan pesan dari suatu diagram, pesan diberi label dengan nomor kronologis dan ditempatkan di dekat link pesan yang dikirim. Cara untuk membaca communication diagram yaitu mulai dari pesan 1 dan mengikuti pesan dari objek ke objek.

Simbol-simbol yang digunakan pada communication diagram dapat dilihat pada Tabel 2.7

Tabel 2.7 Simbol Communication Diagram

No	Simbol	Keterangan
1.	<p>Frame</p> 	Frame communication diagram dapat ditampilkan dalam bingkai persegi panjang dengan nama dalam kotak terpisah di sudut kiri atas
2.	<p>Lifeline</p>  <p><i>Lifiline anonim class x</i></p>  <p><i>Lifiline data pada class x</i></p>  <p><i>Lifiline data pada class x selektor k</i></p>	Lifeline adalah spealisasi dari elemen bernama yang mewakili peserta individu di dalam interaksi.
3.	<p>Message</p> 	Message dalam communication diagram ditampilkan sebagai garis dengan ekspresi urutan dan panah di atas garis.

No	Simbol	Keterangan
4.	Aktor 	Dapat berkomunikasi dengan object, maka actor dapat diurutkan sebagai kolom.
5.	Object 	Object digambarkan sebagai sebuah class (kotak) dengan nama object di dalamnya, di awali dengan sebuah titik koma.

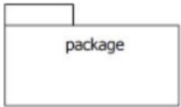
Sumber : Salamah Permadyanti Putri, 2021

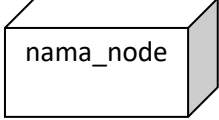


2.4.8 Deployment Diagram

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. Deployment diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

Simbol-simbol yang digunakan pada deployment diagram dapat dilihat pada Tabel 2.8

Tabel 2.8 Simbol Deployment Diagram

No	Simbol	Keterangan
1.	Package 	Package merupakan simbol bungkus dari satu atau lebih.

No	Simbol	Keterangan
2.	Node 	Biasanya mengacu pada perangkat keras (hardware), perangkat lunak yang tidak dibuat sendiri (software), jika didalam mode disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang di ikut sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3.	Depedency 	Kebergantungan antar node. Arah panah mengarah pada node yang digunakan.
4.	Link 	Relasi antar node.

Sumber : MARLIANSYAH, 2022

2.5 Metode Certainty Factor

Teori *Certainty Factor* diusulkan oleh *Shortliffe* dan *Buchanan* pada 1975 untuk mengakomodasi ketidakpastian kepemilikan (*inexact reasoning*) seorang pakar. Metode *Certainty Factor* ini dipilih ketika dalam menghadapi suatu masalah, sering ditemukan jawaban yang tidak memiliki kepastian penuh. Untuk mengakomodasi hal ini maka digunakan *Certainty Factor* guna menggambarkan

tingkat keyakinan pakar terhadap masalah yang sedang dihadapi ketidakpastian ini berupa probabilitas yang tergantung dari hasil suatu kejadian (Indrayani, dkk., 2020).

Agar mendapatkan nilai tingkat keyakinan *CertaintyFactor* dari sebuah *rule*, dapat dilakukan dengan mewawancarai seorang pakar. Nilai *CertaintyFactor* (*Rule*) didapat dari interpretasi “*term*” dari pakar, yang diubah menjadi nilai *Certainty Factor* tertentu sesuai Tabel 2.5 berikut (Narulita, 2021):

Tabel 2.5 Tingkat Keyakinan *Certainty Factor*

Kondisi Tidak Pasti (<i>Uncertain Term</i>)	CF
Tidak Tahu	0
Hampir Mungkin	0,2
Mungkin	0,4
Kemungkinan Besar	0,6
Hampir Pasti	0,8
Pasti	1,0

Metode *Certainty Factor* ini hanya bisa mengolah 2 bobot dalam sekali perhitungan. Untuk bobot yang lebih dari 2 banyaknya, untuk melakukan perhitungan tidak terjadi masalah apabila bobot yang dihitung teracak, artinya tidak ada aturan untuk mengkombinasikan bobotnya, karena untuk kombinasi seperti apapun hasilnya akan tetap sama. Untuk mengetahui apakah seorang pasien tersebut menderita penyakit atau tidak, itu dilihat dari hasil perhitungan bobot setelah semua keluhan keluhan diinputkan dan semua bobot dihitung dengan menggunakan metode *Certainty Factor*. Pasien yang divonis mengidap penyakit adalah pasien yang memiliki bobot mendekati +1 dengan keluhan-

keluhan yang dimiliki mengarah kepada penyakit. Sedangkan pasien yang mempunyai bobot mendekati -1 adalah pasien yang dianggap tidak mengidap penyakit, serta pasien yang memiliki bobot sama dengan 0 diagnosisnya tidak diketahui atau *unknown* atau bisa disebut dengan netral (Perangin-angin, 2021).

Certainty Factor menggambarkan derajat kepercayaan atau ketidakpercayaan, dimana hasil dari penjumlahan keduanya tidak selalu berjumlah 1. *Certainty Factor* menggunakan $MB[H, E]$ untuk menggambarkan nilai kepercayaan dari hipotesis H, Gejala E, dan $MD[H, E]$ untuk nilai ketidakpercayaan dari hipotesis H, gejala E. Karena keterangan atau fakta bagian dari gejala salah satunya menyangkal hipotesis, $MB[H, E]$ atau $MD[H, E]$ maka nilainya harus nol untuk setiap H dan E. Jadi rumus untuk *Certainty factor* adalah sebagai berikut (Rosid, 2022):

Dimana :

$$CF[H, E] = MB[H, E] - MD[H, E]$$

$CF[H, E]$: *Certainty Factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB[H, E]$: Ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

$MD[H, E]$: Ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E. Bentuk dasar rumus *Certainty Factor* sebuah aturan.

H : Hipotesis atau dugaan penyakit.

E : Evidence (Peristiwa atau fakta).

Secara umum *rule* di representasikan sebagai berikut :

Rumus : IF E1 AND E2 AND En THEN H (CF Rule)

Atau

IF E1 OR E2 OR En THEN H (CF Rule)

Dimana :

E1 En : Fakta – fakta (*evidence*) yang ada.

H : Hipotesis atau konklusi yang dihasilkan.

CF Rule : Tingkat keyakinan terjadinya hipotesis H akibat adanya fakta-fakta

E1.....En.

1. Untuk menghitung nilai CF dari 1 gejala menggunakan rumus :

$$CF[H, E] = MB[H, E] - MD[H, E]$$

2. Untuk menghitung nilai CF lebih dari 1 gejala menggunakan rumus :

$$CF(Rule1, Rule2) = CF(Rule1) + CF(Rule2) * [1 - CF(Rule1)]$$

2.5.1 Cara Kerja Certainty Factor

Proses kerja CF dimulai dengan penentuan nilai certainty untuk setiap aturan atau bukti yang ada dalam sistem. Nilai CF ini berkisar antara -1 hingga 1, di mana nilai 1 menunjukkan keyakinan penuh terhadap kebenaran informasi, nilai -1 menunjukkan keyakinan penuh bahwa informasi tersebut salah, dan nilai 0 menggambarkan ketidakpastian atau keadaan netral. Nilai-nilai ini mewakili tingkat kepercayaan sistem terhadap informasi atau bukti yang ada, yang dapat berupa gejala medis, kondisi pasar, atau fakta lainnya yang relevan dengan sistem pakar tersebut.

Nilai CF ditetapkan untuk setiap bukti, langkah berikutnya adalah menggabungkan nilai-nilai tersebut untuk menghasilkan keputusan akhir yang mencerminkan keyakinan terhadap suatu prediksi atau hasil (Aditama et al, 2022).

2.6 Bahasa Pemrograman PHP

PHP adalah sebuah bahasa pemrograman *server side scripting* yang bersifat *open source*, sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses (Salmaa Awwaabiin, 2021).

PHP sering dipakai para programmer untuk membuat situs web yang bersifat dinamis karena gratis dan berguna dalam merancang aplikasi web.

Supono dan Putratama mengemukakan bahwa "PHP adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam HTML". Sedangkan, menurut Solichin mengemukakan bahwa "PHP merupakan salah satu bahasa pemrograman berbasis web yang ditulis oleh dan untuk pengembang web". PHP merupakan bahasa (*script*) pemrograman yang sering digunakan pada sisi server sebuah web.

Kumpulan kutipan diatas menerangkan bahwa *hypertext preprocessor* (PHP) merupakan bahasa pemrograman untuk membuat/mengembangkan aplikasi berbasis web dan bersifat *open soure* dan ditanamkan ke dalam *script* HTML.

2.6.1 Sintaks PHP

Pengertian Sintaks PHP adalah aturan penulisan agar mampu dimengerti dengan benar oleh *compiler* saat membaca bahasa pemrograman. Dalam penulisan PHP yang benar diawali dengan “<?php” dan diakhiri dengan “?>”. Dan di dalam File PHP juga dapat berisi tag seperti HTML dan skrip sisi klien seperti JavaScript (Beon, 2022).

2.6.2 Komponen Dasar PHP

Komponen dasar PHP mencakup berbagai elemen penting yang membentuk struktur dan fungsi bahasa pemrograman ini. Salah satu komponen utama adalah PHP opening dan closing tags, yaitu <?php untuk membuka dan ?> untuk menutup kode PHP (Mahendra, 2023).

Instruksi PHP harus ditempatkan di dalam tag ini agar dapat diproses oleh server. selain itu, PHP mendukung penggunaan variabel, yang dimulai dengan tanda dolar (\$), yang digunakan untuk menyimpan data sementara selama eksekusi program. fungsi juga merupakan komponen dasar lainnya, memungkinkan programmer untuk mengorganisir kode menjadi bagian yang dapat dipanggil dan digunakan kembali (Dimas et al., 2023).

2.6.3 Pengenalan HTML

HTML merupakan bahasa yang digunakan untuk menulis halaman web. Biasanya mempunyai ekstensi .htm, .html, .shtml. HTML tersusun dari atas tag-tag, digunakan untuk menentukan tampilan dari kumpulan HTML yang diterjemahkan oleh browser. Tag HTML tidak *casesensitive*. Jadi bisa

menggunakan <HTML> atau <html>. Keduanya menghasilkan output yang sama (Anamisa & Mufarroha, 2022).

HTML berasal dari bahasa SGML (*Standart Generalized Mark up Language*) yang penulisannya disederhanakan. HTML dapat dibaca oleh berbagai macam platform. HTML juga merupakan bahasa pemrograman yang fleksibel, dapat disiplin/digabungkan dengan bahasa pemrograman lain, seperti PHP, ASP, JSP, JavaScript, dan lainnya. Jika ada kesalahan pada penulisan HTML, browser tidak akan memperlihatkan syntax error, tetapi hanya tidak menampilkannya. HTML terus berkembang seiring perkembangan browser (Amilia Trianasari, 2023).

2.6.4 Pengenalan CSS

CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda (Diky Setiawan, dkk., 2021).

CSS dapat mengatur ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouseover, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. Sama halnya styles dalam aplikasi wordprocessing seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya heading, sub-bab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (Diky Setiawan, dkk., 2021).

2.7 Xampp

Xampp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan Xampp ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat website secara offline untuk masa coba-coba di komputer sendiri. Dapat disimpulkan xampp adalah aplikasi tools untuk menyediakan paket berisi konfigurasi Web Server, Apache, PHP, MySql untuk membantu kita dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam program web (Danang, dkk., 2022).

2.8 MySQL

MySQL (*My Structured Query Language*) adalah “Suatu sistem basis data *relation* atau *Relational Database management System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga sangat digunakan untuk aplikasi *multi user* (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan *closed source* atau komersial” (Andryansyah, 2023).

MySQL merupakan database yang paling digemari di kalangan Programmer Web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah database server yang mampu untuk manajemen database dengan baik, MySQL terhitung merupakan database yang paling digemari dan

paling banyak digunakan dibanding database lainnya, Berikut ini ada beberapa tipe data pada MySQL adalah sebagai berikut menurut (Addilla S.A.S, 2021) :

1. Tipe Data Angka (*Numerik*)

Tipe data *numerik* (angka) adalah tipe data yang bisa digunakan untuk menyimpan nilai berupa angka, baik bilangan bulat, bilangan pecahan, positif maupun negatif. Contoh dari Tipe data *Numerik* yaitu *Tinyint*, *Smallint*, *Mediumint*, *Int*, *Bigint*, *Float*, *Double*, *Real*, *Decimal* dan *Numeric*. Tipe data ini hampir sama kegunaannya namun yang membedakan mereka adalah fungsi, jangkauan, dan ukurannya. dapat dilihat pada Tabel 2.6 Berikut ini :

Tabel 2.6 Tipe Data Angka (*Numerik*)

No	Nama	Fungsi	Jangkauan	Ukuran
1.	<i>Tinyint</i>	Menyimpan data bilangan bulat positif dan negatif	-128 s/d 127	1 byte (8 bit).
2.	<i>Smallint</i>	Menyimpan data bilangan bulat positif dan negatif	-32.768 s/d 32.767	2 byte (16 bit).
3.	<i>Mediumint</i>	Menyimpan data bilangan bulat positif dan negatif	-8.388.608 s/d 8.388.607	3 byte (24 bit).
4.	<i>Int</i>	Menyimpan data bilangan bulat positif dan negatif	-2.147.483.648 s/d 2.147.483.647	4 byte (32 bit).
5.	<i>Bigint</i>	Menyimpan data	+ 9,22 X 10 ¹⁸	8 byte

No	Nama	Fungsi	Jangkauan	Ukuran
		bilangan bulat positif dan negatif		(64 bit).
6.	<i>Float</i>	Menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+3	4 byte (32 bit).
7.	<i>Double</i>	Menyimpan data bilangan pecahan positif dan negatif presisi ganda	-1.79..E+308 s/d - 2.22..E-308,0, dan 2.22..E-308 s/d 1.79..E+308.	5 byte (64 bit).
8.	<i>Real</i>	Menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-1.79..E+308 s/d - 2.22..E-308,0, dan 2.22..E-308 s/d 1.79..E+308.	6 byte (64 bit).
9.	<i>Decimal</i>	Menyimpan data bilangan bulat positif dan negatif	-1.79..E+308 s/d - 2.22..E-308,0, dan 2.22..E-308 s/d 1.79..E+308.	7 byte (64 bit).
10.	<i>Numeric</i>	Menyimpan data bilangan bulat positif dan negatif	-1.79..E+308 s/d - 2.22..E-308,0, dan 2.22..E-308 s/d	8 byte (64 bit).

No	Nama	Fungsi	Jangkauan	Ukuran
			1.79..E+308.	

2. Tipe Data Teks (*String*)

Tipe Data *String* (Teks) adalah tipe data yang bisa digunakan untuk menampung hingga 255 karakter. Tipe data teks ini biasanya digunakan untuk menyimpan data berupa tulisan, maupun angka yang akan dibaca *string*. Contoh dari tipe data *String* yaitu *Char*, *Varchar*, *Tinytext*, *Text*, *Mediumtext*, *Longtext*. Tipe data ini semua hampir sama kegunaannya namun yang membedakan mereka adalah fungsi, dan jangkauan nya. dapat dilihat pada Tabel 2.7 Berikut ini :

Tabel 2.7 Tipe Data Teks (*String*)

No	Nama	Fungsi	Jangkauan
1.	<i>Char</i>	Menyimpan data string ukuran tetap	0 s/d 255 karakter
2.	<i>Varchar</i>	Menyimpan data string ukuran dinamis	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
3.	<i>Tinytext</i>	Menyimpan data text	1 s/d 255 karakter (versi 4.1), 0 s/d 65.535
4.	<i>Text</i>	Menyimpan data text	0 s/d 65.535
5.	<i>Mediumtext</i>	Menyimpan data text	0 s/d 224 -1 karakter
6.	<i>Longtext</i>	Menyimpan data text	1 s/d 224 -1 karakter

3. Tipe Data Tanggal dan Waktu (*Date And Time*)

Tipe Data *Date and Time* adalah tipe data yang bisa digunakan untuk data dalam bentuk waktu seperti tanggal, tahun, bulan. Contoh dari tipe data *date and time* yaitu *Date*, *Time*, *Datetime*, *Year*. Tipe data ini semua hampir sama kegunaannya namun yang membedakan mereka adalah fungsi, jangkauan, dan ukurannya, Namun berdasarkan nama tipe datanya kamu pasti sudah tau fungsi dari masing-masing tipe data tersebut. dapat dilihat pada Tabel 2.8 Berikut ini :

Tabel 2.8 Tipe Data Tanggal dan Waktu (*Date And Time*)

No	Nama	Fungsi	Jangkauan	Ukuran
1.	<i>Date</i>	Menyimpan data tanggal	1000-01-01s/d9999-12-31(YYYY-MM-DD)	3 byte
2.	<i>Time</i>	Menyimpan data waktu	-838:59:59 s/d +838:59:59(HH:MM:SS)	3 byte
3.	<i>Datetime</i>	Menyimpan data tanggal dan waktu	1000-01-01s/d9999-12-31 23:59:59	8 byte
4.	<i>Year</i>	Menyimpan data bilangan bulat positif/ negatif	1900 s/d 2155	1 byte

4. Tipe Data *BLOB*

Tipe Data *BLOB* adalah tipe data yang dapat digunakan untuk menampung data multimedia seperti video, gambar, musik dan lain-lain. Contoh dari tipe data *BLOB* yaitu *Bit*, *Tinyblob*, *Blob*, *Mediumblob*, *Longblob*. Tipe data ini

semua hampir sama kegunaannya namun yang membedakan mereka adalah fungsi, dan jangkauan nya. dapat dilihat pada Tabel 2.9 Berikut ini:

Tabel 2.9 Tipe Data *BLOB*

No	Nama	Fungsi	Jangkauan
1.	<i>Bit</i>	Menyimpan data biner	64 digit biner
2.	<i>Tinyblob</i>	Menyimpan data biner / gambar ukuran kecil	255 byte
3.	<i>Blob</i>	Menyimpan data biner / gambar	4
4.	<i>Mediumblob</i>	Menyimpan data biner / gambar ukuran sedang	244-1 byte
5.	<i>Longblob</i>	Menyimpan data biner / gambar ukuran besar	232-1 byte