

BAB II

KAJIAN TEORITIS

A. Kajian Pustaka

1. Sistem Pakar

Menurut Penelitian Hakim (2020:59), bahwa sistem pakar adalah teknologi yang meniru kemampuan seorang ahli untuk memberikan jawaban dan menyelesaikan masalah. Melalui interaksi dialog, sistem ini memberikan solusi kepada pengguna. Sistem ini dapat memberikan bantuan kepada orang yang tidak memiliki kemampuan untuk menjawab pertanyaan, mengatasi masalah, dan membuat keputusan seperti seorang ahli.

Berdasarkan penelitian Minanti & Munti, (2022 : 61), bahwa sistem pakar merupakan sistem yang dibuat untuk meniru kemampuan seorang pakar untuk memberikan jawaban dan menyelesaikan masalah. Sistem ini berinteraksi dengan pengguna melalui dialog untuk memberikan solusi atas masalah yang dihadapi. Dengan menggunakan sistem pakar, orang yang tidak memiliki keahlian khusus dapat mengatasi pertanyaan, memecahkan masalah, dan membuat keputusan sebagaimana yang biasanya dilakukan oleh seorang pakar.

Sedangkan berdasarkan sumber dari buku menurut Rossnelly (2012 : 2-3), bahwa Sistem pakar adalah sistem komputer yang dibuat untuk meniru semua kemampuan dalam pengambilan keputusan seorang pakar. Memanfaatkan pengetahuan khusus secara optimal seperti yang dilakukan oleh seorang pakar untuk menyelesaikan suatu masalah. Bisa juga disebut

orang yang memiliki keahlian khusus / pengetahuan yang tidak banyak orang memiliki pengetahuan / keahlian tersebut.

Dari ketiga penelitian diatas maka dapat disebutkan bahwa sistem pakar adalah sistem komputer yang dirancang untuk meniru kemampuan pengambilan keputusan seorang ahli. Sistem ini berfungsi untuk memberikan solusi kepada pengguna dengan memanfaatkan pengetahuan khusus yang dimiliki oleh seorang ahli. Melalui interaksi dialog, sistem pakar membantu pengguna yang bukan ahli dalam menjawab pertanyaan, memecahkan masalah, dan membuat keputusan yang biasanya dilakukan oleh seorang pakar. Sistem ini secara maksimal memanfaatkan pengetahuan khusus untuk menyelesaikan berbagai masalah.

2. Kecanduan Game Online

Menurut penelitian dari Akbar (2020 : 43), bahwa game online adalah salah satu komponen internet yang sangat populer dan sering digunakan, bahkan bisa menyebabkan kecanduan dengan intensitas yang tinggi. Kecanduan adalah kondisi di mana seseorang sangat bergantung secara fisik atau psikologis pada suatu aktivitas dan merasa tidak nyaman jika tidak bisa melakukannya. Oleh karena itu, kecanduan game online bisa juga disebut keadaan di mana penderita menjadi sangat terikat pada kebiasaan bermain game online dan sulit untuk berhenti. Seiring waktu, frekuensi, durasi, atau intensitas bermain akan meningkat, tanpa memperhatikan dampak negatif yang mungkin timbul.

Berdasarkan penelitian dari Subandi et al., (2022 : 247), bahwa ketika seorang *gamers* mengalami penghayatan yang mendalam, mereka seolah-olah terlibat langsung dalam permainan dan semakin menikmatinya, yang pada gilirannya meningkatkan kecanduan. *Immersion* (penghayatan) adalah keinginan untuk memahami dan mendalami game serta strategi - strategi yang bisa diterapkan. Perkembangan game online yang terus berlanjut membuat anak-anak lupa akan kewajiban mereka sebagai pelajar. Banyak dari mereka mengabaikan pendidikan demi bermain game. Kurangnya perhatian dari orang tua memungkinkan anak-anak berbohong dan menyalahgunakan waktu yang seharusnya dihabiskan untuk belajar di sekolah

Berdasarkan dari kedua penelitian tersebut bisa disebut bahwa game online yang sangat populer dapat menyebabkan kecanduan, di mana pemain sulit berhenti dan semakin meningkatkan intensitas bermain tanpa memedulikan dampak negatif. Penghayatan mendalam pada game memperburuk kecanduan ini, membuat anak-anak mengabaikan kewajiban pendidikan dan sering kali kurang diawasi oleh orang tua.

3. *Website*

Menurut penelitian dari Dutonde (2022 : 359) , bahwa ada tiga jenis spesialisasi dalam pengembangan *web* : pengembang *front-end*, pengembang *back-end*, dan pengembang *full-stack*. Pengembang *front-end* fokus pada tampilan dan interaksi yang dialami oleh pengguna di *browser*, sedangkan pengembang *back-end* bertugas mengelola *server*.

Pengembangan *web* juga mencakup aspek jaringan dan penjualan pribadi yang erat. Situs *web* kini tidak hanya berfungsi sebagai alat untuk pekerjaan atau perdagangan, tetapi juga sebagai media komunikasi dan jejaring sosial yang luas. *Platform* seperti *Facebook* dan *Twitter* memungkinkan pengguna untuk berkomunikasi, serta memungkinkan organisasi untuk berinteraksi dengan masyarakat secara personal dan interaktif. Serta menurut sumber dari buku Sukri et al., (2023 : 1) bahwa *website* merupakan kumpulan halaman *web* yang saling terhubung dan dapat diakses secara global selama terhubung dengan internet. Setiap halaman *website* memiliki *URL (Uniform Resource Locator)*. Situs *web* dapat memuat beragam jenis informasi, termasuk teks, gambar, video, dan audio.

Berdasarkan penelitian dari Sudrajat Juwanto & Meisella Kristania., (2023 : 64), bahwa *Web* adalah platform yang menyajikan berbagai jenis dokumen multimedia seperti teks, gambar, suara, animasi, dan video dengan memanfaatkan protokol *HTTP (Hypertext Transfer Protocol)*. Untuk membukanya, dibutuhkan perangkat lunak yang disebut *browser*.. Dengan demikian, *web* dapat didefinisikan sebagai kumpulan halaman informasi yang bisa diakses melalui internet dengan memasukkan alamat (*link*) yang dituju. Sedangkan sumber dari buku Widia & Asriningtias, (2021 : 3) bahwa sebuah *website* adalah sekumpulan halaman *web* yang berisi teks dalam format *Hyper Text Markup Language (HTML)*. Halaman-halaman ini disimpan di *server hosting* dan dapat diakses melalui *browser* menggunakan jaringan internet dengan alamat yang dikenal

sebagai *Uniform Resource Locator (URL)*. Pengguna mengakses website melalui *browser* dengan menggunakan *protokol Hyper Text Transfer Protocol (HTTP)*. *HTTP* adalah jalur enkripsi yang digunakan untuk menjaga keamanan dan privasi konten yang terdapat dalam sebuah *website*.

Berdasarkan dari empat penelitian diatas bisa disebut bahwa terdapat tiga jenis spesialisasi pengembang web: *front-end*, *back-end*, dan *full-stack*. Pengembang *front-end* fokus pada tampilan dan interaksi di *browser* pengguna, sedangkan pengembang *back-end* menangani *server*. Pengembangan *web* mencakup aspek jaringan dan penjualan pribadi yang erat, menjadikan situs *web* tidak hanya sebagai alat kerja atau perdagangan, tetapi juga sebagai sarana komunikasi dan jejaring sosial, seperti yang dilakukan oleh *platform* seperti *Facebook* dan *Twitter*. Selain itu, *web* adalah aplikasi yang berisi dokumen multimedia yang diakses menggunakan protokol *HTTP* melalui *browser*, memungkinkan pengguna untuk mengakses informasi dengan memasukkan link tertentu di internet.

4. Forward Chaining

Menurut Penelitian dari Orun et al., (2022 : 326 - 328), bahwa *forward-chaining* adalah pendekatan utama dalam pemrosesan di dalam sistem pengambil keputusan, yang secara logis mengikuti proses aplikasi dari modus ponens. Berlawanan dengan *backward-chaining*, *forward-chaining* dimulai dengan data yang ada dan menggunakan aturan inferensi untuk menghasilkan data tambahan sampai mencapai tujuan atau kesimpulan yang diinginkan. Mesin inferensi yang menggunakan *forward*

chaining mencari aturan inferensi yang cocok dengan kondisi awal (*antecedent*) dari aturan *IF-THEN*. Ketika aturan yang sesuai ditemukan, mesin pengambil keputusan dapat menarik kesimpulan atau konsekuensi dengan memanfaatkan informasi tambahan dari data yang tersedia.

Berdasarkan Penelitian dari Putri et al., (2023 : 11), bahwa *Artificial Intelligence (AI)* dapat memfasilitasi proses diagnosis dengan memanfaatkan algoritma *forward chaining* sebagai alat bantu bagi para ahli. Algoritma *forward chaining* merupakan metode penalaran yang bergerak maju dengan menggunakan fakta-fakta yang ada untuk menghasilkan pengetahuan yang digunakan sebagai dasar untuk mencapai kesimpulan. Proses ini dimulai dengan fakta-fakta yang dijadikan premis dalam sistem pakar, yang kemudian digunakan oleh pengguna untuk mendukung proses diagnosis.

Berdasarkan dari 2 penelitian sebelumnya bisa disebutkan bahwa *forward-chaining* adalah metode utama dalam mesin pengambil keputusan yang menggunakan aturan-aturan inferensi untuk mencapai tujuan atau kesimpulan berdasarkan data yang tersedia. Metode ini dimulai dengan fakta yang ada dan terus bergerak maju hingga mendapatkan kesimpulan yang diinginkan, berbeda dengan *backward-chaining*. Dalam konteks kecerdasan buatan (*AI*), *forward-chaining* dapat mendukung proses diagnosis dengan menggunakan fakta-fakta yang ada sebagai premis dalam sistem pakar, membantu para ahli mencapai kesimpulan yang diperlukan.

5. *HTML*

Menurut Penelitian Dari Ganapathy et al., (2021 : 2431), bahwa halaman *web* konvensional umumnya dibangun dengan menggunakan kerangka *HTML* yang tujuannya adalah menyediakan struktur dasar, sedangkan *cascading style sheet* digunakan untuk mengatur tampilan halaman. Setelah proyek pembuatan halaman *web* selesai, biasanya tidak ada lagi perubahan yang terjadi. Namun, pemrogram masih dapat melakukan pembaruan dan modifikasi konten *HTML* dengan menambahkan artikel berita atau konten lainnya, serta melakukan perubahan desain *HTML* jika diperlukan.

Berdasarkan penelitian dari Čabarkapa, (2022 : 2), bahwa *HTML* adalah bahasa *markup* standar yang digunakan untuk membuat dokumen yang akan ditampilkan di *browser web*. Teknologi lain seperti *Cascading Style Sheets (CSS)* dan *JavaScript* sering digunakan bersama *HTML* untuk meningkatkan tampilan dan fungsi dari halaman web. *HTML* sangat penting karena membantu dalam membangun struktur dasar dari sebuah situs *web* dan merupakan bagian utama dari *front-end* sebuah situs *web*.

Menurut pendapat Penelitian dari Ghimire, (2020 : 4) , bahwa *HTML* merupakan bahasa *markup* yang digunakan untuk membuat struktur dasar dari halaman *web* dan dianggap sebagai kerangka dasar dari sebuah situs *web*. Kode *HTML* membentuk dokumen elektronik yang disebut halaman web dan memastikan pemformatan teks dan gambar yang tepat agar dapat ditampilkan sesuai dengan keinginan pembuatnya oleh *browser* internet.

Cascading Style Sheet (CSS) digunakan untuk mengatur tampilan halaman *web* dan biasanya ditempatkan di atas *HTML*. *HTML* terdiri dari tag pembuka dan penutup serta nama tag yang digunakan untuk menentukan struktur dan konten dari halaman *web*. Sedangkan menurut dari buku Anamisa & Mufarroha, (2020 : 21) bahwa *HTML* merupakan singkatan dari *Hypertext Markup Language*, yang memungkinkan pengguna untuk menciptakan dan mengatur elemen-elemen. Contohnya seperti paragraf, *heading*, tautan, dan *blockquote* untuk halaman *web* dan aplikasi. *HTML* adalah bahasa *markup* yang digunakan untuk merancang halaman *web*.

Berdasarkan keempat penelitiannya bisa disebutkan bahwa *HTML* adalah bahasa *markup* yang digunakan secara umum untuk membangun kerangka dasar halaman *web*, seringkali didampingi oleh *CSS* dan *JavaScript* untuk meningkatkan penampilan dan fungsionalitasnya. *HTML* membentuk dokumen elektronik yang disebut halaman *web*, menentukan format teks dan gambar agar sesuai dengan keinginan pembuatnya. *CSS* bertanggung jawab atas penataan tampilan halaman *web* dan biasanya disisipkan di atas kode *HTML*. Meskipun biasanya halaman *web* konvensional tidak banyak mengalami perubahan setelah pembuatannya, pemrogram masih memiliki kemampuan untuk melakukan pembaruan dan modifikasi konten dan desain *HTML* bila dibutuhkan.

6. *Tailwind CSS*

Menurut penelitian Lutfi & Latipah, (2023 : 86) Bahwa *tailwind CSS* adalah *framework CSS* yang digunakan untuk mengembangkan tampilan antarmuka yang responsif pada situs *web* dengan pendekatan berbasis utilitas. *Tailwind CSS* sangat mendukung pengembangan desain berorientasi komponen. Salah satu alasan pemilihan *tailwind CSS* dalam pengembangan komponen untuk tugas akhir ini adalah karena pendekatan utilitasnya berbeda dari kebanyakan *framework CSS* lainnya.

Berdasarkan pendapat dari Eldina, (2023 : 18-19) bahwa *tailwind CSS* memiliki konfigurasi yang fleksibel dengan beragam opsi yang memungkinkan penyesuaian warna, ukuran, dan properti *CSS* lainnya sesuai dengan kebutuhan desain pengembang. *tailwind CSS* juga menyediakan kelas-kelas responsif yang memudahkan pengembang mengatur tampilan elemen berdasarkan perangkat atau ukuran layar yang berbeda. Misalnya, kelas `md:text-xl` dapat digunakan untuk mengubah ukuran teks pada layar berukuran sedang ke atas, sehingga *tailwind CSS* mendukung desain yang responsif.

Menurut penelitian Karim, (2022 : 14), bahwa *tailwind CSS* didefinisikan sebagai *framework CSS* yang bersifat *utility-first* untuk membangun desain antarmuka dengan cepat yang dikemas dalam bentuk *class HTML*. Dengan mengusung konsep *utility-first*, *tailwind CSS* memudahkan pengguna dalam memodifikasi antarmuka aplikasi *web* dengan menyusun *class-class utility* milik *tailwind CSS* untuk membentuk

komponen antarmuka sesuai selera. *Framework* ini sangat ideal untuk pengembangan cepat dan efisien tanpa harus menulis banyak *CSS custom*.

Berdasarkan penelitian Watoni, (2023 : 18-19), bahwa *tailwind CSS* dapat digunakan untuk membangun antarmuka pengguna yang responsif dan mempercepat pengembangan aplikasi pengenalan daun bawang merah. Dengan menggunakan kelas utilitas dan kemampuan responsifnya, Tailwind CSS memudahkan pengaturan tampilan elemen antarmuka pengguna sesuai kebutuhan proyek. Ini memungkinkan pengembang untuk fokus pada fungsionalitas dan estetika aplikasi tanpa perlu menyusun kode CSS yang kompleks.

Berdasarkan keempat penelitiannya, bisa disebutkan bahwa *tailwind CSS* adalah *framework CSS* yang sangat mendukung pengembangan tampilan antarmuka responsif dengan pendekatan berbasis utilitas. *Framework* ini memudahkan pengembang dalam menyesuaikan warna, ukuran, dan properti CSS lainnya sesuai dengan kebutuhan proyek. *Tailwind CSS* memungkinkan pengaturan tampilan elemen *UI* berdasarkan perangkat atau ukuran layar yang berbeda, mendukung desain responsif, dan memfasilitasi pengembangan cepat serta efisien tanpa perlu menulis banyak *CSS custom*. Dengan konsep *utility-first*, *Tailwind CSS* sangat ideal untuk membangun komponen antarmuka sesuai selera pengguna, mempercepat proses pengembangan, dan fokus pada fungsionalitas serta estetika aplikasi.

7. *Database Management System*

Menurut Buku Raissa Amanda Putri, (2022:3) *Database Management System (DBMS)* adalah *software* yang diciptakan untuk menyederhanakan tugas-tugas yang sebelumnya rumit. Dengan menyimpan data dalam DBMS daripada sebagai sekumpulan file dalam sistem operasi, fitur-fitur DBMS memungkinkan pengelolaan data secara efisien dan efektif. Ketika volume data dan jumlah pengguna meningkat menjadi ratusan gigabyte data dan ribuan pengguna dalam database perusahaan, dukungan DBMS menjadi krusial.

Kemudian menurut Siswandi & Muhidin, (2022:200) menjelaskan bahwa untuk mengatur sebuah basis data, diperlukan aplikasi yang disebut DBMS atau Database Management System. Contoh aplikasi DBMS meliputi MySQL, SQL Server, Microsoft Access, Oracle, dan lain-lain. Sistem basis data menyediakan dua jenis bahasa, yaitu bahasa data definition (DDL) yang digunakan untuk membuat skema basis data, dan bahasa data manipulation (DML) yang digunakan untuk mengambil dan mengubah data dalam basis data.

Pemahaman menurut Dalimunthe, (2022:4) menjelaskan bahwa MySQL, dikembangkan oleh MySQL AB sekitar tahun 1994 di Swedia, awalnya ditujukan untuk mendukung aplikasi berbasis web pada klien. MySQL merupakan perangkat lunak sistem manajemen basis data SQL (DBMS) yang multithread, dapat digunakan oleh banyak pengguna secara bersamaan, dengan lebih dari 6 juta instalasi di seluruh dunia. Software ini

tersedia secara gratis di bawah lisensi GPL (GNU General Public License). MySQL adalah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan di bawah lisensi GPL, memungkinkan penggunaan bebas dengan batasan tidak untuk kepentingan komersial. MySQL mendasarkan konsep utamanya pada SQL, sebuah bahasa yang digunakan untuk mengoperasikan basis data, terutama untuk pengambilan data (query) dan manipulasi data, yang memudahkan pengguna dalam mengelola data dengan efisien.

Berdasarkan ketiga penelitiannya bisa disebutkan bahwa *MySQL* sebagai *Database Management System (DBMS)* menonjol dalam keandalan dan kinerja cepat saat menjalankan *query data*. *DBMS*, termasuk *MySQL*, berperan penting dalam pengelolaan basis data untuk mendukung efisiensi dan skalabilitas dalam lingkungan perusahaan. *MySQL*, dikembangkan sejak tahun 1994, gratis di bawah lisensi *GPL*, dan konsep utamanya berkaitan dengan *SQL*, memfasilitasi operasi data dengan mudah. Keseluruhan, *MySQL* dan *DBMS* umumnya menjadi alat krusial dalam mengelola data perusahaan dengan efisien.

8. *Unified Modeling Language (UML)*

Menurut penelitian dari Medikano et al., (2023 : 54) , bahwa bahasa pemodelan terpadu (*UML*) adalah standar bahasa yang luas digunakan dalam dunia bisnis untuk mendefinisikan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (OOP). Sebagai bahasa visual, *UML* digunakan untuk

memodelkan dan mengkomunikasikan sistem melalui diagram dan teks pendukung. *UML* muncul karena adanya kebutuhan untuk pemodelan visual yang dapat menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak. *UML* digunakan secara eksklusif untuk pemodelan dan tidak terbatas pada metodologi tertentu, meskipun dalam praktiknya, *UML* paling sering diterapkan dalam metodologi berorientasi objek.

Serta penjelasan penelitian dari Limantoro & Kristiadi, (2021 : 8) dengan kata lain, *UML* dapat digunakan untuk mendeskripsikan berbagai hal. *Diagram Use Case* bersifat statis, menampilkan kumpulan use case dan aktor (sejenis kelas khusus). Diagram ini sangat penting untuk mengatur dan memodelkan perilaku sistem yang diharapkan dan dibutuhkan oleh pengguna. Sebaliknya, Diagram Interaksi dan *Sequence* bersifat dinamis, menyoroti pengiriman pesan dalam urutan waktu tertentu untuk menunjukkan dinamika interaksi dalam sistem.

Berdasarkan 2 penelitian sebelumnya bisa disebutkan bahwa *UML* adalah bahasa standar yang sering digunakan dalam bisnis untuk mendefinisikan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Sebagai bahasa visual, *UML* memungkinkan pemodelan sistem melalui diagram dan teks. Keberadaannya dipicu oleh kebutuhan akan pemodelan visual untuk sistem perangkat lunak. *UML* tidak terbatas pada satu metodologi saja dan sering digunakan dalam metodologi berorientasi objek. Simbol-simbol dan



proses dalam UML membantu dalam menjelaskan hasil yang diharapkan, memberikan kerangka kerja untuk berbagai aspek terkait sistem yang dimodelkan.

9. *Use Case Diagram*





Berdasarkan penelitian dari Viktoria, (2022 : 101), bahwa *Use Case Diagram* adalah representasi grafis yang menampilkan beberapa atau semua aktor, *use case*, dan interaksinya. Diagram ini memperkenalkan suatu sistem untuk dikembangkan lebih lanjut dalam *use case* yang disajikan. Ada empat proses utama yang dilakukan dalam sistem informasi monitoring yang akan dibuat.

Menurut Farhan & Leman, (2023 : 20) bahwa *Use case diagram* menggambarkan fungsi yang diharapkan dari sebuah sistem, dengan fokus pada "apa" yang dilakukan sistem, bukan "bagaimana" caranya. *Diagram* ini menampilkan interaksi antara aktor dan sistem. *Use case Diagram* dapat digambarkan menggunakan simbol-simbol seperti yang ditunjukkan pada tabel 2.1.

Tabel 2. 1 *Use Case Diagram*

Simbol	Deskripsi
<i>Use Case</i> 	Unit-unit yang berkomunikasi satu sama lain di antara aktor-aktor.
<i>Actor</i> 	Orang, proses, atau sistem lain yang berhubungan dengan sistem informasi.

Tabel 2. 1 lanjutan *Use Case Diagram*.

Simbol	Deskripsi
<i>Association</i> 	Interaksi antara aktor dan use case yang terlibat.
<i>Ekstensi/extend</i> 	Hubungan use case tambahan.
<i>Generalisasi / Generalized</i> 	Hubungan generalisasi dan spesialisasi (umum menuju ke khusus).
<i>Included</i> 	Hubungan antara use case tambahan dan sebuah use case lain di mana use case tambahan memerlukan use case tersebut untuk menjalankan fungsinya atau sebagai prasyarat untuk menjalankan use case tersebut.






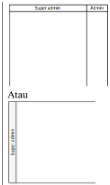
10. Activity Diagram

Penelitian menurut Farhan & Leman, (2023 : 22), bahwa *activity diagram* menggambarkan berbagai proses aktivitas dalam sistem yang sedang dikembangkan, termasuk bagaimana setiap proses dimulai, keputusan - keputusan yang mungkin terjadi, dan bagaimana proses tersebut berakhir. Diagram ini juga menunjukkan proses atau tahapan paralel yang mungkin terjadi di beberapa bagian eksekusi. Selain itu, diagram ini membantu dalam memvisualisasikan alur kerja dan interaksi antara aktivitas-aktivitas tersebut.

Sedangkan penelitian menurut Viktoria, (2022 : 101), bahwa *activity diagram* adalah model yang digunakan untuk menggambarkan aktivitas

dalam suatu sistem yang sedang berjalan. Diagram ini berfungsi sebagai penjelasan aktivitas program tanpa memerlukan detail coding atau tampilan. *Activity diagram* dapat diilustrasikan dengan simbol-simbol seperti yang ditunjukkan pada gambar tabel 2.2.

Tabel 2. 2 *Activity Diagram*

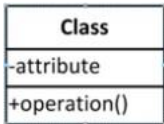

Simbol	Deskripsi
Status Awal / <i>Initial State</i> 	Status awal atau initial state adalah kondisi awal ketika sistem mulai beroperasi.
Status Akhir / Final State 	Status akhir atau final state adalah kondisi terakhir dari siklus hidup.
Aktivasi 	Aktivasi adalah tindakan yang dilakukan di dalam sistem, biasanya dimulai dengan kata kerja.
Percabangan / <i>Decision</i> 	Percabangan adalah proses di mana terdapat opsi atau pilihan kegiatan yang dapat diambil.
Penggabungan / <i>Join</i> 	Asosiasi penggabungan adalah kombinasi dari beberapa aktivitas menjadi satu kesatuan.
Swimlane 	Memisahkan unit organisasi bisnis yang bertugas mengelola aktivitas yang berlangsung.

11. Class Diagram






Menurut penelitian Viktoria, (2022 : 101), bahwa *class diagram* menggambarkan struktur statis dari kelas-kelas dalam sebuah sistem. Kelas merepresentasikan entitas yang dikelola oleh sistem dan dapat berhubungan dengan kelas lain melalui berbagai cara: asosiasi (satu kelas bergantung pada kelas lain), spesialisasi (satu kelas merupakan spesialisasi dari kelas lain), atau pengelompokan (dikumpulkan bersama sebagai satu unit). *Class Diagram* membantu memvisualisasikan hubungan dan hierarki antara kelas-kelas dalam sistem tersebut.

Berdasarkan penelitian dari Farhan & Leman, (2023 : 21), bahwa *class diagram* adalah perincian yang, ketika diinstansiasi, menghasilkan sebuah objek dan menjadi inti dari pengembangan serta desain berorientasi objek. Diagram ini menggambarkan bentuk dan deskripsi kelas, paket, dan objek, serta hubungan di antara mereka, seperti containment, pewarisan, asosiasi, dan lainnya. *Class diagram* dapat digambarkan menggunakan simbol-simbol seperti yang ditunjukkan pada tabel 2.3.

Tabel 2. 3 *Class Diagram*

Simbol	Deskripsi
Kelas / <i>Class</i> 	Kelas adalah representasi dari struktur sistem. Atribut menggambarkan kondisi atau karakteristik dari suatu objek. Operasi menggambarkan fungsi atau aktivitas yang dapat dilakukan oleh objek tersebut.
Antar muka / <i>Interface</i> 	Mirip dengan konsep interface dalam pemrograman berorientasi objek.

Tabel 2.3 Lanjutan *Class Diagram*







Simbol	Deskripsi
Asosiasi / <i>Association</i> 	Hubungan antara kelas dalam konteks umum, asosiasi sering kali memiliki multiplicity yang menggambarkan banyaknya instans dari kelas yang terlibat dalam hubungan tersebut.
Asosiasi berarah / <i>Directed association</i> 	Hubungan antar kelas di mana satu kelas digunakan oleh kelas lain, asosiasi sering kali memiliki multiplicity yang menggambarkan berapa kali atau seberapa banyak kelas yang menggunakan kelas tersebut.
Generalisasi 	Hubungan antar kelas yang menggambarkan generalisasi-spesialisasi, di mana kelas umum memiliki karakteristik yang dimiliki oleh kelas khusus yang turunannya.
Kebergantungan / <i>Dependency</i> 	Hubungan antara kelas di mana satu kelas membutuhkan kelas lain untuk melakukan tugas tertentu atau untuk menyediakan data yang diperlukan, menggambarkan kebergantungan antar kelas.
Agregasi / <i>Aggregation</i> 	Hubungan antara kelas di mana sebuah kelas utuh terdiri dari bagian-bagian yang lebih kecil atau lebih spesifik, menggambarkan hubungan semua-bagian (whole-part).

12. *Sequence diagram*

Menurut Viktoria, (2022 : 101), bahwa *sequence diagram* menggambarkan interaksi antara berbagai objek dan waktu penggunaannya, menunjukkan urutan pesan yang dikirim antara objek serta interaksi yang terjadi pada saat tertentu dalam suatu sistem. Diagram ini membantu memvisualisasikan alur komunikasi dan proses yang berlangsung antara objek-objek tersebut. *Sequence Diagram* dapat memberikan wawasan yang jelas tentang bagaimana sistem beroperasi dalam konteks waktu dan urutan.

Sequence diagram dapat digambarkan menggunakan simbol-simbol seperti yang ditunjukkan pada tabel 2.4.

Tabel 2. 4 *Sequence Diagram*

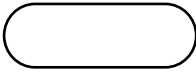


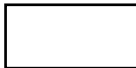
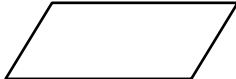

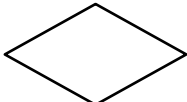
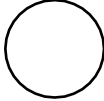
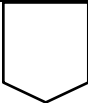
Simbol	Deskripsi
<p><i>Actor</i></p> 	Menyajikan gambaran tentang seseorang yang sedang menggunakan sistem untuk berinteraksi.
<p>Entity Class</p> 	Menjelaskan hubungan antara entitas-entitas dan aktivitas yang akan dilakukan.
<p>Boundary Class</p> 	Menjelaskan keterkaitan antara elemen yang berbeda satu sama lain..
<p>Control Class</p> 	Menjelaskan koneksi antara batas (boundary) dengan tabel.
<p>A focus of control & a life line</p> 	Menjelaskan titik awal dan akhir suatu pesan.
<p>Message</p> 	Menjelaskan detail tentang komunikasi antara objek yang mencakup informasi tentang aktivitas yang terjadi di dalamnya.

13. *Flowchart*


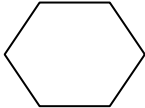
Menurut Penelitian dari Zalukhu et al., (2023:63), bahwa *flowchart* adalah representasi grafis dari langkah-langkah dan urutan prosedur dalam sebuah program. *Flowchart* sistem menunjukkan urutan proses dalam sistem, serta menggambarkan alat-alat media *input*, *output*, dan jenis media yang digunakan untuk penyimpanan selama pengolahan data. Di sisi lain,

flowchart program menggunakan simbol-simbol tertentu untuk secara detail menggambarkan urutan proses dan hubungan antara satu proses (instruksi) dengan proses lainnya dalam sebuah program. *Flowchart* ini dapat digambarkan menggunakan simbol-simbol seperti yang ditunjukkan pada tabel 2.5.

Tabel 2. 5 Simbol *flowchart*

Simbol	Nama	Keterangan
	Terminator	Awal dan akhir dari sebuah program.
	Garis Alir	Penyusunan jalur atau urutan dari aliran program.
	<i>Preparation</i>	Langkah awal dalam proses atau penetapan nilai awal.
	<i>Process</i>	Langkah-langkah dalam menghitung atau memproses data.
	<i>Input/Output Data</i>	Proses penerimaan atau pengiriman data, parameter, atau informasi.
	<i>Predefined Process (Sub Process)</i>	Langkah awal dalam menjalankan subprogram atau proses memulai subprogram.
	<i>Decision</i>	Perbandingan pernyataan atau seleksi data yang memberikan opsi untuk langkah berikutnya.
	<i>On Page Conector</i>	Koneksi antara elemen-elemen flowchart yang terletak dalam satu halaman.
	<i>Off Page Conector</i>	Koneksi antara elemen-elemen flowchart yang terletak pada halaman yang berbeda.

Tabel 2.5 lanjutan Simbol *Flowchart*

Simbol	Nama	Keterangan
	Document atau print	Simbol yang digunakan untuk mencetak keluaran dalam format dokumen.
	<i>Preparations</i> atau <i>save</i>	Simbol yang berguna untuk menyimpan data dalam penyimpanan (storage).

14. Black Box

Menurut penelitian dari Praniffa et al., (2023 : 4), bahwa pengujian *black box* memiliki kelebihan dan kekurangan dalam pelaksanaannya. Salah satu kelebihannya adalah membantu mengidentifikasi aspek-aspek yang tidak sesuai dengan spesifikasi kebutuhan yang telah ditetapkan dalam pengembangan perangkat lunak. Namun, kekurangannya adalah pengujian ini tidak bisa dilakukan secara menyeluruh karena penguji memiliki pengetahuan yang terbatas tentang perangkat lunak yang diuji.

Menurut penelitian dari Azimi & Rinjani, (2024 : 44) , bahwa pengujian *black box* adalah metode yang hanya memerlukan informasi tentang batas bawah dan batas atas dari data yang dibutuhkan, membuatnya menjadi cara yang sederhana untuk diterapkan. Dengan mempertimbangkan jumlah bidang data *input* yang akan diuji, persyaratan input, serta skenario batas bawah dan atas, kita dapat memperkirakan jumlah data yang akan diuji. Selain itu, pendekatan ini memungkinkan kita untuk mengevaluasi apakah fungsionalitas sistem masih dapat menangani input data yang tidak valid, yang dapat mempengaruhi validitas data yang disimpan.

Berdasarkan dari 2 penelitian diatas bisa disebutkan bahwa pengujian *black box* memiliki keunggulan dalam mengidentifikasi ketidaksesuaian dengan spesifikasi kebutuhan yang telah ditetapkan, namun terbatas dalam cakupan karena pengetahuan terbatas penguji terhadap perangkat lunak yang diuji. Metode ini hanya membutuhkan informasi tentang batas bawah dan batas atas dari data yang diperlukan, menjadikannya mudah diterapkan. Dengan mempertimbangkan jumlah bidang data input, persyaratan input, dan skenario batas bawah dan atas, kita dapat memperkirakan jumlah data yang akan diuji. Selain itu, pendekatan ini memungkinkan evaluasi terhadap kemampuan sistem dalam menangani input data yang tidak valid, yang berdampak pada validitas data yang disimpan.

15. System Usability Scale (SUS)

Menurut penelitian dari Tri et al., (2023 : 51-52), bahwa Metode System Usability Scale (SUS) adalah sebuah kuesioner yang sudah dirancang atau siap digunakan untuk mengumpulkan data penelitian. Metode ini berdasarkan tanggapan responden terhadap survei yang menggunakan Skala Likert. Terdapat 10 pertanyaan dalam kuesioner SUS, di mana jawabannya ditentukan oleh perasaan pengguna terhadap aplikasi. Responden akan menilai setiap pertanyaan dengan skor dari 1 hingga 5, tergantung pada sejauh mana mereka setuju dengan pernyataan dalam kuesioner. Jika responden merasa ragu dalam menentukan jawaban yang tepat, mereka dapat memilih titik tengah pada Skala Likert, yaitu netral.

Berdasarkan penelitian dari Arjiansa & Sutabri, (2023 : 116), bahwa System Usability Scale (SUS) adalah metode evaluasi yang digunakan untuk mengukur kemudahan penggunaan sebuah produk atau sistem. Metode ini melibatkan kuesioner yang terdiri dari 10 pertanyaan dengan skala 5 poin, yang dirancang untuk menilai aspek-aspek seperti kepuasan pengguna, kompleksitas, dan efisiensi. Hasil dari pengukuran SUS dapat digunakan untuk menentukan tingkat kemudahan penggunaan sistem dan memberikan rekomendasi perbaikan yang diperlukan.

Berdasarkan dari 2 penelitian diatas bisa disebutkan bahwa System Usability Scale (SUS) bisa didefinisikan sebagai metode evaluasi berbasis kuesioner yang terdiri dari 10 pertanyaan dengan skala 5 poin. Metode ini digunakan untuk mengukur kemudahan penggunaan sebuah produk atau sistem dengan menilai aspek kepuasan pengguna, kompleksitas, dan efisiensi, serta memberikan rekomendasi perbaikan berdasarkan hasilnya.

B. Kajian Teori

Berikut ini merupakan penelitian yang relevan untuk dijadikan acuan dalam penelitian ini :

Menurut penelitian dari Andika & Purnomo, (2023 : 136), bahwa penelitian ini berfokus pada diagnosis kecanduan gadget pada anak usia dini. Ketika penggunaan gadget menjadi berlebihan dan melampaui kebutuhan normal, pengguna cenderung menjadi ketagihan dan terus-menerus memainkan gadget mereka, bahkan tanpa alasan mendesak. Hal ini sangat berbahaya jika terjadi pada anak sekolah yang seharusnya aktif dalam kegiatan

yang dapat meningkatkan perkembangan kognitif, motorik, dan interaksi sosial mereka. Oleh karena itu, dibuatlah sistem pakar untuk mendiagnosis kecanduan gadget pada anak usia dini.

Menurut penelitian dari Yani et al., (2023 : 1), bahwa beberapa penelitian menunjukkan tingkat ketergantungan terhadap penggunaan internet terus meningkat meskipun dampak COVID-19 sudah mulai mereda. Remaja sering kali kesulitan memilih aktivitas online yang bermanfaat dan lebih rentan terhadap pengaruh negatif dari lingkungan pergaulan. Ketersediaan psikolog untuk memberikan bantuan tetap sangat terbatas, terutama di kota-kota kecil. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pakar yang bisa melakukan diagnosis kecanduan internet.

Menurut penelitian dari Wiyono et al., (2022 : 527), bahwa penggunaan media sosial saat ini sangat penting dalam kehidupan sehari-hari karena memudahkan berbagai aktivitas. Namun, penggunaan media sosial yang berlebihan cenderung memicu perubahan signifikan dan mempengaruhi pola hidup masyarakat. Aktivitas ini dapat menjadi kebiasaan yang, jika dilakukan terus-menerus, menyebabkan kecanduan dan membuat pengguna sulit melepaskan diri dari media sosial. Oleh karena itu, dikembangkanlah sebuah aplikasi sistem pakar yang dapat membantu masyarakat mendiagnosis kecanduan media sosial.

Menurut penelitian dari Iksan et al., (2023 : 273), bahwa remaja sering kali kesulitan memilih aktivitas online yang bermanfaat dan rentan terhadap

pengaruh negatif dari lingkungan pergaulan. Mereka memerlukan dukungan dari psikolog untuk mencegah dan mendiagnosis kecanduan internet sebelum menjadi parah. Namun, jumlah psikolog yang tersedia, di kota-kota kecil, ketersediaan psikolog sangat terbatas, dan faktor ekonomi serta keterbatasan waktu orangtua dalam mendampingi anak-anak mereka ke psikolog menjadi hambatan utama. Untuk mengatasi tantangan ini dan memberikan akses pelayanan psikologis kepada remaja, salah satu solusi yang diusulkan adalah pengembangan sistem pakar yang dapat berfungsi serupa dengan psikolog. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pakar yang dapat mendiagnosis kecanduan internet menggunakan pendekatan metode runut mundur.

Menurut penelitian dari Rizaldi, (2020 : 76), bahwa karena informasi mengenai penyakit yang disebabkan oleh penggunaan rokok elektrik masih sangat terbatas, diperlukan sistem terkomputerisasi untuk memberikan data akurat tentang masalah ini, yaitu sistem pakar. Oleh karena itu, dibuatlah aplikasi sistem pakar berbasis web untuk mengidentifikasi penyakit akibat penggunaan *vape* (rokok elektrik) dengan menggunakan metode *Certainty Factor*. Dengan adanya sistem ini, penyakit yang disebabkan oleh penggunaan *vape* dapat diidentifikasi dengan lebih tepat.

Beberapa penelitian tersebut memiliki kesamaan dalam tujuan untuk mengembangkan sistem pakar diagnosa berbasis web guna mendukung penelitian sistem pakar. Metode penelitian dan pengembangan digunakan dalam semua penelitian, dengan melibatkan validasi ahli. Hasilnya

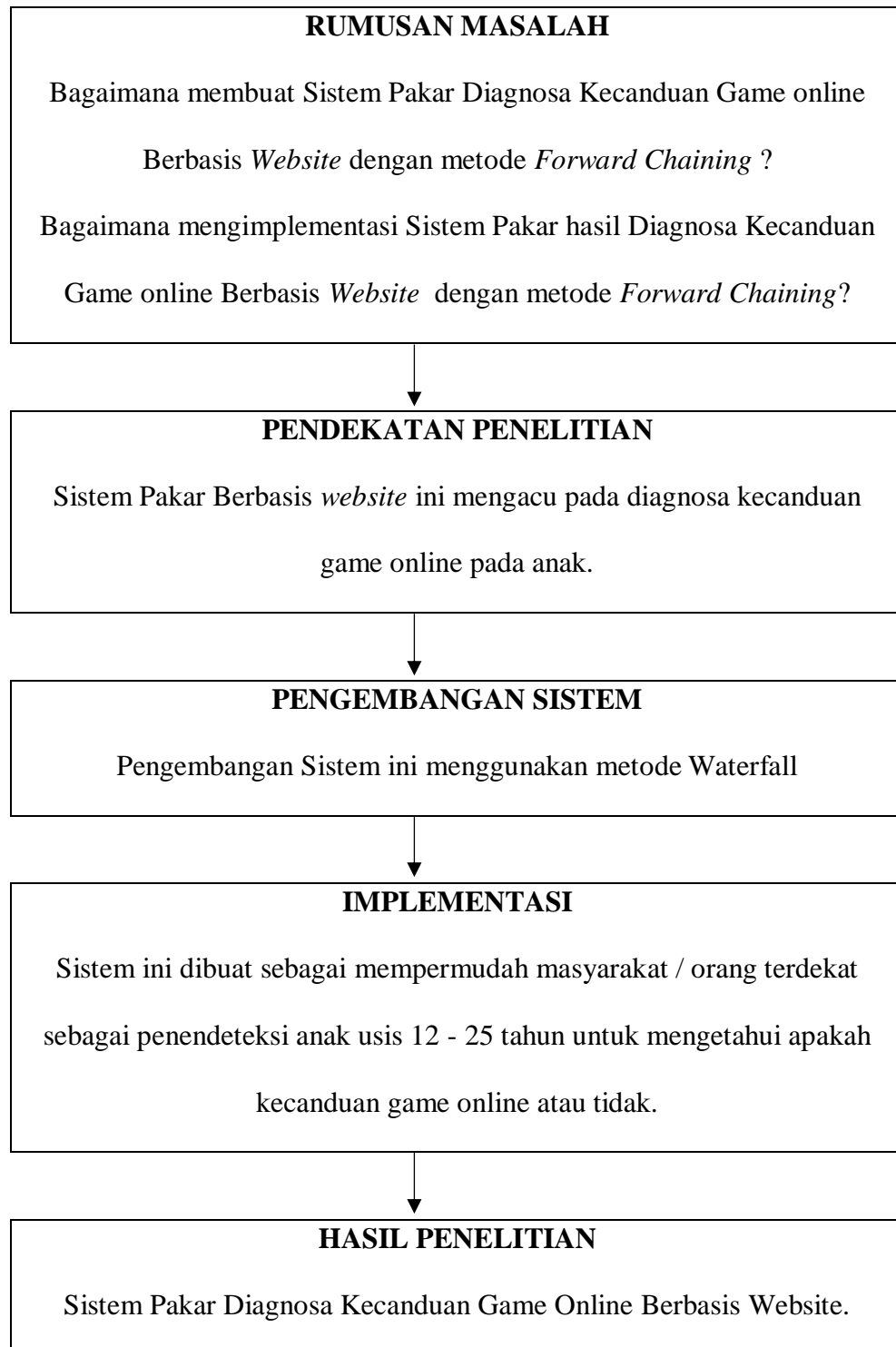
menunjukkan bahwa sistem pakar yang dikembangkan layak digunakan. Meskipun demikian, terdapat perbedaan dalam fokus diagnosa penyakit, tujuan penggunaan, dan metode evaluasi yang digunakan.

C. Kerangka Berfikir

Pada saat ini game online di Indonesia meningkat pesat. Terutama pada kalangan remaja masih merajalela di mana - mana. Saat ini sering diketahui oleh banyak masyarakat rata rata anak muda/remaja yang nongkrong di cafe, angkringan, dll. Untuk mencari wifi dan bermain game online. Game online saat ini sangat diminati oleh beragam kalangan dan profesi, mulai dari anak-anak hingga orang dewasa. Secara umum, alasan mengapa pelajar tertarik untuk bermain game online adalah untuk hiburan dan memenuhi rasa ingin tahu mereka terhadap teknologi. Sehingga menimbulkan dampak negatif kecanduan game online seperti kurangnya minat belajar, kesulitan mengatur waktu, pengeluaran uang jajan yang tidak efisien, penggunaan bahasa kasar, perilaku agresif, gangguan sosial, kesehatan fisik dan mental juga bisa terpengaruh negatif, sehingga menimbulkan kekhawatiran bagi orangtua.

Salah satu cara untuk mengatasi permasalahan kecanduan game pada anak. Maka penulis mengangkat / membuat sistem pakar kecanduan Game online berbasis *website*. Dengan adanya sistem ini diharapkan orangtua / penderita dapat mendiagnosa lebih dini tentang kecanduan game online berbasis *website*. Serta memudahkan masyarakat ketika diarea sekitar rumah kita sulit untuk mendapatkan psikolog.

Adapun kerangka berpikir pada penelitian ini sebagai gambar 2.1 berikut :



Gambar 2. 1 Kerangka Berfikir

