

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori dasar dapat diartikan sebagai pernyataan terstruktur secara sistematis dengan variabel yang kuat. Teori dasar substantif terdiri dari teori dan hasil penelitian, dimana teori dan hasil penelitian yang digunakan berfungsi sebagai kerangka teori bagi peneliti untuk melakukan penelitiannya.

1. Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “ *Artificial Intelligence* ” yang di singkat AI. *Intelligence* sendiri merupakan kata sifat yang berarti cerdas, *artificial* berarti buatan. Di sini, kecerdasan buatan mengacu pada mesin yang dapat berpikir, menimbang tindakan, dan membuat keputusan seperti manusia. Selain itu, ahli matematika Inggris Alan Turing, juga disebut sebagai bapak komputer modern dan pengurai kode Nazi selama Perang Dunia II pada tahun 1950, mendefinisikan kecerdasan buatan: "Ketika komputer tidak dapat dibedakan dari manusia ketika mereka menggunakan terminal komputer untuk berbicara, maka .”bisa disebut komputer. Dia cerdas, dia memiliki kecerdasan.”

Para filsuf telah berusaha mempelajari kecerdasan manusia selama bertahun-tahun. Dari pemikiran ini, kecerdasan buatan muncul sebagai cabang ilmu yang mencoba mempelajari dan meniru kecerdasan manusia. Sejak itu, para ilmuwan mulai memikirkan pengembangan kecerdasan buatan, sehingga teori dan prinsip terus berkembang hingga saat ini.

2. JST (Jaringan Saraf Tiruan)

Jaringan saraf tiruan adalah paradigma yang terinspirasi oleh sistem saraf biologis sebagai proses yang terjadi di otak manusia. JST mirip dengan cara orang bekerja: belajar dengan contoh. JST dikonfigurasi melalui proses pembelajaran untuk aplikasi spesifik seperti pengenalan pola atau klasifikasi data.

JST memiliki kemampuan luar biasa untuk mengekstraksi wawasan dari data yang kompleks atau tidak tepat, memecahkan masalah yang tidak terstruktur dan sulit dipahami, belajar dari pengalaman, mendapatkan pengetahuan meskipun tidak pasti, menggeneralisasi dari model data yang diberikan dan memanfaatkannya. , mampu membuat model pengetahuan melalui pengaturan diri atau kemampuan belajar (*self-organization*), mampu memilih data masukan dari kategori tertentu (klasifikasi), mampu mendeskripsikan objek secara keseluruhan, meskipun hanya ada yang diberikan adalah bagian dari objek data (asosiasi), memiliki kemampuan mengolah data input tanpa tujuan (*self-organization*), dan mampu menemukan jawaban terbaik untuk meminimalkan fungsi biaya (optimasi).

3. Fuzzy Logic

Konsep logika fuzzy diperkenalkan oleh Prof. Lotfi Astar Zadeh pada tahun 1962. Logika fuzzy adalah metodologi pemecahan masalah sistem kontrol yang cocok untuk implementasi dalam sistem sederhana, sistem kecil, sistem tertanam, jaringan PC, workstation, dan sistem kontrol data berbasis multi-channel atau akuisisi. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya.

Dalam logika klasik, semuanya disebut biner, yaitu hanya ada dua kemungkinan: "Ya" atau "Tidak", "Benar atau Salah", "Benar atau Salah" dan lain-lain. . Oleh karena itu, semua dapat memiliki nilai keanggotaan 0 atau 1. Namun, dalam logika fuzzy, nilai keanggotaan dapat berada di antara 0 dan 1.

2.2 Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem komputer yang meniru kemampuan pengambilan keputusan pakar manusia. Sistem pakar dirancang untuk memecahkan masalah yang kompleks dengan memperoleh pengetahuan yang terutama diwakili oleh aturan jika-maka daripada kode prosedural konvensional. Sistem pakar pertama kali dikembangkan pada tahun 1970-an dan kemudian menyebar luas pada tahun 1980-an.(Tinggi, Informatika, and Komputer 2022).

Sistem pakar memiliki kemampuan dan kelebihan tertentu, namun ada juga beberapa kekurangannya.

keuntungan:

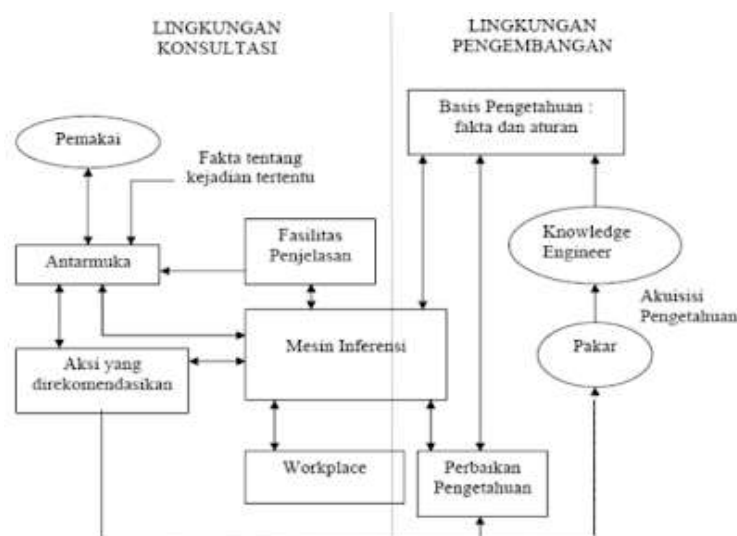
1. Meningkatkan produktivitas karena sistem pakar dapat bekerja lebih cepat dari manusia.
2. Menjadikan orang awam bekerja seperti seorang ahli.
3. Meningkatkan kualitas melalui nasehat yang konsisten dan mengurangi kesalahan.
4. Menangkap pengetahuan dan pengalaman seseorang.
5. Ini dapat digunakan di lingkungan berbahaya.
6. Memfasilitasi akses keahlian.
7. Andal.Sistem pakar tidak pernah bosan, lelah atau sakit.

Kekurangan:

1. Sangat mahal untuk membangun dan memelihara.
2. Sulit dikembangkan karena keterbatasan pengetahuan dan ketersediaan tenaga ahli.

2.2.1 Struktur Sistem Pakar

Kerangka sistem pakar terdiri dari dua bagian utama, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan sistem pakar berfungsi untuk mengintegrasikan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh non-ahli untuk memperoleh pengetahuan pakar.. (Alam et al. 2021).



Gambar 2. 1 Struktur Sistem Pakar
Sumber : KajianPustaka (2023)

Keterangan :

1. Antarmuka Pengguna (*User Interface*)

Antarmuka adalah mekanisme komunikasi antara pengguna dan sistem pakar. Antarmuka menerima informasi dari pengguna dan mengubahnya menjadi bentuk yang dapat diterima oleh sistem. Selain itu, antarmuka menerima data dari sistem dan menyajikannya dalam bentuk yang dapat dipahami pengguna.

2. Basis Pengetahuan

Basis pengetahuan berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan, dan memecahkan masalah.

3. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keterampilan pemecahan masalah dari sumber pengetahuan ke dalam program komputer. Pada fase ini, perancang pengetahuan mencoba menambahkan pengetahuan ke basis pengetahuan untuk kemudian ditransfer. Pengetahuan diperoleh dari para ahli, dilengkapi dengan buku, database, laporan penelitian, dan pengalaman pengguna.

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

Komponen ini berisi mekanisme berpikir dan penalaran yang digunakan oleh pemecah masalah berpengalaman. Mesin inferensi adalah program komputer yang menyediakan metodologi untuk menyimpulkan informasi dan merumuskan kesimpulan yang tersedia di basis pengetahuan dan ditempat kerja.

5. Workplace/Blackboard

Workplace adalah area memori kerja yang digunakan untuk menyimpan kejadian terkini, termasuk keputusan sementara.

6. Fasilitas Penjelasan

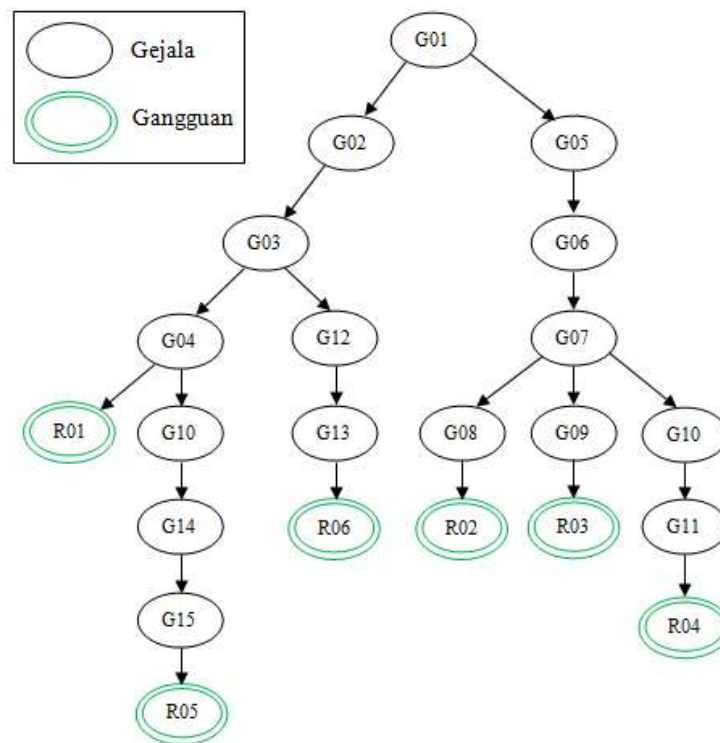
Fasilitas penjelasan memperluas kemampuan sistem pakar dan digunakan untuk melacak tanggapan dan memberikan penjelasan tentang perilaku sistem pakar secara interaktif melalui pertanyaan.

7. Perbaikan Pengetahuan

Ahli memiliki kemampuan untuk menganalisis dan memperbaiki hasil mereka sendiri dan belajar darinya. Keterampilan ini penting dalam pembelajaran berbasis komputer agar program dapat menganalisis alasan keberhasilan dan kegagalan serta menilai apakah pengetahuan yang ada masih dapat digunakan di masa mendatang.

2.2.2 Pohon Keputusan

Pohon keputusan adalah bentuk representasi pengetahuan yang digunakan untuk memodelkan masalah yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Pohon keputusan dibangun dengan mereferensikan tabel keputusan.



Gambar 2. 2 Pohon Keputusan
Sumber : widuri-raharja (2023)

Berdasarkan pohon keputusan pada gambar di atas, sistem melacak tingkat depresi yang ada dengan mengajukan pertanyaan kepada pengguna aplikasi (dalam hal ini gejala depresi yang dialami siswa). Pengguna harus menjawab "Ya" atau "Tidak". Namun, jika dia tidak yakin jawaban mana yang harus diberikan, dia harus menjawab "ya" atau "tidak", tergantung jawaban mana yang paling dekat dengan pertanyaan atau gejala yang diberikan. Berdasarkan respon pengguna, sistem terus mencari gejala selanjutnya dan menyajikan pertanyaan atau gejala berikutnya untuk penilaian akhir (dalam hal ini adalah tingkat depresi siswa).

2.3 *Certainly Factor*

Certainly Factor adalah metode untuk menghadapi ketidakpastian dalam sistem berbasis aturan. Pada pertengahan 1970-an, Shortliffe dan Buchanan

(1975) mengembangkan model CF untuk MYCIN, sebuah sistem pakar untuk diagnosis dan pengobatan meningitis dan infeksi aliran darah. Sejak saat itu, model CF telah menjadi pendekatan standar untuk manajemen ketidakpastian sistem berbasis aturan. (Tinggi et al. 2022). Perhitungan nilai *Certainly Factor* adalah sebagai berikut :

1. Menghitung Nilai CF

$$CF[H,E]= MB[H,E] - MD[H,E] \dots\dots (1)$$

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

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

MD(H,E):Ukuran ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E H: Hipotesa atau konklusi yang dihasilkan (antara 0 dan 1) E: Evidence atau peristiwa atau fakta (gejala) Bentuk dasar rumus *Certainly Factor* adalah sebuah aturan

JIKA E MAKA H seperti ditunjukkan oleh persamaan 2 berikut :

$$CF (H, e) = CF (E, e) * CF (H, E) \dots (2)$$

Keterangan : CF (H, e): *Certainly Factor*

hipotesis yang dipengaruhi oleh evidence e.

CF (E, e): *Certainly Factor* evidence E yang dipengaruhi oleh evidence e.

CF (H, E) : *Certainly Factor*

hipotesis dengan asumsi evidence diketahui dengan pasti, yaitu ketika $CF(E, e) = 1$

2. Menghitung Nilai CFcombine

$CF_{combine} [CF1, CF2] = CF1 + CF2 * [1 - CF1]$ (3)

2.4 Variabel

Variabel penelitian adalah atribut, properti, atau nilai seseorang, objek, organisasi, atau aktivitas yang menunjukkan variasi spesifik yang ditentukan oleh peneliti untuk dipelajari dan ditarik kesimpulannya. Depresi merupakan gangguan jiwa yang umum terjadi di masyarakat. Stres yang tidak terselesaikan dapat menyebabkan seseorang masuk ke fase depresi. Gejala ini sering diabaikan karena dianggap akan hilang dengan sendirinya tanpa pengobatan. Orang yang menderita depresi umumnya menderita gangguan yang melibatkan gerakan emosional, motivasional, fungsional, dan kognitif-perilaku. Ada 4 tingkat depresi: gangguan mood, depresi ringan, depresi sedang, dan depresi berat.

1. *Mood Disorder* (Gangguan Mood)

Gangguan suasana hati atau *Mood Disorder* adalah gangguan mental yang mempengaruhi keadaan emosional seseorang. Gangguan ini menyebabkan seseorang mengalami kebahagiaan yang luar biasa, kesedihan yang luarbiasa, atau keduanya untuk jangka waktu yang lama. Perubahan suasana hati adalah normal untuk semua orang. Bergantung pada situasi saat ini, seseorang bisa sedih, marah, dan bahagia dari waktu ke waktu.

2. *Mild Depression* (Depresi Ringan)

Pada level ini, gejala seringkali berdampak pada aktivitas sehari-hari orang yang mengalaminya, seperti Kurangnya minat terhadap hal-hal yang sering dilakukan, mudah tersinggung dan berkurangnya motivasi kerja. Depresi ini tidak memprihatinkan, tetapi perlu diobati untuk mencegah kondisi menjadilebih buruk.

3. *Middle Depression* (Depresi Sedang)

Depresi ini menyebabkan seseorang mengalami kesulitan dalam aktivitas sosial, pekerjaan. Secara umum, dengan depresi sedang, seseorang kurang percaya diri dan/atau harga diri, mengakibatkan kurangnya motivasi untuk melakukan sesuatu. Seringkali seseorang mengkhawatirkan hal-hal sensitif yang tidak perlu dan cenderung merasa sakit hati atau tersinggung dalam hubungan pribadi.

4. *.Severe Depression* (Depresi Berat)

Dengan depresi ini, orang tersebut mengalami kesusahan yang parah, seperti kehilangan harga diri atau perasaan tidak berharga dan bersalah, serta keinginan untuk bunuh diri. Seseorang yang menderita depresi berat tidak dapat mengendalikan emosinya dan karena itu mudah mengalami perasaan putus asa. Orang dengan episode depresi berat juga dapat mengalami delusi, halusinasi, atau pingsan depresi.

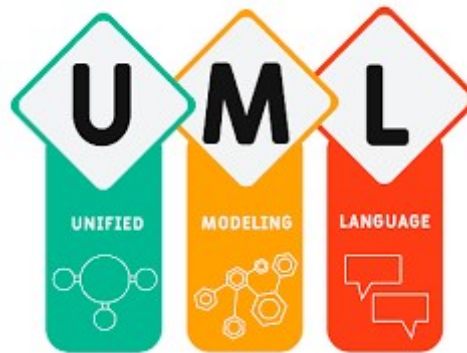
2.5 Desain Perancangan

Desain perancangan digunakan untuk proses perencanaan, pengembangan dan pembuatan desain untuk membuat suatu sistem.

1. UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah bahasa berbasis grafik/gambar untuk memvisualisasikan, menentukan, menulis, dan mendokumentasikan sistem pengembangan perangkat lunak berorientasi objek (OO).(Mubarak 2019).

UML sendiri juga menyediakan standar penulisan desain sistem yang mencakup konsep proses bisnis, penulisan kelas dalam bahasa pemrograman tertentu, skema basis data, dan komponen yang diperlukan dalam sistem perangkat lunak. UML sebagai bahasa yang menyediakan kosakata dan urutan penulisan kata dalam “MS Word” untuk keperluan komunikasi. Bahasa model adalah bahasa yang memiliki kosa kata dan seperangkat aturan penulisan konseptual dan mewakili representasi fisik dari suatu sistem. Sama seperti UML adalah bahasa pengembangan perangkat lunak standar, ia dapat mengajarkan bagaimana model dibuat dan dibentuk, tetapi tidak mengajarkan apa dan kapan membuat model, yang merupakan salah satu proses implementasi pengembangan perangkat lunak. Selain sebagai bahasa pemrograman visual, UML juga dapat dihubungkan langsung ke berbagai bahasa pemrograman seperti JAVA, C++, Visual Basic, atau bahkan langsung ke database berorientasi objek.



Gambar 2. 3 Logo UML
Sumber : Dreamstime (2023)






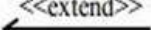
2. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use Case Diagram* bisa sangat berguna ketika kita menangkap persyaratan untuk sistem, mengkomunikasikan desain ke pelanggan, dan merancang test case untuk semua fungsi sistem.

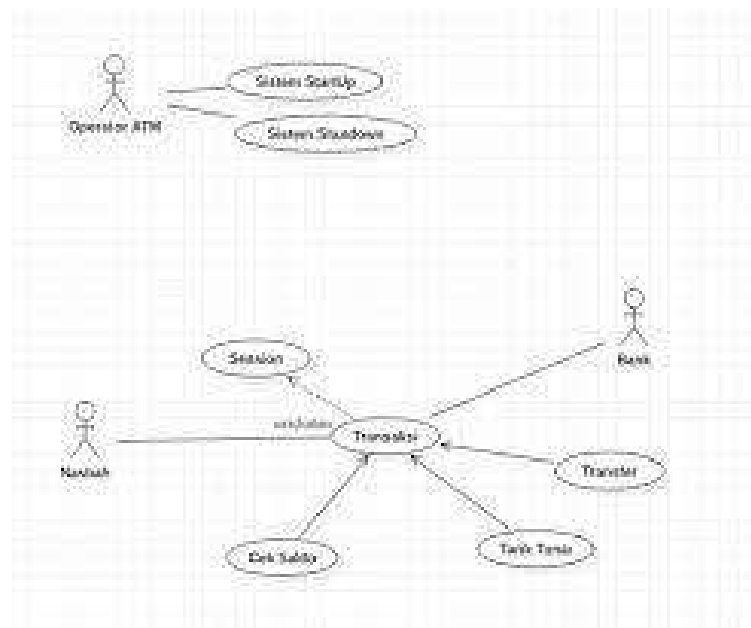
Sebuah use case dapat mencakup fungsionalitas use case lain sebagai bagian dari proses internalnya sendiri. Secara umum, use case yang disertakan diasumsikan dipanggil setiap kali use case yang disertakan berjalan normal. Sebuah use case dapat dimuat di dalam lebih dari satu use case lainnya, sehingga duplikasi fungsionalitas dapat dihindari dengan mengekstraksi fungsionalitas

umum. Sebuah use case juga dapat memperluas use case lain dengan perilakunya sendiri. Sedangkan hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari use case lainnya.

Tabel 2. 1 *Use Case Diagram*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Sumber : Dicoding (2023)



Gambar 2. 4 Contoh *Use Case Diagram*
 Sumber : Dicoding (2023)


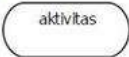



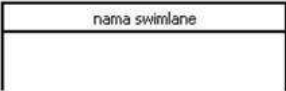


3. Activity Diagram

Activity Diagram menggambarkan aliran aktivitas yang berbeda dalam sistem yang akan dirancang, bagaimana setiap aliran dimulai, keputusan apa yang dapat dibuat, dan bagaimana akhirnya. Diagram aktivitas juga dapat menggambarkan proses paralel yang dapat berlangsung selama beberapa kaliproses. Diagram aktivitas adalah diagram keadaan khusus di mana sebagian besar keadaan adalah tindakan dan sebagian besar transisi dipicu oleh penyelesaian keadaan sebelumnya (pemrosesan internal). Oleh karena itu, diagram aktivitas tidak secara tepat menggambarkan perilaku internal suatu sistem (dan interaksi antar subsistem), melainkan menggambarkan proses dan jalur aktivitas dari tingkat yang lebih tinggi secara umum.

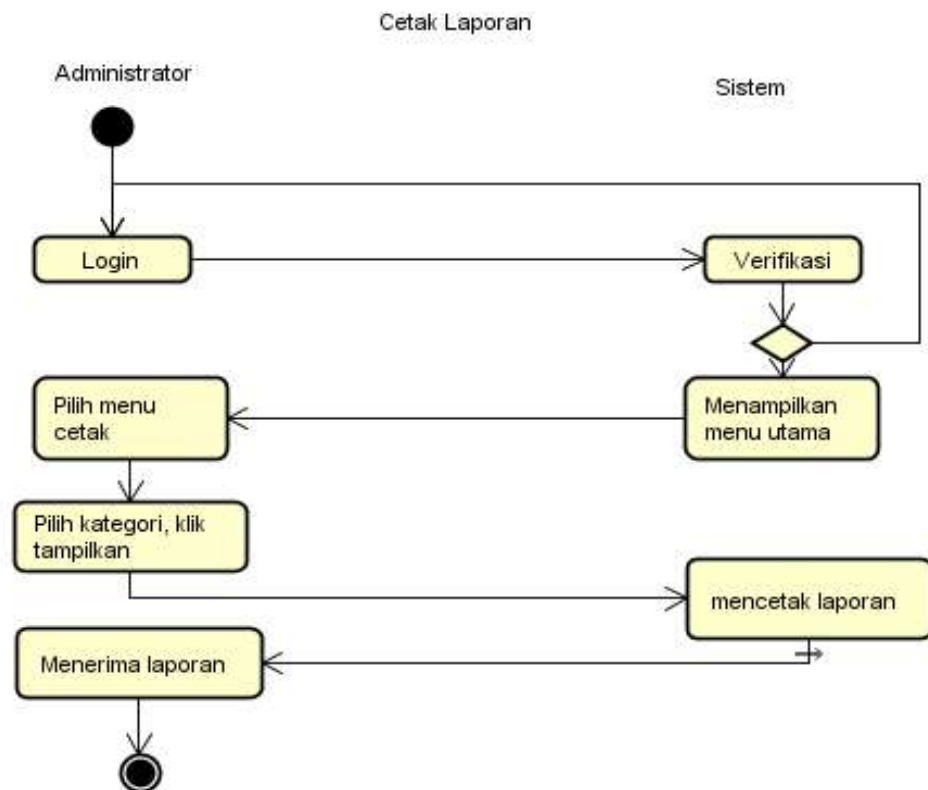
Sebuah aktivitas dapat dieksekusi oleh satu atau lebih use case. Aktivitas mendeskripsikan proses yang sedang berlangsung, sedangkan use case

mendeskripsikan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Untuk status, standar UML menggunakan persegi panjang dengan sudut membulat untuk merepresentasikan aktivitas. Keputusan digunakan untuk menggambarkan perilaku dalam kondisi tertentu. Untuk mengilustrasikan proses paralel (fork dan join), titik sinkronisasi digunakan, yang dapat berupatitik, garis horizontal atau vertikal. Diagram aktivitas dapat dipecah menjadi relobjek untuk menunjukkan objek mana yang bertanggung jawab atas aktivitas tertentu.

Tabel 2. 2 Activity Diagram

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / decision 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / join 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
fork, 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
join, 	digunakan utk menunjukkan kegiatan yg digabungkan

Sumber : Badoy Studio (2023)



Gambar 2. 5 Contoh Activity Diagram







umber : PelajarIndo (2023)

4. Sequence Diagram

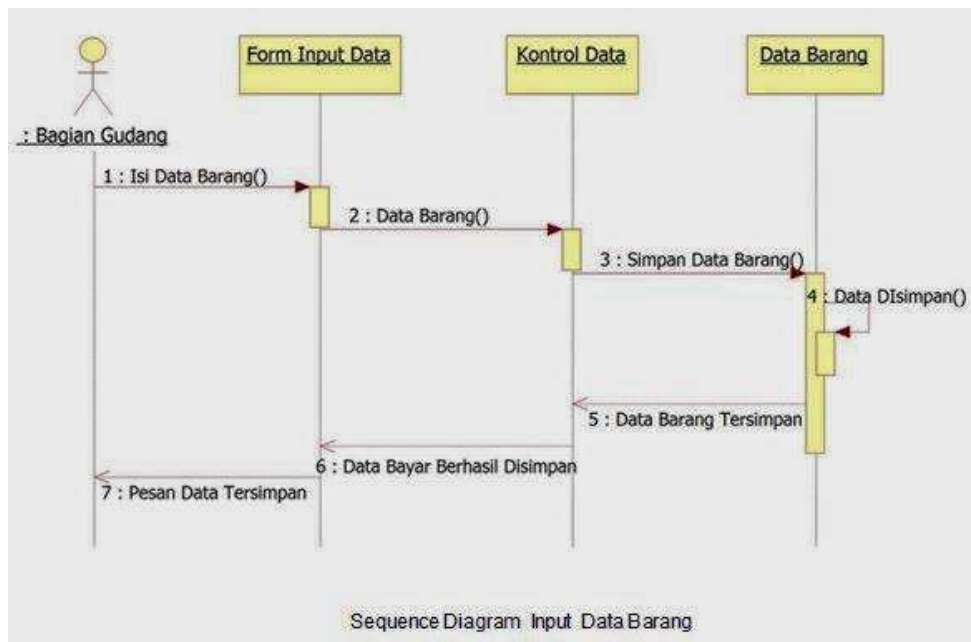
Sequence Diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, layar, dll.) dalam kaitannya dengan pesan yang diplot terhadap waktu. *Sequence Diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek terkait). *Sequence Diagram* biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons terhadap suatu peristiwa untuk mencapai hasil tertentu. Apa yang memicu aktivitas, proses dan perubahan apa yang terjadi secara internal dan produk apa yang diproduksi.

Objek apa pun, termasuk aktor, memiliki garis hidup vertikal. Pesandirepresentasikan sebagai garis dengan panah dari satu objek ke objek lainnya. Pada fase desain berikutnya, pesan dipetakan ke operasi/metode kelas. Bilah aktivasi menunjukkan durasi eksekusi proses, biasanya dimulai dengan kedatangan pesan. Untuk objek dengan properti khusus, standar UML mendefinisikan simbol khusus untuk objek batas, pengontrol, dan entitaspersisten.

Tabel 2. 3 Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Sumber : Ilmu Teknologi (2023)



Gambar 2. 6 Contoh *Sequence Diagram*
Sumber : Pinhome (2023)

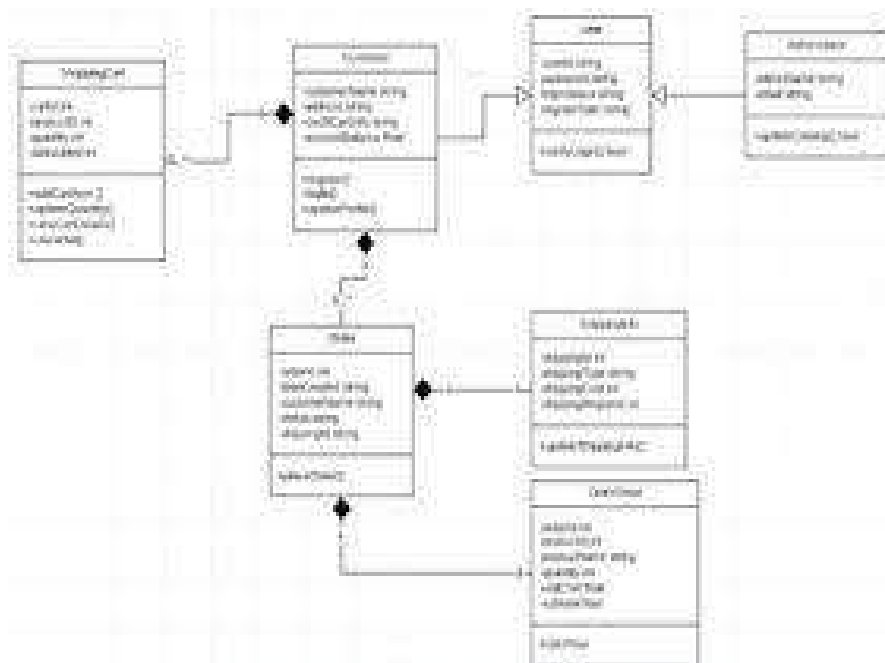
5. Class Diagram

Kelas adalah spesifikasi yang membuat objek ketika dibuat dan merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) dari suatu sistem dan menawarkan layanan untuk memanipulasi keadaan ini (metode/fungsi). Diagram kelas menggambarkan struktur dan deskripsi kelas, paket, dan objek serta hubungannya satu sama lain, seperti warisan, asosiasi, dll. Kelas bisa menjadi implementasi dari sebuah antarmuka, yaitu kelas abstrak yang hanya memiliki metode. Antarmuka tidak dapat dibuat instance-nya secara langsung, tetapi pertama-tama harus diimplementasikan sebagai kelas. Oleh karena itu, antarmuka mendukung resolusi metode saat runtime. Menurut evolusi model kelas, kelas dapat dikelompokkan ke dalam paket.

Tabel 2. 4 Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1	— —	Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	◇	Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3	▭	Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4	○	Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5	←-----	Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
6	----->	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	—	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : Pinhome (2023)



Gambar 2. 7 Contoh Class Diagram

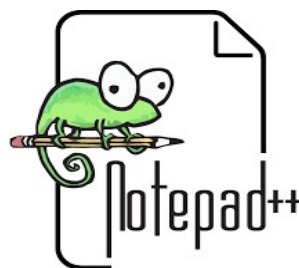
Sumber : waskhas (2023)

2.6 Software Pendukung

Perangkat lunak pendukung ini digunakan dalam proses pembangunan sistem. Dimana nantinya akan dilakukan proses perancangan dan pengkodean sistem.

1. Notepad++

Notepad++ adalah program aplikasi pengembang yang berguna untuk mengedit teks dan skrip kode pemrograman. Versi terbaru program ini adalah Notepad++ v5.9, yang dirilis pada tanggal 06 April 2012. Software Notepad++ dibuat dan dikembangkan oleh Tim Notepad++. Perangkat lunak komputer ini memiliki kelebihan pada peningkatan kemampuan sebuah program text editor, lebih dari sekedar program Notepad bawaan Windows. Notepad++ bisa mengenal tag dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut dan pengeditan teks yang tersedia juga cukup ampuh, sangat membantu tugas seorang programmer atau developer dalam menyelesaikan skrip kode programnya. Program Notepad++ banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi desktop dan web. Notepad++ merupakan software gratis (opensource).



Gambar 2. 8 Logo Notepad++

Sumber : wikipedia (2023)

2. PHP

PHP atau singkatnya *Hypertext Preprocessor*, adalah bahasa pemrograman sumber terbuka yang sangat cocok untuk atau secara khusus diarahkan untuk pengembangan web dan dapat disematkan dalam skrip HTML. Dapat dikatakan bahwa bahasa PHP menggambarkan berbagai bahasa pemrograman seperti C, Java dan Perl dan mudah untuk dipelajari. PHP adalah bahasa skrip sisi server di mana pemrosesan data dilakukan sisi server. Sederhananya, itu adalah server yang menerjemahkan skrip program dan hasilnya dikirim ke klien yang membuat permintaan.

**Gambar 2. 9** Logo PHP

Sumber : Flaticon (2023)

3. MYSQL

MySQL adalah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara bebas di bawah GPL (General Public License). Setiap pengguna bebas menggunakan MySQL dengan peringatan bahwa perangkat lunak tersebut tidak boleh digunakan sebagai produk turunan komersial. MySQL sebenarnya merupakan turunan dari salah satu konsep utama database lama; SQL (Bahasa Kueri Terstruktur).



Gambar 2. 10 Logo MYSQL
Sumber : Icon-Icons (2023)

4. XAMP

XAMPP adalah perangkat lunak yang dikembangkan, berjalan di banyak sistem operasi dan merupakan perangkat lunak gratis. XAMPP berfungsi sebagai server terpisah yang disebut localhost. Ada beberapa kombinasi perangkat lunak dalam XAMPP yaitu Pemrograman PHP, Pemrograman Perl, Apache HTTP Server dan Database MySQL. XAMPP banyak digunakan oleh programmer yang ingin membuat aplikasi online karena kemudahan penggunaannya. Pada umumnya pemula yang ingin coding sangat disarankan untuk menggunakan XAMPP.(Dalimunthe 2020)



Gambar 2. 11 Logo XAMP
Sumber : Nesabamedia (2023)

5. Framework Bootstrap

Kerangka kerja Bootstrap adalah pustaka CSS yang dibuat khusus untuk mengembangkan ujung depan situs web. Bootstrap juga dikenal sebagai kerangka kerja CSS, HTML, dan Javascript yang digunakan oleh pengembang situs web untuk mengembangkan situs web dan membuatnya lebih responsif. Tentunya dengan Bootstrap, halaman website bisa diubah ukurannya sesuai ukuran perangkat pengawasan. Baik akses melalui perangkat seluler, tablet, atau komputer desktop. Awalnya, Bootstrap menjuluki dirinya Cetak Biru Twitter, dibuat dan dikembangkan oleh Jacob Thornton dan Mark Otto di Twitter untuk alat kerja yang dapat meningkatkan konsistensi dalam alat internal mereka. Tentunya dengan menggunakan Bootstrap Framework, seorang developer dapat membuat pembuatan front end pada sebuah website menjadi lebih mudah dan cepat.. (Martani, Saripuddin, and Ikhsan 2022)



Gambar 2. 12 Logo Bootstrap
Sumber : getbootstrap (2023)

6. Framework JQuery

Jquery JQuery dengan semboyannya “write less, do more”, merupakan javascript library yang instan serta ringan untuk menangani dokumen HTML, menangani event, membuat animasi dan interaksi ajax. JQuery bertujuan untuk

mengubah cara menulis javascript anda mejadi ringkas. Untuk memahami JQuery anda harus mempelajari dahulu dasar mengenai HTML, CSS dan Javascript. (Nurul Utama et al. 2022).



Gambar 2. 13 Logo JQuery
Sumber : Alfindo Informatika (2023)

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak diperlukan untuk mengkonfirmasi dan memvalidasi hasil pengembangan perangkat lunak. Pengujian dilakukan secara sistematis menggunakan sejumlah strategi pengujian perangkat lunak. Pengujian perangkat lunak digunakan untuk mencegah kesalahan manusia dalam suatu sistem, atau dengan kata lain untuk mencegah kesalahan pada perangkat lunak yang ada. Pengujian perangkat lunak adalah metode untuk memastikan bahwa perangkat lunak memenuhi persyaratan yang ada dan memastikan bahwa perangkat lunak tersebut bebas dari kesalahan. Pengujian perangkat lunak diimplementasikan menggunakan komponen manual atau otomatis untuk mengevaluasi satu atau lebih properti yang dipilih. Pengujian perangkat lunak diperlukan untuk mendeteksi kekurangan dan kesalahan yang dibuat selama *fase* pengembangan.. (Huda et al. 2022) Pengujian ini dapat dilakukan menggunakan *black box* atau *white box*..

1. *Black box* testing bisa disebut juga sebagai pengujian fungsional yang dibuat berdasarkan spesifikasi dari klien dan penguji dalam *black box* testing tidak memiliki akses ke dalam kode program.
2. *White box*. testing bisa disebut juga sebagai pengujian kotak kaca atau pengujian struktural dimana pengujian yang dikembangkan berdasarkan pada kode program.

2.8 Penelitian Terdahulu

Sebagai bahan pertimbangan dan referensi dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, sebagai berikut jurnal terdahulu :

- 1) (Koli et al. 2022) **SISTEM PAKAR IDENTIFIKASI TINGKAT STRES PADA MAHASISWA SELAMA MASA PANDEMIC COVID - 19 MENGGUNAKAN METODE FORWARD CHAINING DAN CERTAINLY FACTOR BERBASIS WEBSITE.** Covid-19 pertama kali menyerang kehidupan manusia pada akhir tahun 2019. WHO menetapkan Covid-19 menjadi pandemic global sejak Maret 2020. Dalam rangka pencegahan terhadap penyebaran Covid-19 di Indonesia, Menteri Pendidikan dan Kebudayaan Republik Indonesia memberlakukan pembelajaran daring bagi mahasiswa. Perubahan sistem pembelajaran ini tentu sangat memengaruhi kehidupan mahasiswa. Dalam proses beradaptasi dengan sistem baru, mahasiswa menghadapi berbagai jenis tantangan. Tantangan yang dihadapi berpotensi menjadi stresor. Tujuan dari penelitian ini adalah untuk membantu mahasiswa mengetahui tingkat stres yang dialaminya. Analisis dilakukan

dengan mengidentifikasi tingkat stres pada mahasiswa menggunakan metode pada sistem pakar yakni metode Forward Chaining, serta menggunakan perhitungan *Certainly Factor* untuk menambahkan tingkat keyakinan hasil identifikasi. Hasil akhir dari website yang dibuat berupa keterangan tingkat stres yang dialami mahasiswa tersebut.

- 2) (Purnomo et al. 2022) **Sistem Pakar Mendiagnosa Gangguan Mental pada Diri Seseorang Menggunakan Metode *Certainly Factor***. Penderita gangguan mental sangat mulai banyak di Indonesia, hasil riset kesehatan pada tahun 2015, 1.193.151 orang atau 0,467 % penduduk yang mengidap penyakit gangguan mental. Penyakit gangguan mental ini tidak boleh didiamkan lagi, harus ditindaklanjuti ataupun diminimalisir agar bisa meningkatkan kembali kondisi kesehatan di Indonesia. Tingkat kemiskinan di Indonesia sampai saat sekarang ini masih tergolong tinggi dan membuat masyarakat susah dalam penanganan penyakit gangguan mental tersebut. Adapun untuk menimalisir penanganan gangguan mental ini, sangat cocok dibangun suatu sistem pakar menggunakan metode *Certainly Factor*, yang merupakan suatu metode yang mendefinisikan ukuran kapasitas terhadap suatu fakta atau aturan, dalam mengekspresikan tingkat keyakinan seorang pakar terhadap suatu masalah yang sedang dihadapi seorang pasien atau penderita dan juga dapat digunakan sebagai tolak ukur dari hasil nilai diagnosa terhadap suatu penyakit yang diderita. Dengan pembuatan sistem pakar ini dapat bermanfaat bagi masyarakat banyak dan dapat mengetahui dengan jelas tentang penyakit gangguan mental dari gejala dan solusinya dengan berbasis website dan menggunakan

pemrograman PHP dengan database MySQL. Hasil penelitian menggunakan metode *Certainly Factor* dengan pengujian yang telah dilakukan oleh user atau pengguna adalah bahwa jenis gangguan mental yang diderita yaitu jenis Skizofrenia dengan Tingkat Kepastian 56%.

- 3) (Santi and Andari 2019) **Sistem Pakar Untuk Mengidentifikasi Jenis Kulit Wajah dengan Metode *Certainly Factor***. Peran ahli kecantikan wajah sangat penting untuk mengidentifikasi jenis kulit wajah serta memberikan solusi perawatan yang tepat untuk setiap jenis kulit wajah. Dari ini, sistem pakar diperlukan untuk membantu memberikan solusi dengan membangun sistem pakar yang dapat mengidentifikasi jenis kulit wajah dengan memasukkan solusi perawatan. Metode *Certainly Factor* memilih berdasarkan pertimbangan dalam proses perhitungan, dan metode ini mencari kombinasi nilai kepercayaan tertinggi. Pada awal studi pengumpulan data dilakukan pada 40 responden wanita yang diperoleh 100% responden tidak memahami jenis kulit wajah dan 76% mengatakan mereka membutuhkan ahli, 95% membutuhkan aplikasi sistem pakar. Sementara hasil penilaian sistem aplikasi yang telah dibangun oleh responden menyatakan bahwa 88% dari desain sistem sangat baik dan sangat baik 91% dari sistem mudah digunakan, dan 98% mengatakan operasi itu dengan apa yang dibutuhkan.
- 4) (Rahmadani, Setianingsih, and Dirgantara 2022) **Tes Gangguan Depresi, Kecemasan, Dan Stres Pada Mahasiswa Selama Masa Covid- 19 Menggunakan Algoritma Naïve Bayes Testing Depression, Anxiety, And Stress Disorders In Students During The Covid-19 Pandemic Using Naïve**

Bayes Algorithm. Pada masa pandemi Covid-19 di Indonesia melakukan pembelajaran daring dari rumah sebagai upaya pembatasan sosial selama pandemi. Proses pembelajaran daring ini dinilai masih kurang efektif dan efisien dan mengakibatkan beberapa pelajar khususnya mahasiswa memiliki tumpukan pekerjaan rumah pada masa pembelajaran daring. Hasil survei mengenai kesehatan mental pada masa pandemi yang dilakukan oleh Perhimpunan Dokter Spesialis Kedokteran Jiwa Indonesia (PDSKJI) menunjukkan 64,8% responden mengalami masalah psikologi dengan kelompok usia 19-24 tahun dan lebih 60 tahun. Dalam penelitian tugas akhir ini penulis akan membuat sistem Tes Gangguan Depresi, Kecemasan dan Stres pada Mahasiswa. Hasil dari tes ini berupa tingkat keparahan dari masing-masing gangguan psikologis serta rekomendasi treatment berdasarkan hasil tes tersebut. Skala psikologi yang digunakan pada penelitian ini adalah DASS- 42 (Depression, Anxiety, and Stress Scale) yang memiliki 42 pernyataan. Sistem Tes Gangguan Depresi, Kecemasan, dan Stres pada Mahasiswa ini menggunakan metode Naïve Bayes dengan akurasi dari dataset yang didapatkan sebesar 86.44%. Untuk akurasi uji alfa sebesar 100% dan ini siswa akurasi sistem pada implementasi website sebesar 88%.

- 5) (Tinggi et al. 2022) **SISTEM PAKAR DIAGNOSA STRESS PADA MAHASISWA TINGKAT AKHIR DENGAN MENGGUNAKAN METODE *CERTAINLY FACTOR* BERBASIS ANDROID.** Sistem Pakar Diagnosa Stres pada Mahasiswa Tingkat Akhir dengan menggunakan Metode *Certainly Factor* merupakan sebuah aplikasi berbasis android yang dapat

mengukur tingkat stres yang dialami oleh mahasiswa tingkat akhir. Stres merupakan salah satu yang dapat menyebabkan keterhambatan dalam pengerjaan skripsi pada mahasiswa tingkat akhir. Dengan dibuatnya aplikasi ini diharapkan dapat membantu mahasiswa dalam mengukur tingkat stres yang dialami serta dapat segera mengatasinya. Aplikasi yang dibuat menggunakan bahasa pemrograman Kotlin, IDE Android Studio, dan Firebase Firestore sebagai Database.

6) (Wubshet et al. 2022) **Prevalence and Associated Factors of Poststroke**

Depression among Outpatient Stroke Patients Who Have a Follow-Up at the Outpatient Neurology Clinic of Zewditu Memorial Hospital in Addis Ababa, Ethiopia Tsion. Poststroke depression is the most common and burdensome poststroke psychiatric complication. Studies showed discrepancies in reporting frequencies and risk factors for poststroke depression. Updated local data are relevant for efficient strategies of poststroke depression screening and prevention. Objectives. To determine the prevalence and associated factors of poststroke depression among outpatient stroke patients from the outpatient neurology clinic of Zewditu Memorial Hospital in Addis Ababa, Ethiopia. Methods. An institution-based cross-sectional study was conducted on 249 stroke patients. Data was collected through structured questionnaire using interviews and a review of medical charts. PHQ-9 depression questionnaire was used to diagnose poststroke depression. Descriptive analysis was used to see the nature of the characteristics of interests. Bivariate analysis was used to sort out variables at p values less than 0.05 for multivariate logistic regression.

Significance level was obtained using an odds ratio with 95% CI and p value < 0.05. Results. Point prevalence for poststroke depression was 27.5 percent. Female gender, unemployment, low social support level, diabetes mellitus, and poststroke period under 2 years were statistically significant and independent predictors for poststroke depression. Conclusions. The point prevalence estimate of poststroke depression was comparable with other studies. Low social support levels increased the odds for poststroke depression by more than eight folds. It appeared that external factors are more important in the pathogenesis of poststroke depression in the African population. Detection and prevention programs should consider disparities of poststroke depression incidence and risk factors.

- 7) (Tomaszek and Muchacka-Cymerman 2022) **Student Burnout and PTSD Symptoms: The Role of Existential Anxiety and Academic Fears on Students during the COVID 19 Pandemic Katarzyna**. It is well known that student burnout is a serious mental health problem, caused by chronic stress related to the educational area. However, in the COVID 19 pandemic, young people have to struggle with additional threats that affect their overall functioning and perception of the world. The main purpose of this study was to investigate the mediating effects of existential anxiety and academic fears on the relationship between academic burnout and posttraumatic stress disorder symptoms. The findings confirmed that academic burnout, existential anxiety, and academic fear were significantly associated with higher posttraumatic symptoms. Existential anxiety and academic fear played a mediating role in the

association between academic burnout and posttraumatic stress disorder symptoms. In conclusion, it is considered that student burnout and anxiety indicators are important risk factors for the trauma experienced by students and may increase its symptoms.

2.9 Kerangka Pemikiran

Untuk mengetahui tingkat depresi pada siswa perlu dilakukan tes untuk menentukan derajat depresi pada siswa sebagai bahan referensi dan menghindari kesalahan dalam menentukan tingkat depresi pada siswa tanpa dasar apapun. Sistem pakar ini merupakan metode pengujian yang dapat digunakan untuk mengetahui tingkat depresi pada siswa.



Gambar 2. 14 Kerangka Pemikiran
Sumber : Peneliti (2023)

Berdasarkan kerangka penelitian di atas, bahwa dalam penelitian ini berjalan, sebagai berikut :

1. Input : Memasukkan data penyebab atau gejala depresi siswa yang dialami siswa. Seperti Merasa sedih atau Mood yang buruk, Kadang kala melamun atau menghayal berkepanjangan, Mudah bosan dan kesal, Konsentrasi dan perhatian yang berkurang, Mudah merasa lelah dalam beraktivitas, Tidak berantusiasme, Selalu merasa khawatir atau pesimis,

Putus asa dan kehilangan harapan masa depan , Wajah sedih dan kadang berlinang air mata, Insomnia atau susah tidur, Selalu gelisah, Tidak puas dengan diri sendiri dan merasa tidak berharga, Terusik dengan segala hal, Selalu terlihat lesu dan tidak bertenaga, Menjadi tidak tertarik dengan minat dan hobi yang disenangi, Kesunyian, Memiliki pikiran rasa bersalah, Memiliki pikiran dianiaya, Benci dan selalu mengkritik diri sendiri, Gampang tersinggung dan tersentuh, Nafsu makan hilang, Cemas akan performa, Sangat sensitive dan tidak terkontrol emosi, *Introvert*, Selalu berpikir untuk mengakhiri hidup, Tidak mudah menentukan pilihan, Tidak mudah beraktivitas dengan baik , Berat badan yang tidak stabil, Insecure atau kurang percaya diri.

2. Proses : Melakukan pengkodean dengan perhitungan yang telah ada sesuai dengan data. Menggunakan Notepad++ untuk membuat skrip kode pemrograman dan juga menggunakan php, MySQL digunakan untuk mengelola informasi database serta menggunakan xampp, Framework Bootstrap dan JQuery sebagai library CSS. Metode perhitungan dilakukan dengan *Certainly Factor* dimana *Certainly Factor* akan mendefinisikan ukuran kepastian terhadap fakta itu pasti atau tidak pasti.
3. Output : *Expert System* akan secara otomatis mengeluarkan hasil diagnosa setelah dilakukan sebuah tes pada web *Expert System* mendiagnosa tingkat depresi siswa menggunakan metode *Certainly Factor*. Dengan cara penggunaanya adalah siswa menjawab pertanyaan dari gejala-gejala depresi yang telah diinput ke sistem yang telah diberi nilai kepastian oleh

pakar dan selanjutnya juga akan dilakukan perhitungam untuk mendapatkan nilai kepastian dalam mendiagnosa tingkat depresi siswa. Setelah itu, kita bisa melihat hasil diagnosa.