

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dwi Rahma Putri et al. (2024) mengembangkan sistem identifikasi kesehatan mental bernama SIMAHIR untuk mahasiswa tingkat akhir di UPN Veteran Jawa Timur. Sistem ini menggabungkan metode *Forward Chaining* dan *Backward Chaining*, di mana *Forward Chaining* dipakai untuk mendeteksi gejala awal, lalu *Backward Chaining* menghitung persentase kecocokan penyakitnya. Penelitian ini melibatkan 105 responden dari 21 program studi dan mampu mendiagnosis 10 jenis gangguan mental. Bedanya, gejala yang dipakai bukan dari instrumen baku, melainkan hasil wawancara dengan psikolog. Penelitian ini mirip dengan penelitian sekarang dari sisi objek (mahasiswa tingkat akhir) dan penggunaan sistem pakar, tapi berbeda pada sumber acuan gejalanya karena memakai data buatan sendiri.

Penelitian lain yang relevan adalah milik Armalivia et al. (2024), yang juga memakai *Forward Chaining* untuk mendeteksi gangguan mental mahasiswa tingkat akhir di Universitas Almarisah Madani. Bedanya, mereka sudah menggunakan kuesioner DASS-21 sebagai dasar penentuan gejala dan aturan sistem, dengan fokus pada tiga masalah yaitu depresi, kecemasan, dan stres. Hasil pengujian blackbox menunjukkan sistem ini valid dan mampu mengenali gejala awal dengan akurat. Penelitian ini menegaskan bahwa *Forward Chaining* bisa digabungkan langsung dengan instrumen psikometrik baku seperti DASS-21, sehingga basis gejala sistem jadi lebih kuat dibanding gejala buatan sendiri.

Penggunaan instrumen DASS juga terlihat pada penelitian Ain et al. (2025) namun cakupannya lebih sempit karena hanya fokus pada deteksi tingkat stres. Data yang dipakai berasal dari 14 pernyataan subskala stres pada DASS-42 dengan acuan psikolog klinis RSUD Jombang. Cara kerjanya sederhana: skor jawaban pengguna dijumlahkan, lalu dipetakan ke lima tingkatan mulai dari normal sampai sangat berat, kemudian sistem menampilkan saran penanganan. Hasil pengujian menunjukkan kalkulasi sistem cocok seratus persen dengan hitungan manual, dan pengujian blackbox juga valid. Penelitian ini membuktikan aturan logika sederhana pada *Forward Chaining* cukup efektif kalau berbasis ambang skor.

Berbeda dari ketiga penelitian sebelumnya, Chairani & Rosdiana (2018) memilih jalur perangkat keras dengan membuat alat ukur stres berbasis *Fuzzy Logic* Sugeno. Alat ini tidak memakai kuesioner mandiri, melainkan membaca sinyal fisiologis pengguna langsung lewat tiga sensor, yaitu detak jantung, suhu tubuh (DS18B20), dan Galvanic Skin Response. Data dikirim lewat modul NodeMcU ESP8266 dan ditampilkan di layar LCD serta ponsel Android. Hasil pengukuran alat ini dibandingkan dengan skor DASS-42 dan tingkat kecocokannya rata-rata 56%. Penelitian ini menarik sebagai pembandingan karena menunjukkan pendekatan objektif berbasis respons fisik tubuh bisa dikontraskan dengan pendekatan subjektif berbasis kuesioner.

Haura et al. (2026) juga memanfaatkan logika fuzzy, tepatnya varian *Fuzzy Logic* Mamdani, untuk deteksi dini kesehatan mental. Seluruh 42 butir instrumen DASS-42 diadaptasi untuk tiga variabel utama, masing-masing dengan lima level keparahan. Perhitungannya memakai fungsi keanggotaan segitiga dan trapesium, lalu keputusan akhir diperoleh lewat defuzzifikasi centroid. Metode ini terbukti efektif mengolah data psikologis yang sifatnya tidak pasti, dan hasil keputusan sistem sangat mendekati hasil analisis pakar sungguhan. Ini menunjukkan keunggulan logika fuzzy, yaitu derajat gejala bisa dipetakan lebih fleksibel dibanding aturan baku pada metode chaining biasa.

Sebagai pelengkap, Nisa (2026) menawarkan pendekatan komputasional murni lewat metode *Ordinal Logistic Regression* untuk skrining stres berbasis DASS-21 pada 367 mahasiswa. Penelitian ini tidak merancang sistem pakar, melainkan memodelkan lima tingkatan keparahan gejala secara ordinal memakai pendekatan cumulative logit, tanpa perlu menetapkan jarak tetap antar kategori tingkatan. Nilai quadratic weighted Cohen kappa yang didapat cukup memuaskan, yaitu 0,850 untuk depresi, 0,883 untuk kecemasan, dan 0,905 untuk stres. Meski akurasi klasifikasi nominalnya berada di level sedang, penelitian ini menunjukkan bahwa pendekatan statistik murni tanpa aturan pakar tetap mampu menjaga hierarki tingkatan keparahan gejala.

Tabel 2. 1 Penelitian Terdahulu

Kategori	(Dwi Rahma Putri et al., 2024)	(Armalivia et al., 2024)	(Ain et al., 2025)	(Chairani & Rosdiana, 2018)	(Haura et al., 2026)	(Nisa, 2026)
Masalah	Belum ada sistem diagnosis kesehatan mental mahasiswa tingkat akhir yang menggabungkan dua arah penalaran sekaligus	Deteksi dini gangguan kesehatan mental mahasiswa tingkat akhir masih jarang dilakukan secara mandiri dan mudah diakses	Dibutuhkan alat bantu deteksi tingkat stres yang cepat berbasis pengetahuan psikolog klinis	Diperlukan instrumentasi objektif berbasis sinyal fisiologis untuk mengukur tingkat stres	Rendahnya kesadaran & akses masyarakat terhadap deteksi dini kesehatan mental yang mudah dan mandiri	Model klasifikasi tingkat stres konvensional mengabaikan sifat ordinal dari kategori keparahan DASS
Metode	<i>Forward Chaining & Backward Chaining</i>	<i>Forward Chaining</i>	<i>Forward Chaining</i>	<i>Fuzzy Logic</i> Sugeno + sensor (GSR, suhu, detak jantung)	<i>Fuzzy Logic</i> Mamdani (fuzzifikasi segitiga/trapesium, defuzzifikasi centroid)	<i>Ordinal Logistic Regression</i> (cumulative logit/ordered logit)

Kategori	(Dwi Rahma Putri et al., 2024)	(Armalivia et al., 2024)	(Ain et al., 2025)	(Chairani & Rosdiana, 2018)	(Haura et al., 2026)	(Nisa, 2026)
Instrumen	Kuesioner gejala mandiri hasil wawancara psikolog (bukan DASS); 46 gejala & 10 jenis gangguan	DASS-21 (3 subskala: Depresi, Kecemasan, Stres; 21 item)	DASS-42 (subskala Stres, 14 item)	DASS-42 (sebagai pembanding hasil alat)	DASS-42 (3 subskala: Depresi, Kecemasan, Stres; 42 item)	DASS-21 (3 subskala; 21 item; 367 responden mahasiswa)
Platform	Website SIMAHIR (Next.js & Tailwind CSS)	Website (form konsultasi & cetak hasil)	Website (HTML, CSS, JavaScript)	Perangkat keras (Arduino Uno, NodeMcu ESP8266) + Android/LCD	Sistem pakar berbasis rule base	Model statistik/komputasional (bukan sistem pakar berbasis rule)
Hasil	Sistem mampu mendiagnosa 10 jenis gangguan mental berdasarkan 105	Sistem berhasil mengidentifikasi gejala awal depresi, kecemasan, dan stres secara	Perhitungan sistem sesuai perhitungan manual; lima tingkat	Tingkat kesesuaian rata-rata alat dengan DASS-42 sebesar 56%	Output crisp menghasilkan kategori tingkat mental (contoh Z=40 → Sedang),	Kappa kuadratik tinggi (0,850-0,905); akurasi tertinggi pada kecemasan 75,5% dengan MAE 0,33

Kategori	(Dwi Rahma Putri et al., 2024)	(Armalivia et al., 2024)	(Ain et al., 2025)	(Chairani & Rosdiana, 2018)	(Haura et al., 2026)	(Nisa, 2026)
	data uji, sesuai studi literatur	akurat; blackbox testing valid	stres (Normal-Sangat Berat); blackbox testing valid		mendekati penilaian pakar	

Berdasarkan penelitian sebelumnya pada Tabel 2.1, dapat diketahui bahwa penelitian-penelitian tersebut telah membahas topik yang hampir sama dengan penelitian ini. Meskipun demikian, masih terdapat beberapa hal yang dapat dikembangkan, seperti penggunaan metode yang masih terbatas pada satu metode, belum memanfaatkan instrumen psikologi yang baku, hanya berfokus pada satu aspek kesehatan mental, serta belum memberikan rekomendasi tindak lanjut berdasarkan tingkat keparahan kondisi pengguna. Oleh karena itu, penelitian ini mengembangkan sistem deteksi dini kesehatan mental mahasiswa berbasis *Progressive Web App* (PWA) menggunakan instrumen DASS-42 dan menggabungkan metode *Logistic Regression*, *Forward Chaining*, serta *Fuzzy Logic*. Kombinasi metode tersebut bertujuan agar hasil klasifikasi lebih terukur, dapat menyimpulkan kondisi kesehatan mental secara keseluruhan, serta memberikan rekomendasi tindak lanjut yang sesuai dengan tingkat keparahan kondisi pengguna.

2.2 Landasan Teori

2.2.1 Kesehatan Mental

Kesehatan mental merupakan suatu kondisi yang mencakup kesejahteraan emosional, psikologis, dan sosial pada diri seseorang. *World Health Organization* (WHO) mendefinisikan kesehatan mental sebagai suatu keadaan di mana setiap individu mampu menyadari potensi dirinya, mampu mengatasi tekanan hidup sehari-hari dengan baik, mampu bekerja secara produktif, serta mampu memberikan kontribusi bagi lingkungan dan komunitasnya (Negara, 2024). Pada mahasiswa, kondisi kesehatan mental banyak dipengaruhi oleh beban akademik, kualitas tidur, tingkat kelelahan mental, serta persepsi dan kenyamanan mereka dalam mengakses layanan konseling maupun psikolog di lingkungan kampus (Widyawati et al., 2025). Penelitian tersebut menunjukkan bahwa sebagian besar mahasiswa mengalami stres akademik yang tinggi dan kecemasan tanpa penyebab yang jelas. Jika tidak ditangani dengan baik, kondisi tersebut dapat mengganggu konsentrasi, emosi, dan produktivitas belajar mahasiswa (Widyawati et al., 2025). Penelitian ini menjadi salah satu alasan pentingnya melihat kondisi psikologis mahasiswa secara menyeluruh, sekaligus menjadi dasar dikembangkannya sistem skrining atau deteksi dini kesehatan mental berbasis teknologi, agar kondisi stress,

kecemasan dan depresi pada mahasiswa dapat dikenali lebih cepat dan sistematis.

2.2.2 *Skrining Awal Kesehatan Mental*

Skrining atau penapisan awal merupakan proses penilaian awal yang bertujuan untuk mengidentifikasi kemungkinan adanya risiko gangguan psikologis sebelum dilakukan pemeriksaan lebih lanjut oleh tenaga profesional. Berbeda dengan diagnosis klinis yang memberikan kepastian terhadap suatu kondisi, skrining hanya digunakan sebagai tahap awal untuk mengetahui apakah seseorang memerlukan pemeriksaan lanjutan. Di lingkungan perguruan tinggi, skrining kesehatan mental menjadi salah satu upaya deteksi dini untuk mengenali gejala stres, kecemasan, dan depresi pada mahasiswa sebelum berkembang menjadi masalah yang lebih serius serta memengaruhi aktivitas akademik (Setyanto, 2023).

Skrining kesehatan mental umumnya dilakukan menggunakan kuesioner *self-report*, yaitu instrumen yang diisi sendiri oleh responden berdasarkan kondisi yang dirasakan dalam periode tertentu. Pendekatan ini banyak digunakan karena mudah dilakukan, dapat diakses secara mandiri oleh mahasiswa, dan tidak memerlukan keterlibatan tenaga profesional pada tahap awal. Hasil skrining kemudian digunakan untuk mengetahui tingkat keparahan gejala dan menentukan tindak lanjut yang diperlukan, tanpa menggantikan diagnosis resmi dari psikolog atau psikiater.

2.2.3 *Depression Anxiety Stress Scales (DASS-42)*

Menurut Ardlina et al. (2022) Depression anxiety stress scale (DASS-42) merupakan tiga set skala penilaian diri untuk mengukur emosi negatif berupa depresi, kecemasan dan stres yang dirasakan dalam satu minggu terakhir. Selain itu, menurut Andriana & Prihantini (2021) Tingkat stres bisa dinilai ringan, sedang, maupun berat diukur dengan menggunakan *Depression Anxiety Stress Scale 42* (DASS 42) dari (LOVIBOND & LOVIBOND, 2022) yang terdiri dari 42 pertanyaan.

Lebih lanjut, Andriana & Prihantini (2021) menjelaskan DASS-42 adalah skala subjektif untuk mengukur status emosional negatif dari depresi, kecemasan, dan stres, dengan kategori tingkat stres berupa normal, ringan, sedang, berat, dan sangat berat. DASS-42 terdiri dari 42 item yang mencakup 3 *subvariabel*, yaitu fisik, emosi/psikologis, dan perilaku.

Setiap gangguan kondisi kesehatan mental memiliki beberapa gejala yang berbeda yang dapat dilihat pada tabel 2.2 yang berisi 42 gejala terkait dengan jenis penyakit kesehatan mental.

Tabel 2. 2 Tes DASS-42

No	Kode	Pertanyaan	Kategori
1	G01	Saya merasa bahwa diri saya menjadi marah karena hal-hal sepele.	Stres
2	G02	Saya merasa bibir saya sering kering.	Kecemasan
3	G03	Saya sama sekali tidak dapat merasakan perasaan positif.	Depresi
4	G04	Saya mengalami kesulitan bernapas (misalnya: seringkali terengah-engah atau tidak dapat bernapas padahal tidak melakukan aktivitas fisik sebelumnya).	Kecemasan
5	G05	Saya sepertinya tidak kuat lagi untuk melakukan suatu kegiatan.	Depresi
6	G06	Saya cenderung bereaksi berlebihan terhadap suatu situasi.	Stres
7	G07	Saya merasa goyah (misalnya, kaki terasa mau 'copot').	Kecemasan
8	G08	Saya merasa sulit untuk bersantai.	Stres
9	G09	Saya menemukan diri saya berada dalam situasi yang membuat saya merasa sangat cemas dan saya akan merasa sangat lega jika semua ini berakhir.	Kecemasan
10	G10	Saya merasa tidak ada hal yang dapat diharapkan di masa depan.	Depresi
11	G11	Saya menemukan diri saya mudah merasa kesal.	Stres
12	G12	Saya merasa telah menghabiskan banyak energi untuk merasa cemas.	Stres
13	G13	Saya merasa sedih dan tertekan.	Depresi
14	G14	Saya menemukan diri saya menjadi tidak sabar ketika mengalami penundaan (misalnya: kemacetan lalu lintas, menunggu sesuatu).	Stres
15	G15	Saya merasa lemas seperti mau pingsan.	Kecemasan
16	G16	Saya merasa saya kehilangan minat akan segala hal.	Depresi
17	G17	Saya merasa bahwa saya tidak berharga sebagai seorang manusia.	Depresi

No	Kode	Pertanyaan	Kategori
18	G18	Saya merasa bahwa saya mudah tersinggung.	Stres
19	G19	Saya berkeringat secara berlebihan (misalnya: tangan berkeringat), padahal temperatur tidak panas atau tidak melakukan aktivitas fisik sebelumnya.	Kecemasan
20	G20	Saya merasa takut tanpa alasan yang jelas.	Kecemasan
21	G21	Saya merasa bahwa hidup tidak bermanfaat.	Depresi
22	G22	Saya merasa sulit untuk beristirahat.	Stres
23	G23	Saya mengalami kesulitan dalam menelan.	Kecemasan
24	G24	Saya tidak dapat merasakan kenikmatan dari berbagai hal yang saya lakukan.	Depresi
25	G25	Saya menyadari kegiatan jantung, walaupun saya tidak sehabis melakukan aktivitas fisik (misalnya: merasa detak jantung meningkat atau melemah).	Kecemasan
26	G26	Saya merasa putus asa dan sedih.	Depresi
27	G27	Saya merasa bahwa saya sangat mudah marah.	Stres
28	G28	Saya merasa saya hampir panik.	Kecemasan
29	G29	Saya merasa sulit untuk tenang setelah sesuatu membuat saya kesal.	Stres
30	G30	Saya takut bahwa saya akan 'terhambat' oleh tugas-tugas sepele yang tidak biasa saya lakukan.	Kecemasan
31	G31	Saya tidak merasa antusias dalam hal apapun.	Depresi
32	G32	Saya sulit untuk sabar dalam menghadapi gangguan terhadap hal yang sedang saya lakukan.	Stres
33	G33	Saya sedang merasa gelisah.	Stres
34	G34	Saya merasa bahwa saya tidak berharga.	Depresi
35	G35	Saya tidak dapat memaklumi hal apapun yang menghalangi saya untuk menyelesaikan hal yang sedang saya lakukan.	Stres
36	G36	Saya merasa sangat ketakutan.	Kecemasan
37	G37	Saya melihat tidak ada harapan untuk masa depan.	Depresi
38	G38	Saya merasa bahwa hidup tidak berarti.	Depresi
39	G39	Saya menemukan diri saya mudah gelisah.	Stres

No	Kode	Pertanyaan	Kategori
40	G40	Saya merasa khawatir dengan situasi dimana saya mungkin menjadi panik dan mempermalukan diri sendiri.	Kecemasan
41	G41	Saya merasa gemetar (misalnya: pada tangan).	Kecemasan
42	G42	Saya merasa sulit untuk meningkatkan inisiatif dalam melakukan sesuatu.	Depresi

Tabel 2. 3 Klasifikasi Berdasarkan Kategori

Kategori	Kode Gejala
Stres	G01, G06, G08, G11, G12, G14, G18, G22, G27, G29, G32, G33, G35, G39
Kecemasan	G02, G04, G07, G09, G15, G19, G20, G23, G25, G28, G30, G36, G40, G41
Depresi	G03, G05, G10, G13, G16, G17, G21, G24, G26, G31, G34, G37, G38, G42

2.2.4 Skoring dan Interpretasi DASS-42

Penilaian atau skoring pada instrumen DASS-42 dilakukan dengan memberikan nilai pada setiap jawaban responden menggunakan skala 0 sampai 3, sebagaimana ditunjukkan pada Tabel 2.2.

Tabel 2. 4 Skala Tes DASS-42

Skor	Keterangan
0	Tidak sesuai dengan saya sama sekali, atau tidak pernah.
1	Sesuai dengan saya sampai tingkat tertentu, atau kadang-kadang.
2	Sesuai dengan saya sampai batas yang dapat dipertimbangkan, atau lumayan sering.
3	Sangat sesuai dengan saya, atau sering sekali.

Bobot pada setiap butir pernyataan tersebut kemudian dijumlahkan berdasarkan kelompok dimensinya, yaitu stres, kecemasan, atau depresi, sesuai dengan pengelompokan kode gejala pada Tabel 2.4. Total skor pada masing-masing dimensi inilah yang menjadi dasar penentuan tingkat keparahan kondisi responden.

Tingkat keparahan pada setiap dimensi selanjutnya dikategorikan berdasarkan rentang skor tertentu, sebagaimana disajikan pada Tabel 2.5, yang

mengelompokkan kondisi responden ke dalam lima kategori tingkatan, yaitu normal, ringan, sedang, parah, dan sangat parah.

Tabel 2. 5 Skor Setiap Tingkatan Kode

Aspek	Kode	Kategori Tingkat	Total Skor
Stres	T1	Normal	0 – 14
	T2	Ringan	15 – 18
	T3	Sedang	19 – 25
	T4	Parah	26 – 33
	T5	Sangat Parah	≥ 34
Kecemasan	T6	Normal	0 – 7
	T7	Ringan	8 – 9
	T8	Sedang	10 – 14
	T9	Parah	15 – 19
	T10	Sangat Parah	≥ 20
Depresi	T11	Normal	0 – 9
	T12	Ringan	10 – 13
	T13	Sedang	14 – 20
	T14	Parah	21 – 27
	T15	Sangat Parah	≥ 28

2.2.5 Sistem Pakar

Menurut Muafi et al. (2020) sistem pakar merupakan aplikasi komputer yang dirancang untuk meniru proses penalaran seorang ahli dalam menyelesaikan masalah tertentu dan menghasilkan suatu keputusan atau kesimpulan. Pengetahuan yang dimiliki oleh pakar disimpan ke dalam basis pengetahuan (*knowledge base*) sehingga dapat digunakan oleh sistem untuk membantu proses pemecahan masalah. Dasar dari sistem pakar adalah memindahkan pengetahuan seorang ahli ke dalam komputer, kemudian menggunakannya sebagai dasar dalam mengambil keputusan dan menarik kesimpulan.

Sementara itu, menurut Sondang (2023) menjelaskan bahwa sistem pakar merupakan program kecerdasan buatan yang menggabungkan basis pengetahuan (*knowledge base*) dengan sistem inferensi. Sistem ini dikembangkan untuk meniru

kemampuan seorang ahli pada bidang tertentu. Dalam penerapannya, sistem pakar berperan sebagai konsultan atau pemberi saran yang cerdas berdasarkan kumpulan pengetahuan yang diperoleh dari beberapa ahli. Dengan adanya sistem pakar, pengguna yang bukan ahli tetap dapat memperoleh bantuan dalam menyelesaikan masalah yang dihadapi.

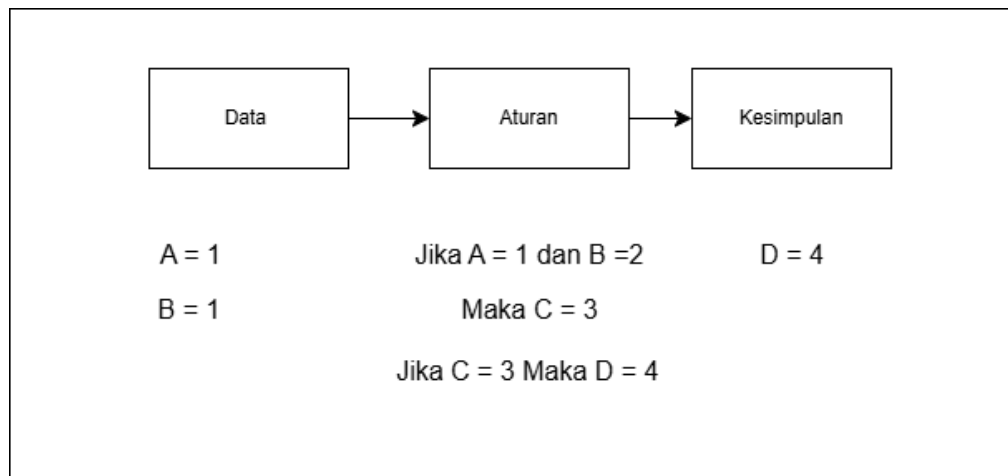
2.2.6 *Forward Chaining*

Forward Chaining merupakan metode penelusuran pengetahuan yang dimulai dari kumpulan fakta untuk memperoleh suatu kesimpulan berdasarkan fakta tersebut. Metode ini bekerja dengan mencocokkan fakta yang tersedia dengan aturan yang telah ditentukan di dalam basis pengetahuan hingga menghasilkan sebuah kesimpulan.

Sistem yang menggunakan metode *Forward Chaining* memiliki karakteristik sebagai berikut:

Sistem diaktifkan oleh satu atau lebih kondisi awal.

1. Sistem diawali dengan satu atau lebih kondisi awal.
2. Untuk setiap kondisi awal, sistem akan mencari aturan pada basis pengetahuan yang sesuai dengan bagian IF dari aturan tersebut.
3. Jika aturan terpenuhi, sistem akan menghasilkan kondisi baru pada bagian THEN.
4. Kondisi baru tersebut ditambahkan ke dalam kumpulan fakta yang sudah ada, kemudian diproses kembali untuk mencari aturan lain yang sesuai. Proses ini terus berlanjut hingga diperoleh kesimpulan yang diinginkan atau tidak ada lagi kondisi baru yang dapat dihasilkan.



Gambar 2. 1 *Forward Chaining*

Tahap selanjutnya adalah menyusun aturan (*rule*) untuk penerapan sistem pakar menggunakan metode *Forward Chaining*. Aturan tersebut digunakan untuk memetakan proses deteksi kesehatan mental pada mahasiswa. Perhitungan dilakukan menggunakan rumus yang digunakan untuk menghitung total skor berdasarkan jawaban yang diberikan pengguna pada setiap gejala.

Berikut perancangan rule metode *Forward Chaining* untuk setiap tingkatan

Tabel 2. 6 *Rule Forward Chaining*

Rule (R)	Kondisi (IF)	Hasil (THEN)
1	$0 \leq \sum \text{Skor Gejala} \leq 14$	T1
2	$15 \leq \sum \text{Skor Gejala} \leq 18$	T2
3	$19 \leq \sum \text{Skor Gejala} \leq 25$	T3
4	$26 \leq \sum \text{Skor Gejala} \leq 33$	T4
5	$\sum \text{Skor Gejala} \geq 34$	T5

Keterangan: T1–T5 merupakan tingkat stres berdasarkan total skor gejala ($\sum \text{Skor Gejala}$).

Rule (R)	Kondisi (IF)	Hasil (THEN)
6	$0 \leq \sum \text{Skor Gejala} \leq 7$	T6
7	$8 \leq \sum \text{Skor Gejala} \leq 9$	T7
8	$10 \leq \sum \text{Skor Gejala} \leq 14$	T8

Rule (R)	Kondisi (IF)	Hasil (THEN)
9	$15 \leq \sum \text{Skor Gejala} \leq 19$	T9
10	$\sum \text{Skor Gejala} \geq 20$	T10

Keterangan: T6–T10 merupakan tingkat kecemasan berdasarkan total skor gejala ($\sum \text{Skor Gejala}$).

Rule (R)	Kondisi (IF)	Hasil (THEN)
11	$0 \leq \sum \text{Skor Gejala} \leq 9$	T11
12	$10 \leq \sum \text{Skor Gejala} \leq 13$	T12
13	$14 \leq \sum \text{Skor Gejala} \leq 20$	T13
14	$21 \leq \sum \text{Skor Gejala} \leq 27$	T14
15	$\sum \text{Skor Gejala} \geq 28$	T15

Keterangan: T6–T10 pada tabel ini merupakan tingkat depresi berdasarkan total skor gejala ($\sum \text{Skor Gejala}$), dengan skala penilaian terpisah dari kategori kecemasan.

2.2.7 Machine Learning

Machine learning merupakan cabang ilmu yang memungkinkan sistem komputer belajar langsung dari data. Sistem ini mampu melakukan klasifikasi maupun prediksi secara otomatis tanpa perlu instruksi pemrograman yang kaku untuk setiap kondisinya. Berdasarkan cara kerjanya, metode ini terbagi menjadi dua kategori. Kategori pertama adalah *supervised learning*, yang memanfaatkan data berlabel untuk melatih model klasifikasi. Sebaliknya, *unsupervised learning* bekerja secara mandiri untuk menemukan pola tersembunyi pada sekumpulan data tanpa label (Nurhalizah et al., 2024).

Untuk keperluan klasifikasi, *supervised learning* memang lebih sering digunakan. Algoritma seperti *Logistic Regression*, *K-Nearest Neighbor*, *Naive Bayes*, dan *Support Vector Machine* menjadi pilihan yang lazim karena tingkat akurasi yang sudah terbukti (Nurhalizah et al., 2024). Pemanfaatannya di lapangan pun sangat luas. (Retnoningsih & Pramudita, 2020) mencontohkan bahwa pendekatan ini sukses diterapkan pada banyak kasus, mulai dari sebatas identifikasi spesies bunga hingga pengelompokan data kependudukan.

2.2.8 Logistic Regression

Sebagai bagian dari algoritma klasifikasi, *Logistic Regression* berfungsi

memprediksi probabilitas terjadinya suatu peristiwa. Pemodelan ini bekerja dengan memetakan hubungan antara fitur sebagai variabel independen dengan variabel dependen *biner*. Hasil keluarannya selalu berupa nilai probabilitas yang berada pada rentang 0 hingga 1 (Pratama et al., 2023).

Inti dari perhitungan algoritma ini terletak pada penggunaan fungsi logistik atau fungsi sigmoid. Fungsi tersebut bertindak menahan nilai luaran prediksi agar tidak melewati batas rasio. Dengan begitu, angka yang muncul bisa langsung ditafsirkan sebagai peluang masuknya suatu data ke kelas tertentu (Fahmuddin et al., 2023). Algoritma ini tergolong sangat populer karena kemampuannya memberikan hasil klasifikasi yang konsisten (Sutarman et al., 2024) Di bidang kesehatan, metode ini kerap diandalkan untuk mendiagnosis penyakit berdasarkan kumpulan gejala dan rekam medis pasien.

2.2.9 Fuzzy Logic

Lotfi A. Zadeh pertama kali memperkenalkan konsep logika *Fuzzy* pada tahun 1965. Metode ini dirancang sebagai solusi untuk mengolah variabel yang sifatnya samar atau tidak tegas. Pendekatan ini sangat berbeda dengan logika Boolean konvensional. Jika logika Boolean hanya mengenali kondisi mutlak benar atau salah, logika *Fuzzy* memungkinkan sebuah variabel memiliki derajat keanggotaan (*membership degree*). Nilainya pun bersifat kontinu dengan rentang antara 0 hingga 1 (Ega Santosa & Ratna Indra Astutik, 2021).

Secara operasional, sistem penalaran *Fuzzy* melewati tiga tahapan utama. Proses selalu diawali dengan tahap fuzzifikasi. Data kemudian diteruskan ke tahap inferensi untuk diproses menggunakan basis aturan IF-THEN. Tahapan ini diakhiri dengan defuzzifikasi, yang berfungsi mengubah derajat keanggotaan tersebut kembali menjadi luaran numerik yang tegas (P et al., 2008). Kemampuan logika *Fuzzy* dalam memproses ketidakpastian dan menghasilkan luaran bertingkat (*gradual*) membuatnya sangat relevan. Karakteristik inilah yang menjadikannya ideal saat diimplementasikan pada sistem deteksi dan pemberian rekomendasi kesehatan mental mahasiswa (Nawindah & Lydiani, 2020).

2.2.10 Integrasi Metode (*Forward Chaining*, *Logistic Regression*, dan *Fuzzy Logic*)

Penelitian ini mengintegrasikan tiga metode, yaitu *Forward Chaining*, *Logistic Regression*, dan *Fuzzy Logic*, yang masing-masing memiliki peran yang saling melengkapi dalam proses identifikasi kondisi kesehatan mental mahasiswa. *Forward Chaining* berperan sebagai mekanisme sistem pakar berbasis aturan (rule-based) untuk menentukan tingkat depresi, kecemasan, dan stres berdasarkan skor hasil pengisian kuesioner DASS-42. *Logistic Regression* berperan sebagai model klasifikasi awal berbasis pembelajaran mesin (data-driven) yang digunakan untuk mengevaluasi tingkat konsistensi hasil identifikasi yang diperoleh dari sistem pakar. Selanjutnya, *Fuzzy Logic* berperan dalam menghasilkan rekomendasi tindak lanjut yang disesuaikan dengan tingkat kedaruratan kondisi kesehatan mental mahasiswa berdasarkan hasil identifikasi dari kedua metode sebelumnya. Dengan demikian, integrasi ketiga metode tersebut diharapkan dapat menghasilkan sistem identifikasi kesehatan mental yang lebih andal, karena hasil identifikasi berbasis aturan dapat divalidasi oleh pendekatan berbasis data, sementara keluaran akhir berupa rekomendasi dapat disesuaikan secara lebih fleksibel melalui penalaran fuzzy.

2.2.11 *Progresissive Web App*

Menurut Nurwanto (2019), *Progressive Web App* (PWA) merupakan teknologi pengembangan aplikasi berbasis web yang dirancang untuk mengatasi keterbatasan browser seluler dan aplikasi *native*. PWA memanfaatkan berbagai teknologi web modern, konsep desain, serta API sehingga aplikasi web dapat memberikan pengalaman penggunaan yang menyerupai aplikasi *native*. Selain itu, PWA dapat dijalankan langsung melalui browser tanpa perlu diunduh melalui *Play Store* atau *App Store*, dapat ditambahkan ke layar utama perangkat, serta tetap dapat digunakan meskipun koneksi internet terbatas.

2.2.12 *React.js*

Menurut Iswari & Nasution (2021) *React* adalah *open-source library* JavaScript deklaratif, efisien dan fleksibel untuk membangun antarmuka pengguna. *React* memungkinkan untuk membuat *user interface* yang kompleks dengan set

kode kecil yang terisolasi yang disebut "komponen". React JS ini digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dan pengembangan *mobile application*. React JS dikelola oleh *facebook*, *instagram*, komunitas pengembang dan korporasi. React berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur yang paling mencolok adalah JSX, Komponen Stateful, Model Objek Dokumen Virtual.

2.2.13 Node.js

Menurut FAJRIN (2017) Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi *web*. Node.JS dapat juga disebut sebagai runtime *environment*. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model *event driven* (basis event) dan *asynchronous I/O*. Tidak seperti kebanyakan bahasa *JavaScript* yang dijalankan pada *web browser*, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari *V8 Engine* buatan *Google* dan beberapa modul bawaan yang terintegrasi seperti modul *http*, modul *filesystem*, modul *security* dan beberapa modul penting lainnya.

Menurut (Nugroho & Kartadie, 2016) NodeJS adalah sebuah platform untuk membangun *real-time application*. NodeJS dapat menangani *event input output server*, dengan kata lain NodeJS dapat memungkinkan para *developer Javascripts* untuk membuat *event driven servers* dalam JavaScript.

2.2.14 MongoDB

Menurut (Putra & Rahmayeni, 2016) *MongoDB* merupakan sebuah sistem basis data yang berbasis dokumen (*Document Oriented Database*) dan termasuk sistem basis data yang menganut paham *NoSQL*. *NoSQL* singkatan dari *Not Only SQL*, artinya sebuah sistem basis data tidak hanya harus menggunakan perintah *SQL* untuk melakukan proses manipulasi data.

MongoDB merupakan sistem basis data yang menggunakan konsep *key-value*, artinya setiap dokumen dalam *mongodb* pasti memiliki *key*. Hal ini berbeda dalam RDBMS yang bisa tidak menggunakan *primary key* ketika membuat sebuah tabel. Sehingga walaupun kita membuat sebuah dokumen tanpa menggunakan *primary key*, tapi secara otomatis *mongodb* memberinya sebuah *key*.

Berikut adalah kelebihan *MongoDB*:

1. Performa yang ditawarkan *MongoDB* lebih cepat disebabkan oleh memcached dan format dokumennya yang berbentuk seperti JSON
2. Kita tidak perlu membuat struktur tabel, karena *MongoDB* akan otomatis membuatkan struktur tabelnya pada saat proses *insert* (fleksibel skema) .

Berikut adalah kekurangan *MongoDB*:

1. Belum banyak hosting yang *support*.
2. Fleksibilitas dalam *query* (sebagai contoh tidak adanya *JOIN*).
3. Selain itu, karena bervariasinya produk dan format penyimpanan, berpindah antar satu produk database ke produk noSQL lainnya perlu waktu untuk (Maharani, 2022) .

2.2.15 *BlackBox Testing*

Blackbox testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak tanpa memperhatikan struktur kode program di dalamnya, dan bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan performansi, maupun kesalahan inisialisasi dan terminasi (Wijaya & Astuti, 2021). Pengujian *black box* dilakukan hanya untuk mendapatkan hasil eksekusi melalui pengujian data serta memverifikasi kesesuaian fungsionalitas perangkat lunak terhadap spesifikasi yang telah ditentukan (Pramudita, 2020). Salah satu teknik yang umum digunakan dalam *black box testing* adalah *equivalence partitioning*, yaitu teknik pengujian yang membagi domain data masukan program ke dalam kelas-kelas data, sehingga kasus uji dapat dibentuk secara representatif berdasarkan kondisi valid maupun tidak valid tanpa harus menguji seluruh kemungkinan data satu per satu (Ilham et al., 2023).

2.2.16 *WhiteBox Testing*

White box testing merupakan metode pengujian yang menggunakan struktur kendali dan desain prosedur untuk memeriksa detail kode sumber, alur algoritma, dan jalur (*path*) suatu program, sehingga mampu memberikan gambaran yang lebih rinci mengenai jalannya sistem dibandingkan pengujian pada *level* fungsional saja .Pengujian ini umumnya dilakukan setelah tahap implementasi, dengan pertimbangan bahwa sebuah program tidak akan berjalan dengan benar apabila

logika atau alur algoritmanya keliru, meskipun secara fungsional tampak berjalan (Eriana, 2020). Dibandingkan dengan *black box testing* yang berfokus pada fungsionalitas dan keluaran perangkat lunak dari sudut pandang pengguna, *white box testing* menitikberatkan pengujian pada tingkat alur kode program serta kesesuaian *input* dan *output* terhadap spesifikasi teknis yang dibutuhkan (Jamaliyah, 2022). Kedua metode pengujian ini memiliki keunggulan yang saling melengkapi, sehingga kombinasi keduanya dinilai lebih efektif untuk memastikan kualitas perangkat lunak secara menyeluruh, baik dari sisi fungsionalitas maupun struktur internal kode (Wintana et al., 2022).

2.2.17 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) merupakan tahap pengujian sistem yang dilakukan langsung oleh pengguna akhir (*end user*) untuk memastikan bahwa sistem yang dikembangkan telah berjalan sesuai dengan kebutuhan dan dapat diterima oleh penggunanya, dengan melibatkan responden yang menjawab kuesioner penilaian menggunakan skala *Likert* (Aliyah Aliyah et al., 2024). Berbeda dengan pengujian pada tahap pengembangan yang umumnya dilakukan oleh tim pengembang, UAT bertujuan memperoleh bukti langsung bahwa sistem telah berjalan sesuai kebutuhan dari sudut pandang pengguna sebelum sistem digunakan secara operasional (Hady et al., 2020). Dalam praktiknya, pengujian sebuah sistem informasi dapat menggabungkan pendekatan *black box testing* dan *white box testing* secara bersamaan pada objek penelitian yang sama, sebagaimana ditunjukkan pada penelitian pengujian sistem informasi parkir berbasis web yang menerapkan kedua metode tersebut untuk memastikan sistem berhasil dan *valid*, baik dari sisi fungsionalitas maupun struktur kode program yang mendasarinya (Praniffa et al., 2023).