

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang membahas dan merancang sebuah sistem pakar dengan studi kasus penyakit stress telah dilakukan sebelumnya oleh berbagai pihak dengan beberapa gejala yang berbeda-beda hingga metode yang berbeda. Selain itu, penelitian tersebut menghasilkan berbagai macam hasil akurasi. Berikut merupakan beberapa penelitian yang telah dilakukan sebelumnya diantaranya sebagai berikut.

1. Perancangan Sistem Pakar untuk Mendiagnosa Tingkat Stress Belajar pada Siswa SMA dengan Menggunakan Metode *Forward Chaining*”. Penelitian yang telah dilakukan oleh Cut Rizki Putri Amalia, dan Mahyuddin pada tahun 2022. Penelitian ini membahas tentang penerapan metode Forward Chaining dalam studi kasus penyakit stress. Pada penelitian ini memiliki tujuan yaitu perancangan sistem pakar untuk mendiagnosa tingkat stress belajar pada siswa SMA dengan menggunakan metode *Forward Chaining* berbasis web. Alur proses sistem pakar ini dimulai dengan tahap penginputan data oleh admin. Admin akan menginputkan data, lalu siswa akan menginput datanya dan memilih gejala-gejala yang sering dialami, setelah semua data berhasil dimasukkan maka akan diproses oleh sistem. Output dari penelitian ini berupa file hasil rekap yang akan mengeluarkan hasil diagnosa dan sistem memberikan rekap atau laporan keseluruhan. Penelitian ini memberikan kontribusi dalam penggunaan sistem pakar untuk membantu mendiagnosa masalah psikologis di kalangan pelajar dengan teknologi berbasis web, meskipun masih ada ruang untuk perbaikan lebih lanjut, terutama dalam meningkatkan akurasi diagnosa dengan metode tambahan. Penelitian Putri dan Mahyuddin (2022) menerapkan Forward Chaining untuk mendiagnosis tingkat stres belajar pada siswa SMA melalui sistem berbasis web. Penelitian ini menunjukkan bahwa FC dapat digunakan untuk mengklasifikasi gejala stres berdasarkan aturan pakar. Namun, akurasi diagnosis sangat bergantung pada kualitas aturan dan tidak

melibatkan pembelajaran dari data, sehingga berpotensi menghasilkan tingkat akurasi yang rendah pada kasus-kasus baru. Keterbatasan inilah yang menjadi dasar penelitian saat ini untuk membandingkan kinerja FC dengan metode berbasis machine learning seperti Random Forest, sekaligus melihat apakah integrasi keduanya dapat meningkatkan performa sistem.

2. Analisa Diagnosa Penyakit Berdasarkan Riwayat Medis menggunakan Algoritma Random Forest Studi Kasus Rumah Sakit Padjongadg Ngalle Kabupaten Takalar [6] Penelitian ini membahas tentang penerapan algoritma Random Forest di Rumah Sakit Padjonga Dg Ngalle untuk menganalisis dan mediagnosis penyakit berdasarkan riwayat medis yang ada disana. Penelitian ini memberikan kesimpulan mengenai efektivitas model Random Forest dalam mendiagnosis penyakit berdasarkan riwayat medis dan hasil tes laboratorium. Penelitian ini menunjukkan bahwa Random Forest dalam menganalisis riwayat medis pasien menunjukkan potensi untuk menangani data kompleks, namun masih ada ruang untuk perbaikan dalam akurasi. Tingkat akurasi sebesar 48,50% menandakan bahwa perlu perbaikan lebih lanjut untuk meningkatkan kinerjanya. Peneliti menyarankan agar menggunakan algoritma lain atau bisa juga dengan membandingkan 2 algoritma atau lebih. Penelitian mengenai diagnosis penyakit menggunakan Random Forest di RS Padjonga Dg Ngalle menunjukkan bahwa RF mampu menangani data medis yang kompleks, namun tingkat akurasinya hanya 48,50%. Rendahnya akurasi ini mengindikasikan perlunya optimasi, misalnya melalui seleksi fitur atau teknik penyeimbangan data. Hal ini mendukung penelitian saat ini untuk menguji apakah penerapan RFE dapat memaksimalkan performa RF pada kasus deteksi stres dan apakah optimasi tersebut lebih unggul dibanding metode tunggal.
3. “Landslide susceptibility mapping using hybrid random forest with GeoDetector and RFE for factor optimization” [7]. Penelitian ini menyimpulkan bahwa penggunaan RFE dalam kombinasi dengan Random Forest dapat mengoptimalkan prediksi dalam pemetaan longsor.

Geodetectoe dan RFE dalam penelitian ini digunakan untuk memilih faktor paling berpengaruh dalam pemetaan longsor, dan hybrid pada penelitian ini dapat mengatasi masalah overfitting. Penelitian pemetaan longsor dengan pendekatan RF yang dioptimasi RFE dan GeoDetector menunjukkan bahwa kombinasi hybrid tersebut dapat mengurangi overfitting dan meningkatkan akurasi prediksi. Meskipun konteksnya berbeda (geospasial), mekanisme seleksi fitur RFE terbukti efektif untuk meningkatkan performa model berbasis pohon keputusan. Relevansinya bagi penelitian ini adalah memberikan dasar ilmiah bahwa RFE dapat meningkatkan kemampuan Random Forest dalam kasus lain, termasuk kasus deteksi stres.

4. Sistem Pakar Deteksi Dini Tingkat Stres Mahasiswa Terhadap Tugas Akhir Menggunakan Metode Forward Chaining dan Certainty Factor Berbasis Android” [8]. Penelitian ini membahas tentang perbedaan cara seseorang dalam merespon stres, karena adanya perbedaan besar mengenai cara merespons stres maka penulis membuat sebuah sistem pakar untuk deteksi dini tingkat stres pada mahasiswa akhir dengan mengimplementasikan metode Forward Chaining dan Certainty Factor berbasis Android dengan tujuan untuk mempermudah mahasiswa dalam melakukan deteksi dini tingkat stres terhadap dirinya ketika mereka sedang mengerjakan tugas akhir. Metode Forward Chaining pada sistem pakar ini sebagai klasifikasi gejala yang dimasukkan oleh pengguna sesuai dengan aturan yang sudah ditetapkan oleh pakar, kemudian aplikasi akan menghitung nilai keyakinan disetiap aturan dengan menggunakan metode Certainty Factor. Pada penelitian ini dapat ditarik kesimpulan bahwa kedua metode tersebut dapat diterapkan dengan baik pada aplikasi sistem pakar, dan berdasarkan hasil pengujian fungsionalitas menggunakan Black Box didapatkan bahwa semua fitur yang ada pada aplikasi dapat berjalan dengan baik, dan hasil pengujian usabilitas didapatkan skor sebesar 84% yang masuk dalam kategori dapat diterima. Penelitian lain yang menggabungkan Forward Chaining dan Certainty Factor untuk deteksi stres mahasiswa menunjukkan bahwa metode berbasis aturan dapat berjalan efektif dari sisi fungsionalitas

aplikasi, dengan skor usabilitas sebesar 84%. Namun, penelitian ini belum mengukur akurasi prediksi secara komprehensif dan tidak membandingkan metode alternatif. Hal ini memperkuat alasan penelitian ini untuk tidak hanya mengimplementasikan FC, tetapi juga menguji performanya terhadap RF dan RF+RFE.

5. Penelitian dengan judul “Sistem Identifikasi Masalah Kesehatan Mental Mahasiswa Tingkat Akhir Dengan Metode Forward Chaining dan Backward Chaining (Studi Kasus : UPN “Veteran” Jawa Timur)”. Penelitian yang telah dilakukan oleh Septiani Dwi Rahma Putri pada 2024. Penelitian ini membahas tentang penerapan Forward Chaining dan Backward Chaining dalam studi kasus kesehatan mental. Pada penelitian ini memiliki tujuan yaitu untuk mengembangkan sistem identifikasi masalah kesehatan mental, dimana metode forward chaining bertujuan untuk memulai tes kesehatan mental dengan memilih gejala yang dirasakan, dan memfasilitasi identifikasi awal kesehatan mental. Sedangkan metode backward chaining untuk menghasilkan persentase penyakit terpilih berdasarkan gejala yang terpenuhi. Alur proses sistem ini akan dimulai dengan pengguna melakukan konsultasi kepada sistem dengan memilih gejala yang sesuai, kemudian akan didapatkan hasil input gejala yang dirasakan oleh user. Data gejala yang telah dipilih oleh user akan diproses menggunakan forward chaining untuk menyimpulkan data penyakit berdasarkan dengan gejala, data tersebut lalu akan dikirimkan ke backward chaining untuk dicek dan dibandingkan jumlah fakta atau gejala yang terpenuhi dengan gejala penyakit yang terpilih sehingga menghasilkan persentase penyakit yang terpilih. Hasil dari backward chaining tersebut akan dikirimkan ke sistem untuk diproses. Penelitian ini memberikan kontribusi dalam penggunaan sistem pakar untuk mendiagnosis awal penyakit gangguan kesehatan mental pada mahasiswa tingkat akhir, namun masih ada ruang untuk perbaikan dan pengembangan sistem dengan menggunakan metode selain forward chaining dan backward chaining. Penelitian Putri (2024) menggabungkan Forward Chaining dan Backward Chaining untuk identifikasi gangguan

kesehatan mental mahasiswa. Sistem berhasil memberikan hasil diagnosis awal, namun pendekatan ini masih sepenuhnya berbasis aturan tanpa dukungan machine learning. Kekurangannya adalah tidak dapat belajar dari pola data dan tidak mampu menangani kasus baru yang tidak dicakup aturan pakar. Kondisi ini sangat relevan dengan penelitian saat ini yang ingin membandingkan metode rule-based murni dengan metode berbasis data.

6. Optimasi Klasifikasi Curah Hujan Menggunakan Support Vector Machine dan Recursive Feature Elimination [9]. Penelitian ini dirancang untuk menyelesaikan permasalahan klasifikasi curah hujan dengan beberapa kategori hujan, untuk menyelesaikan masalah tersebut peneliti menggunakan metode SVM (Support Vector Machine) dengan metode optimasi seleksi fitur menggunakan RFE (Recursive Feature Elimination). Dengan penggunaan optimasi RFE ini atribut yang dihasilkan berubah menjadi 3 dari total 10 atribut. Dan setelah dilakukan evaluasi dengan Confusion Matrix menghasilkan tingkat akurasi yang lebih tinggi jika menggunakan RFE sebanyak 2% , dengan total akurasi 79% dengan RFE dan 77% tanpa RFE. Penelitian ini menunjukkan bahwa penggunaan RFE akan lebih unggul dan dapat meningkatkan akurasi klasifikasi hujan. Penelitian optimasi klasifikasi curah hujan menggunakan SVM dan RFE menunjukkan bahwa RFE mampu meningkatkan akurasi model sebesar 2% dibanding tanpa seleksi fitur. Hasil ini menunjukkan bahwa jumlah fitur yang berlebihan dapat menurunkan kinerja model, dan RFE efektif dalam memilih fitur yang paling relevan. Relevansinya terhadap penelitian ini adalah memberikan bukti pendukung bahwa penerapan RFE pada Random Forest berpotensi meningkatkan akurasi klasifikasi tingkat stres.

Berdasarkan keenam penelitian terdahulu diatas, dapat disimpulkan bahwa masing-masing metode yaitu Forward Chaining, Random Forest, dan Recursive Feature Elimination memiliki kelebihan dan kekurangannya masing-masing. *Forward Chaining* unggul dalam penalaran berbasis aturan namun kurang adaptif pada data baru, sedangkan Random Forest lebih fleksibel namun memerlukan optimasi fitur agar mencapai kinerja yang maksimal. Penelitian

sebelumnya menunjukkan bahwa RFE mampu meningkatkan akurasi model. Namun belum ada penelitian yang secara khusus membandingkan kinerja *Forward Chaining*, *Random Forest*, dan RFE dalam konteks deteksi stres pada Gen Z, sehingga *research gap* inilah yang menjadi penelitian ini.

2.2 Landasan Teori

2.2.1 Generasi Z

Generasi Z atau kerap disebut dengan Gen Z merupakan kelompok individu yang lahir antara pertengahan 1990-an sampai 2010-an [2], merupakan kelompok yang tumbuh dan berkembang dalam konteks sosial, ekonomi, dan teknologi yang dinamis. Gen Z tumbuh dan berkembang di era kemajuan teknologi digital, dimana internet dan teknologi menjadi suatu kebutuhan dalam kehidupan sehari-hari. Berbeda dengan generasi sebelumnya, Gen Z lebih cepat beradaptasi dengan teknologi baru dan sangat bergantung pada perangkat digital baik dalam aspek pendidikan, komunikasi, hingga hiburan.

Gen Z merupakan generasi yang tumbuh di era kemajuan teknologi digital, dimana akses informasi melalui internet dan sosial media sudah menjadi bagian dari kehidupan sehari-hari. Gen Z tumbuh di tengah pesatnya perkembangan teknologi digital, di mana internet dan media sosial menjadi bagian keseharian mereka. Generasi ini sangat bergantung pada perangkat digital serta media sosial untuk berinteraksi, memperoleh informasi, dan bersosialisasi. Meskipun demikian, penggunaan media sosial yang tinggi di kalangan Gen Z dapat membawa dampak negatif, khususnya bagi kesehatan mental. Banyak dari mereka merasakan tekanan untuk menampilkan versi ideal diri di dunia maya, serta mencari validasi melalui like, komentar, dan pengikut di platform seperti Instagram, TikTok, dan Twitter. Akibatnya, hal ini dapat memicu masalah psikologis seperti kecemasan dan kurangnya kepercayaan diri, terutama ketika mereka merasa tidak mampu memenuhi standar sosial yang ada.

Dalam konteks perkembangan teknologi, ketidakpastian ekonomi, dan perubahan sosial, Gen Z menghadapi berbagai tantangan unik yang dapat

mempengaruhi pola pikir dan pandangan mereka terhadap kehidupan. Dalam menghadapi masa depan yang penuh ketidakpastian, kekhawatiran Gen Z menjadi subjek yang cukup menarik untuk dipelajari. Gen Z memiliki karakteristik yang unik, diantaranya lebih mandiri, berorientasi pencapaian, serta sangat mengandalkan teknologi sebagai alat komunikasi dan pembelajaran. Mereka juga dianggap lebih sadar akan isu-isu sosial dan lingkungan, serta cenderung mendukung gerakan-gerakan sosial yang positif.

Secara keseluruhan, Gen Z merupakan generasi yang tumbuh ditengah kemajuan teknologi dan dinamika sosial-ekonomi yang setiap harinya terus berubah. Generasi ini banyak menghadapi tekanan terkait ekspektasi sosial dan kekhawatiran terhadap masa depan, akan tetapi pada saat yang bersamaan mereka juga memiliki potensi untuk menjadi awal perubahan generasi yang lebih maju dan berkembang berkat kesadaran yang cukup tinggi dan ketergantungan mereka pada teknologi untuk mencapai tujuan hidup.

2.2.2 Stres

Stres Stres merupakan respons psikologis yang timbul ketika individu merasa tertekan oleh tuntutan, tekanan, atau beban yang berasal dari lingkungan maupun diri sendiri. Dalam kehidupan modern, berbagai faktor dapat memicu munculnya stres, salah satunya adalah penggunaan media sosial. Bagi Generasi Z, media sosial tidak hanya menjadi ruang hiburan, tetapi juga arena perbandingan sosial yang intens. Mereka sering kali menilai pencapaian diri berdasarkan apa yang dilihat dari unggahan orang lain, sehingga memunculkan rasa cemas, minder, hingga tekanan psikologis yang berujung pada stres. Selain itu, stres juga kerap dialami oleh siswa sekolah dan mahasiswa akibat tuntutan akademik yang tinggi, seperti tugas, ujian, target prestasi, dan tekanan dari lingkungan belajar [10]. Kondisi ini menjadikan stres sebagai fenomena yang sangat umum di kalangan anak muda saat ini.

Dalam konteks Gen Z, situasi menjadi semakin kompleks ketika mereka menghadapi perbandingan sosial, cyberbullying, atau tekanan untuk tampil sempurna secara daring. Ketika individu merasa tidak memiliki kemampuan yang

cukup untuk menghadapi tantangan tersebut, mereka dapat mengalami rasa terancam yang memicu stres berlebih. Respons setiap individu terhadap stres pun berbeda-beda; ada yang mampu mengelola tekanan dengan baik, namun ada pula yang kesulitan dan menunjukkan reaksi emosional negatif. Perbedaan ini dipengaruhi oleh pengalaman pribadi, ketahanan mental, dukungan sosial, dan lingkungan tempat mereka tumbuh [11]. Dengan memahami bagaimana stres bekerja pada generasi ini, penelitian mengenai deteksi dini tingkat stres menjadi semakin relevan untuk membantu mereka mengelola kondisi psikologis secara tepat dan efektif.

A. Jenis Stres Berdasarkan Tingkat Keparahan

Stres dapat dikategorikan berdasarkan tingkat keparahannya, yaitu terdapat stres ringan, sedang, hingga berat. Setiap tingkatan tersebut mempunyai karakteristik dan dampak yang berbeda terhadap individu. Berikut merupakan jenis-jenis stres:

Tabel 2.1 Jenis Stress

Jenis Stress	Definisi	Karakteristik	Dampak
Stress ringan	Merupakan bentuk stres yang bersifat sementara dan biasanya terjadi akibat situasi yang menantang atau membutuhkan perhatian segera.	Perasaan gelisah, waspada saat berada di situasi tertentu.	Tidak berbahaya apabila dapat mengelola dengan baik
Stress sedang	Stres sedang merupakan tingkat stres yang lebih signifikan, yang dapat berlangsung lebih lama dan mulai mempengaruhi emosi seseorang	Kesulitan untuk konsentrasi, perasaan lelah oleh tanggung jawab seperti pekerjaan, gangguan pola tidur seperti insomnia ringan, serta mudah lelah	Dapat mengganggu kegiatan atau produktivitas, dan dapat berubah menjadi stres berat apabila tidak dapat mengelola emosi.

Jenis Stress	Definisi	Karakteristik	Dampak
Stress berat	Merupakan bentuk stres dengan level yang sudah parahm berkelanjutan, dan dapat memiliki dampak yang serius pada kesehatan mental dan fisik penderitanya	Perasaan cemas yang berkepanjangan, tidak dapat fokus terhadap kegiatan yang dilakukan, tidak memiliki ketertarikan melakukan aktivitas sehari-hari.	Berisiko menyebabkan gangguan kesehatan mental seperti depresi, dapat memicu penyakit seperti jantung dan gangguan metabolismik.

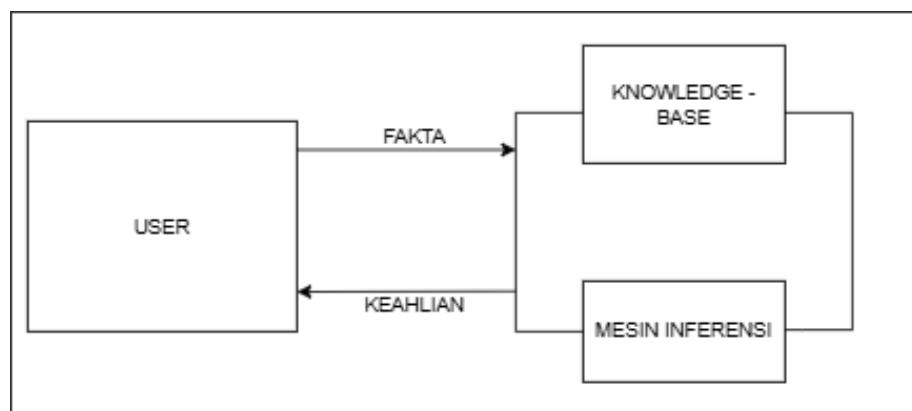
B. Gejala Stres Menurut Ahli

Berdasarkan jurnal berjudul Rancang Bangun Sistem Pakar Deteksi Stres Berbasis Web [12] maka ada beberapa gejala stres yaitu menjadi mudah marah, berasiki berlebihan terhadap sesuatu, sulit untuk bersantai, merasa kesal, menghabiskan banyak energi untuk cemas, tidak sabar, mudah tersinggung, sulit untuk beristirahat, mudah marah, sulit tenang, sulit sabar, gelisah, tidak memakleumi hal yang dianggap menghalangi, gelisah.

2.2.3 Sistem Pakar

Sistem pakar merupakan sebuah sistem yang kinerjanya mengadopsi keahlian yang dimiliki oleh seorang pakar dalam bidang tertentu ke dalam sistem atau program komputer yang disajikan dengan tampilan yang dapat digunakan oleh pengguna yang bukan seorang pakar sehingga dengan sistem tersebut pengguna bisa membuat sebuah keputusan atau menentukan kebijakan layaknya seorang pakar [18]. Sistem pakar atau biasa disebut dengan *expert system* merupakan salah satu cabang dari AI atau *Artificial Intelligence* yang menggunakan pengetahuan untuk penyelesaian masalah manusia dengan menggunakan pakar. Tujuan dari sistem pakar yaitu agar orang awam dapat menyelesaikan masalah atau sekedar mencari informasi yang akan dipandang berhasil ketika mampu mengambil keputusan seperti pakar.

Sistem pakar merupakan program komputer yang didalamnya terdapat pengetahuan khusus dari para ahli di bidang tertentu [22]. Program ini pertama kali dikembangkan oleh peneliti kecerdasan buatan pada tahun 1960-an dan 1970-an, serta mulai diterapkan secara komersial pada dekade 1980-an. Sistem ini bekerja dengan menggunakan serangkaian aturan untuk menganalisis informasi yang biasanya diberikan oleh pengguna, serta melakukan analisis matematis terhadap masalah tersebut. Sistem pakar memiliki kontribusi besar di berbagai sektor seperti industri, layanan keuangan, telekomunikasi, kesehatan, video game, dst.



Gambar 2.4 Konsep Dasar Sistem Pakar

Gambar 2.4 menggambarkan konsep dasar sistem pakar. Gambar tersebut menjelaskan bahwa user atau pengguna akan menyampaikan fakta atau informasi ke dalam sistem pakar, kemudian fakta tersebut akan disimpan di *knowledge-base* dan diolah oleh mesin inferensi, sistem dapat memberikan feedback ke user berupa jawaban berdasarkan pengetahuan yang disampaikan sebelumnya. Penjelasan komponen-komponen yang ada pada struktur sistem pakar adalah:

1. Basis Pengetahuan (*Knowledge Base*)

Knowledge Base merupakan inti dari sebuah sistem pakar karena basis pengetahuan merupakan presentasi pengetahuan. Basis pengetahuan adalah sebuah basis data yang menyimpan aturan-aturan tentang suatu *domain knowledge* atau pengetahuan tertentu.

2. Mesin Inferensi

Mesin Inferensi merupakan bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar.

A. Kelebihan Sistem Pakar

Sistem pakar mempunyai beberapa kelebihan seperti Increased *Availability* atau pengetahuan seorang pakar yang sudah diadaptasi dalam bentuk software dapat diperbanyak dan disebarluaskan dalam jumlah yang tidak terbatas, Reduced cost atau mengurangi biaya dimana pembuatan sistem pajar ini mampu mengurangi biaya yang harus dikeluarkan untuk membayar pajak dan ahli, Reduced danger atau mengurangi bahaya yaitu sistem pakar dapat digunakan dalam lingkungan yang mungkin berbahaya untuk manusia, Multiple expertise atau beberapa keahlian yaitu pengetahuan dari beberapa ahli dapat dibuat tersedia untuk bekerja secara simultan dan terus menerus pada masalah di setiap saat, siang atau malam hari.

B. Kekurangan Sistem Pakar

Adapun kekurangan dari sistem pakar yaitu daya kerja dan produktivitas manusia menjadi berkurang karena semua hal dilakukan oleh sistem, pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional, dan pengetahuan seorang pakar yang sudah diadaptasi dalam bentuk software dapat diperbanyak dalam jumlah yang tidak terbatas.

C. Unsur Penting Sistem Pakar

Dalam pengembangan sistem pakar terdapat beberapa unsur penting dalam pengembangan sistem pakar, yaitu perancangan sistem pakar berdasarkan pakar dan dalam perancangannya akan menggunakan beberapa metode seperti metode forward, backforward chaining, dan certain faktor ataupun dapat menggunakan fuzzy logic berdasarkan masalah yang didasari oleh pakar. Sistem yang mendukung dari segi hardware maupun software, sistem tersebut berbasis object dan dapat menggunakan phpmyadmin, Java, SQL, dan lain sebagainya.

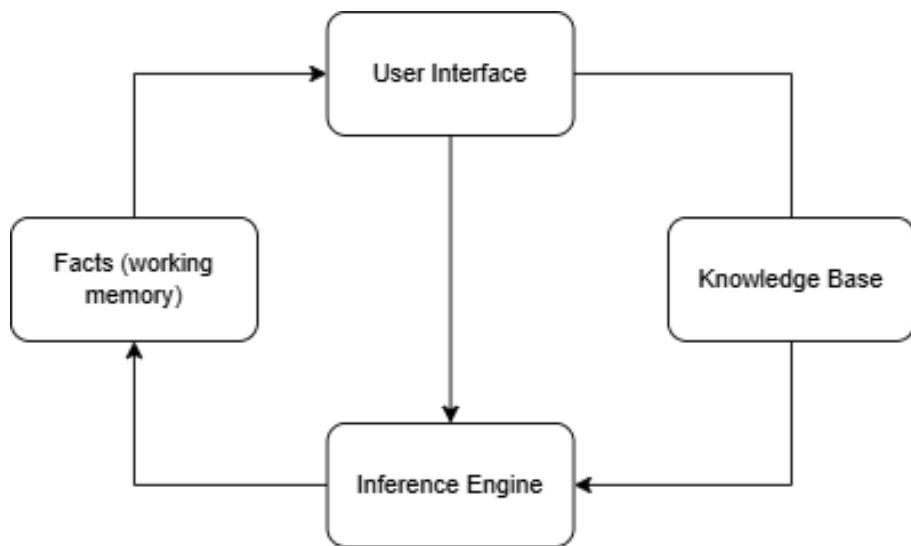
2.2.4 Website

Website merupakan kumpulan halaman yang bisa diakses melalui internet yang berisikan informasi dalam bentuk gambar, teks, video, dan format lainnya. Jenis-jenis website sendiri terbagi menjadi beberapa macam seperti website statis (setiap adanya perubahan dilakukan secara manual oleh pengembang), website dinamis (website yang isinya bisa diperbarui otomatis dengan database), website berita (menyajikan beberapa berita), blog (berisikan artikel atau jurnal pribadi). Dalam penelitian ini, website digunakan sebagai platform untuk menampilkan hasil deteksi stres pada Gen Z.

2.2.5 Forward Chaining

Metode *Forward Chaining* merupakan salah satu metode yang sering digunakan untuk penelitian sistem pakar. *Forward Chaining* yaitu metode pencarian ke depan dengan menampilkan data data yang mengarah pada kesimpulan akhir [21]. Metode ini dipilih karena memungkinkan proses pencarian dimulai dengan memilih kumpulan data atau fakta, kemudian akan dilakukan penyusunan konklusi berdasarkan fakta-fakta yang telah dipilih untuk mencapai tujuan [22]. Metode *Forward Chaining* merupakan strategi pencarian melalui proses pencarian dari sekumpulan data atau fakta, kemudian dari data-data tersebut akan disimpulkan solusi apa yang dipilih untuk masalah tersebut [22].

Forward Chaining memulai proses pencarian dengan data, inferensinya diawali dengan informasi atau data yang diberikan pengguna. Pada sistem pakar yang menggunakan *Forward Chaining*, proses akan dimulai dengan mengumpulkan data dan fakta mengenai masalah yang akan dianalisis. Setelah data terkumpul, mesin inferensi akan mencari aturan-aturan dalam basis pengetahuan yang cocok dengan fakta yang ada. Setiap kali aturan cocok dengan fakta, maka aturan tersebut akan dieksekusi dan akan muncul kesimpulannya [23]. Proses ini berlanjut hingga semua fakta diproses dan akan dihasilkan kesimpulan akhir [24]



Gambar 2.5 Arsitektur *Forward Chaining*

Pada gambar diatas dimulai dengan *user interface* atau tempat pengguna akan berinteraksi dengan sistem, pada bagian ini terdapat input fakta awal, kemudian fakta-fakta tersebut akan disimpan pada bagian *facts* atau memori fakta. Setelah itu sistem akan membuatkan aturan berupa logika yang digunakan oleh sistem untuk membuat kesimpulan. Lalu mesin inferensi akan menjalankan proses *Forward Chaining*.

A. Keunggulan Forward Chaining

Forward Chaining mempunyai beberapa keunggulan yang membuat metode ini menjadi efektif untuk digunakan dalam sistem pakar. Forward chaining dinilai sederhana dan efisien karena proses pencarian berbasis data membuat metode ini cukup mudah untuk diimplementasikan terutama untuk masalah yang memiliki banyak fakta awal tetapi memerlukan inferensi langsung menuju solusi. Selain itu, metode ini kontekstual berdasarkan fakta karena bekerja dari fakta yang ada, sistem dapat memberikan solusi yang relevan dengan kondisi nyata pengguna, hal ini sangat bermanfaat dalam beberapa situasi salah satunya seperti diagnosis medis, dimana fakta-fakta gejala harus dianalisis untuk memberikan diagnosis yang tepat.

B. Keterbatasan Forward Chaining

Meskipun forward chaining mempunyai keunggulan, adapula beberapa keterbatasan yang perlu dipertimbangkan dalam metode ini. Kinerja yang lambat pada masalah komplek seperti pada kasus dengan jumlah fakta dan aturan yang sangat banyak, Forward Chaining dapat menjadi tidak efisien karena harus memeriksa setiap aturan yang mungkin cocok dengan fakta yang diberikan. Ini dapat memperlambat kinerja sistem dalam situasi yang sangat kompleks. Metode forward chaining dinilai memerlukan basis pengetahuan yang lengkap, keberhasilan forward chaining sangat bergantung pada kualitas dan kelengkapan basis pengetahuan. Jika basis pengetahuan tidak lengkap atau terdapat aturan yang tidak mencakup semua kemungkinan, hasil yang diberikan oleh sistem dapat kurang akurat.

C. Cara Kerja dan Contoh Forward Chaining

Forward Chaining bekerja dengan melakukan penalaran dari fakta awal menuju suatu kesimpulan melalui serangkaian aturan berbentuk *IF–THEN*. Metode ini memulai proses inferensi dari informasi yang diberikan pengguna, kemudian mesin inferensi akan mencari aturan mana yang cocok dengan fakta tersebut. Jika suatu aturan terpenuhi seluruhnya, maka aturan tersebut dieksekusi dan menghasilkan fakta baru. Fakta baru tersebut kembali dibandingkan dengan aturan-aturan lain, sehingga proses berjalan secara berantai (*chaining*) sampai sistem menemukan kesimpulan akhir atau tidak ada lagi aturan yang dapat diproses. Dengan kata lain, *Forward Chaining* menggunakan pendekatan *data- driven reasoning*, yaitu bergerak dari data menuju kesimpulan.

Sebagai ilustrasi, aturan *Forward Chaining* biasanya disusun dalam bentuk *IF–THEN*, misalnya pada penelitian [25] yang menyusun aturan diagnosis stres berdasarkan kombinasi gejala. Contoh aturan tersebut antara lain:

- a. Rule 1: IF (GS48 AND GS29 AND GS03 AND GS44 AND GS38 AND GS27 AND GS45) THEN Stres = Ringan
- b. Rule 2: IF (GS16 AND GS18 AND GS05 AND GS07 AND GS04 AND GS08 AND GS28) THEN Stres = Sedang
- c. Rule 3: IF (GS06 AND GS33 AND GS21 AND GS15 AND GS20

AND GS40 AND GS10 AND GS41) THEN Stres = Berat

Aturan-aturan diatas menggambarkan bahwa *Forward Chaining* mengharuskan seluruh gejala yang tercantum dalam rule terpenuhi agar sistem dapat menghasilkan suatu diagnosis. Jika pengguna memilih gejala yang sesuai dengan Rule 1, misalnya GS48, GS29, GS03, GS44, GS38, GS27, dan GS45, maka mesin inferensi akan mengeksekusi Rule 1 dan menghasilkan kesimpulan “Stres Ringan”. Jika sebagian gejala tidak terpenuhi, maka aturan tidak akan dijalankan sehingga sistem akan melanjutkan pencarian aturan lain yang relevan. Mekanisme ini menegaskan bahwa *Forward Chaining* sepenuhnya bergantung pada kelengkapan aturan pakar, berbeda dengan metode seperti *Random Forest* yang mampu mempelajari pola dari data dan memberikan hasil yang lebih adaptif.

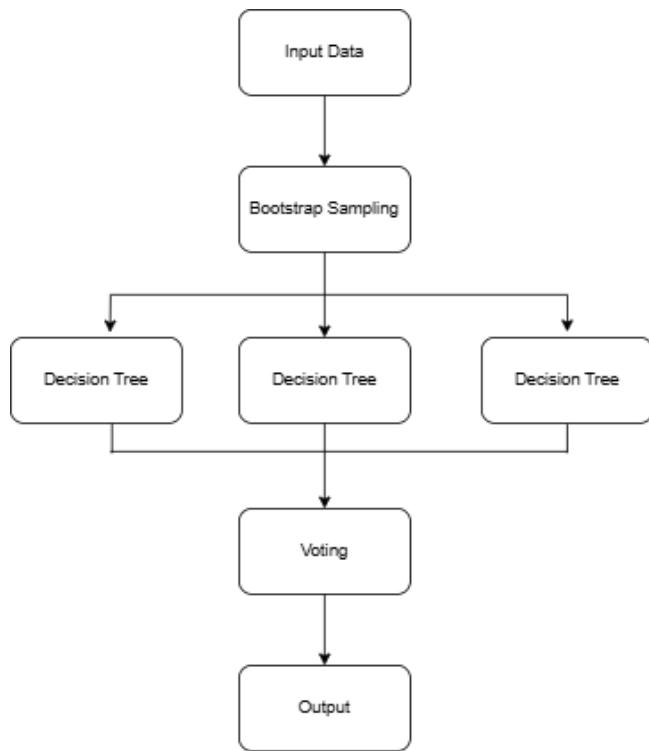
2.2.6 Random Forest

Random Forest merupakan salah satu metode klasifikasi populer dalam machine learning yang terbukti mampu menghasilkan performa tinggi pada berbagai jenis dataset [13]. Algoritma ini diperkenalkan oleh Breiman pada tahun 2001 [14] sebagai pengembangan dari teknik ensemble learning, yaitu pendekatan yang menggabungkan beberapa model untuk meningkatkan akurasi dan stabilitas prediksi. Dalam *Random Forest*, sejumlah pohon keputusan (*decision trees*) dibangun dari data sampel yang diambil secara acak melalui teknik bootstrap sampling. Setiap pohon dibentuk secara independen, tanpa dipengaruhi oleh pohon lainnya, dan berakhir pada node daun yang menghasilkan prediksi kelas [15]. Setelah semua pohon terbentuk, algoritma melakukan proses *majority voting* untuk menentukan hasil klasifikasi akhir. Pendekatan ini membuat *Random Forest* menjadi metode yang kuat, mudah digunakan, dan mampu mencapai akurasi tinggi walaupun tidak memerlukan banyak penyesuaian parameter.

Secara teknis, *Random Forest* sangat bergantung pada dua konsep inti, yaitu bagging dan pemilihan fitur secara acak. Konsep bagging atau *bootstrap aggregating* dilakukan dengan mengambil sampel data secara berulang dengan pengembalian (*replacement*), sehingga setiap pohon dilatih menggunakan

kombinasi data yang berbeda. Ini menghasilkan variasi antar pohon dan membantu menurunkan variansi model, sehingga mengurangi kemungkinan terjadinya overfitting. Selain itu, pada setiap node dalam pohon, algoritma memilih subset fitur secara acak untuk menentukan pemisahan terbaik. Pemilihan fitur secara acak ini membuat setiap pohon memiliki struktur yang unik dan memperkaya keragaman dalam ensemble, yang pada akhirnya membuat model lebih stabil dan tangguh dalam menghadapi data berdimensi besar maupun kompleks.

Keunggulan lain dari *Random Forest* adalah fleksibilitasnya yang dapat digunakan baik untuk tugas klasifikasi maupun regresi serta kemampuannya dalam menghadapi noise dan ketidakseimbangan data. *Random Forest* juga menyediakan informasi mengenai tingkat kepentingan fitur melalui perhitungan *Mean Decrease in Impurity (Gini Importance)*, sehingga peneliti dapat memahami fitur mana yang paling berkontribusi dalam proses klasifikasi. Dalam berbagai penelitian empiris, *Random Forest* terbukti memiliki performa yang unggul dibandingkan beberapa algoritma lain seperti *Support Vector Machine*, *Logistic Regression*, *Neural Network*, *Naïve Bayes*, *K-Nearest Neighbors (KNN)*, *XGBoost*, *C5.0*, dan *CART* [17]. Berdasarkan keunggulan tersebut, *Random Forest* menjadi pilihan yang tepat dalam penelitian ini, terutama untuk menganalisis pola gejala dan mengklasifikasikan tingkat stres pada Generasi Z secara akurat.



Gambar 2.6 Arsitektur Random Forest

Arsitektur diatas dimulai dari input data berupa dataset awal yang berisi fitur, kemudian sistem akan mengambil sampel dataset secara acak, setiap pohon akan memperoleh dataset yang berbeda (*bootstrap sample*). Dari setiap bootstrap sample tersebut akan dibangun pohon keputusan pada proses ini fitur yang dipilih hanya sebagian dan acak pada setiap nodenya. Setelah semua pohon selesai dibuat, sistem akan menggabungkan prediksi dari semua pohon, dan akan menggabungkan prediksi dari semua pohon. Hasil akhirnya berupa prediksi berdasarkan hasil penggabungan pohon.

Adapun langkah-langkah dalam membangun model Random Forest adalah sebagai berikut:

1. Pemilihan jumlah pohon keputusan: Tahapan awal dalam *Random Forest* yaitu menentukan jumlah pohon keputusan yang akan digunakan. Jumlah pohon yang lebih besar biasanya akan meningkatkan nilai akurasinya, dan juga meningkatkan waktu komputasinya.

2. Pembuatan subset data: Dari dataset yang ada, beberapa subset data dibuat secara acak dengan pengembalian atau *replacement*. Tiap subset data tersebut akan digunakan untuk melatih masing-masing pohon keputusan.
3. Pemilihan fitur secara acak: Pada setiap percabangan pohon, algoritma akan memilih acak subset fitur dari total fitur yang ada. Pada tahapan ini bertujuan untuk mengurangi korelasi antar pohon.
4. Pembangunan pohon keputusan: Setiap pohon akan dilatih menggunakan subset data yang berbeda sehingga akan mencapai kriteria penghentian tertentu, seperti kedalaman maksimal pohon atau jumlah sample minimal pada daun.
5. Voting atau rata rata hasil: Hasil akan akan ditentukan melalui voting mayoritas dari tiap pohon (untuk klasifikasi).

2.2.7 Recursive Feature Elimination

Recursive Feature Elimination atau RFE adalah teknik seleksi fitur yang digunakan untuk memilih fitur yang paling relevan dengan menghapus fitur yang tidak memberikan kontribusi signifikan terhadap kinerja model. Pada setiap iterasinya, fitur dengan bobot atau nilai persentase terendah akan dihapus. Dengan menggunakan RFE, model akan jauh lebih sederhana dan hanya berisi fitur yang penting namun tidak akan mengurangi nilai akurasinya. Dalam penelitian ini, RFE digunakan untuk mengoptimalkan deteksi stres pada Gen Z dengan memilih fitur terbaik yang akan digunakan dalam model antara Forward Chaining dan Random Forest.

Proses RFE bekerja dengan cara membangun model pelatihan menggunakan seluruh fitur yang ada, kemudian secara bertahap menghapus fitur yang paling tidak signifikan berdasarkan suatu kriteria tertentu, biasanya berdasarkan bobot fitur dalam model yang dilatih. Proses ini diulang hingga jumlah fitur yang tersisa mencapai nilai yang diinginkan, atau hingga tercapai kinerja model yang optimal.

Langkah-langkah implementasi RFE dalam penelitian ini adalah sebagai berikut:

- a. Preprocessing Data: Sebelum melakukan seleksi fitur dengan RFE, dataset akan dipersiapkan melalui beberapa tahap *preprocessing*, seperti penanganan *missing values*, normalisasi fitur, dan pembagian dataset menjadi data pelatihan (*training*) dan data pengujian (*testing*).
- b. Penerapan Random Forest: Model *Random Forest* digunakan untuk menilai kepentingan setiap fitur. Dalam RFE, *Random Forest* akan dilatih menggunakan seluruh fitur dalam dataset untuk menentukan bobot setiap fitur berdasarkan seberapa besar kontribusinya terhadap prediksi model. Fitur-fitur yang dianggap kurang penting oleh model akan dieliminasi satu per satu.
- c. Eliminasi Fitur: Fitur yang tidak memberikan kontribusi signifikan akan dihapus secara iteratif. Proses ini akan dilakukan beberapa kali hingga hanya tersisa fitur-fitur yang dianggap paling penting dalam memprediksi tingkat stres pada Gen Z.
- d. Evaluasi Kinerja: Setiap kali RFE menghapus fitur, model akan diuji kembali untuk melihat perubahan kinerja, yang diukur menggunakan metrik-metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Proses ini memastikan bahwa hanya fitur-fitur yang benar-benar relevan yang dipilih, sehingga model yang dihasilkan lebih efisien dan akurat.
- e. Penerapan pada Model: Setelah seleksi fitur selesai, fitur yang terpilih akan diterapkan pada yang menggunakan dua metode yaitu *Forward Chaining* dan *Random Forest* untuk mengoptimalkan deteksi stres. *Forward Chaining* digunakan untuk membuat aturan berdasarkan fakta yang ada, sementara *Random Forest* memberikan kemampuan klasifikasi yang kuat. Integrasi kedua metode ini diharapkan dapat meningkatkan akurasi dan efisiensi model dalam mendeteksi stres pada Gen Z.

2.2.8 Synthetic Minority Over-Sampling Technique

Ketidakseimbangan kelas atau *class imbalance* adalah salah satu permasalahan klasik yang sering muncul dalam penerapan algoritma klasifikasi. Kondisi ini terjadi ketika satu atau lebih kelas memiliki jumlah sampel yang jauh lebih sedikit dibandingkan kelas lainnya, sehingga distribusi data menjadi tidak seimbang. Situasi seperti ini dapat menyebabkan model cenderung belajar lebih banyak dari kelas mayoritas dan mengabaikan pola pada kelas minoritas. Akibatnya, meskipun model tampak memiliki akurasi tinggi secara keseluruhan, performanya seringkali buruk dalam mendeteksi kelas yang jarang muncul. Ketidakmampuan model untuk mengenali kelas minoritas ini dapat menurunkan kemampuan generalisasi, terutama ketika model diterapkan pada data nyata yang membutuhkan prediksi presisi pada seluruh kelas, termasuk yang minoritas [19].

Untuk mengatasi permasalahan tersebut, salah satu pendekatan yang efektif adalah dengan menerapkan *Synthetic Minority Over-sampling Technique* (SMOTE). SMOTE merupakan teknik *oversampling* yang bekerja dengan cara menambah jumlah data pada kelas minoritas melalui pembuatan sampel sintetis baru. Tidak seperti metode oversampling tradisional yang hanya melakukan duplikasi data, SMOTE menghasilkan data baru melalui proses interpolasi antara sebuah sampel minoritas dengan beberapa *nearest neighbors*-nya. Dengan strategi ini, distribusi setiap kelas menjadi lebih seimbang tanpa meningkatkan risiko overfitting secara signifikan, karena sampel tambahan yang dihasilkan bukan sekadar salinan data, tetapi benar-benar representasi baru yang berada di sekitar ruang fitur kelas minoritas [19]. Hal ini membantu model lebih memahami pola kelas minoritas dan meminimalkan bias terhadap kelas mayoritas.

Efektivitas SMOTE telah dibuktikan dalam berbagai penelitian, terutama dalam meningkatkan kinerja model machine learning terkait akurasi, presisi, dan recall. Salah satu penelitian dalam klasifikasi kepribadian MBTI menunjukkan bahwa penerapan SMOTE dapat meningkatkan performa model Logistic Regression dari 75% menjadi 80% [20]. Peningkatan ini menandakan bahwa model menjadi lebih sensitif terhadap kelas yang underrepresented tanpa mengorbankan

kinerja keseluruhan. Dengan demikian, penggunaan SMOTE sangat relevan dalam penelitian ini yang juga menghadapi ketidakseimbangan pada kelas stres, khususnya kelas Stres Sedang yang jumlah datanya jauh lebih sedikit dibandingkan dua kelas lainnya. Penerapan SMOTE membantu memastikan bahwa model dapat belajar secara optimal pada seluruh kategori dan memberikan hasil prediksi yang lebih akurat serta stabil.

2.2.9 Confusion Matrix

Confusion matrix merupakan alat evaluasi yang digunakan dalam klasifikasi machine learning untuk mengukur kinerja model prediktif. Biasanya berbentuk tabel yang menampilkan perbandingan antara hasil prediksi model dengan label sebenarnya dari data. Kegunaan confusion matrix adalah untuk mengevaluasi akurasi model klasifikasi, dan menghitung metrik evaluasi seperti akurasi, presisi, recall, F1-Score. Komponen utama dari confusion matrix ada 4 yaitu:

1. True Positive (TP): Data positif yang diprediksi dengan benar
2. False Positive (FP): Data negatif yang salah diklasifikasikan sebagai positif
3. True Negative (TN): Data negatif yang diprediksi dengan benar
4. False Negative (FN): Data positif yang salah diklasifikasikan sebagai negatif

Tabel 2.2 Contoh Confusion Matrix

		Kelas Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	TP	FN

Rumus untuk menghitung akurasi, presisi, dan recall pada confusion matrix adalah sebagai berikut:

$$akurasi = \frac{TP+TN}{\text{Total}} \quad (2.1)$$

Akurasi merupakan nilai yang didapatkan dari perbandingan antara jumlah data yang diprediksi dengan benar dan total keseluruhan data. Tujuan akurasi yaitu untuk menilai sejauh mana ketepatan model algoritma dalam hasil klasifikasinya

$$presisi = \frac{TP}{TP+FP} \quad (2.2)$$

Presisi merupakan nilai hasil perbandingan dari jumlah data yang diprediksi dengan benar positif dengan total seluruh data yang diprediksi positif. Tujuan presisi yaitu untuk mengukur seberapa sering prediksi tersebut benar ketika model algoritma menunjukkan prediksi positif.

$$recall = \frac{TP}{TP+FN} \quad (2.3)$$

Recall merupakan nilai hasil perbandingan dari jumlah data yang diprediksi dengan benar positif dengan total seluruh data yang diprediksi benar. Recall digunakan untuk mengukur seberapa sering prediksi tersebut positif ketika model algoritma menunjukkan prediksi benar.

$$F1 Score = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (2.4)$$

F1 Score digunakan untuk mengukur nilai rata-rata dari nilai presisi dan recall.

2.2.10 Cross Validation

Cross Validation adalah metode validasi model yang digunakan untuk menilai kemampuan generalisasi model terhadap data baru. Teknik ini bekerja dengan membagi dataset menjadi beberapa bagian atau folds, kemudian akan membagi secara bergantian satu bagian dijadikan data uji dan sisanya digunakan sebagai data latih. Proses ini akan diulang hingga seluruh bagian pernah menjadi data uji, dan hasil evaluasi dihitung sebagai rata-rata dari seluruh pengujian.

Salah satu yang paling umum adalah penggunaan K-Fold Cross Validation, dataset ala dibagi menjadi k- bagian. Nilai k- biasanya antara 5 hingga 10, tergantung berapa datatset dan variasinya. Metode ini memberikan estimasi akurasi model yang lebih stabil. Pada konteks sistem pakar deteksi stres, penerapan K-Fold Cross Validation ini dinilai sangat berguna untuk memastikan bahwa model Random Forest dan dengan RFE tidak hanya unggul pada subset tertentu, tetapi juga memiliki performa yang konsisten di seluruh data.