

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan secara menyeluruh mengenai metodologi yang digunakan dalam proses pengembangan, mulai dari pendekatan penelitian, tahapan pelaksanaan, alur pengembangan sistem, hingga arsitektur teknis dari sistem yang akan dirancang.

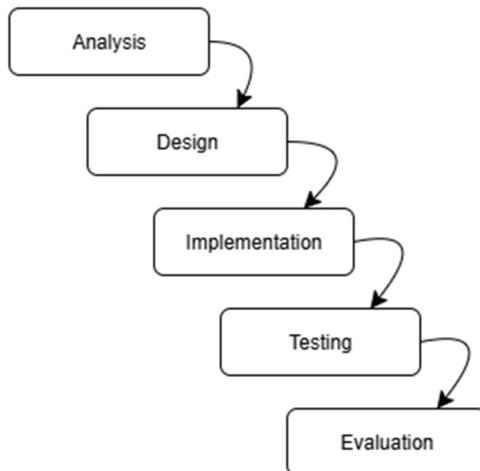
3.1 Jenis dan Pendekatan Penelitian

Jenis penelitian yang digunakan adalah *Research and Development* (R&D). Pendekatan R&D dipilih karena kesesuaian dengan tujuan utama penelitian ini, yaitu untuk menghasilkan dan mengembangkan produk teknologi yang dapat digunakan secara praktis. Dengan pendekatan ini, penulis dapat melalui tahapan yang terprediksi dan terukur, dimulai dari perumusan masalah hingga evaluasi dan penyempurnaan produk.

Pengelolaan data yang dipergunakan pada penelitian ini yaitu kombinasi metode kualitatif dan kuantitatif. Metode kualitatif digunakan untuk menggali informasi dari *expert* melalui wawancara, khususnya dalam proses perolehan pengetahuan untuk membangun basis pengetahuan pada sistem pakar. Sementara itu, pendekatan kuantitatif digunakan untuk pengujian sistem dengan menggunakan *Black Box Testing* dan mengukur kepuasan pengguna dengan menggunakan *System Usability Scale* (SUS) melalui kuesioner.

Pendekatan penelitian adalah suatu proses dalam sebuah penelitian yang bermanfaat untuk menciptakan suatu penelitian yang berkualitas. Salah satu rancangan pengembangan sistem pakar yang selalu dipergunakan yaitu jenis model *Waterfall*. Model *Waterfall* dipilih sebagai pendekatan dalam penelitian ini karena sesuai dengan karakteristik pengembangan perangkat lunak yang bersifat sistematis dan berurutan. Pada model ini, setiap tahapan dilakukan secara berurutan mulai dari perencanaan, perancangan (pemodelan), implementasi (konstruksi), hingga pengujian, di mana masing-masing tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Alur model penelitian ini digambarkan

pada Gambar 3.1 berikut:



Gambar 3. 1 Pendekatan *Waterfall*

Proses pengembangan ini dianalogikan seperti aliran air terjun (*waterfall*) yang mengalir ke bawah secara bertahap, sehingga cocok digunakan dalam pengembangan sistem pakar yang memiliki tahapan kerja yang jelas dan terstruktur. Model *Waterfall* dimulai dari analisa kebutuhan, dilanjutkan dengan perancangan sistem, implementasi, pengujian, hingga evaluasi akhir.

3.2 Deskripsi Umum Sistem

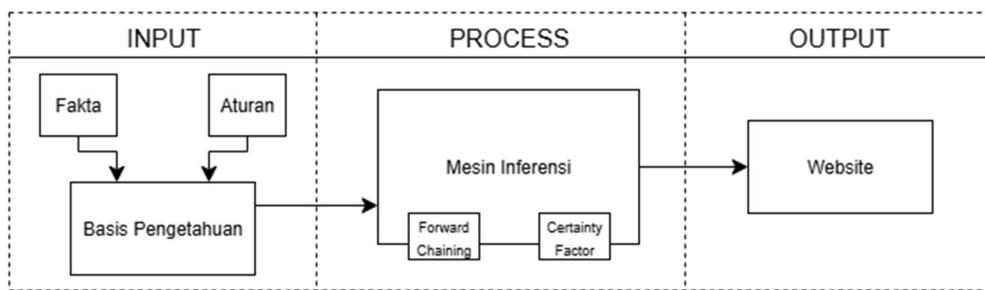
Sistem yang dirancang dan dikembangkan pada penelitian ini adalah sistem pakar berbasis *Forward Chaining* dan *Certainty Factor*, yang bertujuan untuk memberikan rekomendasi tindakan *maintenance* pada mesin Kraft di PT Dirgantara Indonesia. Sistem ini dirancang untuk membantu teknisi pemula, seperti karyawan magang dalam melakukan mitigasi awal terhadap kondisi mesin berdasarkan gejala yang teridentifikasi di lapangan.

Sistem ini diimplementasikan dalam bentuk *website* sederhana agar mudah dipahami dan digunakan oleh pengguna. Hasil rekomendasi ditampilkan dengan jelas dan langsung menunjukkan tindakan yang bisa dilakukan.

Fokus utama dari sistem ini adalah untuk memberikan hasil rekomendasi mitigasi berupa rekomendasi tindakan *maintenance*, yang disusun berdasarkan basis pengetahuan pakar dan dihitung menggunakan metode *Certainty Factor*. Nilai

Certainty Factor tersebut didapat berdasarkan tingkat keyakinan pakar terhadap setiap tindakan *maintenance* yang direkomendasikan. Dengan pendekatan ini, sistem mampu memberikan tingkat keyakinan terhadap setiap rekomendasi, sehingga mendukung pengambilan keputusan teknis secara lebih terarah dan efisien.

Untuk memberikan pemahaman menyeluruh mengenai cara kerja sistem, berikut disajikan gambaran umum melalui *block diagram* sistem. Rancangan *block diagram* sistem pada sistem ini sebagai berikut:



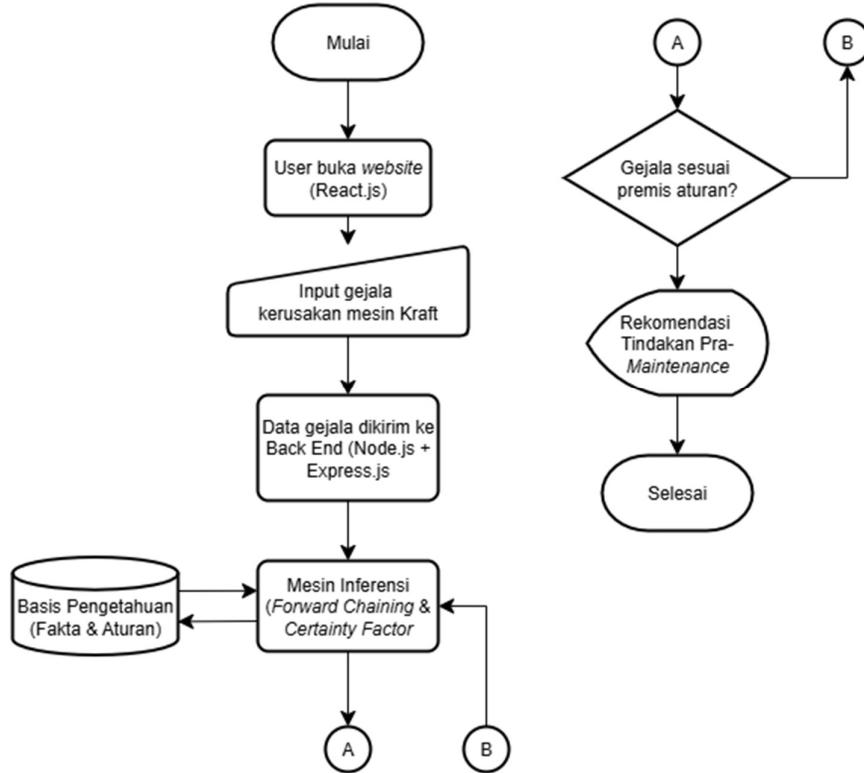
Gambar 3. 2 *Block Diagram* Sistem Pakar

Berdasarkan Gambar 3.2, *block diagram* sistem pada sistem pakar ini terdiri dari tiga bagian, yaitu bagian *input*, *process*, dan *output*. Pada bagian *input*, sistem menerima data yaitu basis pengetahuan yang berisi kumpulan fakta dan aturan yang telah disusun oleh *expert engineer*. Basis pengetahuan menjadi fondasi logika yang digunakan untuk menghasilkan keputusan.

Selanjutnya, pada bagian *process*, sistem menggunakan mesin inferensi dengan metode *forward chaining* yang dilengkapi dengan pendekatan *certainty factor* (CF) untuk menangani ketidakpastian. Mesin inferensi ini akan menelusuri aturan dari basis pengetahuan, lalu menghitung tingkat kepastian dari setiap kemungkinan rekomendasi. Proses ini memastikan bahwa hasil yang diberikan tidak hanya logis, tetapi juga memiliki bobot keyakinan yang terukur.

Terakhir, pada bagian *output*, sistem menghasilkan rekomendasi tindakan *maintenance* yang disertai dengan nilai CF. Nilai ini menunjukkan seberapa yakin sistem terhadap rekomendasi yang diberikan, sehingga pengguna dapat mempertimbangkan tindakan berdasarkan tingkat kepastian yang relevan.

Untuk memberikan gambaran proses kerja sistem secara menyeluruh dan terstruktur, berikut adalah rancangan *flowchart* sistem pada sistem ini:



Gambar 3. 3 *Flowchart* Sistem Pakar *Forward Chaining & Certainty Factor*

Berdasarkan Gambar 3.3, *flowchart* sistem pada sistem ini dimulai dengan aktivitas pengguna membuka *website* yang dibangun menggunakan *framework React.js*. Setelah halaman antarmuka terbuka, pengguna diminta untuk memasukkan gejala kerusakan pada mesin Kraft melalui *form input* yang telah disediakan.

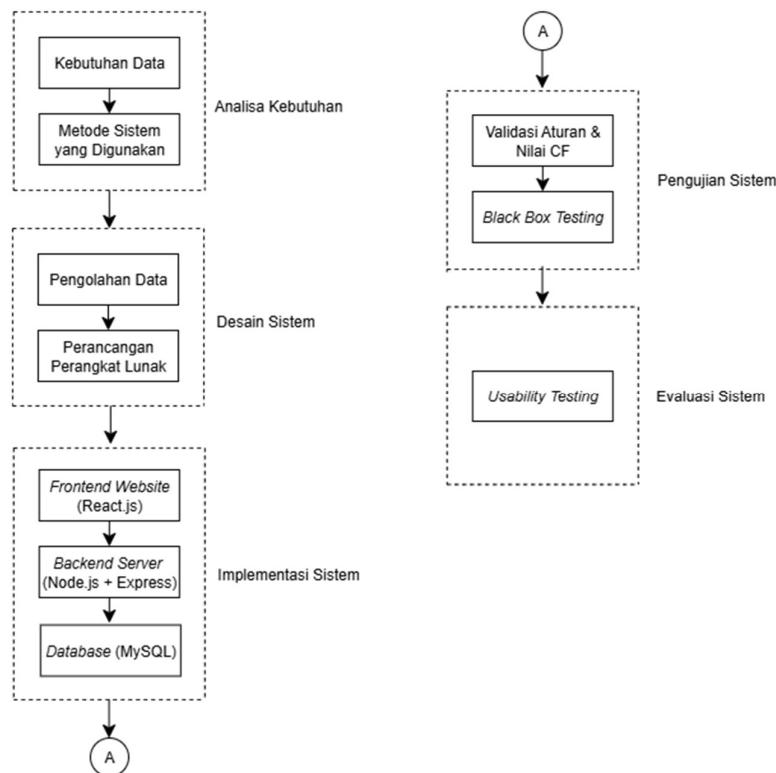
Data gejala yang dimasukkan kemudian dikirim ke sisi *backend*, yang dibangun menggunakan *Node.js* dan *Express*, untuk diproses lebih lanjut. Di tahap ini, sistem melakukan pencocokan gejala terhadap basis aturan yang telah ditentukan. Jika gejala tidak sesuai dengan aturan yang ada, sistem akan mengarahkan pengguna untuk memasukkan ulang gejala agar proses diagnosis dapat dilakukan secara akurat.

Jika gejala dinyatakan sesuai, sistem akan melanjutkan ke proses mesin inferensi, yang menggunakan metode *Forward Chaining* dan *Certainty Factor* untuk menentukan tingkat kepastian dari diagnosis yang dihasilkan. Mesin inferensi ini bekerja berdasarkan basis pengetahuan yang terdiri dari fakta dan aturan yang telah dirancang sebelumnya.

Hasil dari proses inferensi kemudian digunakan untuk menghasilkan rekomendasi tindakan *maintenance*, yang ditampilkan kepada pengguna sebagai *output* akhir dari sistem. Setelah rekomendasi diberikan, proses dinyatakan selesai.

3.3 Prosedur Penelitian

Prosedur penelitian ini mengikuti tahapan-tahapan sistematis yang ditetapkan dalam alur model *Waterfall* pada Gambar 3.1. Namun, dalam konteks penelitian ini, setiap tahap disesuaikan dengan kebutuhan pengembangan sistem pakar, sebagaimana ditunjukkan pada Gambar 3.4. Setiap tahapan dilaksanakan secara berurutan dan iteratif untuk memastikan pengembangan solusi yang valid.

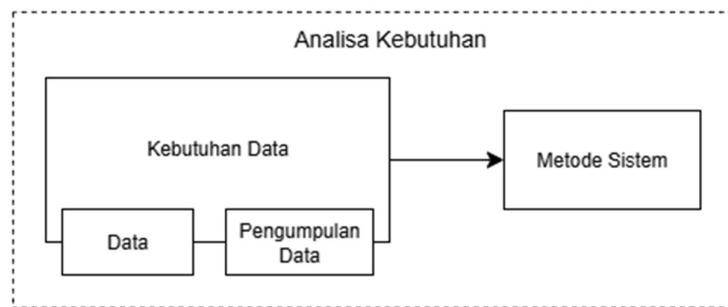


Gambar 3. 4 Prosedur Penelitian

Berdasarkan Gambar 3.4, setiap tahapan mencerminkan proses kerja yang terstruktur, dimulai dari analisa kebutuhan hingga evaluasi sistem. Diagram ini juga menampilkan komponen teknis yang digunakan, seperti teknologi *frontend* dan *backend*, serta metode pengujian dan evaluasi yang mendukung validitas sistem secara menyeluruh.

3.3.1 Analisa Kebutuhan

Tahap awal dalam penelitian ini adalah mengidentifikasi kebutuhan penelitian. Pada tahap ini, kebutuhan yang diperlukan dalam penelitian mulai dari data dan metode sistem yang akan digunakan. Struktur analisa kebutuhan dalam penelitian ini dapat divisualisasikan melalui Gambar 3.5, yang menunjukkan keterkaitan antara komponen data yang dikumpulkan dan metode sistem yang diterapkan dalam pengembangan sistem pakar.



Gambar 3. 5 Proses Analisa Kebutuhan

Berdasarkan Gambar 3.5, proses analisa kebutuhan dalam penelitian ini terdiri dari dua komponen utama, yaitu kebutuhan data dan metode sistem. Kebutuhan data mencakup jenis informasi yang diperlukan untuk membangun basis pengetahuan, serta proses pengumpulan data dari sumber yang valid. Hasil dari analisa kebutuhan ini menjadi landasan dalam pemilihan metode sistem, yaitu *Forward Chaining* dan *Certainty Factor*, sebagaimana dijelaskan pada subbab berikutnya.

3.3.1.1 Kebutuhan Data

1. Data

Dalam penelitian ini, data yang digunakan berasal dari *dataset* historis *maintenance* mesin Kraft yang mencakup data dari tahun 2004 hingga 2024.

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

Dataset historis ini berisi informasi terkait *Functional Architecture Hierarchy*, jenis (*Failure Type*), tindakan perbaikan (*Action Type*), serta *total downtime* mesin yang telah terdokumentasi selama dua dekade.

Data historis tersebut akan dimanfaatkan sebagai basis pengetahuan (*knowledge base*) dalam pengembangan sistem pakar. Selanjutnya, informasi yang terdapat pada data tersebut dianalisis untuk mengidentifikasi pola hubungan antara jenis kerusakan, akar penyebab (*Root Cause*), dan tindakan teknis yang dilakukan. Hasil analisis ini kemudian diformulasikan menjadi sejumlah aturan (*rule*) yang digunakan dalam proses inferensi. Metode yang digunakan untuk membangun sistem inferensi adalah *Forward Chaining*, karena mampu menelusuri fakta-fakta awal secara progresif untuk menghasilkan kesimpulan yang sesuai dengan kondisi aktual mesin. Dengan pendekatan ini, sistem pakar yang dibangun diharapkan mampu memberikan rekomendasi tindakan *maintenance* secara otomatis berdasarkan pola historis yang telah teridentifikasi, sehingga dapat membantu teknisi dalam melakukan diagnosis awal dan perencanaan perawatan secara lebih efektif.

2. Pengumpulan Data (*Collecting Data*)

Tahap awal pada desain sistem adalah proses pengumpulan data. Data yang dikumpulkan mencakup informasi mesin seperti sub system, device, component, jenis kerusakan, jenis tindakan, penyebab, total downtime, kebutuhan tenaga kerja, dan atribut pendukung lainnya yang dapat mempengaruhi *maintenance* mesin Kraft dari tahun 2004 – 2024. Sumber data berasal dari *Computerized Maintenance Management System* (CMMS) di Departemen *Production Facility Maintenance* (DM6000). Tahapan ini memiliki peran penting dalam memastikan bahwa sistem pakar dibangun di atas fondasi data yang valid terhadap kondisi mesin di lapangan.

3.3.1.2 Metode Sistem yang Digunakan

Penelitian ini akan menggunakan metode *Forward Chaining* sebagai pendekatan inferensi dalam sistem pakar, serta metode *Certainty Factor* sebagai teknik penanganan ketidakpastian dalam proses pengambilan keputusan. Penggabungan kedua metode ini memungkinkan sistem pakar menghasilkan

rekomendasi tindakan *maintenance* yang tidak hanya berbasis fakta historis, tetapi juga mempertimbangkan tingkat keyakinan terhadap kesimpulan yang diberikan.

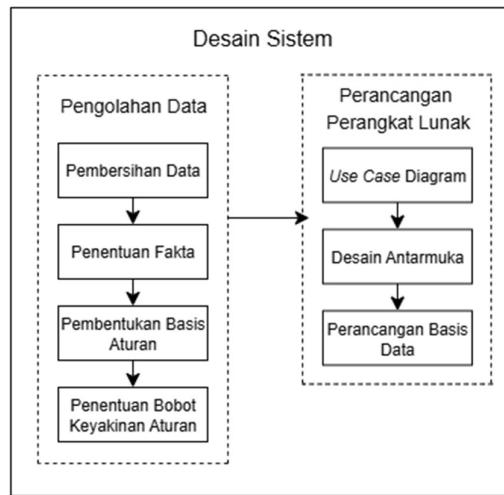
Basis aturan dalam sistem ini disusun melalui analisis terhadap data historis kerusakan mesin Kraft yang telah diklasifikasikan ke dalam struktur hierarki *Sub System - Device - Component*. Setiap baris data dianalisis untuk mengidentifikasi pola hubungan antara jenis kerusakan (*Failure Type*), akar penyebab (*Root Cause*), dan tindakan teknis yang dilakukan (*Action Type*). Informasi tambahan seperti *total downtime*, dan kebutuhan tenaga kerja juga digunakan untuk memperkuat dasar pengambilan keputusan dalam rekomendasi yang dihasilkan oleh sistem.

Dengan pendekatan ini, sistem pakar yang dibangun diharapkan mampu memberikan rekomendasi tindakan *maintenance* secara otomatis untuk membantu teknisi berdasarkan pola historis yang telah teridentifikasi, sehingga dapat mendukung teknisi dalam pengambilan keputusan yang lebih cepat dan akurat.

3.3.2 Desain Sistem

Tahap desain sistem merupakan proses lanjutan setelah analisa kebutuhan, yang bertujuan untuk mengolah data dan merancang struktur teknis sistem pakar. Pada tahap ini, dilakukan pengolahan data kuantitatif dan kualitatif untuk membentuk basis aturan, serta perancangan struktur sistem yang akan diimplementasikan secara digital.

Untuk memberikan gambaran menyeluruh mengenai alur dan komponen yang terlibat dalam proses desain sistem, berikut disajikan diagram tahapan desain sistem secara terstruktur.



Gambar 3. 6 Desain Sistem

Berdasarkan Gambar 3.6, disajikan tiga tahapan utama dalam desain sistem pakar, yaitu pengolahan data, perancangan struktur sistem, dan perancangan perangkat lunak. Ketiga tahapan ini saling berkesinambungan dan membentuk fondasi sistem pakar yang mampu memberikan rekomendasi teknis secara cepat dan akurat.

3.3.2.1 Pengolahan Data

Tahap ini merupakan fondasi awal dalam proses desain sistem pakar, di mana data mentah dari domain permasalahan diolah menjadi informasi yang terstruktur dan siap digunakan dalam proses inferensi. Proses ini mencakup beberapa langkah utama, yaitu sebagai berikut:

1. Pembersihan Data (*Data Cleaning*)

Setelah proses pengumpulan data, selanjutnya dilakukan proses pembersihan data untuk seleksi data sehingga data yang terbentuk memiliki performa yang baik. Performa data yang baik akan mempengaruhi hasil dari penelitian yang dilakukan. Langkah ini meliputi nilai kosong (*null*), duplikasi data, kesalahan format, dan deteksi anomali yang dapat mengganggu proses analisis. Data yang telah melalui proses *cleaning* akan digunakan sebagai dasar dalam pembentukan basis aturan dan pengembangan modul inferensi.

2. Penentuan Fakta

Fakta-fakta awal ditentukan berdasarkan data historis dan hasil wawancara dengan pakar. Fakta ini digunakan sebagai *input* dalam proses inferensi berbasis *Forward Chaining*, yang menelusuri kondisi awal untuk menghasilkan kesimpulan teknis.

3. Pembentukan Basis Aturan

Basis aturan merupakan inti dari sistem pakar yang berfungsi sebagai fondasi dalam proses inferensi. Aturan-aturan ini dibentuk berdasarkan data yang telah di bersihkan, serta didukung oleh pengetahuan domain yang diperoleh melalui wawancara dari pakar. Setiap aturan disusun dalam format *IF–THEN*, yang merepresentasikan hubungan antara kondisi (premis) dan tindakan atau kesimpulan (konklusi).

Aturan-aturan ini disimpan dalam basis pengetahuan dan diakses oleh modul inferensi untuk menghasilkan rekomendasi teknis yang sesuai dengan kondisi lapangan. Dengan struktur aturan yang konsisten dan relevan, sistem pakar dapat membantu teknisi dalam menentukan tindakan yang tepat secara lebih cepat dan terarah.

4. Penentuan Bobot Keyakinan Aturan

Setiap aturan yang dibentuk dilengkapi dengan bobot keyakinan (*certainty factor*) yang merepresentasikan tingkat kepercayaan terhadap kesimpulan yang dihasilkan. Bobot ini ditentukan berdasarkan validitas data historis dan tingkat konsistensi hasil wawancara dengan pakar, sehingga sistem dapat memberikan rekomendasi yang tidak hanya logis tetapi juga proporsional terhadap tingkat kepastian.

3.3.2.2 Perancangan Perangkat Lunak

Tahapan ini berfokus pada perancangan teknis sistem pakar yang akan dikembangkan, dimulai dari struktur sistem hingga perangkat lunak pendukung. Komponen utama yang dirancang meliputi *block diagram* sistem dan *flowchart* sistem. Setelah struktur sistem dirancang, proses dilanjutkan dengan perancangan perangkat lunak, dimulai dengan perancangan *data flow diagram*, *use case*

Jenni Febiyola Sari, 2025

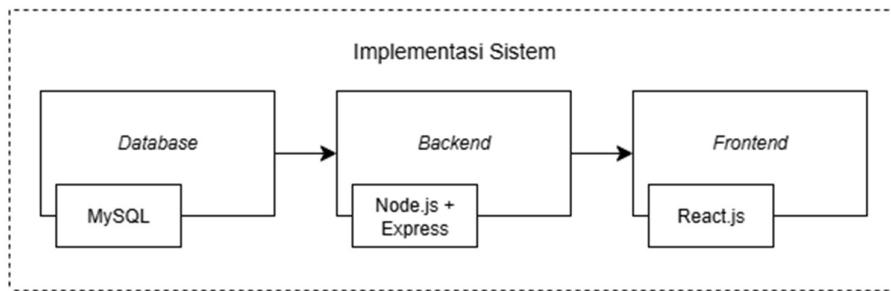
SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

diagram, mendesain antarmuka berbasis *website* menggunakan *Figma*, serta perancangan basis data yang dijelaskan lebih lanjut pada subbab berikutnya. Seluruh rancangan ini menjadi acuan dalam tahap implementasi sistem yang dibahas pada bab selanjutnya.

3.3.3 Implementasi Sistem

Tahap implementasi dilakukan setelah seluruh proses perancangan sistem dan perangkat lunak selesai disusun secara menyeluruh. Pada fase ini, rancangan yang sebelumnya bersifat konseptual mulai diwujudkan dalam bentuk sistem yang dapat dijalankan dan diuji. Struktur implementasi sistem pakar berbasis *website* ditunjukkan pada Gambar 3.7 berikut:



Gambar 3.7 Implementasi Sistem

Berdasarkan Gambar 3.7, arsitektur implementasi sistem pakar berbasis *website* yang dibangun dengan pendekatan *client-server*. Diagram ini terdiri dari tiga komponen utama yang saling terhubung: *frontend*, *backend*, dan *database*.

Pada bagian *database*, sistem menggunakan MySQL sebagai *Database Management System (DBMS)*. *Database* ini menyimpan berbagai informasi yang menjadi dasar kerja sistem pakar, seperti data gejala, komponen, aturan berbasis *if-then*, serta nilai pendukung lainnya. Dengan basis data yang terstruktur, proses pencarian dan pengolahan data dapat dilakukan secara cepat dan akurat.

Bagian *backend* dikembangkan menggunakan Node.js dengan *framework* Express.js. *Backend* berperan sebagai pusat pengolahan logika sistem pakar, yang menghubungkan antara basis data dan lapisan antarmuka. *Backend* mengelola komunikasi data melalui API sehingga integrasi antar komponen dapat berjalan secara konsisten dan terukur. Implementasi *backend* juga mencakup penerapan

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

mesin inferensi untuk mengolah aturan yang tersimpan dalam *database*. Struktur modular pada backend memudahkan proses debugging dan pengembangan berkelanjutan sesuai kebutuhan sistem.

Sementara itu, *frontend* dibangun menggunakan React.js sebagai antarmuka. Implementasi pada lapisan ini berfokus pada penyajian hasil pemrosesan dari *backend* dalam bentuk yang terstruktur dan mudah dipahami. Selain itu, penggunaan React.js memberikan fleksibilitas dalam pengembangan tampilan sehingga sistem pakar dapat disajikan secara interaktif dan responsif.

Dengan integrasi yang terstruktur antara basis data, *backend*, dan *frontend*, sistem pakar ini mampu menjalankan proses inferensi secara efisien dan menyajikan hasil rekomendasi yang akurat kepada pengguna. Arsitektur yang diterapkan tidak hanya mendukung skalabilitas dan kemudahan pengembangan, tetapi juga memastikan bahwa setiap komponen berfungsi secara sinergis dalam mendukung pengambilan keputusan berbasis pengetahuan.

3.3.4 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur utama berfungsi sesuai dengan rancangan teknis. Metode yang digunakan adalah *black box testing*, yaitu pengujian berdasarkan validasi *input* dan *output* tanpa melihat struktur internal kode. Pengujian dilakukan secara lokal dan melalui *endpoint* ngrok untuk mensimulasikan kondisi akses jarak jauh.

Hasil pengujian menunjukkan bahwa sistem dapat menangani berbagai kombinasi *input* dan menghasilkan *output* yang sesuai dengan basis pengetahuan yang telah dimasukkan. Seluruh fitur inti, termasuk proses inferensi dan penyajian hasil diagnosis, berjalan sesuai dengan ekspektasi. Pengujian ini bersifat teknis dan dilakukan oleh pakar serta *staff maintenance planning and control*, bukan oleh pengguna akhir. Fokus utama pengujian sistem ini adalah validasi fungsionalitas sistem, bukan aspek *usability* atau pengalaman pengguna.

3.3.5 Evaluasi Sistem

Evaluasi sistem merupakan tahap akhir dalam proses pengembangan yang bertujuan untuk menilai apakah sistem yang dibangun telah sesuai dengan tujuan **Jenni Febiyola Sari, 2025**

**SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI
TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA**

penelitian. Evaluasi dilakukan berdasarkan *Usability Testing*, yang masing-masing mengukur kesesuaian fungsi sistem serta kenyamanan penggunaan dari perspektif pengguna akhir. Tahap ini menjadi krusial untuk memastikan bahwa sistem tidak hanya bekerja secara teknis, tetapi juga relevan secara praktis bagi pengguna.

Melalui evaluasi ini, dilakukan analisis terhadap efektivitas sistem dalam menjalankan fungsinya, serta identifikasi terhadap bagian-bagian yang belum optimal. Jika ditemukan ketidaksesuaian pada evaluasi ini, maka akan dilakukan identifikasi terhadap bagian dari sistem yang perlu diperbaiki, beserta penyebabnya.

Evaluasi ini mempertimbangkan aspek pengalaman pengguna sebagai indikator keberhasilan sistem. Hal ini mencakup kemudahan navigasi, kejelasan antarmuka, dan pemahaman terhadap istilah teknis yang digunakan. Dengan demikian, evaluasi tidak hanya berfungsi sebagai validasi akhir, tetapi juga sebagai dasar untuk pengembangan sistem yang lebih adaptif dan berkelanjutan.

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui dua pendekatan utama, yaitu dengan metode dokumentasi dan wawancara dengan pakar. Metode ini dipilih karena data yang dibutuhkan merupakan data historis yang telah terdokumentasi secara digital, sehingga relevan untuk penelitian sistem pakar berbasis *Forward Chaining*. Dokumentasi digital memungkinkan proses ekstraksi dan analisis data dilakukan secara sistematis dan efisien, serta mendukung pembentukan basis pengetahuan yang terstruktur.

Selain dokumentasi, penelitian ini juga menggunakan metode wawancara dengan pakar teknis yang memiliki pengalaman langsung dalam pengoperasian dan perawatan mesin Kraft. Wawancara ini bertujuan untuk memperoleh pemahaman mendalam mengenai cara kerja mesin, pola kerusakan yang umum terjadi, serta tindakan mitigasi yang biasanya dilakukan.

Kombinasi antara dokumentasi dan wawancara memungkinkan sistem pakar yang dibangun memiliki landasan data yang kuat sekaligus mempertimbangkan pengetahuan praktis dari lapangan. Dengan demikian, sistem dapat memberikan rekomendasi yang tidak hanya berbasis data, tetapi juga mencerminkan pengalaman

dan intuisi teknis dari para ahli.

3.5 Sumber dan Deskripsi Data

Untuk membangun basis pengetahuan yang akurat dan relevan dalam sistem pakar, diperlukan data historis yang mencerminkan kondisi nyata di lapangan. Oleh karena itu, tahap awal dalam pengembangan sistem ini dimulai dengan identifikasi sumber data peninjauan struktur, atribut, dan konteks operasional yang terkandung di dalamnya.

3.5.1 Sumber Data

Penelitian ini menggunakan data historis *maintenance* mesin Kraft yang diperoleh dari *Computerized Maintenance Management System* (CMMS) di Departemen *Production Facility Maintenance* (DM6000). Data tersebut mencakup periode dari tahun 2004 – 2024. Setiap entri dalam CMMS dilengkapi dengan kategori kerusakan, dan tindakan perbaikan yang dilakukan, sehingga memungkinkan analisis yang komprehensif terhadap pola kerusakan dan efektivitas tindakan *maintenance*.

3.5.2 Deskripsi Data

Penelitian ini menggunakan data historis *maintenance* mesin Kraft untuk membangun sistem pakar berbasis *Forward Chaining* dan *Certainty Factor* untuk rekomendasi tindakan *maintenance*. *Dataset* disimpan dalam format Excel (.xlsx) dengan ukuran sebesar 115KB. *Dataset* ini terdiri dari 477 baris data, di mana setiap baris merepresentasikan satu kejadian atau entri *maintenance* yang terdokumentasi secara sistematis. Data historis mesin Kraft dirincikan pada Tabel 3.1 berikut:

Tabel 3. 1 Deskripsi Dataset

No.	Nama Atribut	Tipe Data	Deskripsi
1	ResponsibleCode	String	Kode divisi <i>maintenance</i>
2	Codification	String	Kode klasifikasi mesin
3	Description	String	Jenis mesin
4	Manufacturer	String	Nama manufaktur mesin

No.	Nama Atribut	Tipe Data	Deskripsi
5	FacType	String	Nama mesin
6	WorkOrderNo	String	Nomor work order <i>maintenance</i>
7	IssuedDate	Date	Tanggal awal kerusakan mesin
8	IssuedTime	Time	Waktu awal kerusakan mesin
9	Status	String	Status mesin
10	DeliveryDate	Date	Tanggal akhir kerusakan mesin
11	DeliveryTime	Time	Waktu akhir kerusakan mesin
12	ProblemDesc	String	Deskripsi kerusakan yang terjadi
13	ActionDescription	String	Jenis tindakan <i>maintenance</i>
14	ActionTakenDesc	String	Deskripsi detail tindakan yang dilakukan
14	Penyebab	String	Penyebab kerusakan
15	FailureType	String	Jenis kerusakan
16	SubSystem	String	Sub System Mesin
17	TotDownTime	Float	Total downtime (jam)
18	ManPower	Integer	Jumlah teknisi

Tabel 3.1 menjelaskan atribut beserta tipe data dan deskripsinya. Meskipun demikian, tidak semua atribut akan dimanfaatkan menjadi fakta dalam basis pengetahuan sistem pakar. Hanya atribut-atribut yang memiliki nilai relevansi terhadap gejala yang terdeteksi yang akan dikonversi menjadi fakta. Proses seleksi ini dilakukan untuk memastikan bahwa basis pengetahuan yang dibangun bersifat efisien, terfokus, dan mendukung proses inferensi secara optimal.

3.6 Teknik Pengolahan Data

Teknik pengolahan data merupakan tahapan krusial dalam pengembangan sistem pakar, karena menentukan kualitas basis pengetahuan yang akan digunakan dalam proses inferensi. Proses ini dilakukan secara bertahap, dimulai dari pembersihan data mentah hingga penetapan bobot keyakinan pada setiap aturan yang dibentuk.

3.6.1 Pembersihan Data (*Data Cleaning*)

Tahap pertama dalam pengolahan data adalah proses pembersihan atau *data cleaning*, yang bertujuan untuk memastikan bahwa data historis yang diperoleh memiliki kualitas dan integritas yang memadai untuk digunakan dalam proses analisis dan pembentukan basis pengetahuan. Data mentah sering kali mengandung berbagai permasalahan teknis, seperti nilai kosong (*null values*), entri duplikat, kesalahan format, dan anomali operasional yang tidak sesuai dengan pola kerja normal mesin.

Setelah proses pembersihan dilakukan, data historis yang semula berjumlah 477 baris mengalami penyusutan karena penghapusan entri yang tidak valid, duplikat, atau mengandung nilai kosong yang tidak dapat diperbaiki. Hasil akhir dari proses *data cleaning* menghasilkan sebanyak 244 baris data bersih yang siap digunakan dalam tahap analisis dan pembentukan basis pengetahuan.

Selain pengurangan jumlah baris, proses pembersihan juga disertai dengan penyesuaian struktur data, termasuk penambahan beberapa atribut baru yang diperlukan untuk mendukung proses inferensi. Penambahan atribut ini dilakukan berdasarkan hasil analisis kebutuhan sistem pakar serta integrasi informasi dari dokumen teknis pendukung, seperti dokumen *maintenance instruction*. Tujuannya adalah agar data yang digunakan tidak hanya bersih secara sintaksis, melainkan juga mengandung informasi yang cukup dan terstruktur untuk mendukung proses inferensi berbasis aturan.

Data yang telah dibersihkan dan direstrukturisasi kemudian disusun dalam bentuk *dataset* final yang digunakan sebagai dasar dalam penentuan fakta dan pembentukan basis aturan. Struktur lengkap dari *dataset* hasil pembersihan

ditampilkan pada Tabel 3.2 sebagai berikut:

Tabel 3. 2 Deskripsi *Dataset* setelah di *Cleaning*

No.	Nama Atribut	Tipe Data	Deskripsi
1	ResponsibleCode	String	Kode divisi <i>maintenance</i>
2	Codification	String	Kode klasifikasi mesin
3	Description	String	Jenis mesin
4	Manufacturer	String	Nama manufaktur mesin
5	FacType	String	Nama mesin
6	WorkOrderNo	String	Nomor <i>work order maintenance</i>
7	IssuedDate	Date	Tanggal awal kerusakan mesin
8	IssuedTime	Time	Waktu awal kerusakan mesin
9	Status	String	Status mesin
10	DeliveryDate	Date	Tanggal akhir kerusakan mesin
11	DeliveryTime	Time	Waktu akhir kerusakan mesin
12	ProblemDesc	String	Deskripsi kerusakan yang terjadi
13	ActionType	String	Jenis tindakan <i>maintenance</i>
14	ActionTakenDesc	String	Deskripsi detail tindakan yang dilakukan
14	RootCause	String	Penyebab kerusakan
15	FailureType	String	Jenis kerusakan
16	SubSystem	String	<i>Sub System</i> Mesin
17	Device	String	<i>Device</i> Mesin
18	Component	String	<i>Component</i> Mesin

No.	Nama Atribut	Tipe Data	Deskripsi
19	TotDownTime	Float	<i>Total downtime (jam)</i>
20	ManPower	Integer	Jumlah teknisi

Berdasarkan struktur data yang telah dibersihkan dan ditampilkan pada Tabel 3.2, terlihat bahwa atribut Device dan Component ditambahkan pada *dataset* yang telah di bersihkan. Device merupakan turunan dari atribut Sub System dan Component merupakan turunan dari atribut Device. Kedua atribut tersebut ditambahkan dari dokumen teknis pendukung dan hasil wawancara bersama *staff maintenance planning and control*. Tahap selanjutnya adalah melakukan identifikasi fakta-fakta kunci yang akan digunakan sebagai dasar dalam pembentukan basis pengetahuan sistem pakar.

3.6.2 Penentuan Fakta

Setelah data dibersihkan, tahap berikutnya adalah penentuan fakta, yaitu proses identifikasi kondisi-kondisi awal yang akan digunakan sebagai *input* dalam sistem pakar. Fakta merupakan elemen penting dalam metode inferensi *Forward Chaining*, karena sistem akan menelusuri fakta-fakta ini untuk menghasilkan kesimpulan teknis yang relevan.

Fakta ditentukan melalui dua pendekatan utama. Pertama, dilakukan ekstraksi informasi dari data historis yang telah melalui proses pembersihan. Informasi ini mencakup jenis kerusakan yang sering terjadi, frekuensi kejadian, waktu respons terhadap perawatan, serta tindakan korektif yang dilakukan. Kedua, fakta divalidasi dan dilengkapi melalui wawancara dengan pakar teknis yang memiliki pengalaman langsung dalam menangani mesin Kraft. Pendekatan ini memastikan bahwa fakta yang digunakan tidak hanya bersifat historis, tetapi juga mencerminkan pengetahuan praktis yang aktual.

Fakta-fakta yang telah ditentukan tidak hanya digunakan sebagai premis dalam proses inferensi, tetapi juga menjadi landasan utama dalam pembentukan rekomendasi teknis. Dengan menelusuri kondisi awal yang tercermin dalam fakta, sistem pakar dapat menyimpulkan tindakan paling sesuai berdasarkan pola historis.

Fakta-fakta yang telah ditentukan dari hasil ekstraksi data historis dan validasi pakar kemudian diklasifikasikan dan disusun dalam bentuk terstruktur. Tabel 3.3 berikut menyajikan daftar fakta yang digunakan sebagai premis dalam proses inferensi sistem pakar.

Tabel 3. 3 Atribut yang dipilih menjadi Fakta

No.	Atribut <i>dataset</i> yang dipilih menjadi Fakta
1	Sub System
2	Device
3	Component
4	Failure Type
5	Root Cause
6	Action Type

Dengan tersusunnya daftar fakta pada Tabel 3.3, sistem pakar memiliki fondasi awal yang kuat untuk menjalankan proses inferensi berbasis *Forward Chaining*. Fakta-fakta ini akan berperan sebagai premis dalam penelusuran aturan, sehingga memungkinkan sistem menghasilkan rekomendasi teknis yang sesuai dengan kondisi operasional aktual.

3.6.3 Pembentukan Basis Aturan

Tahap ketiga adalah pembentukan basis aturan, yang merupakan inti dari sistem pakar. Basis aturan berfungsi sebagai representasi formal dari pengetahuan domain, disusun dalam format logika *IF–THEN* menggunakan metode *Forward Chaining* yang menggambarkan hubungan antara kondisi dan tindakan. Aturan-aturan ini memungkinkan sistem untuk melakukan inferensi secara sistematis dan menghasilkan rekomendasi teknis yang sesuai dengan kondisi lapangan.

Pembentukan aturan dilakukan dengan mengidentifikasi pola hubungan antara fakta-fakta yang telah ditentukan dan tindakan teknis yang relevan. Setiap aturan dirancang untuk menangkap pola hubungan antara kondisi awal (premis) dan tindakan teknis yang sesuai (konklusi), sehingga sistem pakar dapat memberikan rekomendasi yang relevan terhadap situasi operasional mesin Kraft. Berikut ini

adalah beberapa contoh aturan yang dibentuk berdasarkan fakta-fakta yang telah ditentukan, sebagaimana ditampilkan pada Tabel 3.4 berikut:

Tabel 3. 4 Aturan yang Terbentuk dari Fakta-Fakta

No.	Aturan
1	IF [Sub System = Hydraulic] AND [Device = Actuator Hydraulic] AND [Component = Accumulator] AND [FailureType = TRBL – Trouble] AND [RootCause = Rusak] THEN [ActionType = Repairing] AND [Component = Accumulator]
2	IF [Sub System = Motor Blower] AND [Device = Motor Blower] AND [Component = Bearing Motor] AND [FailureType = FTRB – Function Imperfect] AND [RootCause = Tidak Berfungsi] THEN [ActionType = Replacing] AND [Component = Bearing Motor]
3	IF [Sub System = Motor Blower] AND [Device = Motor Blower] AND [Component = Motor Blower] AND [FailureType = TRBL - Trouble] AND [RootCause = Terbakar] THEN [ActionType = Replacing] AND [Component = Motor Blower]

Aturan-aturan ini kemudian disimpan dalam basis pengetahuan dan digunakan oleh modul inferensi untuk menghasilkan rekomendasi tindakan *maintenance*. Basis aturan ini menjadi fondasi logika sistem pakar dalam menelusuri kondisi dan menentukan tindakan yang paling tepat secara otomatis.

3.6.4 Penetapan Bobot Keyakinan Aturan

Tahap berikutnya dalam pembangunan sistem adalah penetapan bobot keyakinan atau *certainty factor* pada setiap aturan yang telah dibentuk. Setelah fakta ditentukan dan aturan disusun berdasarkan pola hubungan antara kondisi dan tindakan, sistem pakar memerlukan mekanisme untuk merepresentasikan tingkat keyakinan terhadap setiap kesimpulan yang dihasilkan. Di sinilah peran nilai CF menjadi krusial, karena memberikan dimensi probabilistik dalam proses inferensi.

Bobot ini merepresentasikan tingkat kepercayaan pakar terhadap validitas hubungan antara premis dan kesimpulan dalam setiap aturan. Penetapan nilai

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

certainty factor dilakukan setelah seluruh aturan dalam basis pengetahuan selesai dirumuskan. Nilai ini tidak diberikan pada fakta secara individual, melainkan pada hubungan logis yang dibentuk oleh aturan. Dengan kata lain, bobot keyakinan merepresentasikan seberapa kuat pakar meyakini bahwa kombinasi kondisi tertentu akan menghasilkan tindakan atau kesimpulan yang tepat.

Penetapan CF dilakukan secara kualitatif berdasarkan penilaian pakar, kemudian dikonversi ke dalam skala kuantitatif antara 0 sampai 1. Misalnya, jika pakar menyatakan sangat yakin terhadap suatu hubungan, maka CF dapat ditetapkan sebesar 0.8 atau lebih. Sebaliknya, jika hubungan tersebut dianggap lemah atau masih meragukan, maka CF dapat bernilai lebih rendah, seperti 0.4 atau 0.6. Setiap nilai CF aturan dalam basis pengetahuan disusun dalam format:

IF [fakta-fakta] THEN [kesimpulan] WITH CF = [nilai]

Nilai CF ini digunakan oleh modul inferensi untuk menentukan tingkat keyakinan terhadap kesimpulan yang dihasilkan. Jika satu set fakta memicu lebih dari satu aturan, maka sistem akan menghasilkan beberapa kesimpulan, masing-masing dengan nilai CF.

Perlu ditekankan bahwa meskipun beberapa aturan memiliki premis atau fakta yang identik, kesimpulan yang dihasilkan dapat berbeda secara sah. Hal ini terjadi karena setiap aturan merepresentasikan pendekatan pakar yang berbeda terhadap kondisi yang sama. Dengan kata lain, fakta yang sama dapat memicu lebih dari satu aturan, masing-masing dengan struktur logis dan bobot keyakinan tersendiri. Perbedaan kesimpulan bukanlah bentuk kontradiksi, melainkan refleksi dari fleksibilitas sistem dalam menangani ketidakpastian dan menyediakan alternatif tindakan yang relevan secara teknis.

Masing-masing aturan merepresentasikan pendekatan pakar yang berbeda terhadap kondisi yang sama, sehingga menghasilkan tindakan yang berbeda pula. Perbedaan ini bukanlah bentuk kontradiksi, melainkan cerminan dari fleksibilitas sistem dalam menangani ketidakpastian dan menyediakan alternatif solusi teknis yang sah. Setiap kesimpulan diberikan bobot keyakinan (*certainty factor*) sesuai

dengan tingkat kepercayaan pakar terhadap validitas hubungan antara premis dan tindakan. Di sisi lain, terdapat pula aturan yang memiliki nilai CF sebesar 1.0, yang menunjukkan keyakinan penuh terhadap satu kesimpulan tunggal tanpa alternatif tindakan lain. Kedua jenis aturan ini baik yang menghasilkan beberapa kesimpulan dengan CF berbeda maupun yang menghasilkan satu kesimpulan pasti ditampilkan pada Tabel 3.5 berikut:

Tabel 3. 5 Bobot Nilai *Certainty Factor* pada Aturan

No.	Aturan	Nilai CF	Keterangan
1	IF [Sub System = Hydraulic] AND [Device = Valve Hydraulic] AND [Component = Valve One Way] AND [FailureType = TRBL – Trouble] AND [RootCause = Rusak] THEN [ActionType = Repairing] AND [Component = Valve One Way]	0.8	Nilai <i>certainty factor</i> sebesar 0.8 merepresentasikan tingkat keyakinan pakar yang tinggi terhadap inferensi tindakan <i>repairing</i>
2	IF [Sub System = Hydraulic] AND [Device = Valve Hydraulic] AND [Component = Valve One Way] AND [FailureType = TRBL – Trouble] AND [RootCause = Rusak] THEN [ActionType = Replacing] AND [Component = Valve One Way]	0.7	Nilai <i>certainty factor</i> sebesar 0.7 merepresentasikan tingkat keyakinan pakar yang cukup tinggi terhadap inferensi tindakan <i>replacing</i>
3	IF [Sub System = Motor Blower] AND [Device = Motor Blower] AND [Component = Motor Blower] AND [FailureType = TRBL - Trouble] AND [RootCause = Terbakar] THEN	1.0	Nilai <i>certainty factor</i> sebesar 0.8 merepresentasikan tingkat keyakinan pakar yang sangat

No.	Aturan	Nilai CF	Keterangan
	[ActionType = Replacing] AND [Component = Motor Blower]		tinggi terhadap inferensi tindakan <i>replacing</i>

Dengan adanya penetapan nilai CF, sistem pakar mampu menangani ketidakpastian secara terukur, menjadikan rekomendasi teknis lebih realistik dan kontekstual.

3.6.5 Penambahan Saran

Selain aturan utama yang menentukan tindakan teknis, sistem pakar juga dilengkapi dengan saran yang memberikan estimasi kuantitatif terhadap *total downtime* dan kebutuhan *man power* berdasarkan pola historis. Aturan ini tidak bersifat deterministik, melainkan berbasis agregasi data dari kasus-kasus serupa yang telah terjadi sebelumnya.

Setiap saran dirancang untuk melengkapi aturan utama dengan informasi tambahan yang bersifat prediktif. Formatnya mengikuti struktur:

Tabel 3. 6 Saran berdasarkan Aturan

No.	Aturan	Saran
1	IF [Sub System = Hydraulic] AND [Device = Actuator Hydraulic] AND [Component = Accumulator] AND [FailureType = TRBL – Trouble] AND [RootCause = Rusak] THEN [ActionType = Repairing] AND [Component = Accumulator]	Total Downtime = 56.5 jam Man Power = 7 orang
2	IF [Sub System = Motor Blower] AND [Device = Motor Blower] AND [Component = Bearing Motor] AND [FailureType = FTRB – Function Imperfect] AND [RootCause = Tidak Berfungsi]	Total Downtime = 44 jam Man Power = 2 orang

No.	Aturan	Saran
	THEN [ActionType = Replacing] AND [Component = Bearing Motor]	
3	IF [Sub System = Motor Blower] AND [Device = Motor Blower] AND [Component = Motor Blower] AND [FailureType = TRBL - Trouble] AND [RootCause = Terbakar] THEN [ActionType = Replacing] AND [Component = Motor Blower]	Total Downtime = 29 jam Man Power = 6 orang

3.7 Perancangan Perangkat Lunak

Perancangan perangkat lunak dalam penelitian ini bertujuan untuk membangun sistem pakar yang terstruktur, modular, dan mudah dikembangkan. Proses perancangan mencakup pengembangan *data flow diagram*, antarmuka pengguna, basis data, serta pemrograman perangkat lunak. Setiap komponen dirancang untuk mendukung alur inferensi dan penyajian hasil secara efisien, dengan mempertimbangkan kebutuhan fungsional dan kenyamanan pengguna akhir.

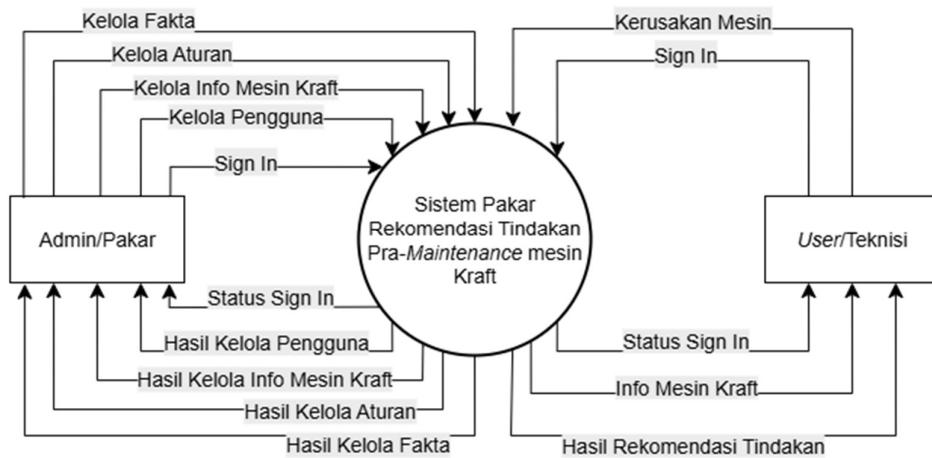
3.7.1 Perancangan *Data Flow Diagram*

Data Flow Diagram (DFD) adalah sebuah diagram yang digunakan untuk menggambarkan suatu sistem bekerja, yang dipresentasikan dalam bentuk aliran data agar lebih mudah dipahami. DFD digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan.

Secara umum, DFD terbagi atas 3 diagram yaitu *Context Diagram* (DFD Level 0), DFD Level 1, dan DFD Level 2 yang menjelaskan proses-proses lebih rinci. Berikut adalah penjabaran *context diagram* Sistem Pakar berbasis *Forward Chaining* dan *Certainty Factor* untuk Rekomendasi Tindakan *Maintenance* mesin Kraft.

3.7.1.1 Context Diagram (DFD Level 0)

DFD Level 0 adalah sebuah diagram dimana pada tahap ini menggambarkan seluruh proses yang terjadi dalam suatu sistem dan menggambarkan interaksi antara entitas dan proses. *Data flow diagram* Level 0 dapat ditunjukkan pada Gambar 3.8 berikut:



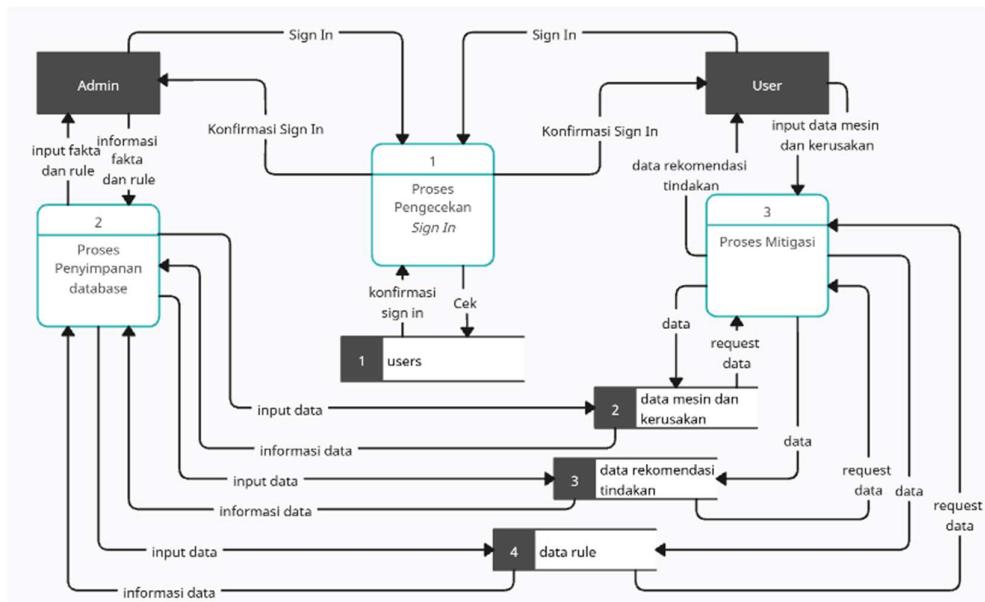
Gambar 3. 8 Context Diagram (DFD Level 0)

Berdasarkan Gambar 3.8, terdapat interaksi antara *user* dan *admin* dengan sistem pakar. Pada sisi Admin, peran utamanya adalah melakukan pengelolaan terhadap berbagai data yang menjadi basis pengetahuan sistem, seperti data mesin, data kerusakan, data aturan, serta data rekomendasi tindakan. Seluruh data tersebut kemudian dimanfaatkan oleh mesin inferensi sebagai bahan dalam proses penalaran untuk menghasilkan rekomendasi *maintenance* secara otomatis.

Sementara itu, pada sisi *user*, perannya adalah sebagai penerima informasi yang disediakan oleh sistem. *User* dapat mengakses informasi terkait mesin, detail kerusakan, serta hasil rekomendasi tindakan mitigasi yang dihasilkan oleh mesin inferensi. Dengan demikian, interaksi antara *admin* dan *user* melalui sistem pakar ini membentuk siklus kerja yang saling melengkapi, di mana *admin* menjaga kelengkapan dan keakuratan data, sedangkan *user* memanfaatkan hasil rekomendasi untuk mendukung proses pengambilan keputusan *maintenance*.

3.7.1.2 Data Flow Diagram Level 1

DFD Level 1 adalah sebuah diagram dimana pada tahap ini akan menjelaskan lebih detail mengenai proses yang terjadi pada DFD Level 0. *Data flow diagram* Level 1 dapat ditunjukkan pada Gambar 3.9 berikut:



Gambar 3.9 DFD Level 1

Berdasarkan Gambar 3.9, terdapat tiga proses utama yang membentuk arsitektur sistem pakar rekomendasi tindakan *maintenance* mesin Kraft, yaitu sebagai berikut:

1. Proses Pengecekan *Sign In*

Proses pengecekan *sign in* berfungsi untuk memverifikasi identitas dan hak akses dari entitas *Admin* maupun *User*. Proses ini menghasilkan status autentikasi yang menentukan aksesibilitas terhadap proses-proses berikutnya.

2. Proses Penyimpanan *Database*

Proses penyimpanan *database* mencakup pengelolaan data oleh Admin, meliputi data pengguna, data mesin dan kerusakan, data aturan inferensi, serta data rekomendasi tindakan. Seluruh data ini disimpan dalam entitas penyimpanan terpisah dan digunakan sebagai basis pengetahuan dalam proses *reasoning*.

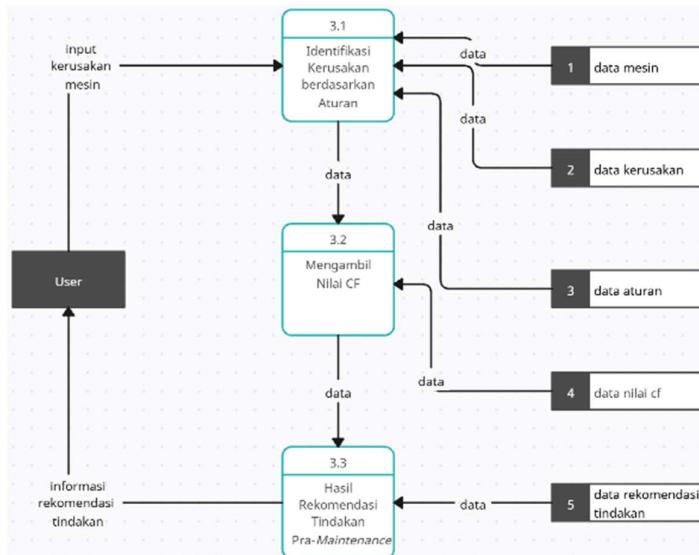
3. Proses Mitigasi

Proses mitigasi dijalankan ketika *User input* data kerusakan mesin. Sistem pakar akan mengakses fakta dan aturan yang relevan, melakukan penalaran berbasis *forward chaining*, serta menghitung tingkat kepastian menggunakan *certainty factor*. Hasil dari proses ini berupa rekomendasi tindakan mitigasi yang dikirimkan kembali kepada *User*.

Aliran data antar proses dan entitas penyimpanan menunjukkan keterkaitan logis antara *input*, proses *reasoning*, dan *output* sistem. Struktur ini memastikan bahwa setiap rekomendasi yang dihasilkan bersumber dari basis pengetahuan yang telah divalidasi, serta mendukung pengambilan keputusan teknis secara efisien dan terukur.

3.7.1.3 Data Flow Diagram Level 2

Data Flow Diagram (DFD) Level 2 merupakan dekomposisi lebih lanjut dari proses-proses utama yang ditampilkan pada DFD Level 1. Pada level ini, sistem pakar dijabarkan secara rinci hingga ke sub-proses inferensi, pengambilan nilai *certainty factor*, dan penyusunan rekomendasi tindakan. Tujuannya adalah untuk memvisualisasikan logika *reasoning* sistem secara modular dan terstruktur. *Data flow diagram* Level 2 dapat ditunjukkan pada Gambar 3.10 berikut:



Gambar 3. 10 DFD Level 2

Berdasarkan Gambar 3.10, terdapat tiga sub-proses utama yang membentuk alur inferensi sistem pakar, yaitu sebagai berikut:

1. Proses Identifikasi Kerusakan Berdasarkan Aturan

Sub-proses ini menerima *input* berupa data kerusakan mesin dari *User*. Sistem akan mencocokkan data tersebut dengan basis pengetahuan yang terdiri dari data mesin, data kerusakan, dan data aturan. Hasil dari proses ini adalah pemetaan premis yang valid untuk proses inferensi selanjutnya.

2. Proses Pengambilan Nilai *Certainty Factor* (CF)

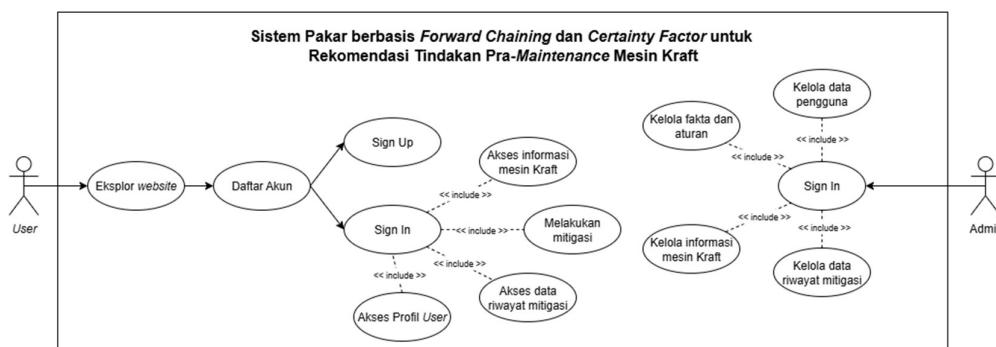
Setelah premis dikenali, sistem akan mengeksekusi *rule* yang relevan dan menghitung nilai *certainty factor* berdasarkan bobot keyakinan pakar terhadap masing-masing premis. Nilai CF ini digunakan untuk mengukur tingkat kepastian terhadap inferensi yang dihasilkan. *Output* dari proses ini adalah data nilai CF yang akan digunakan dalam rekomendasi.

3. Proses Penyusunan Hasil Rekomendasi Tidakan *Maintenance*

Sub-proses ini menggabungkan hasil *reasoning* dan nilai CF untuk menghasilkan rekomendasi tindakan mitigasi. *Output* berupa data rekomendasi tindakan dikirimkan kembali kepada *User* sebagai hasil akhir dari proses penalaran sistem pakar.

3.7.2 Use Case Diagram

Use case diagram digunakan untuk memodelkan interaksi antara aktor dan sistem dalam konteks operasional yang relevan. Rancangan *use case diagram* pada sistem ini sebagai berikut:



Gambar 3. 11 Use Case Diagram

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

Berdasarkan Gambar 3.11, *use case diagram* pada sistem ini memiliki dua peran dalam hal akses *Website*, yaitu sebagai berikut:

1. *Admin* (Administrator)

Admin merupakan seseorang yang dapat mengelola *Website* pada sistem ini, seperti IT atau manajerial yang bertanggung jawab atas konfigurasi, pengaturan pengguna, serta pengelolaan basis pengetahuan. *Admin* memiliki akses terhadap seluruh fitur yang tersedia dalam sistem, mulai dari proses autentikasi seperti *Sign In*, *Sign Up*, dan *Sign Out*, hingga pengelolaan email dan peran pengguna. *Admin* juga bertanggung jawab dalam mengelola fakta dan aturan yang menjadi inti dari sistem pakar, serta memiliki akses terhadap informasi mesin Kraft dan data histori mitigasi rekomendasi.

2. *User* (Pengguna)

User merupakan orang yang dapat mengakses *Website* pada sistem ini, seperti karyawan magang. *User* memiliki akses terhadap seluruh fitur yang tersedia dalam sistem, mulai dari proses autentikasi seperti *Sign In*, *Sign Up*, dan *Sign Out*. *User* dapat mengakses informasi mesin Kraft, melihat data histori mitigasi rekomendasi, serta melakukan mitigasi berdasarkan hasil analisis sistem pakar.

3.7.3 *Wireframe* Antarmuka Pengguna

Perancangan antarmuka pengguna (*User Interface*) bertujuan untuk menciptakan pengalaman pengguna yang intuitif dan efisien dalam menggunakan sistem virtual. *Wireframe* berfungsi sebagai *blueprint* visual yang menggambarkan tata letak dan hierarki elemen dalam antarmuka sebelum tahap implementasi. Rancangan ini mempertimbangkan prinsip *usability* dan *user experience* agar pengguna dapat dengan mudah memahami dan mengoperasikan sistem. Antarmuka pengguna dirancang menggunakan aplikasi Figma pada sistem ini, yaitu sebagai berikut:

1. *Landing Page*

Wireframe halaman awal atau *landing page* merupakan rancangan awal antarmuka pengguna yang berfungsi sebagai titik masuk utama ke dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. *Wireframe landing*

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI
TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

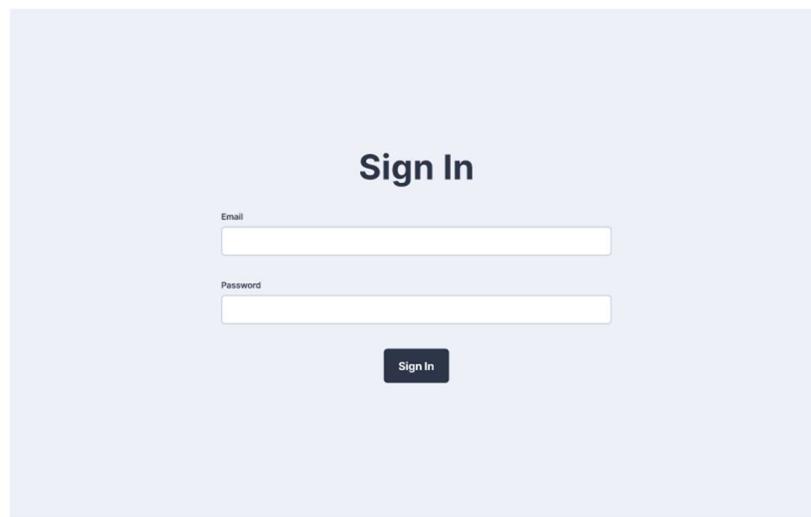
page dapat dilihat pada Gambar 3.12 berikut:



Gambar 3. 12 *Wireframe Landing Page*

2. Halaman Sign In

Wireframe halaman *Sign In* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk memfasilitasi proses autentikasi pengguna sebelum mengakses fitur sistem pakar. *Wireframe* halaman *Sign In* dapat dilihat pada Gambar 3.13 berikut:



Gambar 3. 13 *Wireframe Halaman Sign In*

3. Halaman *Sign Up*

Wireframe halaman *Sign Up* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk memfasilitasi proses pendaftaran pengguna baru ke dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Desain ini bertujuan untuk menyusun alur registrasi secara logis dan efisien, dengan penempatan elemen *input* yang mudah dipahami dan dijangkau. *Wireframe* halaman *Sign Up* dapat dilihat pada Gambar 3.14 berikut:

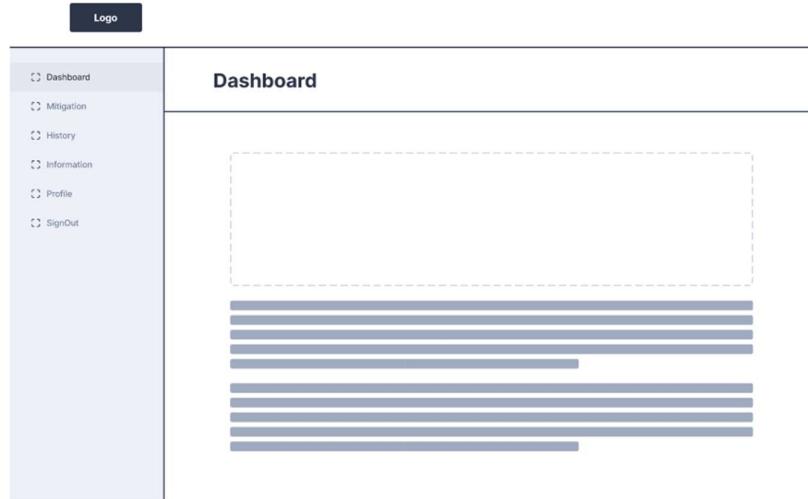
The wireframe shows a 'Sign Up' form with the following layout:

- Name:** A text input field.
- Email:** A text input field.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Sign Up:** A dark blue rectangular button at the bottom center.

Gambar 3. 14 *Wireframe* Halaman *Sign Up*

4. Dashboard – User

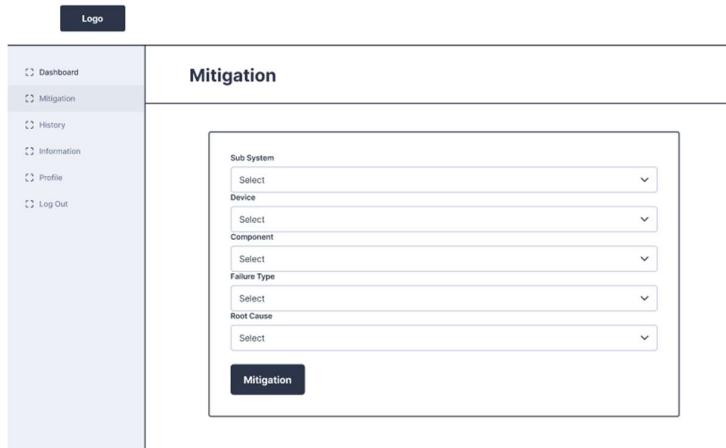
Wireframe Dashboard – User merupakan rancangan awal antarmuka pengguna yang berfungsi sebagai pusat kendali bagi pengguna setelah berhasil melakukan proses autentikasi. Halaman ini menampilkan akses menuju fitur-fitur utama sistem pakar, seperti *input* gejala, histori diagnosis, rekomendasi tindakan, informasi sistem, dan pengaturan profil. *Wireframe Dashboard – User* dapat dilihat pada Gambar 3.15 berikut:



Gambar 3. 15 Wireframe Dashboard – User

5. Halaman *Mitigation* – User

Wireframe halaman *Mitigation* – User merupakan rancangan awal antarmuka pengguna yang berfungsi untuk memfasilitasi proses pemilihan dan rekomendasi tindakan mitigasi berdasarkan hasil diagnosis sistem pakar. Halaman ini memungkinkan pengguna untuk memilih parameter teknis seperti subsistem, perangkat, komponen, jenis kerusakan, dan akar penyebab secara terstruktur melalui *dropdown* menu. Setiap *input* yang dipilih akan menjadi premis dalam proses *reasoning* untuk menentukan tindakan *maintenance* yang sesuai. *Wireframe* halaman *Mitigation* – User dapat dilihat pada Gambar 3.16 berikut:



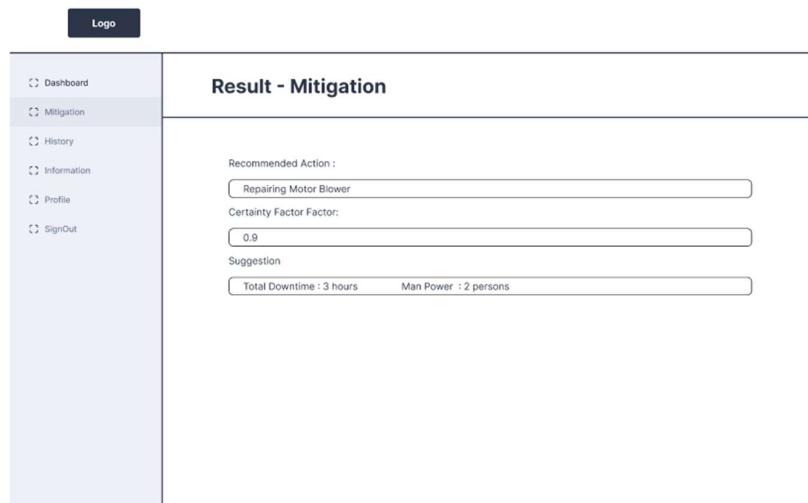
Gambar 3. 16 Wireframe Halaman *Mitigation* – User

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

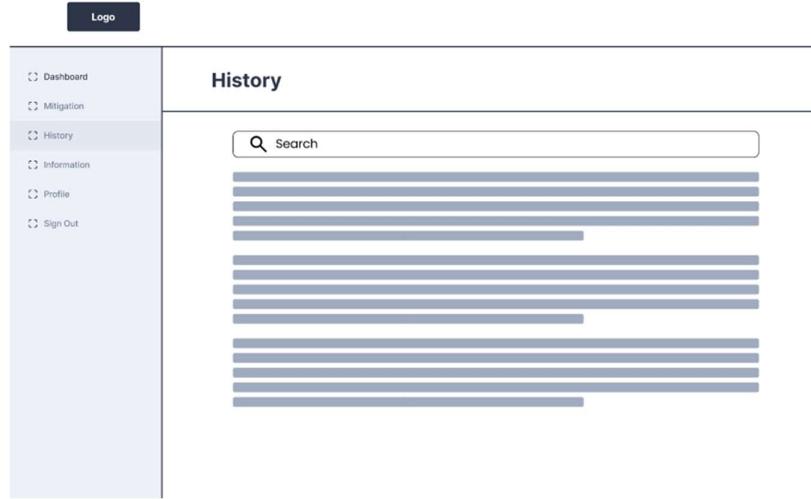
Wireframe hasil rekomendasi pada halaman *Mitigation – User* merupakan bagian dari antarmuka pengguna yang menampilkan output sistem pakar setelah proses *reasoning* dilakukan berdasarkan *input* parameter teknis. Halaman ini menyajikan tindakan mitigasi yang direkomendasikan, nilai *certainty factor*, serta estimasi sumber daya yang dibutuhkan seperti durasi downtime dan jumlah personel. *Wireframe* hasil rekomendasi *Mitigation – User* dapat dilihat pada Gambar 3.17 berikut:



Gambar 3. 17 *Wireframe Result – Mitigation*

6. Halaman *History – User*

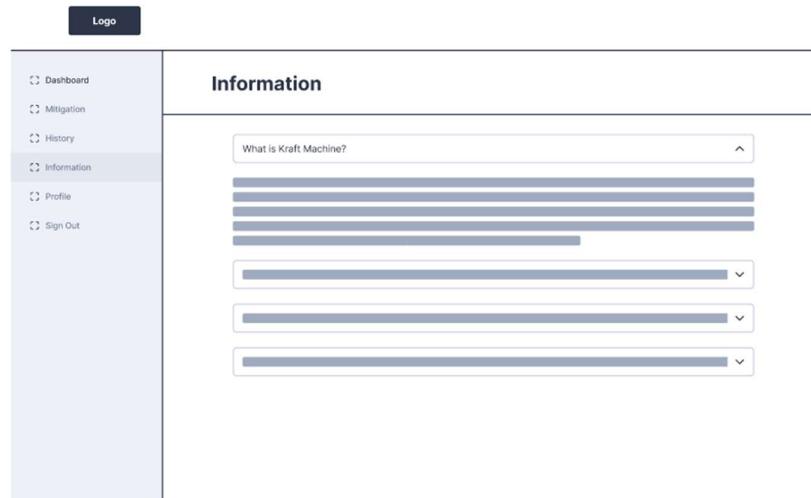
Wireframe halaman *History – User* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk menampilkan riwayat interaksi pengguna dengan sistem pakar. *Wireframe* halaman *History – User* dapat dilihat pada Gambar 3.18 berikut:



Gambar 3. 18 Wireframe Halaman *History* - *User*

7. Halaman *Information* – *User*

Wireframe halaman *Information* – *User* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk menyajikan informasi teknis dan konseptual terkait sistem pakar serta mesin Kraft yang menjadi objek diagnosis. Halaman ini dirancang untuk membantu pengguna memahami konteks kerja sistem, terminologi yang digunakan, dan cara kerja mesin sebelum melakukan proses *input* atau *reasoning*. *Wireframe* halaman *Information* – *User* dapat dilihat pada Gambar 3.19 berikut:



Gambar 3. 19 Wireframe Halaman *Information* - *User*

Jenni Febiyola Sari, 2025

SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

8. Halaman *Profile*

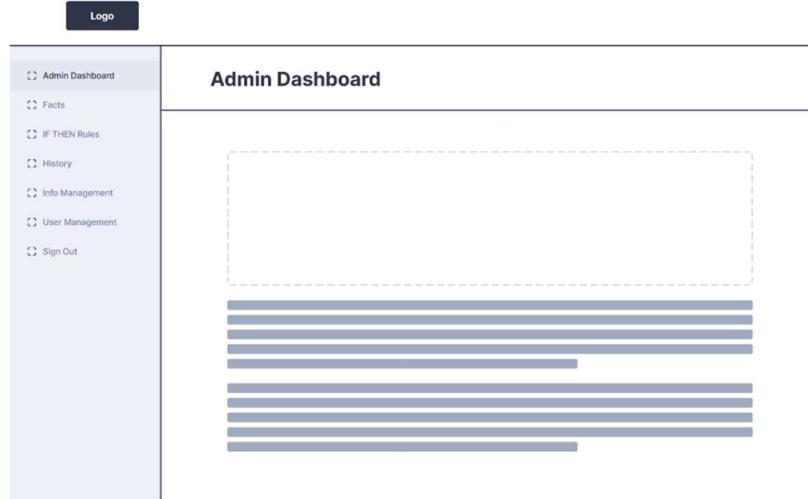
Wireframe halaman *Profile* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk memfasilitasi pengelolaan data akun pengguna dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Melalui halaman ini, pengguna dapat memperbarui informasi pribadi seperti nama, email, dan kata sandi secara mandiri. *Wireframe* halaman *Profile* dapat dilihat pada Gambar 3.20 berikut:

The wireframe shows a user interface for profile management. On the left is a vertical sidebar with a dark header containing a 'Logo'. Below the header are six menu items, each with a circular icon and text: 'Dashboard', 'Mitigation', 'History', 'Information', 'Profile' (which is highlighted with a blue background), and 'Sign Out'. The main content area has a light gray header with the word 'Profile' in bold. Below the header are four input fields with labels: 'Name' (with a single input field), 'Email' (with a single input field), 'New Password' (with a single input field), and 'Confirm New Password' (with a single input field). There is also a large empty space below these fields.

Gambar 3. 20 *Wireframe* Halaman *Profile* – *User*

9. Dashboard – Admin

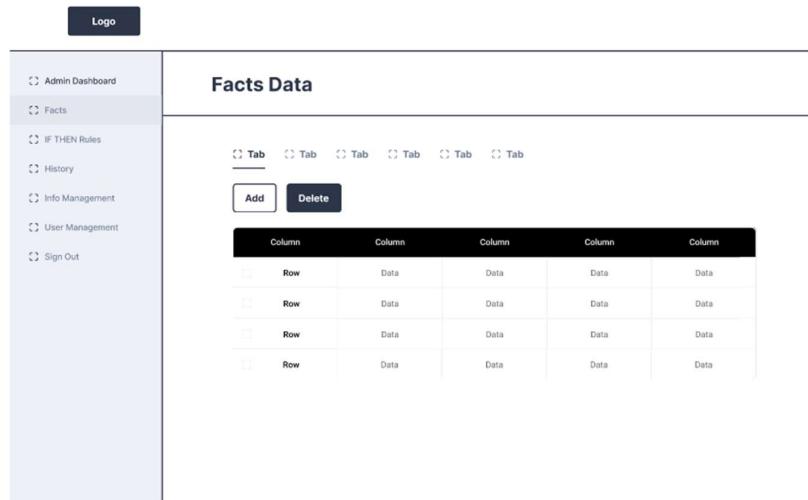
Wireframe halaman *Dashboard – Admin* merupakan rancangan awal antarmuka pengguna yang berfungsi sebagai pusat kendali administratif dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Melalui halaman ini, admin dapat mengakses dan mengelola berbagai komponen sistem seperti basis fakta, aturan *IF-THEN*, histori pengguna, manajemen informasi, dan pengaturan akun pengguna. *Wireframe* halaman *Dashboard – Admin* dapat dilihat pada Gambar 3.21 berikut:



Gambar 3. 21 Wireframe Dashboard - Admin

10. Halaman Facts – Admin

Wireframe halaman *Facts – Admin* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk mengelola basis fakta dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Melalui halaman ini, admin dapat menambahkan, menghapus, dan memodifikasi data fakta yang menjadi premis dalam proses *reasoning* berbasis *forward chaining* dan *certainty factor*. Wireframe halaman *Facts – Admin* dapat dilihat pada Gambar 3.22 berikut:



Gambar 3. 22 Wireframe Halaman Facts - Admin

11. Halaman *IF-THEN Rules – Admin*

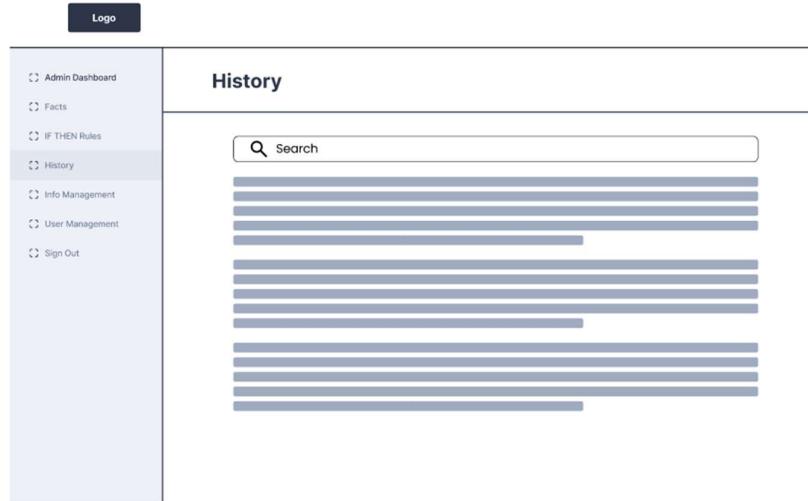
Wireframe halaman *IF-THEN Rules – Admin* merupakan rancangan awal antarmuka pengguna yang berfungsi untuk mengelola basis aturan dalam sistem pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Halaman ini memungkinkan admin untuk menambahkan, menghapus, dan memodifikasi aturan berbentuk logika kondisional *IF-THEN*, yang menjadi fondasi *reasoning* berbasis *forward chaining*. *Wireframe* halaman *IF-THEN Rules – Admin* dapat dilihat pada Gambar 3.23 berikut:

Column	Column	Column	Column	Column
Row	Data	Data	Data	Data
Row	Data	Data	Data	Data
Row	Data	Data	Data	Data
Row	Data	Data	Data	Data

Gambar 3. 23 *Wireframe* Halaman *IF-THEN Rules - Admin*

12. Halaman *History – Admin*

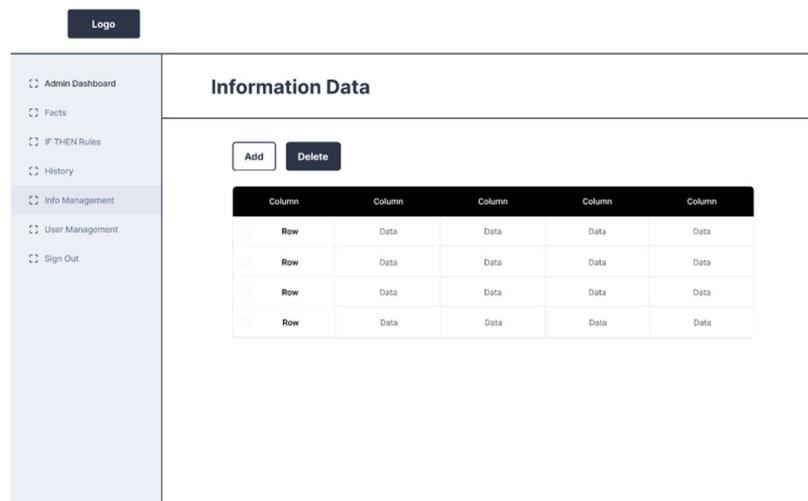
Wireframe halaman *History – Admin* berfungsi sebagai pusat pelacakan aktivitas sistem pakar, khususnya dalam konteks pengelolaan fakta dan aturan berbasis *forward chaining*. Antarmuka ini memungkinkan admin untuk meninjau riwayat interaksi dan eksekusi aturan yang telah terjadi dalam sistem. *Wireframe* halaman *History – Admin* dapat dilihat pada Gambar 3.24 berikut:



Gambar 3. 24 Wireframe Halaman History - Admin

13. Halaman *Info Management – Admin*

Wireframe halaman *Info Management – Admin* dirancang sebagai ruang dokumentasi yang menjelaskan secara konseptual dan teknis seluruh komponen sistem pakar yang digunakan dalam diagnosis gangguan mesin Kraft. Tidak seperti halaman fakta atau aturan yang bersifat inferensial, halaman ini berfungsi sebagai repositori pengetahuan yang menjembatani pemahaman pengguna terhadap cara kerja sistem. *Wireframe* halaman *Info Management – Admin* dapat dilihat pada Gambar 3.25 berikut:



Gambar 3. 25 Wireframe Halaman *Info Management – Admin*

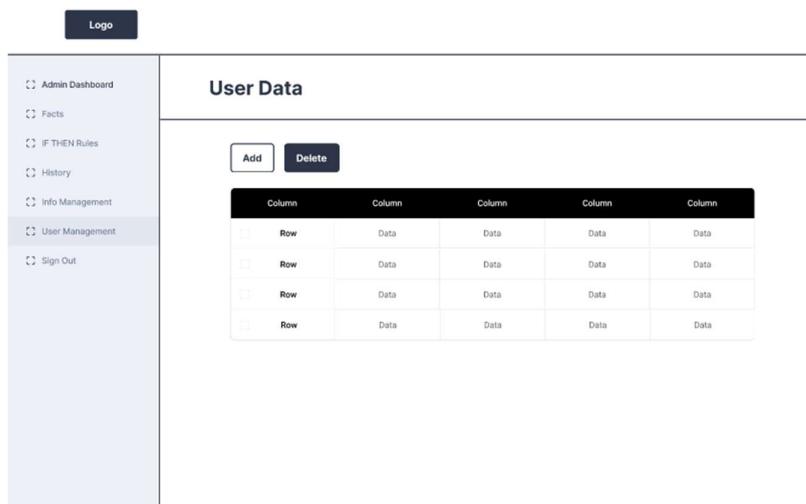
Jenni Febiyola Sari, 2025

**SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI
TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA**

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

14. Halaman *User Management – Admin*

Wireframe halaman *User Management – Admin* dirancang untuk mengelola data pengguna yang memiliki akses terhadap sistem pakar diagnosis gangguan mesin Kraft. Melalui halaman ini, admin dapat menambahkan, menghapus, atau memodifikasi informasi pengguna yang terdaftar, sehingga kontrol terhadap hak akses dan otorisasi sistem dapat dilakukan secara terpusat dan terstruktur. *Wireframe* halaman *User Management – Admin* dapat dilihat pada Gambar 3.26 berikut:



Gambar 3. 26 *Wireframe* Halaman *User Management - Admin*

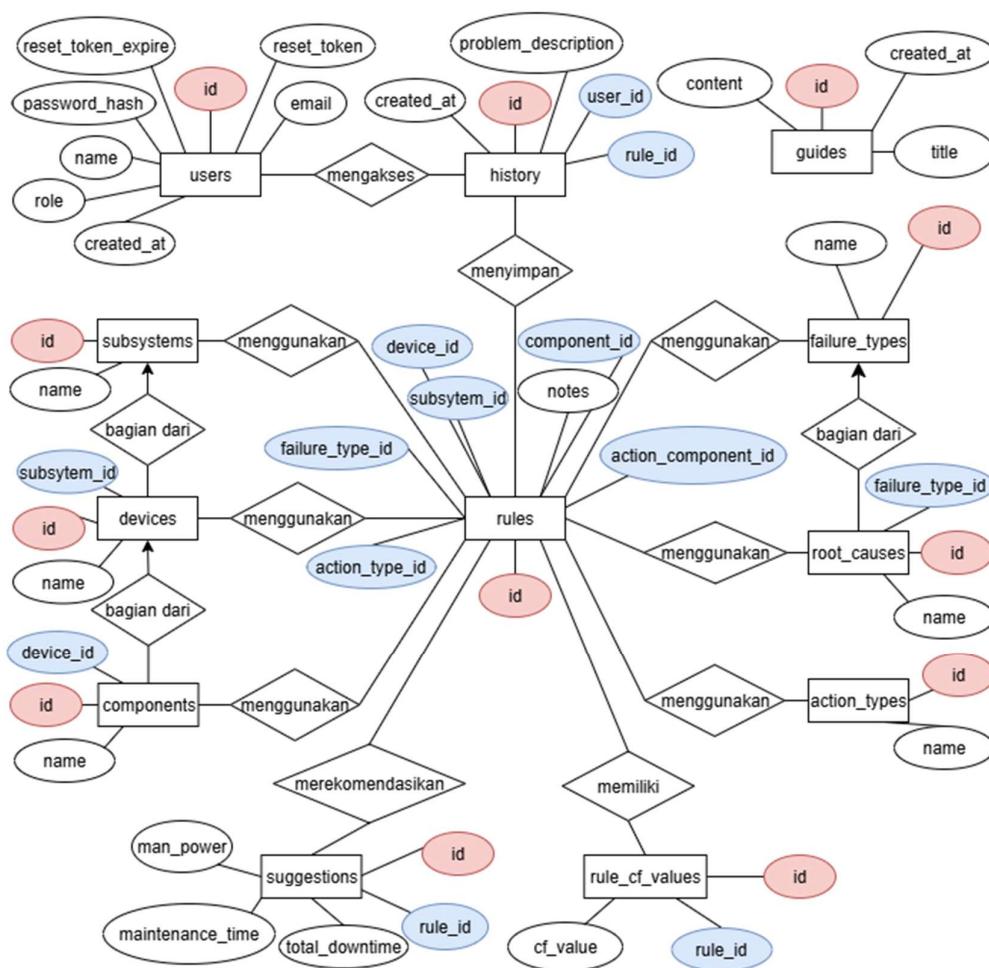
3.7.4 Perancangan Basis Data

Perancangan basis data dilakukan untuk memastikan bahwa seluruh informasi yang dibutuhkan oleh sistem pakar dapat disimpan, diakses, dan dikelola secara efisien. Basis data berfungsi sebagai fondasi penyimpanan fakta, aturan, dan hasil inferensi, serta mendukung integrasi antara modul *input*, modul inferensi, dan modul *output*. Perancangan dilakukan melalui dua tahap utama, yaitu perancangan *Entity Relationship Diagram* (ERD) dan perancangan skema relasional.

3.7.4.1 Perancangan ERD (*Entity Relatioanship Diagram*)

Entity Relationship Diagram (ERD) digunakan untuk memodelkan hubungan antar entitas yang terlibat dalam sistem pakar berbasis *forward chaining* dan

certainty factor untuk rekomendasi tindakan *maintenance* mesin Kraft. ERD memberikan gambaran konseptual mengenai struktur data, termasuk entitas utama seperti *Users*, dan *History*. Setiap entitas memiliki atribut yang merepresentasikan informasi teknis yang relevan dengan proses inferensi berbasis *forward chaining* dan *certainty factor*, seperti *cf_value* dan *rule_id*. Relasi antar entitas menunjukkan keterkaitan logis yang mendukung proses inferensi. ERD pada penelitian ini dirancang untuk merepresentasikan struktur basis data sistem pakar diagnosis kerusakan mesin Kraft secara menyeluruh. Diagram ini ditampilkan pada Gambar 3.27 berikut:



Gambar 3. 27 Entity Relationship Diagram

Berdasarkan Gambar 3.27, *Entity Relationship Diagram* (ERD) dari sistem pakar yang dibangun untuk mendukung proses diagnosis kerusakan dan rekomendasi tindakan mitigasi pada mesin. ERD ini menggambarkan hubungan antar entitas yang merepresentasikan struktur data utama di dalam sistem.

Sistem dimulai dari entitas *users*, yang menyimpan data pengguna seperti nama, email, *password*, dan peran (admin atau *user*). Setiap pengguna dapat melakukan proses diagnosis dan hasilnya disimpan ke dalam entitas *history*. Entitas *history* berfungsi sebagai catatan riwayat penggunaan sistem, dimana setiap baris menghubungkan seorang *user* dengan aturan (*rule*) yang digunakan, serta menyimpan waktu dan deskripsi masalah yang dimasukkan. Dengan demikian, hubungan antara entitas *users* dan *history* bersifat satu ke banyak, karena satu pengguna dapat menghasilkan banyak riwayat diagnosis.

Selanjutnya, sistem pengetahuan dibangun berdasarkan struktur hierarki mesin. Entitas *subsystems* berisi data *sub system* mesin, yang kemudian diturunkan ke entitas *devices*. Setiap subsystem dapat memiliki banyak device, sedangkan device sendiri dapat memiliki banyak komponen yang direpresentasikan dalam entitas *components*. Struktur ini menggambarkan bagian fisik dari mesin yang menjadi objek diagnosis.

Untuk aspek kerusakan, sistem menggunakan entitas *failure_types* yang mendefinisikan jenis-jenis kerusakan utama. Setiap jenis kerusakan memiliki penyebab dasar yang didefinisikan dalam entitas *root causes*, sehingga hubungan antara *failure_types* dan *root causes* juga bersifat satu ke banyak. Selain itu, sistem menyediakan entitas *action_types* yang berisi jenis tindakan perbaikan atau mitigasi yang mungkin dilakukan ketika terjadi kerusakan.

Seluruh pengetahuan sistem pakar kemudian dirangkum dalam entitas *rules*. Entitas ini menjadi inti dari basis pengetahuan, karena setiap rule menggunakan subsystem, device, component, failure type, root cause, action type, hingga komponen yang akan menjadi objek tindakan. Dengan adanya entitas *rules*, sistem mampu melakukan proses inferensi ketika *user* memberikan *input* mengenai kerusakan.

Agar proses inferensi lebih realistik, entitas rules juga dilengkapi dengan nilai kepastian atau *certainty factor* yang disimpan dalam entitas Rule_CF_Values. Selain itu, untuk mendukung analisis perawatan, tersedia entitas Suggestions yang mengaitkan rule dengan informasi tambahan seperti total *downtime*, waktu perawatan, dan jumlah tenaga kerja yang dibutuhkan.

Sementara itu, sistem juga memiliki entitas Guides yang berfungsi menyimpan panduan atau dokumentasi berupa informasi yang bisa diakses oleh pengguna sebagai referensi tambahan. Entitas ini berdiri sendiri dan tidak memiliki relasi langsung dengan entitas lainnya.

Dengan struktur tersebut, ERD ini menunjukkan bagaimana data pengguna, struktur mesin, jenis kerusakan, akar penyebab, serta tindakan mitigasi terhubung melalui basis pengetahuan berupa *rules*. Hasil dari proses inferensi kemudian disimpan dalam history agar dapat ditelusuri kembali. Secara keseluruhan, ERD ini merepresentasikan integrasi antara data teknis mesin, basis pengetahuan sistem pakar, serta interaksi pengguna, sehingga sistem dapat memberikan rekomendasi yang sesuai dengan kondisi kerusakan yang terjadi.

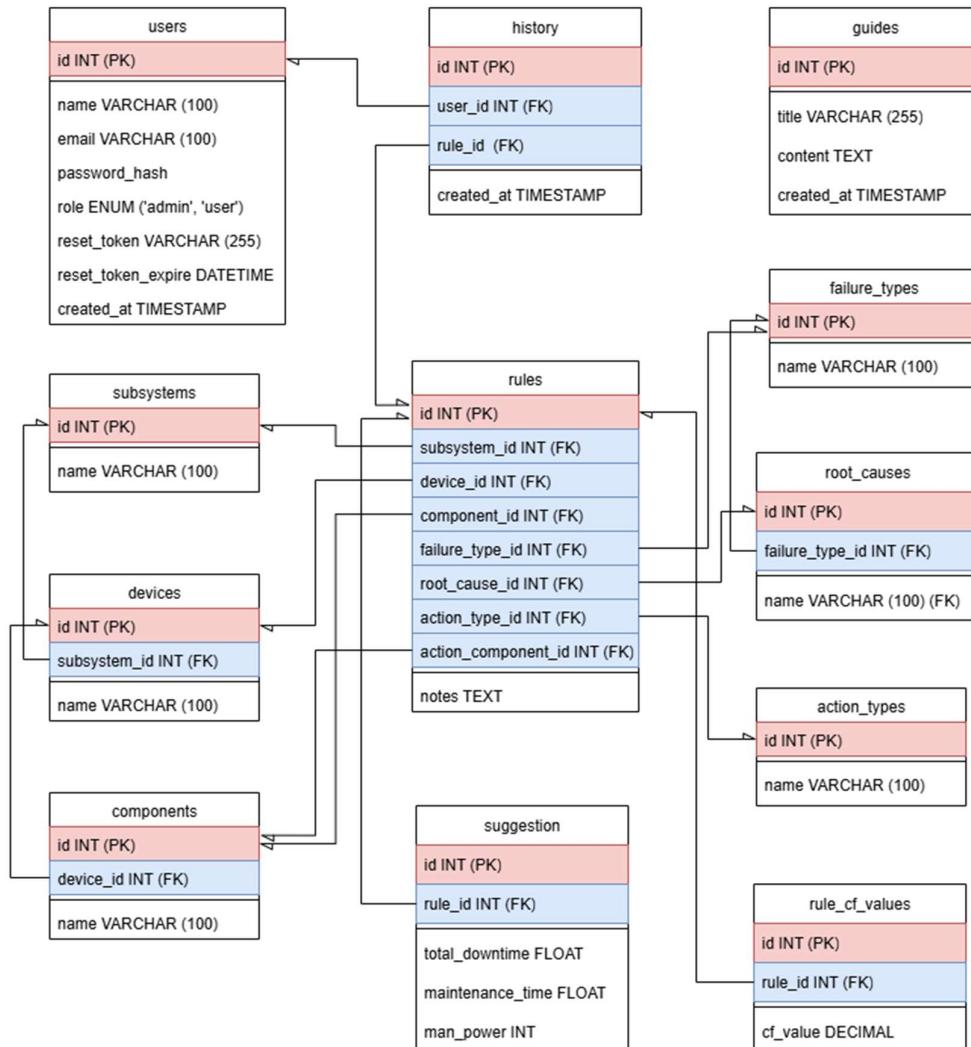
3.7.4.2 Perancangan Skema Relasional

Perancangan skema relasional merupakan tahap lanjutan dari perancangan *Entity Relationship Diagram* (ERD), yang bertujuan untuk mentransformasikan model konseptual menjadi struktur basis data yang implementatif dan siap diintegrasikan ke dalam sistem pakar. Dalam tahap ini, setiap entitas dan relasi yang telah dimodelkan secara semantik dalam ERD dikonversi menjadi tabel relasional dengan atribut yang terdefinisi secara eksplisit, tipe data yang sesuai, serta relasi antar tabel melalui penerapan *Primary Key* dan *Foreign Key*.

Skema relasional ini dirancang untuk memastikan efisiensi penyimpanan, konsistensi data, serta kemudahan integrasi dengan modul inferensi dan antarmuka pengguna. Selain itu, struktur relasional yang baik akan memudahkan proses pemeliharaan sistem, memungkinkan validasi data secara otomatis, dan mendukung interoperabilitas antar komponen sistem pakar. Dengan pendekatan ini, sistem tidak hanya mampu menyimpan data secara terstruktur, tetapi juga dapat melakukan

proses pencarian dan pengambilan informasi secara cepat dan akurat.

Setiap entitas dalam ERD dikonversi menjadi tabel dengan atribut yang telah ditentukan, tipe data yang sesuai, serta relasi antar tabel melalui *primary key* dan *foreign key*. Proses normalisasi juga diterapkan untuk menghindari redundansi dan memastikan integritas data. Berikut adalah penjabaran struktur skema relasional yang digunakan, sebagaimana ditampilkan pada Gambar 3.28 berikut:



Gambar 3. 28 Skema Relasional

Struktur kolom dan penamaan tabel pada skema relasional di atas dirancang secara modular untuk mendukung proses inferensi berbasis aturan dalam sistem

pakar untuk rekomendasi tindakan *maintenance* mesin Kraft. Setiap tabel merepresentasikan entitas yang telah dimodelkan secara konseptual dalam ERD, dan dikonversi ke dalam bentuk relasional dengan atribut yang terdefinisi secara eksplisit serta tipe data yang sesuai. Relasi antar tabel menggunakan *foreign key* untuk memastikan integritas referensial dan efisiensi *query* dalam proses mitigasi.

3.7.5 Pemrograman Perangkat Lunak

Pemrograman perangkat lunak dilakukan untuk membangun sistem pakar diagnosis kerusakan mesin Kraft secara fungsional dan terintegrasi. Proses ini melibatkan pemilihan teknologi yang sesuai dengan kebutuhan sistem, serta pengembangan dua komponen utama: struktur basis data dan antarmuka *website*.

Aplikasi perangkat lunak yang akan digunakan pada penelitian ini ditampilkan pada Tabel 3.7 berikut:

Tabel 3. 7 Perangkat Lunak yang Digunakan

Perangkat Lunak	Keterangan
Sistem Operasi	Window
Bahasa Pemrograman	Javascript, Node.js
Basis Data	MySQL
Browser	Chrome, Microsoft Edge

Pemrograman perangkat lunak pada sistem ini, yaitu sebagai berikut:

1. Struktur *Database MySQL*

Seluruh data pada sistem ini disimpan menggunakan MySQL sebagai basis data relasional. Struktur basis data dirancang untuk mendukung proses inferensi sistem pakar dan pelacakan histori pengguna. Basis data ini terdiri dari beberapa tabel utama yang masing-masing memiliki relasi yang jelas dan terstruktur.

Basis data dalam sistem ini secara umum terbagi menjadi tiga kelompok besar yang saling mendukung dalam proses kerja sistem pakar. Kelompok pertama adalah data fakta dan aturan, yang berfungsi sebagai inti dari basis pengetahuan sistem. Di dalamnya tersimpan informasi mengenai kondisi kerusakan mesin dan aturan

inferensi yang dibangun dengan pendekatan *Forward Chaining*.

Kelompok kedua adalah data hasil diagnosa, yaitu bagian yang menyimpan keluaran dari sistem setelah pengguna memasukkan *input*. Data ini mencatat rekomendasi tindakan yang dihasilkan oleh mesin inferensi, lengkap dengan waktu terjadinya (*timestamp*), identitas pengguna yang melakukan konsultasi, serta detail hasil inferensi. Dengan adanya penyimpanan hasil diagnosa, sistem memungkinkan pelacakan kembali riwayat mitigasi.

Kelompok ketiga adalah data pengguna sistem, yang menyimpan informasi identitas para pengguna *website*, termasuk nama, peran yang dimiliki seperti admin dan *user*, serta *password* yang disimpan dalam bentuk terenkripsi. Bagian ini memastikan hanya pengguna yang sah yang dapat mengakses sistem, serta membedakan hak akses antara pengguna biasa dengan administrator.

Struktur basis data ini dirancang menggunakan skema relasional dan diimplementasikan melalui *query SQL* untuk memastikan integritas data dan efisiensi proses pengambilan keputusan.

2. Pemrograman Website

Website ini dikembangkan menggunakan arsitektur berbasis *JavaScript* dengan *React.js* dengan *Tailwind CSS* sebagai *Frontend* dan *Node.js* dengan *Express* sebagai *backend*, serta *MySQL* sebagai sistem manajemen basis data.

Pada bagian *Frontend*, *React.js* digunakan untuk membangun antarmuka pengguna yang interaktif dan responsif, di mana operator dapat memasukkan gejala, melihat hasil diagnosa, serta menelusuri histori penggunaan. Komponen-komponen dalam *React* dirancang modular sehingga memudahkan pengelolaan tampilan dan interaksi. Di sisi *backend*, *Node.js* bersama *Express* menangani logika inferensi, pemrosesan data, dan komunikasi antara *frontend* dengan basis data. *Backend* juga mengatur autentikasi serta otorisasi pengguna sesuai dengan peran yang dimiliki. Sementara itu, *MySQL* berfungsi sebagai penyimpan utama yang menampung data pengguna, aturan inferensi, hasil diagnosa, hingga riwayat penggunaan sistem.

Dengan pemrograman perangkat lunak yang terstruktur dan teknologi yang saling mendukung, sistem ini diharapkan mampu memberikan hasil diagnosis yang akurat, cepat, dan dapat diakses secara fleksibel oleh pengguna di berbagai kondisi operasional.

3.8 Instrumen Penelitian

Instrumen penelitian yang dilakukan pada penelitian ini meliputi instrumen validasi pakar, *functional suitability testing* dan *usability testing*. Ketiga instrumen ini digunakan untuk menguji dan mengevaluasi apakah sistem telah memenuhi kebutuhan fungsional dan kenyamanan penggunaan sesuai dengan standar yang berlaku.

3.8.1 Validasi Pakar

Validasi pakar merupakan metode evaluasi berbasis penilaian ahli yang digunakan untuk memastikan bahwa sistem diagnosis kerusakan mesin Kraft telah sesuai dengan pengetahuan domain dan logika inferensi yang berlaku [22]. Validasi ini dilakukan oleh pakar yang memiliki kompetensi di bidang teknik mesin dan pemeliharaan industri, dengan fokus pada akurasi premis dan konsekuensi dalam basis aturan sistem.

Instrumen validasi dilakukan melalui pengujian langsung terhadap hasil inferensi sistem menggunakan 10 sampel acak yang merepresentasikan variasi komponen dan kondisi mesin Kraft. Setiap hasil diagnosis dan rekomendasi mitigasi dibandingkan dengan pendekatan teknis yang lazim digunakan dalam praktik industri oleh pakar berpengalaman. Penilaian dilakukan menggunakan metode verifikasi faktual, yaitu mencocokkan hasil sistem dengan rekomendasi aktual pakar tanpa menggunakan skala persepsi.

3.8.2 *Functional Suitability Testing*

Functional Suitability Testing merupakan karakteristik dalam standar ISO/IEC yang digunakan untuk menguji kesesuaian fungsional pada sistem, seperti apakah fitur-fitur pada sistem telah berjalan sesuai dengan kebutuhan fungsional yang telah dirancang [91]. Metode pengujian yang digunakan pada penelitian ini adalah *Black*

Box Testing, yaitu metode pengujian yang dilakukan hanya berdasarkan masukan dan keluaran. Tujuan dari metode ini adalah untuk memastikan bahwa setiap fitur yang tersedia dapat berfungsi sesuai dengan ekspektasi pengguna. Pengujian dilakukan dengan menyusun skenario uji berdasarkan fitur-fitur yang tersedia pada sistem, seperti *landing page*, *sign in*, *sign up*, dan fitur-fitur lainnya yang mendukung proses operasional. Setiap skenario dirancang untuk merepresentasikan interaksi pengguna terhadap sistem, baik dari sisi admin maupun *user*. Hasil pengujian dicatat dalam bentuk tabel yang memuat variabel seperti skenario uji, keluaran yang diharapkan, dan keluaran yang dihasilkan. Jika hasil keluaran sesuai dengan ekspektasi, maka fitur tersebut dinyatakan berfungsi dengan baik.

3.8.3 Usability Testing

Usability testing merupakan karakteristik dalam standar ISO/IEC 25010 digunakan untuk menilai apakah suatu sistem dapat secara mudah dan digunakan oleh pengguna [92]. Pengujian ini berfokus pada kemudahan penggunaan dan kenyamanan interaksi antara pengguna dan sistem.

Metode penilaian yang digunakan pada penelitian ini adalah *System Usability Scale* (SUS) yang dikembangkan oleh Brooke (1996). SUS merupakan instrumen berupa kuesioner yang terdiri atas sepuluh pernyataan yang mencerminkan berbagai aspek pengalaman pengguna. Setiap pernyataan dijawab menggunakan skala Likert 1 hingga 5, mulai dari “Sangat Tidak Setuju” hingga “Sangat Setuju”. Hasil penilaian dicatat dan dikonversi ke nilai 0-100 dengan menggunakan persamaan SUS, kemudian nilai total dianalisis untuk menentukan tingkat *usability* sistem.

Menurut Dumas & Redish (1999), *usability testing* idealnya melibatkan 6–12 partisipan, dengan minimal 3 responden pada setiap kelompok pengguna agar hasilnya representatif [93].

3.9 Teknik Analisis Data

Teknik analisis data yang dilakukan pada penelitian ini meliputi data *functional suitability* dan data *usability testing*.

3.9.1 Data *Functional Suitability Testing*

Data dari hasil *functional suitability testing* dengan menggunakan metode *Black Box Testing* akan dianalisis menggunakan pendekatan Skala Guttman, yaitu pendekatan pengukuran biner untuk menilai kesesuaian fungsional pada sistem dengan kebutuhan fungsional yang telah dirancang. Skala ini dipilih karena sesuai dengan kebutuhan sistem ini, dimana skala ini mampu menilai kemampuan sistem secara terukur, terutama untuk sistem berbasis fitur seperti *Website*. Seluruh fitur *website* akan diuji berdasarkan skenario tertentu, masing masing skenario uji akan diberikan nilai 1 jika sesuai dan 0 jika tidak. Jumlah yang diperoleh akan diukur dengan menggunakan Skala Guttman dengan aturan jika sesuai maka bernilai 1 dan jika tidak maka akan bernilai 0. Presentase kesesuaian sistem akan dihitung dengan menggunakan Persamaan III.1 berikut:

$$\text{Presentase Kelayakan} = \frac{\Sigma \text{Fitur yang sesuai}}{\Sigma \text{Fitur yang diuji}} \times 100 \quad \dots (\text{III.1})$$

Setelah dilakukan perhitungan presentase kelayakan dengan Persamaan III.1, presentase kelayakan yang di peroleh kemudian akan dikonversi dengan Tabel 3.8 berikut:

Tabel 3. 8 Presentase Kelayakan

Total Presentase	Keterangan
0% - 19,99%	Sangat Tidak Layak
20% - 39,99%	Kurang Layak
40% - 59,99%	Cukup Layak
60% - 79,99%	Layak
80%- 100%	Sangat Layak

3.9.2 Data *Usability Testing*

Data dari hasil *usability testing* dengan menggunakan kuesioner SUS akan dianalisis dengan menggunakan rentang nilai 1 sampai 5 untuk mengukur tingkat persetujuan responden terhadap pertanyaan yang diajukan, tabel penilaian dengan menggunakan Skala Likert sebagai berikut:

Tabel 3. 9 Penilaian Skala Likert

Nilai	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Skala ini dipilih karena sesuai dengan kebutuhan sistem ini, dimana skala ini mampu menilai seberapa baik sistem dapat diterima oleh pengguna, baik dari segi kemudahan dan kenyamanan pengguna. Pengujian dilakukan setelah pengguna mencoba sistem yang telah dirancang dan dikembangkan. SUS terdiri dari 10 pertanyaan, yaitu 5 pertanyaan positif dan 5 pertanyaan negatif. Penilaian yang terkumpul dari responden akan dihitung terlebih dahulu nilai rata-rata SUS dengan Persamaan III.2 sebagai berikut:

$$\text{Nilai Rata - Rata SUS} = \frac{\Sigma \text{Nilai SUS}}{\Sigma \text{Responden}} \quad \dots (\text{III.2})$$

Berdasarkan perhitungan nilai rata-rata SUS pada Persamaan III.2, nilai SUS untuk tiap pertanyaan dapat dihitung terlebih dahulu dengan perhitungan sebagai berikut:

1. Pertanyaan Positif

$$\text{Nilai SUS} = \text{Nilai jawaban} - 1 \quad \dots (\text{III.3})$$

2. Pertanyaan Negatif

$$\text{Nilai SUS} = 5 - \text{Nilai jawaban} \quad \dots (\text{III.4})$$

3. Nilai akhir SUS didapat dari hasil perhitungan total nilai rata-rata yang kemudian akan dikali dengan 2.5.

$$\text{Nilai Akhir SUS} = \text{Total Nilai SUS} \times 2.5 \quad \dots (\text{III.5})$$

Setelah dilakukan perhitungan nilai akhir SUS dengan Persamaan III.5, nilai akhir SUS yang diperoleh kemudian akan dihitung rata-ratanya dan dikonversi dengan tabel 3.10.

Jenni Febiyola Sari, 2025

**SISTEM PAKAR BERBASIS FORWARD CHAINING DAN CERTAINLY FACTOR UNTUK REKOMENDASI
TINDAKAN MAINTENANCE MESIN KRAFT DI PT DIRGANTARA INDONESIA**

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

Tabel 3. 10 Konversi Skor SUS ke Interpretasi Kualitatif

Total Nilai	Kelompok Nilai	Keterangan
>80.3	A	Sangat Layak
68 – 80.3	B	Layak
68	C	Netral
51 – 67	D	Tidak Layak
<51	E	Sangat Tidak Layak