

**SISTEM PAKAR PENDETEKSI KERUSAKAN AC
MENGUNAKAN METODE CERTAINTY
FAKTOR UNTUK Mendukung
TEKNISI PEMULA BERBASIS
WEB**

SKRIPSI

DISUSUN OLEH

HELENA RAYANI SARAGIH

NPM. 2209010169



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

**SISTEM PAKAR PENDETEKSI KERUSAKAN AC
MENGUNAKAN METODE CERTAINTY
FAKTOR UNTUK MENDUKUNG
TEKNISI PEMULA BERBASIS
WEB**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

HELENA RAYANI SARAGIH

2209010169

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

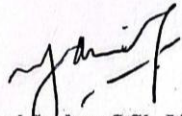
MEDAN

2026

LEMBAR PENGESAHAN

Judul Skripsi : SISTEM PAKAR PENDETEKSI KERUSAKAN AC
MENGUNAKAN METODE CERTAINTY FAKTOR
BERBASIS WEB UNTUK Mendukung Teknisi
Pemula
Nama Mahasiswa : HELENA RAYANI SARAGIH
NPM : 2209010169
Program Studi : SISTEM INFORMASI

Menyetujui
Komisi Pembimbing



(Yohanni Syahra, S.Si., M.Kom)
NIDN. 0129108201

Ketua Program Studi



(Mahardika Abdi Prawira Tanjung,
S.Kom., M.Kom)
NIDN. 0117088902

Dekan



(Dr. Al-Khawalizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FAKTOR UNTUK Mendukung TEKNISI PEMULA BERBASIS WEB

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 09 Maret 2026

Yang membuat pernyataan



Helena Rayani Saragih

NPM. 2209010169

PERNYATAAN PERSETUJUAN PUBLIKASI

KARYA ILMIAH UNTUK KEPENTINGAN

AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Helena Rayani Saragih
NPM : 2209010169
Program Studi : Sistem Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FAKTOR UNTUK Mendukung TEKNISI PEMULA BERBASIS WEB

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 09 Maret 2026

Yang membuat pernyataan



Helena Rayani Saragih

NPM. 2209010169

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Helena Rayani Saragih
Tempat dan Tanggal Lahir : Jakarta, 26 November 2004
Alamat Rumah : Raya Bayu
Telepon/Faks/HP : 0895320109294
E-mail : helenarayani1@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD Negeri 091316 Raya TAMAT: 2016
SMP Negeri 1 Raya TAMAT: 2019
SMA Negeri 1 Raya TAMAT: 2022

KATA PENGANTAR



Assalamualaikum Warahmatullahi Wabarakatuh

Syukur Alhamdulillah penulis lantunkan kehadiran Allah SWT, yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Sistem Pakar Pendeteksi Kerusakan AC Menggunakan Metode Certainty Faktor Untuk Mendukung Teknisi Pemula Berbasis Web”**.

Shalawat dan salam kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman kegelapan menuju zaman yang terang benderang seperti sekarang ini. Penyusunan skripsi ini bertujuan untuk memenuhi salah satu syarat guna mencapai gelar sarjana komputer di Universitas Muhammdiyah Sumatera Utara.

Penulis menyadari bahwa penulisan skripsi ini tidak dapat terselesaikan tanpa bantuan dari berbagai pihak, oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi, Penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.

3. Ibu Dr. Firahmi Rizky, S.Kom., M.Kom sebagai Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi (FIKTI) Sistem Informasi.
4. Bapak Mhd. Basri, M.Kom sebagai Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Bapak Mahardika Abdi Prawira Tanjung, M.Kom sebagai Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
6. Bapak Mulkan Azhari, S.Kom., M.Kom sebagai Sekretaris Prodi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
7. Ibu Yohanni Syahra, S.Si., M.Kom selaku Dosen Pembimbing yang telah dengan sangat sabar, tulus, dan penuh dedikasi memberikan waktunya untuk membimbing penulis. Terima kasih atas segala diskusi yang membuka wawasan, kritik yang membangun, serta arahan yang sangat berharga sehingga penulis dapat menuntaskan penelitian ini dengan lebih terarah. Ilmu dan nasihat yang Ibu berikan tidak hanya membantu penulis menyelesaikan skripsi ini, tetapi juga menjadi bekal berharga bagi masa depan penulis.
8. Ibunda tercinta Siti Rosmida Purba, wanita hebat yang menjadi alasan utama penulis untuk terus berjuang. Terima kasih yang tak terhingga atas setiap tetesan keringat dan deru doa yang Ibu selipkan di antara sujud dan setiap helai sholatmu. Penulis sangat menyadari bahwa selesainya skripsi ini bukan semata – mata karena kecerdasan penulis, melainkan berkat

ketulusan doa – doa Ibu yang menembus langit, yang memudahkan setiap kesulitan, dan menenangkan setiap kegelisahan saat penulis menghadapi kendala dalam penelitian ini. Terima kasih telah menjadi sandaran ternyaman dan sumber cinta yang tidak pernah kering.

9. Alm. Bapak Rimbun Saragih, cinta pertama penulis yang kini telah beristirahat dengan tenang di sisi Allah SWT. Terima kasih sudah menjadi Bapak yang semasa hidupnya selalu berusaha memberikan kehidupan yang baik bagi penulis, kasih sayang yang tiada henti, motivasi, serta doa yang begitu berarti. Penulis sangat merindukan sosokmu hadir di sini, namun penulis yakin Bapak tersenyum bangga di tempat terbaik di sisi-Nya melihat penulis telah sampai di garis *finish* ini.
10. Terima kasih kepada sahabat seperjuangan Amanda Isnaini Putri dan Selasih Sayufi, terima kasih telah menjadi sosok yang selalu ada, baik di saat suka maupun duka, dan memberikan motivasi, dukungan yang sangat berharga sampai selesai. *See you on top guys!!*
11. Kepada teman - teman seperjuangan, terima kasih telah membantu, memberikan motivasi kepada penulis dalam pengerjaan skripsi ini.
12. *Last but not least*, diri saya sendiri Helena Rayani Saragih apresiasi sebesar-besarnya untuk diri sendiri yang telah diam – diam berjuang tanpa henti. Terima kasih telah bekerja keras dan bertahan sejauh ini. Untuk setiap malam yang dihabiskan dalam kelelahan, setiap pagi yang disambut dengan kekhawatiran, namun tetap dijalani dan berhasil dilalui. Terima kasih kepada hati yang tetap ikhlas, meski banyak hal yang terjadi di luar prediksi.

Terima kasih kepada jiwa yang tetap kuat, meski berkali-kali hampir menyerah karena kondisi. Terima kasih kepada raga yang terus melangkah, meski lelah menghadapi tekanan tapi tetap diperjuangkan. Teruslah belajar dan mensyukuri nikmat yang Allah SWT. berikan.

Medan, 09 Maret 2026



Helena Rayani Saragih

SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FAKTOR UNTUK MENDUKUNG TEKNISI PEMULA BERBASIS WEB

ABSTRAK

Air Conditioner (AC) adalah alat yang sering dipakai untuk menjaga kenyamanan suhu ruangan, tetapi dalam operasionalnya sering mengalami berbagai macam kerusakan yang memerlukan penanganan dari teknisi berpengalaman. Teknisi baru sering menghadapi masalah dalam mendiagnosis kerusakan AC akibat minimnya pengetahuan dan pengalaman. Dengan demikian, penelitian ini bertujuan untuk merancang dan mengembangkan sistem pakar deteksi kerusakan AC berbasis web memakai metode *Certainty Factor* (CF) untuk mendukung proses diagnosis kerusakan berdasarkan gejala yang dialami. Informasi mengenai gejala dan tipe kerusakan didapat melalui wawancara dengan ahli (pakar) serta penelitian literatur yang sesuai. Sistem ini dibuat menggunakan Python sebagai bahasa pemrograman dan MySQL sebagai basis data. Temuan penelitian mengindikasikan bahwa sistem pakar yang dirancang dapat memberikan deteksi kerusakan AC beserta level kepastian dan solusi penanganannya, sehingga membantu teknisi pemula dalam melakukan identifikasi awal kerusakan AC dengan lebih cepat dan akurat.

Kata Kunci: Sistem Pakar; Kerusakan AC; *Certainty* Faktor;.

AN EXPERT SYSTEM FOR DETECTING AIR CONDITIONING DAMAGE
USING THE CERTAINTY FACTOR METHOD TO SUPPORT NOVICE
TECHNICIANS USING AWEB-BASED SYSTEM

ABSTRACT

Air Conditioner (AC) is a device that is often used to maintain comfortable room temperatures, but during operation it often experiences various types of damage that require handling by experienced technicians. New technicians often the problems in diagnosing AC damage due to their lack of knowledge and experience. Therefore, this study aims to design and develop a web-based AC fault detection expert system using the Certainty Factor (CF) method to support the diagnosis process based on the symptoms experienced. Information about symptoms and types of faults was obtained through interviews with experts and relevant literature research. This system was developed using Python as the programming language and MySQL as the database. The research findings indicate that the designed expert system can provide AC damage detection along with certainty levels and solutions for handling them, thereby helping novice technicians to perform initial identification of AC damage more quickly and accurately.

Keywords: *Expert System; Air conditioner damage; Certainty Factor;*.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
KARYA ILMIAH UNTUK KEPENTINGAN	iv
AKADEMIS.....	iv
RIWAYAT HIDUP.....	v
KATA PENGANTAR.....	vi
ABSTRAK	ix
ABSTRACT.....	x
DAFTAR ISI.....	xiii
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II.....	7
LANDASAN TEORI	7
2.1 Tinjauan Pustaka	7
2.1.1 Dinamika Penelitian Sistem Pakar.....	7
2.1.2 Sintesis dan Kebaruan Penelitian.....	10
2.2 Teori Sistem Pakar (Expert System).....	12
2.2.1 Definisi dan Konsep Dasar	12
2.2.2 Arsitektur dan Komponen Sistem Pakar.....	13
2.3 Metode Certainty Faktor (CF).....	14
2.3.1 Konsep Ketidakpastian dalam Diagnosa	14
2.3.2 Formulasi Matematis CF.....	15
2.4 AC (Air Conditioner) Split.....	16
2.4.1 Prinsip Kerja Siklus Refigerasi	16

2.4.2	Komponen Utama dan Analisa Kerusakan	17
2.5	Unit Modelling Language (UML).....	19
2.6	Pengembangan Sistem Berbasis Web	23
2.6.1	World Wide Web (WWW) dan HTTP.....	23
2.6.2	HTML (HyperText Markup Language	23
2.6.3	MySQL.....	24
2.6.4	XAMPP	24
2.6.5	Python	24
BAB III	26
ANALISA DAN PERANCANGAN SISTEM	26
3.1	Analisa Permasalahan	26
3.2	Teknik Pengumpulan Data	28
3.2.1	Wawancara.....	28
3.2.2	Analisis Data Gejala dan Kerusakan	29
3.2.3	Basis Pengetahuan (Knowledge base).....	30
3.2.4	Studi Pustaka.....	32
3.3	Algoritma Sistem.....	32
3.3.1	Akuisisi Nilai Kepastian Pakar (MB dan MD).....	32
3.3.2	Penentuan Nilai Keyakinan Pengguna (CF User).....	33
3.3.3	Perhitungan Nilai CF gejala Tunggal	34
3.3.4	Kombinasi Nilai CF (CF <i>Combine</i>).....	34
3.3.6	Analisis Perhitungan Metode Certainty Faktor	35
3.3.7	Contoh kasus deteksi	35
3.4	Pemodelan dan Perancangan Sistem.....	38
3.4.2	Pemodelan dengan Unit Modelling Language (UML).....	38
3.4.4	Design User Interface.....	44
BAB IV	50
IMPLEMENTASI DAN PENGUJIAN SISTEM	50
4.1	Kebutuhan Sistem	50
4.1.1	Kebutuhan Perangkat Keras (<i>Hardware</i>)	50
4.1.2	Kebutuhan Perangkat Lunak (<i>Software</i>)	50
4.2	Implementasi Sistem	52
4.2.1	Implementasi Basis Data	53
4.3	Hasil Pengujian Sistem	55
BAB V	61
PENUTUP	61

5.1 Kesimpulan	61
5.2 Saran	62
DAFTAR PUSTAKA	64
LAMPIRAN	67

DAFTAR TABEL

Tabel 2. 1 Penelitian terdahulu	11
Tabel 2. 2 Simbol - simbol use case diagram	20
Tabel 2. 3 Simbol - simbol activity diagram.....	21
Tabel 2. 4 Simbol - simbol sequence diagram	22
Tabel 2. 5 Simbol - simbol class diagram.....	23
Tabel 3. 1 Daftar Gejala AC	29
Tabel 3. 2 Daftar kerusakan AC.....	30
Tabel 3. 3 Basis pengetahuan kerusakan AC	31
Tabel 3. 4 Skala Terminologi Pakar.....	33
Tabel 3. 5 Nilai CF User	33
Tabel 3. 6 Rancangan Tabel Kerusakan.....	43
Tabel 3. 7 Rancangan Tabel Gejala	43
Tabel 3. 8 Rancangan Tabel Aturan.....	43
Tabel 3. 9 Rancangan Tabel Hasil	44
Tabel 4. 1 Kebutuhan data	52
Tabel 4. 2 Tampilan tb_kerusakan.....	53
Tabel 4. 3 Tampilan tb_gejala.....	54
Tabel 4. 4 Tampilan tb_aturan	54
Tabel 4. 5 Tampilan tb_riwayat hasil.....	54
Tabel 4. 6 Tampilan tb_user	55

DAFTAR GAMBAR

Gambar 3. 1 Alur kerja penelitian	27
Gambar 3. 2 Penulis melakukan wawancara dengan pakar	28
Gambar 3. 3 Use case diagram.....	39
Gambar 3. 4 Activity diagram.....	40
Gambar 3. 5 Sequence diagram	41
Gambar 3. 6 Class diagram	42
Gambar 3. 7 Tampilan Login	45
Gambar 3. 8 Halaman Dashboard Admin.....	46
Gambar 3. 9 Tampilan Kelola Data Gejala.....	46
Gambar 3. 10 Tampilan Dashboard User.....	47
Gambar 3. 11 Tampilan Deteksi Gejala.....	48
Gambar 3. 12 Tampilan Hasil Deteksi.....	48
Gambar 3. 13 Tampilan Riwayat Deteksi.....	49
Gambar 4. 1 Tampilan Halaman login.....	56
Gambar 4. 2 Tampilan Dashboard Admin	57
Gambar 4. 3 Tampilan Kelola Data Gejala.....	57
Gambar 4. 4 Tampilan Dashboard User.....	58
Gambar 4. 5 Tampilan Deteksi Gejala.....	59
Gambar 4. 6 Tampilan Hasil Deteksi.....	59
Gambar 4. 7 Tampilan Riwayat Deteksi.....	60

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan AC (*Air Conditioner*) pada zaman sekarang ini semakin dibutuhkan. Hal ini ditandai dengan hampir semua masyarakat memasangnya baik untuk ruangan kantor maupun di rumah. Mesin ini memiliki fungsi yang penting dalam membuat ruangan terhindar dari hawa panas, gangguan kerusakan pada mesin pendingin ruangan akan mempengaruhi kenyamanan masyarakat, terlebih jika masyarakat tidak mengetahui bagaimana gejala awal kerusakan mesin pendingin ruangan (Nofrianis, 2022).

Masalah utama terjadi pada tahap pengolahan informasi diagnostik oleh teknisi yang baru. Keterbatasan akses ke basis pengetahuan yang terorganisir membuat teknisi baru mengalami kesulitan dalam mengubah fakta-fakta gejala menjadi keputusan yang tepat. Ketidakmampuan untuk mengklasifikasikan data gejala secara sistematis mengakibatkan munculnya mesin informasi dalam mengidentifikasi jenis kerusakan, yang berujung pada tindakan perbaikan yang keliru (*trial and error*). Hal ini mengindikasikan adanya kendala dalam proses informasi dari ahli ke praktisi di lapangan, sehingga diperlukan sebuah solusi yang berbasis Sistem Informasi untuk melaksanakan kodifikasi pengetahuan tersebut.

Kesenjangan kompetensi ini diperburuk oleh tingginya tingkat ketidakpastian data gejala yang terdeteksi di lapangan. Sebuah gejala kualitatif sering kali dapat merujuk pada berbagai jenis kerusakan yang berbeda, sehingga menjadi sulit untuk

mengambil keputusan yang konsisten tanpa adanya sistem pendukung keputusan. Oleh karena itu, diperlukan suatu sistem pakar yang dapat mengintegrasikan pengetahuan ahli ke dalam platform digital untuk mendukung teknisi pemula dalam menganalisis gejala-gejala kerusakan yang terjadi secara terstruktur.

Beberapa penelitian terdahulu mengembangkan sistem pakar. Misalnya (Utomo, 2021) merancang sistem pakar berbasis android untuk mendeteksi kerusakan AC dengan menggunakan metode *Certainty* Faktor dan menunjukkan bahwa metode ini efektif dalam menentukan jenis kerusakan berdasarkan gejala yang tidak pasti. Peneliti lain (Afriansyah & Mawarni, 2022) berhasil mengimplementasikan sistem pakar berbasis web untuk mendeteksi kerusakan AC dengan metode *Forward Chaining*. Namun, sebagian besar penelitian tersebut hanya berfokus pada aspek deteksi umum dan belum secara khusus mengembangkan sistem yang dirancang untuk membantu teknisi pemula dengan proses pembelajaran interaktif dan pengambilan keputusan berbasis web.

Oleh karena itu diperlukan sistem pakar yang dapat membantu teknisi pemula mendeteksi kerusakan AC. Sistem pakar yang akan dikembangkan dalam penelitian ini menggunakan metode *Certainty* Faktor (CF) berbasis web, yang bertujuan untuk menjadi alat bantu yang bermanfaat untuk meningkatkan efisiensi dan akurasi kerja teknisi pemula.

Pemilihan metode *Certainty* Factor (CF) dalam studi ini dijustifikasi oleh kemampuannya dalam mengelola ketidakpastian selama proses pengambilan keputusan. Tidak seperti logika biner yang pasti, metode CF dapat memberikan bobot derajat keyakinan terhadap setiap gejala, sehingga sistem dapat

mengeluarkan output berupa kemungkinan jenis kerusakan beserta tingkat kepastiannya. Ini sangat penting dalam sistem informasi diagnostik untuk mengurangi subjektivitas dan meningkatkan ketepatan keputusan teknis (Ramadhan et al., 2018).

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, dapat diketahui rumusan masalah ini adalah :

1. Bagaimana analisa kebutuhan sistem pakar yang dapat membantu teknisi pemula dalam mendeteksi kerusakan *Air Conditioner* (AC)?
2. Bagaimana rancangan sistem pakar berbasis web untuk mendeteksi kerusakan AC menggunakan metode *Certainty* Faktor?
3. Bagaimana sistem pakar pendeteksi kerusakan AC berbasis web mampu berjalan optimal?

1.3 Batasan Masalah

Untuk memastikan bahwa pembahasan tetap fokus pada rumusan masalah , maka ditetapkan batasan penelitian ini sebagai berikut:

1. Penelitian ini berfokus pada AC tipe split bukan jenis AC industri atau sentral atau jenis pendingin ruangan lainnya.
2. Kemampuan sistem dalam mendeteksi kerusakan terbatas pada aturan yang ada. Jika pengguna memasukkan gejala yang tidak sesuai dengan

basis data pakar, maka sistem secara otomatis tidak akan menampilkan hasil diagnosa.

3. Sistem pakar ini dibuat hanya untuk mendeteksi kerusakan AC tidak melakukan perbaikan langsung.
4. Sistem ini menerapkan metode *Certainty Factor* (CF) untuk menghitung tingkat kepercayaan terhadap kemungkinan kerusakan berdasarkan gejala yang disediakan oleh pengguna
5. Sistem ini hanya digunakan oleh teknisi pemula tidak untuk pelanggan.
6. Sistem pakar yang dikembangkan berbasis web sehingga bisa diakses lewat browser, tetapi pengujian sistem hanya dilakukan di lingkungan pengembangan yang digunakan dalam penelitian
7. Sistem ini hanya menawarkan hasil deteksi berupa tipe kerusakan, tingkat kepercayaan, dan solusi awal, sehingga keputusan akhir untuk melakukan perbaikan tetap ada di tangan teknisi.

1.4 Tujuan Penelitian

Merujuk pada rumusan masalah yang telah diuraikan di atas, tujuan dari penelitian ini dapat dirinci sebagai berikut:

1. Menganalisis kebutuhan sistem pakar yang dapat membantu teknisi pemula dalam mendeteksi kerusakan AC.
2. Merancang sistem pakar berbasis web yang efektif untuk mendeteksi kerusakan AC.
3. Mengimplementasikan metode *Certainty* Faktor untuk mendeteksi kerusakan AC.

4. Menguji sistem pakar berbasis web digunakan untuk memastikan keakuratan dan efektivitas sistem.

1.5 Manfaat Penelitian

Diharapkan penelitian ini akan memberikan kontribusi yang bermanfaat.

Berikut beberapa manfaat yang diharapkan dari penelitian ini adalah :

1. Bagi Kampus
 - a. Salah satu referensi akademis untuk penelitian di bidang kecerdasan buatan (*Artificial Intelligence*), khususnya penerapan metode *Certainty Faktor* (CF) dalam sistem berbasis web.
 - b. Menambah kumpulan karya ilmiah yang dapat dijadikan sebagai panduan bagi mahasiswa lain dalam mengembangkan sistem pakar untuk deteksi kerusakan peralatan elektronik.
 - c. Berkontribusi secara verbal terhadap implementasi pelaksanaan teknologi cerdas yang relevan dengan kebutuhan industri dan Pendidikan vokasi.
 - d. Mendorong kolaborasi antar-dosen dan mahasiswa dalam proyek penelitian terapan, sehingga meningkatkan output publikasi ilmiah kampus di tingkat nasional maupun internasional.
2. Bagi Penulis
 - a. Menambah wawasan tentang *Certainty Factor* dalam analisis deteksi kerusakan AC dan memenuhi penyusunan tugas akhir yaitu skripsi.

- b. Memberikan kesempatan bagi penulis untuk menerapkan secara langsung proses analisis, perancangan, implementasi, dan pengujian sistem pakar berbasis web menggunakan metode *Certainty Factor*.
3. Bagi Masyarakat
- a. Memberikan alat bantu cerdas kepada teknisi pemula dalam mendeteksi kerusakan AC dengan lebih cepat, akurat, dan efisien yang akan mengurangi beban teknisi ahli.
 - b. Membantu penyedia layanan AC dalam meningkatkan kualitas layanan dan akurasi diagnosis melalui sistem berbasis web.
 - c. Masyarakat dapat menggunakan teknologi berbasis kecerdasan buatan dalam layanan teknis sehari-hari

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam pengembangan sebuah sistem pakar yang bertujuan untuk mendeteksi kerusakan pada mesin pendingin ruangan atau *Air Conditioner (AC)*, penelusuran terhadap literatur dan penelitian terdahulu menjadi fondasi krusial. Analisis ini tidak hanya berfungsi untuk memvalidasi kebaruan (*novelty*) dari penelitian yang sedang dilakukan, tetapi juga untuk memetakan perkembangan metode komputasi, khususnya *Certainty Factor (CF)*, dalam menangani ketidakpastian diagnostik pada perangkat elektromekanis. Selama periode 2021 hingga 2025, terlihat tren signifikan dalam integrasi kecerdasan buatan (*Artificial Intelligence*) pada ranah pemeliharaan perangkat keras, di mana sistem pakar telah berevolusi dari sekadar alat pencatat menjadi sistem pendukung keputusan (*Decision Support System*) yang cerdas.

2.1.1 Dinamika Penelitian Sistem Pakar

Penelitian mengenai sistem pakar diagnostik telah menunjukkan diversifikasi yang luas, baik dari sisi objek penelitian maupun metode inferensi yang digunakan. Salah satu penelitian fundamental dalam domain ini dilakukan oleh (Utomo, 2021) yang merancang aplikasi sistem pakar untuk mendeteksi kerusakan AC rumah berbasis Android. Penelitian ini menjadi titik tolak penting karena berhasil mengimplementasikan metode *Certainty Factor* untuk menangani gejala kerusakan AC yang sering kali ambigu. Dalam studinya, Utomo menemukan bahwa

penggunaan CF mampu memberikan tingkat akurasi diagnosa sebesar 80% ketika divalidasi dengan pakar manusia. Temuan ini mengindikasikan bahwa logika fuzzy atau ketidakpastian yang diakomodasi oleh CF sangat relevan untuk diagnosa AC, di mana gejala seperti “kurang dingin” tidak selalu merujuk pada satu penyebab tunggal (misalnya, bisa karena freon habis, kotor, atau kompresor lemah). Namun, keterbatasan utama penelitian ini terletak pada platform Android yang mungkin membatasi integrasi dengan sistem manajemen bengkel yang lebih besar yang biasanya berbasis web, serta antarmuka yang lebih ditujukan untuk pengguna akhir (pemilik rumah) daripada teknisi yang membutuhkan detail teknis lebih mendalam.

Melengkapi perspektif tersebut, (Afriansyah & Mawarni, 2022) melakukan penelitian dengan fokus pada pembangunan sistem pakar kerusakan AC berbasis web pada “Jaya Servis AC”. Berbeda dengan Utomo, penelitian ini menggunakan metode *Forward Chaining*. Metode ini bekerja dengan premis yang pasti, menelusuri fakta-fakta gejala untuk mencapai kesimpulan. Meskipun sistem ini berhasil membantu efisiensi administrasi pencatatan kerusakan di bengkel, pendekatan *Forward Chaining* memiliki kelemahan dalam menangani nuansa ketidakpastian. Dalam konteks teknisi pemula, yang mungkin ragu-ragu dalam menentukan apakah suara mesin “berisik” atau “sangat berisik”, metode deterministik seperti ini kurang fleksibel dibandingkan CF. Studi ini memberikan kontribusi penting dalam hal arsitektur sistem berbasis web yang memudahkan aksesibilitas data servis secara terpusat, sebuah aspek yang diadopsi dalam penelitian skripsi ini.

Selain pada AC, penerapan *Certainty Factor* juga terbukti efektif pada perangkat elektronik lain yang memiliki kemiripan karakteristik kerusakan (gabungan masalah kelistrikan dan komponen fisik). (Thoriq Achmad Alawi, M Ghofar Rohman, 2025) meneliti sistem pakar untuk diagnosa kerusakan televisi berbasis web. Penelitian ini menegaskan bahwa metode CF mampu memberikan nilai kepercayaan yang presisi dalam bentuk persentase, yang sangat membantu pengguna dalam memprioritaskan tindakan perbaikan. Dengan akurasi mencapai 90%, studi ini memvalidasi bahwa CF adalah metode yang robust untuk domain elektronika, di mana satu gejala visual (seperti layar bergaris) bisa memiliki bobot keyakinan berbeda terhadap berbagai kemungkinan kerusakan komponen (*mainboard* vs panel). Prinsip pembobotan gejala yang diterapkan pada diagnosa TV ini sangat relevan untuk diadopsi ke dalam diagnosa komponen elektronik AC seperti modul PCB dan *thermistor*.

Relevansi metode CF semakin diperkuat oleh penelitian (Syarifuddin & Suryadi, 2025) yang mengembangkan sistem pakar untuk diagnosa kerusakan motor matic Honda. Studi terbaru ini menyoroti proses akuisisi pengetahuan dari mekanik ahli dan konversinya menjadi nilai *Measure of Belief* (MB) dan *Measure of Disbelief* (MD). Penelitian ini memberikan kerangka kerja metodologis yang sangat baik tentang bagaimana mentransfer tacit *knowledge* (pengetahuan yang ada di kepala teknisi senior) menjadi *explicit knowledge* (aturan sistem). Mengingat target pengguna sistem dalam skripsi ini adalah teknisi pemula, pendekatan Syarifuddin dalam menyajikan solusi perbaikan yang terstruktur berdasarkan

prioritas nilai CF menjadi referensi utama dalam merancang antarmuka hasil diagnosa.

Senada dengan itu, (Afwan et al., 2022) mengimplementasikan metode CF untuk diagnosa kerusakan software pada smartphone berbasis web. Penelitian ini menonjolkan aspek edukasi, di mana sistem tidak hanya memberikan “vonis” kerusakan, tetapi juga penjelasan logis mengapa kesimpulan tersebut diambil. Hal ini sejalan dengan tujuan skripsi ini untuk "mendukung teknisi pemula", di mana aspek transfer pengetahuan menjadi nilai tambah utama. Tingkat akurasi sistem yang diuji dengan membandingkan hasil diagnosa sistem dan pakar manual menunjukkan konsistensi yang tinggi, memberikan validasi empiris terhadap formula CF yang digunakan.

2.1.2 Sintesis dan Kebaruan Penelitian

Berdasarkan analisis terhadap lima penelitian di atas, dapat ditarik benang merah bahwa metode *Certainty Factor* adalah standar industri yang andal untuk sistem pakar diagnostik yang melibatkan ketidakpastian. Tabel 2.1 berikut merangkum posisi penelitian terdahulu dan membandingkannya dengan penelitian yang diusulkan.

Tabel 2. 1 Penelitian terdahulu

Peneliti (Tahun)	Objek Diagnosa	Metode	Platform	Hasil
Utomo (2021)	AC Rumah	Certainty Faktor	Android	Akurasi 80%, fokus pengguna umum
Afriansyah & Mawarni (2022)	AC (Jaya Service)	Forward Chaining	Web	Administrasi servis, logika deterministic.
Alawi et al. (2024)	Televisi	Certainty Faktor	Web	Akurasi 90%, diagnosa komponen elektronik.
Syarifuddin & Suryadi (2025)	Motor Matic Honda	Certainty Faktor	Dekstop (Java)	Akuisisi pengetahuan mekanik bengkel resmi.
Afwan et al. (2022)	Smartphone	Certainty Faktor	Web	Penjelasan logis diagnosa (edukatif).

Celah Penelitian (Research Gap): Mayoritas penelitian sebelumnya pada objek AC masih terpolarisasi pada dua kutub: sistem berbasis Android untuk pengguna rumahan (Utomo, 2021) atau sistem administrasi bengkel dengan logika sederhana (Afriansyah & Mawarni, 2022). Belum terdapat penelitian yang secara spesifik merancang sistem pakar berbasis web dengan metode *Certainty Factor* yang didedikasikan sebagai alat bantu ajar dan pendukung keputusan (*decision support tool*) bagi teknisi pemula. Teknisi pemula membutuhkan lebih dari sekadar “hasil diagnosa”; mereka memerlukan alur penalaran yang transparan untuk meningkatkan kompetensi mereka. Penelitian ini mengisi celah tersebut dengan mengintegrasikan keandalan CF dalam menangani ketidakpastian gejala AC

dengan platform web yang mudah diakses, serta desain antarmuka yang mendukung proses pembelajaran teknisi.

2.2 Teori Sistem Pakar (Expert System)

2.2.1 Definisi dan Konsep Dasar

Sistem Pakar adalah komponen dari kecerdasan buatan yang meniru cara berpikir manusia. Penggunaan teknologi memungkinkan manusia untuk mendapatkan informasi tanpa batasan ruang dan waktu. Sistem Pakar merupakan perangkat lunak yang meniru cara berpikir serta pengetahuan seorang ahli dalam mengatasi suatu masalah tertentu (Roy, Yohanni Syahra, 2021). Konsep ini berakar pada premis bahwa kepakaran manusia, yang terdiri dari pengetahuan teoretis mendalam dan pengalaman praktis bertahun-tahun, dapat dikodifikasikan ke dalam serangkaian aturan logika computer (Yogi Ainur Rofiq Anggara, Suryo Adi Wibowo, 2025).

Menurut definisi klasik yang sering dikutip dari (Mechael et al., 2025) sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran untuk memecahkan masalah yang biasanya hanya dapat diselesaikan oleh seorang pakar dalam bidang tersebut. Dalam konteks perbaikan AC, seorang “pakar” adalah teknisi senior yang mampu mendiagnosa penyebab AC tidak dingin hanya dengan mendengarkan suara kompresor atau memegang pipa evaporator. Sistem pakar berupaya memindahkan kemampuan intuitif ini ke dalam bentuk algoritma yang dapat diakses oleh teknisi pemula.

Tujuan utama dari sistem pakar bukanlah untuk menggantikan keberadaan pakar manusia sepenuhnya, melainkan untuk:

1. Transfer Pengetahuan: Memasyarakatkan pengetahuan langka yang dimiliki segelintir ahli.
2. Efisiensi: Meningkatkan produktivitas dengan mempercepat proses diagnosa.
3. Konsistensi: Menghindari kesalahan manusiawi (human error) akibat kelelahan atau inkonsistensi emosional yang sering dialami manusia.

2.2.2 Arsitektur dan Komponen Sistem Pakar

Sebuah sistem pakar yang ideal tersusun atas dua lingkungan utama, yaitu lingkungan pengembangan (*development environment*) tempat sistem dibangun, dan lingkungan konsultasi (*consultation environment*) tempat sistem digunakan oleh end-user. Di dalam arsitektur ini, dikutip dari (Yusuf, 2021) terdapat komponen-komponen vital yang saling berinteraksi yaitu :

a. Basis Pengetahuan (*Knowledge Base*)

Ini adalah jantung dari sistem pakar. Basis pengetahuan berisi fakta-fakta spesifik dan aturan-aturan heuristik (rules) yang merepresentasikan pengetahuan pakar.

b. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah “otak” yang memproses pengetahuan. Komponen ini bertugas mencocokkan fakta yang ada dalam basis pengetahuan dengan fakta yang diberikan oleh pengguna untuk menarik kesimpulan logis. Terdapat dua metode inferensi utama:

1. *Forward Chaining* (Pelacakan Maju): Memulai dari sekumpulan fakta (gejala) dan bergerak maju menuju kesimpulan (kerusakan). Metode ini

cocok untuk masalah diagnosa di mana gejala awal menuntun pada hipotesis.

2. *Backward Chaining* (Pelacakan Mundur): Memulai dari hipotesis (dugaan kerusakan) dan mencari fakta-fakta pendukung untuk membuktikannya.

Dalam penelitian ini, mekanisme inferensi akan diperkaya dengan perhitungan *Certainty Factor* untuk mengakomodasi ketidakpastian, sehingga tidak hanya menghasilkan kesimpulan biner (Ya/Tidak) tetapi juga derajat keyakinan.

c. Antarmuka Pengguna (*User Interface*)

Fasilitas yang memungkinkan komunikasi dua arah antara pengguna (teknisi pemula) dan sistem. Antarmuka harus dirancang intuitif agar teknisi dapat dengan mudah memasukkan gejala yang mereka temukan di lapangan dan memahami hasil diagnosa serta saran perbaikan yang diberikan sistem.

d. Fasilitas Penjelas (*Explanation Facility*)

Komponen ini membedakan sistem pakar dengan program konvensional. Fasilitas ini memungkinkan sistem untuk menjelaskan mengapa sebuah pertanyaan diajukan atau bagaimana sebuah kesimpulan diambil.

2.3 Metode Certainty Faktor (CF)

2.3.1 Konsep Ketidakpastian dalam Diagnosa

Dalam dunia nyata, jarang sekali seorang teknisi AC bisa 100% yakin bahwa suatu gejala pasti disebabkan oleh satu kerusakan tertentu. Misalnya, gejala “AC kurang dingin” bisa disebabkan oleh filter kotor, kurang freon, atau kompresor lemah. Ada derajat ketidakpastian dalam hubungan sebab-akibat ini.

Metode logika klasik (Boolean) yang hanya mengenal Benar (1) atau Salah (0) seringkali kaku dan gagal menangkap nuansa ini.

Certainty Factor (CF) diperkenalkan oleh Shortliffe dan Buchanan (1975) dalam pengembangan sistem pakar medis MYCIN untuk mengakomodasi ketidakpastian pemikiran seorang pakar tersebut. CF mendefinisikan ukuran kepastian terhadap suatu fakta atau aturan, menggambarkan seberapa besar kepercayaan seorang pakar terhadap validitas data yang ada. Nilai CF berkisar antara -1 (sangat tidak percaya) hingga +1 (sangat percaya), dengan nilai 0 menunjukkan ketidaktahuan total (*ignorance*) (Latumakulita, 2021).

2.3.2 Formulasi Matematis CF

Secara matematis, *Certainty Factor* diformulasikan sebagai selisih antara ukuran kepercayaan (*Measure of Belief*) dan ukuran ketidakpercayaan (*Measure of Disbelief*). Rumus dasarnya adalah sebagai berikut :

$$CF[H, E] = MB[H, E] - MD[H, E] \dots \dots \dots (2.1)$$

Keterangan:

- $CF[H, E]$: Faktor kepastian (*Certainty Factor*) dari hipotesis H yang dipengaruhi oleh bukti/gejala E.
- $MB[H, E]$: *Measure of Belief*, yaitu ukuran kenaikan kepercayaan terhadap hipotesis H yang dipengaruhi oleh gejala E. Nilainya berkisar antara 0 sampai 1.
- $MD[H, E]$: *Measure of Disbelief*, yaitu ukuran kenaikan ketidakpercayaan terhadap hipotesis H yang dipengaruhi oleh gejala E. Nilainya berkisar antara 0 sampai 1.

2.4 AC (Air Conditioner) Split

Pemahaman mendalam mengenai objek penelitian, yaitu AC Split, sangat krusial agar sistem pakar yang dibangun memiliki validitas teknis. AC Split adalah jenis pengkondisi udara yang paling umum digunakan di perumahan dan perkantoran skala kecil di Indonesia.

2.4.1 Prinsip Kerja Siklus Refigerasi

AC Split bekerja berdasarkan siklus kompresi uap (*vapor compression refrigeration cycle*). Sistem ini memindahkan kalor (panas) dari dalam ruangan ke luar ruangan menggunakan fluida kerja yang disebut refrigeran (freon). Siklus ini terdiri dari empat proses termodinamika utama yang berulang terus-menerus (Firdaus et al., 2025) :

a. Komperesi

Gas refrigeran bertekanan rendah dan bersuhu rendah dari evaporator dihisap oleh kompresor, kemudian ditekan (dikompresi) sehingga menjadi gas bertekanan tinggi dan bersuhu tinggi (*superheated vapor*).

b. Kondensasi

Gas panas tersebut mengalir ke kondensor (unit *outdoor*). Di sini, kalor dilepaskan ke udara lingkungan dengan bantuan kipas (*fan*). Akibat pelepasan kalor, gas refrigeran berubah wujud (mengembun) menjadi cairan bertekanan tinggi.

c. Ekspansi

Cairan refrigeran bertekanan tinggi melewati katup ekspansi atau pipa kapiler. Komponen ini menurunkan tekanan refrigeran secara drastis, yang mengakibatkan

penurunan suhu yang ekstrem. Refrigeran keluar dalam bentuk campuran cairan dan uap dingin.

e. Evaporasi

Refrigeran dingin masuk ke evaporator (unit *indoor*). Di sini, refrigeran menyerap panas dari udara ruangan yang dihembuskan oleh blower. Penyerapan panas ini menyebabkan refrigeran mendidih dan berubah wujud kembali menjadi gas. Udara ruangan yang kehilangan panasnya menjadi dingin dan disirkulasikan kembali (Saputra, 2025)

2.4.2 Komponen Utama dan Analisa Kerusakan

Deteksi kerusakan pada sistem pakar akan berpusat pada komponen-komponen berikut beserta mode kegagalannya (Saputra, 2025) sebagai berikut :

a. Kompresor (*Compressor*)

Kompresor adalah “jantung” dari sistem AC yang terletak di unit outdoor. Fungsinya adalah memompa refrigeran ke seluruh sistem. Jenis kompresor pada AC Split umumnya adalah *rotary* atau *scroll*.

1. Gejala Kerusakan: Suara berdengung keras (*mechanical noise*), amper tinggi (*overload*), bodi kompresor sangat panas, atau mati total.
2. Penyebab: Kapasitor lemah, kekurangan oli, tegangan listrik tidak stabil, atau keausan mekanis.

b. Kondensor (*Condenser*)

Terletak di unit outdoor, kondensor berbentuk pipa berlekuk dengan sirip-sirip (*fins*) aluminium untuk memperluas area pelepasan panas.

1. Gejala Kerusakan: AC kurang dingin, tekanan freon tinggi, kompresor sering mati mendadak (*overheat*).
2. Penyebab: Sirip kondensor tertutup debu/kotoran tebal sehingga pertukaran panas terhambat, atau motor fan outdoor mati.

c. Evaporator (*Evaporator*)

Terletak di unit indoor, berfungsi menyerap panas. Kebersihan evaporator sangat mempengaruhi efisiensi pendinginan dan kualitas udara.

1. Gejala Kerusakan: Angin pelan, timbul bunga es (*nge-blok*), air menetes dari unit indoor, bau apek.
2. Penyebab: Kotoran/lendir menutupi sirip, kebocoran freon pada u-bend, atau filter udara kotor.

d. Pipa Kapiler (*Capillary Tube*)

Pipa tembaga berdiameter sangat kecil yang berfungsi menurunkan tekanan refrigeran.

1. Gejala Kerusakan: Timbul bunga es di awal pipa tekan, arus kompresor rendah tapi AC tidak dingin.
2. Penyebab: Penyumbatan total atau parsial akibat kotoran/oli yang terjebak dalam sistem sirkulasi.

e. Refrigeran (*Refrigerant*)

Zat pendingin (biasa disebut freon, misal R32, R410a, R22). Refrigeran bersifat permanen dalam sistem tertutup dan tidak akan habis kecuali terjadi kebocoran.

1. Gejala Masalah: Pipa kecil membeku, tekanan rendah (PSI rendah), AC tidak dingin sama sekali.
2. Penyebab: Kebocoran pada sambungan (*flaring nut*) atau korosi pada pipa evaporator/kondensor.

f. Komponen Kelistrikan Pendukung

1. Kapasitor: Membantu start awal motor kompresor dan fan. Jika rusak, motor hanya mendengung.
2. *Thermistor*: Sensor suhu yang memutus arus kompresor saat suhu ruangan tercapai. Jika rusak, AC bisa terlalu dingin (tidak mau cut-off) atau kompresor tidak mau hidup sama sekali.
3. Modul PCB: Otak elektronik yang mengatur logika kerja. Kerusakan biasanya ditandai dengan kode error pada display atau lampu indikator berkedip.

2.5 Unit Modelling Language (UML)





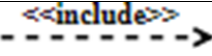
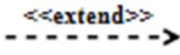
Untuk memastikan sistem pakar yang dibangun memiliki struktur yang logis, terdokumentasi, dan memenuhi standar rekayasa perangkat lunak, penelitian ini menggunakan *Unified Modeling Language* (UML). UML adalah bahasa visual standar untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak berorientasi objek. UML menyediakan serangkaian diagram yang menggambarkan sistem dari berbagai perspektif (Nurlita et al., 2023).

Diagram - diagram UML yang relevan dan akan digunakan dalam perancangan sistem ini meliputi :

1. Use Case Diagram

Diagram ini menggambarkan interaksi antara satu atau lebih aktor dalam sistem, serta membantu dalam memahami fungsi-fungsi yang ada dalam sistem informasi yang sedang dikembangkan (Pangestu et al., 2024). Adapun simbol - simbol *use case* sebagai berikut :




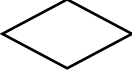


Tabel 2. 2 Simbol - simbol use case diagram

Simbol	Deskripsi
	<i>Use Case</i> : Abstraksi dan interaksi antar sistem dan aktor
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	Asosiasi : Menggambarkan bagaimana aktor berinteraksi dengan <i>use case</i> .
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
	Relasi <i>include</i> : Hubungan antara dua <i>use case</i> untuk menunjukkan adanya perilaku <i>use case</i> yang dimasukkan ke dalam perilaku dari <i>base use case</i>
	Relasi <i>extend</i> : Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya.

2. Activity Diagram

Menunjukkan adanya alur kerja atau aktivitas yang berjalan dari suatu sistem. Diagram ini berfokus pada aliran kontrol (*control flow*) dari satu aktivitas ke aktivitas lainnya, berupa sebuah proses yang didorong oleh *internal processing* (Wayahdi & Ruziq, 2023). Adapun simbol - simbol *activity diagram* sebagai berikut :





Tabel 2. 3 Simbol - simbol activity diagram

Simbol	Deskripsi
	<i>Start</i> : Menjelaskan awal proses kerja dalam <i>activity</i> diagram
	<i>End</i> : Menandai kondisi akhir dari suatu aktivitas dan memrepresentasikan penyelesaian.
	<i>Activity</i> : Menunjukkan kegiatan yang membentuk proses dalam diagram.
	<i>Decision</i> : Percabangan di mana pilihan aktivitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	<i>Connector</i> : Menunjukkan arah aliran atau aliran kontrol dari aktivitas.

3. Sequence Diagram

Sequence diagram merepresentasikan kolaborasi yang dinamis di antara berbagai objek dan menunjukkan rangkaian pesan yang dikirimkan di antara objek serta interaksi yang terjadi di antara objek dalam sistem yang dikembangkan (Narulita et al., 2024). Adapun simbol – simbol *sequence* diagram sebagai berikut:






Tabel 2. 4 Simbol - simbol sequence diagram

Simbol		Deskripsi
		Objek : Komponen utama <i>sequence diagram</i>
		<i>Actor</i> : Menggambarkan orang yang sedang berinteraksi dengan sistem
		<i>Life line</i> : Menggambarkan tempa mulai dan berakhirnya sebuah <i>message</i> .
		<i>Message</i> : Menggambarkan pengiriman pesan.

4. Class Diagram

Menggambarkan struktur statis sistem dengan menunjukkan kelas-kelas (*classes*), atribut, metode, dan hubungan antar kelas tersebut (seperti asosiasi, agregasi, atau pewarisan) (Nugraha & Rosmeida, 2021). Adapun simbol – simbol *class diagram* sebagai berikut :

Tabel 2. 5 Simbol - simbol class diagram

Simbol	Deskripsi
	Asosiasi : Relasi antar kelas dengan makna umum.
	Asosiasi berarah : Relasi antar kelas dengan makna class yang satu digunakan oleh makna class yang lain.
	Generalisasi : Relasi antar kelas dengan makna generalisasi – spesialisasi.
	Dependency : Relasi antar kelas dengan makna keberuntungan antar kelas .
	Agregasi : Relasi antar kelas dengan makna semua bagian (<i>whole part</i>).

2.6 Pengembangan Sistem Berbasis Web

2.6.1 World Wide Web (WWW) dan HTTP

Website adalah kumpulan halaman yang saling terhubung (melalui hyperlink) yang berisi informasi multimedia (teks, gambar, video) dan dapat diakses melalui jaringan internet menggunakan protokol HTTP (*Hypertext Transfer Protocol*). Arsitektur web menggunakan model Client-Server, di mana browser pengguna (*Client*) mengirim permintaan (*request*) ke server, dan server memprosesnya lalu mengirimkan balasan (*response*) berupa halaman web (Agustinus Budi Santoso, S.St., n.d.)

2.6.2 HTML (HyperText Markup Language)

HTML adalah bahasa markah (*markup language*) standar untuk membuat struktur halaman web. HTML5, versi terbaru, menyediakan elemen semantik yang

memungkinkan struktur halaman yang lebih rapi dan dukungan multimedia yang lebih baik.¹ Dalam sistem pakar ini, HTML digunakan untuk membangun kerangka antarmuka pengguna, seperti formulir input gejala dan tabel hasil diagnosa (Sinlae et al., 2024).

2.6.3 MySQL

MySQL adalah Sistem Manajemen Basis Data Relasional (*Relational Database Management System - RDBMS*) yang menggunakan SQL (*Structured Query Language*) untuk mengelola data. MySQL terkenal karena kecepatan, keandalan, dan kemudahan penggunaannya. Dalam sistem ini, MySQL bertugas menyimpan data master (jenis kerusakan, gejala), aturan (basis pengetahuan), data admin, dan riwayat diagnosa user. Relasi antar tabel (misalnya tabel kerusakan berelasi one-to-many dengan tabel gejala) dikelola di sini untuk menjamin integritas data

2.6.4 XAMPP

XAMPP adalah paket perangkat lunak gratis yang menggabungkan Apache (web server), MySQL/MariaDB (database),⁴ dan Perl. XAMPP berfungsi sebagai server lokal (*localhost*) yang memungkinkan pengembang untuk menjalankan dan menguji aplikasi web di komputer lokal sebelum diunggah ke internet (*hosting*) (Chandra & Setyaningsih, 2025)

2.6.5 Python

Python merupakan bahasa pemrograman tingkat tinggi yang ditafsirkan, dikembangkan oleh Guido van Rossum pada tahun 1991, dengan sintaks sederhana menyerupai bahasa Inggris untuk memudahkan pembacaan dan pengembangan

cepat. Linux, serta macOS, didukung pustaka standar luas untuk berbagai aplikasi (Suryaningrat, S.Kom., 2024).

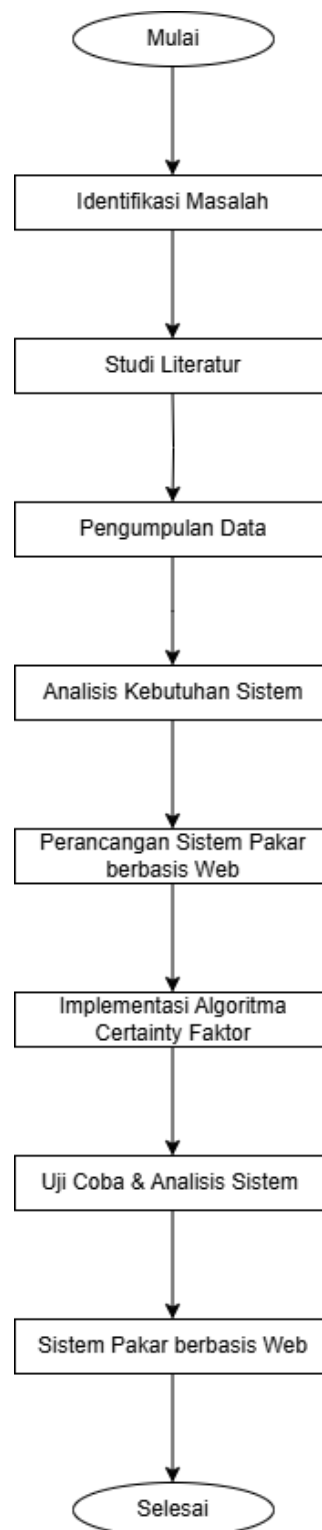
BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Permasalahan

Tahap ini merupakan fase inisiasi krusial di mana peneliti melakukan dekonstruksi terhadap fenomena inefisiensi yang terjadi dalam proses perawatan AC Split. Berdasarkan latar belakang masalah, ditemukan adanya disparitas kompetensi yang signifikan antara teknisi ahli dan teknisi pemula. Teknisi pemula sering kali terjebak dalam metode diagnosa trial and error, yang tidak hanya memboroskan waktu tetapi juga meningkatkan biaya operasional dan risiko kerusakan komponen lebih lanjut.

Pada fase ini, peneliti membatasi ruang lingkup masalah secara tegas pada AC tipe Split - jenis yang paling umum digunakan di perumahan dan menetapkan penggunaan metode *Certainty Factor* (CF) sebagai solusi komputasi untuk menangani ketidakpastian gejala. Masalah diidentifikasi bukan hanya sebagai masalah teknis perbaikan, tetapi sebagai masalah manajemen pengetahuan (*knowledge management*) : bagaimana mentransfer intuisi pakar ke dalam algoritma yang deterministik namun mampu menangani ambiguitas. Output dari tahap ini adalah rumusan masalah yang presisi: bagaimana membangun sistem yang dapat mereplikasi pola pikir pakar dalam mendiagnosa kerusakan AC dengan akurasi yang dapat diterima secara industri.



Gambar 3.1 Alur kerja penelitian

3.2 Teknik Pengumpulan Data

Pengumpulan data merupakan sebuah teknik atau cara yang dilakukan oleh peneliti untuk bisa mengumpulkan data terkait dengan permasalahan dari penelitian yang diambil. Data yang peneliti kumpulkan untuk mendukung penelitian yaitu berkaitan dengan deteksi kerusakan AC (*Air Conditioner*). Teknik pengumpulan data yang dilakukan dalam penelitian ini sebagai berikut:

3.2.1 Wawancara

Wawancara dilakukan untuk mendapatkan informasi berupa data secara langsung dari sumbernya. Wawancara dilakukan dengan pakar teknisi AC Abang Andi di Universitas Muhammadiyah Sumatera Utara (UMSU).



Gambar 3. 2 Penulis melakukan wawancara dengan pakar

3.2.2 Analisis Data Gejala dan Kerusakan

Berdasarkan hasil wawancara yang telah dilakukan dengan pakar (teknisi AC), dilakukan analisis untuk mengidentifikasi data teknis yang akan diterapkan dalam sistem. Data itu terbagi menjadi dua komponen utama, yakni data gejala dan data tipe kerusakan pada unit AC.

a. Data Gejala

Data gejala ini merupakan tanda - tanda atau karakteristik awal yang muncul saat AC mengalami gangguan. Kumpulan data ini diambil langsung dari hasil wawancara bersama pakar teknisi AC yang kemudian disusun dalam Tabel 3.1 di bawah ini.

Tabel 3. 1 Daftar Gejala AC

Kode Gejala	Nama Gejala
G01	AC tidak dingin
G02	Tekanan freon rendah
G03	Kompresor berdengung keras
G04	Kompresor sangat panas (overheat)
G05	Air menetes dari unit indoor
G06	Timbul bunga es pada pipa atau evaporator
G07	Angin keluar lemah dari indoor
G08	AC tidak mau menyala
G09	Lampu indikator berkedip (eror)
G10	AC tidak mau cut – off (terus menyala)
G11	Outdoor fan tidak berputar
G12	Bau tidak sedap dari indoor
G13	AC sering mati sendiri
G14	Pipa kecil membeku
G15	Arus listrik kompresor tinggi (overload)

b. Data Kerusakan

Data jenis kerusakan di bawah ini di peroleh melalui proses pengumpulan data dari ahli atau tahu pakar teknisi AC. Tabel 3.2 berikut merangkum poin-poin kerusakan yang menjadi hasil deteksi akhir sistem setelah memproses gejala – gejala yang diinput oleh pengguna.

Tabel 3. 2 Daftar kerusakan AC

Kode Kerusakan	Nama Kerusakan
K01	Freon habis/kurang
K02	Kompresor rusak/lemah
K03	Kondensor kotor
K04	Evaporator kotor
K05	Kapasitor rusak
K06	Thermistor Rusak
K07	Modul PCB bermasalah
K08	Pipa kapiler tersumbat

3.2.3 Basis Pengetahuan (Knowledge base)

Komponen basis pengetahuan ini berisi logika hubungan antara gejala dan kerusakan yang bersumber langsung dari pengetahuan praktis seorang ahli teknisi AC. Hubungan tersebut telah dianalisis pada sub-bab sebelumnya dan disusun menjadi matriks keputusan pada Tabel 3.3 untuk menunjukkan bagaimana pakar mengambil keputusan berdasarkan tanda-tanda kerusakan yang muncul.

Tabel 3. 3 Basis pengetahuan kerusakan AC

Kode Gejala	K01	K02	K03	K04	K05	K06	K07	K08
G01	✓	✓	✓	✓		✓		✓
G02	✓							
G03		✓			✓			
G04		✓	✓					✓
G05				✓				
G06	✓							
G07				✓				
G08					✓		✓	
G09							✓	
G10						✓		

G11 G12				✓	✓			
G13						✓	✓	
G14								✓
G15								

3.2.4 Studi Pustaka

Studi Pustaka dilakukan dengan mengumpulkan berbagai referensi – referensi dan informasi pendukung dari berbagai sumber seperti internet, jurnal yang terkait dengan sistem pakar, *Certainty* Faktor.

3.3 Algoritma Sistem

Pada bagian ini menjabarkan secara rinci alur logika algoritma *Certainty Factor* (CF) yang diimplementasikan dalam mesin inferensi sistem untuk mengolah data gejala menjadi kesimpulan kerusakan. Proses perhitungan dilakukan melalui tahapan-tahapan berikut :

3.3.1 Akuisisi Nilai Kepastian Pakar (MB dan MD)

Langkah pertama adalah menentukan nilai *Measure of Belief* (MB) dan *Measure of Disbelief* (MD) melalui wawancara terstruktur dengan pakar teknisi AC. MB menggambarkan ukuran kenaikan kepercayaan terhadap suatu kerusakan

jika ditemukan gejala tertentu, sedangkan MD menggambarkan ukuran kenaikan ketidakpercayaan. Pakar memberikan penilaian dalam rentang 0 sampai 1 berdasarkan skala terminologi linguistik berikut:

Tabel 3. 4 Skala Terminologi Pakar

Terminologi Pakar	Nilai MB	Nilai MD
Sangat Pasti	1.0	0.0
Hampir Pasti	0.8	0.1
Kemungkinan Besar	0.6	0.2
Mungkin	0.4	0.4
Sedikit Yakin	0.2	0.6
Tidak Tahu	0.0	0.0

Nilai faktor kepastian dari pakar (CF_{pakar}) untuk setiap aturan dihitung dengan rumus:

$$CF_{pakar}(H, E) = MB(H, E) - MD(H, E) \dots \dots \dots (3.1)$$

3.3.2 Penentuan Nilai Keyakinan Pengguna (CF User)

Saat melakukan deteksi di lapangan, teknisi pemula diminta memilih gejala yang ditemukan dan memberikan tingkat keyakinan mereka terhadap gejala tersebut (CF_{user})

Tabel 3. 5 Nilai CF User

No	Keterangan	Nilai CF User
1	Tidak	0

2	Tidak Tahu	0.1-0.2
3	Mungkin	0.3-0.4
4	Sedikit Yakin	0.5-0.6
5	Hampir Yakin	0.7-0.9
6	Sangat Yakin	0.9-1

3.3.3 Perhitungan Nilai CF gejala Tunggal

Sistem akan menghitung nilai CF untuk setiap gejala yang dipilih oleh pengguna dengan mengalikan bobot kepastian pakar dan bobot keyakinan pengguna:

$$CF_{gejala} = CF_{pakar} \times CF_{user} \dots\dots\dots (3.2)$$

Proses ini menghasilkan nilai kepastian awal untuk satu gejala terhadap satu kemungkinan kerusakan.

3.3.4 Kombinasi Nilai CF (CF Combine)

Mengingat satu kerusakan AC sering kali ditandai oleh lebih dari satu gejala, sistem harus menggabungkan nilai-nilai CF_{gejala} yang dihasilkan. Jika terdapat dua gejala (CF_1 dan CF_2) yang mengarah pada kerusakan yang sama, maka digunakan rumus kombinasi sekuensial :

$$CF_{combine}(CF_1, CF_2) = CF_1 + CF_2 \times (1 - CF_1) \dots\dots\dots (3.3)$$

Jika terdapat gejala ketiga (CF_3), maka nilai $CF_{combine}$ sebelumnya (disebut sebagai CF_{old}) akan dikombinasikan kembali dengan nilai gejala baru tersebut (CF_{new}):

$$CF_{total} = CF_{old} + CF_{new} \times (1 - CF_{old}) \dots\dots\dots (3.4)$$

Proses ini terus berulang hingga seluruh gejala yang dipilih pengguna selesai diproses oleh mesin inferensi. 3.3.5 Perhitungan Persentase Deteksi

Hasil akhir dari perhitungan kombinasi (CF_{total}) dikonversi menjadi nilai persentase untuk memudahkan interpretasi bagi teknisi pemula.

$$CF_{persentase} = CF_{total} \times 100\% \dots \dots \dots (3.5)$$

Kerusakan dengan nilai persentase tertinggi akan ditampilkan sebagai hasil deteksi.

3.3.6 Analisis Perhitungan Metode Certainty Faktor

Di bagian ini, dilakukan simulasi perhitungan secara manual dengan memakai algoritma *Certainty Factor*. Perhitungan ini dimaksudkan sebagai verifikasi untuk memastikan bahwa output yang dihasilkan oleh sistem sesuai dengan teori yang diterapkan.

3.3.7 Contoh kasus deteksi

Sebagai contoh kasus, diasumsikan seorang teknisi pemula (user) melakukan konsultasi pada sistem dan memilih gejala yang mengarah pada kerusakan “Freon habis/kurang” dimana gejala nya adalah sebagai berikut :

G01 (AC tidak dingin) dimana nilai CF pakar : 0.6

G02 (Tekanan freon rendah) dimana nilai CF pakar : 0.8

Langkah – langkah perhitungan :

- a. Menghitung nilai CF untuk masing – masing gejala

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

$$= 0.6 * 0.4$$

$$= 0.24$$

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

$$= 0.8 * 0.7$$

$$= 0.56$$

b. Mengkombinasikan Nilai CF (*Combine*)

$$CF_{combine}(CF_1, CF_2) = CF_1 + CF_2 \times (1 - CF_1)$$

$$= 0.24 + 0.56 * (1 - 0.24)$$

$$= 0.24 + (0.56 * 0.76)$$

$$= 0.24 + 0.4256$$

$$= 0.6656$$

c. Hasil akhir (persentase)

$$CF_{persentase} = CF_{total} \times 100\%$$

$$= 0.6656 * 100 \%$$

$$= 66.56 \%$$

Contoh kasus lain yaitu “Kompresor rusak” dimana gejalanya adalah sebagai berikut :

G01 (AC tidak dingin) dimana nilai CF pakar : 0.6

G03(Kompresor berdengung keras) dimana nilai CF pakar : 0.8

G04 (Kompresor overheat/panas) dimana nilai CF pakar : 0.9

Langkah – langkah perhitungan :

a. Menghitung nilai CF setiap gejala

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

$$= 0.6 * 0.2$$

$$= 0.12$$

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

$$= 0.8 * 0.1$$

$$= 0.08$$

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

$$= 0.9 * 0$$

$$= 0.9$$

b. Mengkombinasikan nilai CF (*Combine*)

$$CF_{combine}(CF_1, CF_2) = CF_1 + CF_2 \times (1 - CF_1)$$

$$= 0.12 + 0.08 * (1 - 0.12)$$

$$= 0.12 + (0.08 * 0.88)$$

$$= 0.12 + 0.0704$$

$$= 0.1904$$

$$CF_{total} = CF_{old} + CF_{new} \times (1 - CF_{old})$$

$$= 0.1904 + 0.9 * (1 - 0.1904)$$

$$= 0.1904 + (0.9 * 0.8096)$$

$$= 0.1904 + 0.72864$$

$$= 0.91904$$

c. Hasil akhir persentase

$$CF_{persentase} = CF_{total} \times 100\%$$

$$= 0.91904 * 100\%$$

$$= 91.904 \%$$

3.4 Pemodelan dan Perancangan Sistem

Bagian ini menjelaskan model perancangan yang digunakan sebagai blueprint pengembangan sistem pakar berbasis web. Perancangan dilakukan secara komprehensif mulai dari analisis kebutuhan, pemodelan visual, hingga struktur data.

3.4.1 Analisis Kebutuhan Sistem

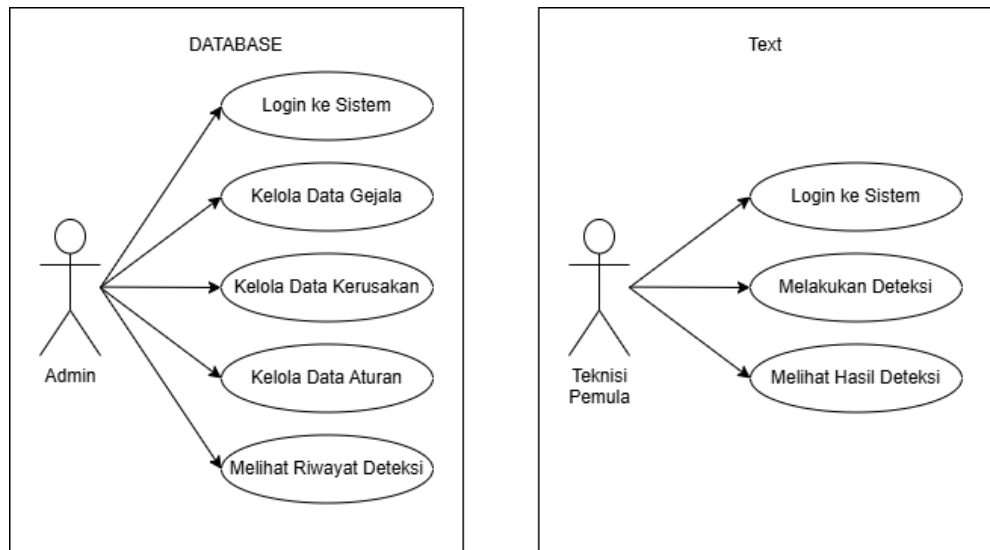
Sebelum melakukan pemodelan, peneliti melakukan identifikasi kebutuhan sistem yang dibagi menjadi dua bagian utama:

- a. Kebutuhan Fungsional: Mencakup kemampuan sistem untuk mengolah basis pengetahuan (gejala, kerusakan, aturan), menyediakan modul konsultasi bagi teknisi, melakukan perhitungan algoritma *Certainty Factor*, dan menampilkan hasil deteksi beserta persentase keyakinan.
- b. Kebutuhan Non-Fungsional: Mencakup ketersediaan sistem berbasis web yang responsif, keamanan akses melalui login aktor (admin, pakar, user), serta kemudahan penggunaan (*usability*) bagi teknisi di lapangan.

3.4.2 Pemodelan dengan Unit Modelling Language (UML)

a. Use Case Diagram

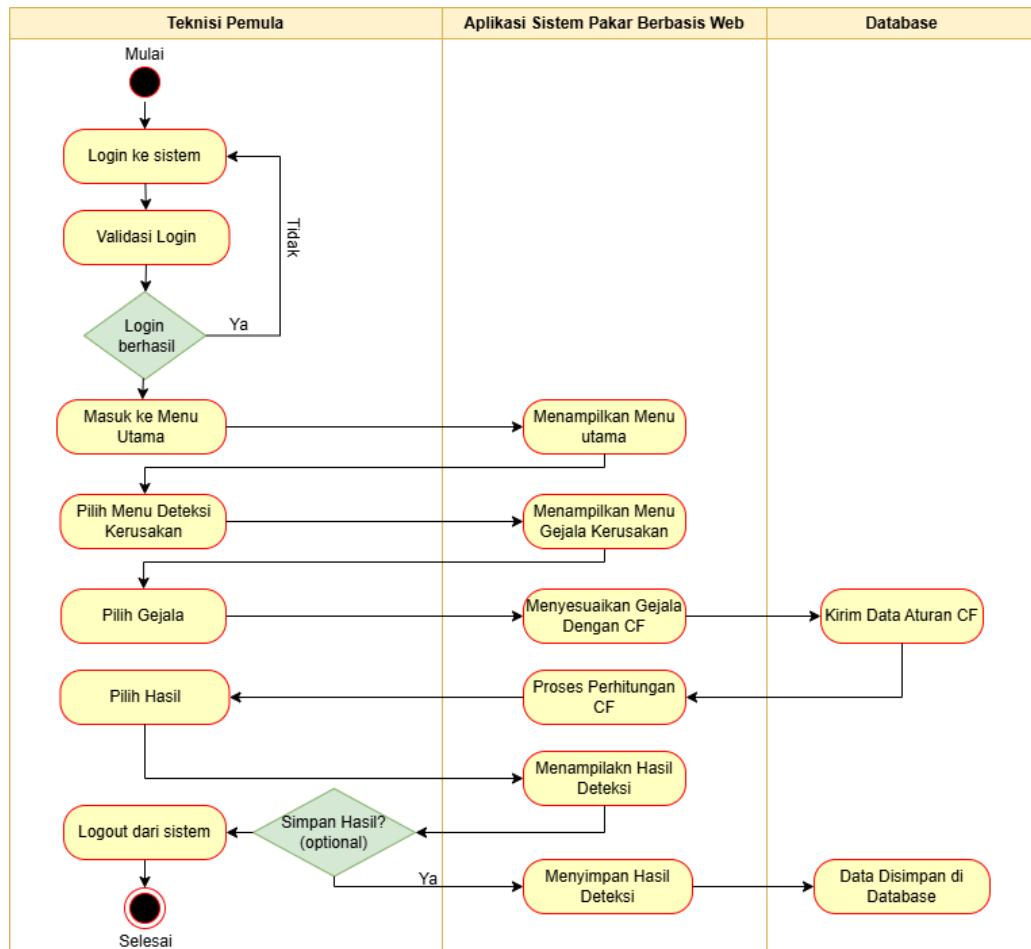
Diagram ini membagi peran aktor menjadi tiga yaitu , Admin mengelola data pengguna, Pakar memiliki otoritas penuh untuk menambah atau mengubah gejala, kerusakan, dan nilai MB/MD, sedangkan Teknisi Pemula (*User*) dapat melakukan input gejala konsultasi, melihat riwayat deteksi, dan melihat hasil deteksi.



Gambar 3. 3 Use case diagram

b. Activity Diagram

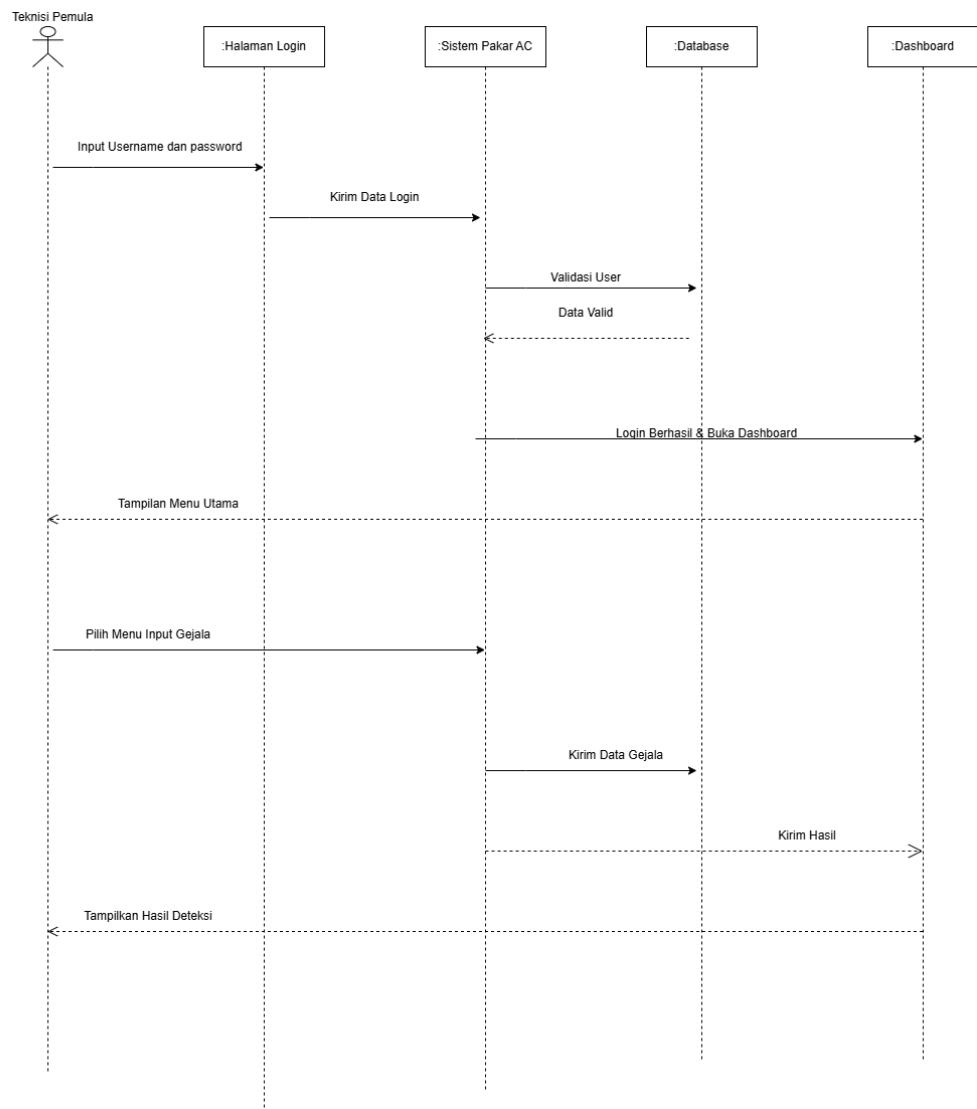
Menggambarkan alur kerja proses deteksi. Dimulai dari user login, memilih gejala teknis AC, sistem memvalidasi gejala dengan basis pengetahuan, melakukan kalkulasi *Certainty Factor* (perkalian *CF User* dan *CF Pakar* serta penggabungan (*CF Combine*), dan berakhir pada penampilan hasil deteksi tertinggi.



Gambar 3. 4 Activity diagram

c. Sequence Diagram

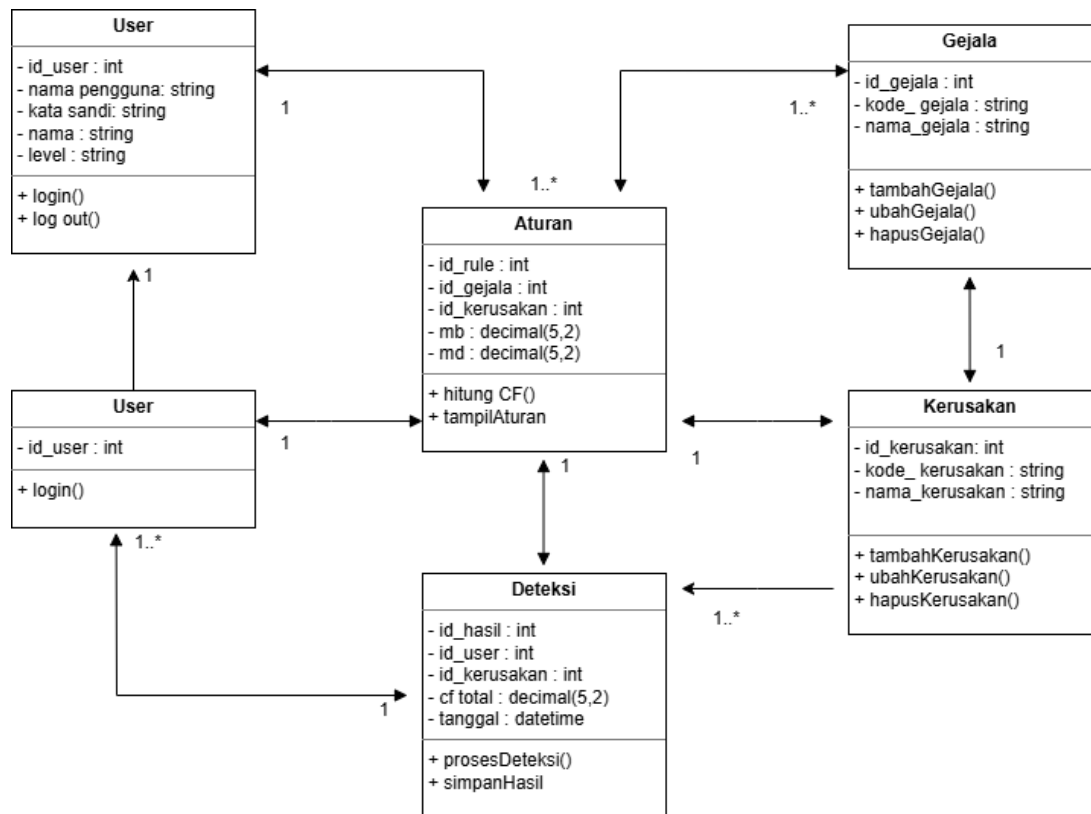
Menggambarkan alur interaksi dan urutan komunikasi antara aktor serta objek-objek yang ada dalam sistem pakar pendeteksi kerusakan AC berbasis web. Sequence diagram ini menggambarkan cara teknisi pemula sebagai pengguna sistem berinteraksi dengan sistem dari tahap login hingga sistem menunjukkan hasil deteksi kerusakan AC.



Gambar 3. 5 Sequence diagram

d. Class Diagram

Menggambarkan struktur statis sistem dengan menunjukkan kelas-kelas (*classes*) utama seperti Gejala, Kerusakan, Aturan, dan Hasil deteksi, beserta atribut dan hubungan asosiasi antar kelas tersebut.



Gambar 3. 6 Class diagram

3.4.3 Perancangan Basis Data (Database Design)

Dalam sistem pakar berbasis aturan, desain basis data harus mampu merepresentasikan hubungan kompleks antara gejala dan kerusakan. Menggunakan prinsip *relational database design*, berikut adalah skema tabel yang dirancang :

Tabel 3. 6 Rancangan Tabel Kerusakan

Tabel ini menyimpan data jenis kerusakan.

Field	Tipe Data	Keterangan
id_kerusakan	Integer (PK)	Identifikasi unitk kerusakan
kode_kerusakan	Varchar (5)	Kode (misal: K01, K02)
nama_kerusakan	Varchar (100)	Nama kerusakan (misal: Kompresor Overheat)

Tabel 3. 7 Rancangan Tabel Gejala

Tabel ini menyimpan data gejala kerusakan AC.

Field	Tipe Data	Keterangan
id_gejala	Integer (PK)	Identifikasi unik gejala
kode_gejala	Varchar (5)	Kode (misal: G01, G02)
nama_gejala	Varchar (255)	Deskripsi Gejala Fisik

Tabel 3. 8 Rancangan Tabel Aturan

Tabel ini berfungsi Menghubungkan gejala dan kerusakan serta menyimpan nilai MB dan MD dari pakar.

Field	Tipe Data	Keterangan
id_rule	Integer (PK)	Identifikasi unik aturan
id_kerusakan	Integer (FK)	Referensi ke tabel kerusakan
ide_gejala	Integer (FK)	Referensi ke tabel gejala
mb	Float	Nilai <i>Measure of Belief</i> (0.0 – 1.0)
md	Float	Nilai <i>Measure of Disbelief</i> (0.0 – 1.0)

Tabel 3. 9 Rancangan Tabel Hasil

Tabel ini Menyimpan riwayat deteksi user beserta nilai persentase CF.

Field	Tipe Data	Keterangan
id_hasil	Integer (PK)	Nomor identitas unik hasil diagnosa
id_user	Integer (FK)	Relasi ke tabel pengguna (teknisi yang melakukan konsultasi)
Tanggal	Datetime	Waktu proses diagnosis dilakukan
Id_kerusakan	Integer (FK)	Relasi ke tabel kerusakan hasil deteksi
Cf_total	Decimal (5,2)	Nilai total keyakinan hasil deteksi (0-1 atau dalam %)

3.4.4 Design User Interface

Design User Interface merupakan rancangan antar muka yang akan digunakan untuk membuat tampilan dalam perangkat lunak yang dikembangkan. Tujuan design *User Interface* untuk membuat rancangan antar muka pada Sistem Pakar Pendeteksi Kerusakan AC Menggunakan *Metode Certainty Faktor* Untuk Mendukung Teknisi Pemula. Adapun *user interface* nya sebagai berikut :

a. Halaman Login

Halaman login berperan sebagai pintu masuk utama untuk mengakses sistem, di mana pengguna harus mengisi username dan password yang valid. Admin dan teknisi yang baru akan diarahkan ke halaman dashboard sesuai dengan hak akses yang mereka miliki.

SISTEM PAKAR PENDETEKSI KERUSAKAN AC

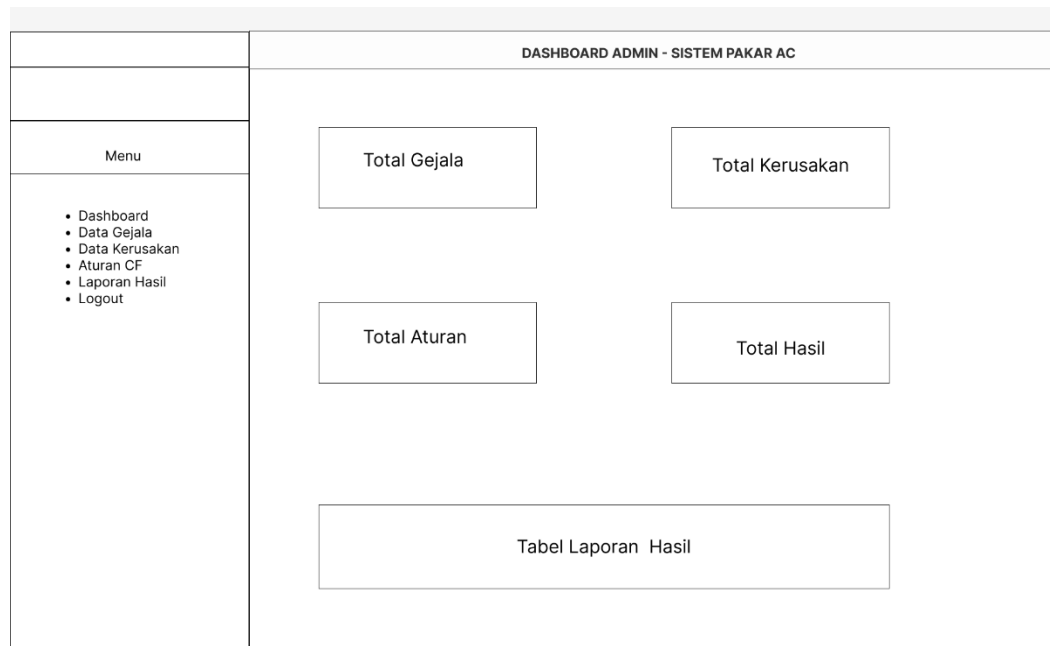
Password :

Username :

Gambar 3. 7 Tampilan Login

b. Halaman Dashboard Admin

Halaman dashboard admin menyajikan ringkasan informasi sistem, termasuk total gejala, total kerusakan, jumlah aturan CF, serta jumlah hasil deteksi yang tersimpan. Fungsi utama dari halaman ini adalah sebagai panel kontrol bagi admin untuk mengawasi semua aktivitas dalam sistem pakar.



Gambar 3. 8 Halaman Dashboard Admin

c. Halaman Kelola Data Gejala

Halaman ini dimanfaatkan oleh admin untuk mengatur basis pengetahuan sistem, terutama bagian gejala.

The image shows a form titled "Kelola Data Gejala". It contains three input fields with labels: "Kode Gejala :", "Nama Gejala :", and "Deskripsi :". Below the input fields are two buttons: "Simpan" and "Batal".

Gambar 3. 9 Tampilan Kelola Data Gejala

d. Halaman Dashboard User

Halaman dasbor pengguna berperan sebagai halaman utama bagi teknisi baru setelah masuk ke sistem. Tampilan ini menyediakan akses cepat ke fitur Deteksi Gejala dan Riwayat Deteksi.

DASHBOARD USER - SISTEM PENDETEKSI KERUSAKAN AC	
Menu	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Hasil Deteksi Akhir </div> Kerusakan : xxxxxx Persentase Keyakinan : xxxx
<ul style="list-style-type: none"> • Dashboard • Data Gejala • Riwayat Deteksi • Logout 	<div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="border: 1px solid black; padding: 5px 15px;">Mulai Deteksi Baru</div> <div style="border: 1px solid black; padding: 5px 15px;">Lihat Riwayat</div> </div>

Gambar 3. 10 Tampilan Dashboard User

e. Tampilan Deteksi Gejala

Halaman ini dimanfaatkan oleh teknisi pemula untuk menentukan gejala-gejala yang tampak pada unit AC. Sistem akan memanfaatkan input gejala tersebut sebagai landasan perhitungan metode *Certainty Factor* (CF) untuk menetapkan tingkat kepastian terhadap kemungkinan kerusakan.

DETEKSI GEJALA - SISTEM PAKAR PENDETEKSI KERUSAKAN AC	
MENU	Pilih Gejala Yang Terjadi Pada AC
<ul style="list-style-type: none"> • Dashboard • Deteksi Gejala • Riwayat Deteksi • Logout 	<ul style="list-style-type: none"> • XXXXXXXXXXXX • XXXXXXXXXXXX • XXXXXXXXXXXX • XXXXXXXXXXXX • XXXXXXXXXXXX <div style="text-align: right; margin-top: 20px;"> <div style="border: 1px solid black; padding: 5px; display: inline-block;">Proses Deteksi</div> </div>

Gambar 3. 11 Tampilan Deteksi Gejala

f. Tampilan Hasil Deteksi

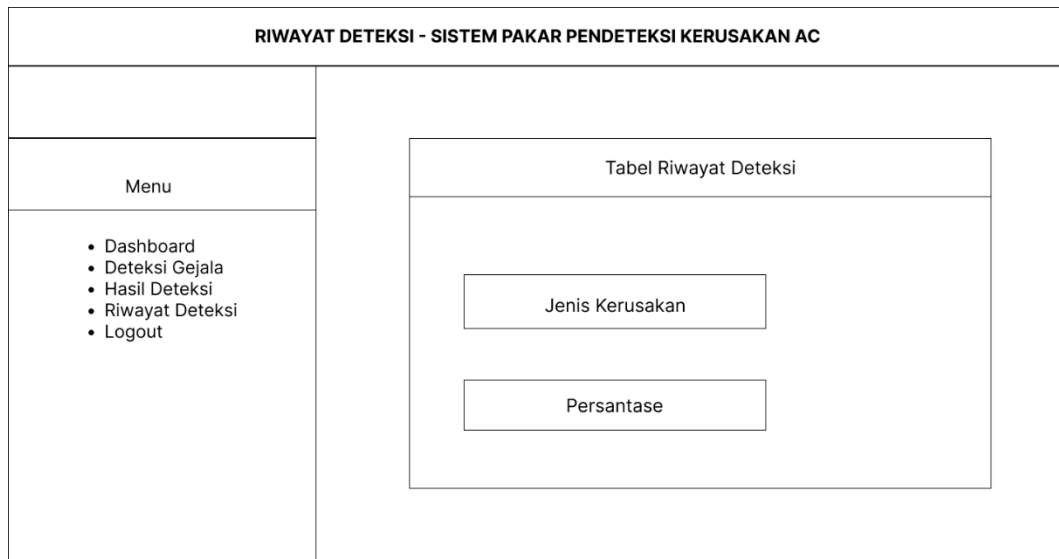
Halaman ini menunjukkan hasil analisis sistem pakar mengenai jenis kerusakan AC yang paling berpotensi terjadi beserta persen tingkat keyakinan (nilai CF).

HASIL DETEKSI - SISTEM PAKAR PENDETEKSI KERUSAKAN AC	
Menu	Hasil Analisis Sistem
<ul style="list-style-type: none"> • Dashboard • Deteksi Gejala • Hasil Deteksi • Riwayat Deteksi • Logout 	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Kerusakan Yang Mungkin Terjadi : xxxxxxxxxxxx Tingkat Keyakinan : xxxxxxxx </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Rician Gejala Yang Dipilih : <ul style="list-style-type: none"> • XXXXXXXX • XXXXXXXX </div> <div style="text-align: right; margin-top: 20px;"> <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-right: 20px;">Simpan Hasil</div> <div style="border: 1px solid black; padding: 5px; display: inline-block;">Deteksi Ulang</div> </div>

Gambar 3. 12 Tampilan Hasil Deteksi

g. Tampilan Riwayat Deteksi

Fitur ini mendukung teknisi dalam menilai hasil pekerjaan dan mengawasi pola kerusakan AC yang sering muncul.



Gambar 3. 13 Tampilan Riwayat Deteksi

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Kebutuhan Sistem

Pada bagian ini akan dibahas mengenai fasilitas pendukung yang digunakan untuk merealisasikan desain sistem yang telah disusun di bab sebelumnya. Pengadaan kebutuhan sistem ini sangat krusial untuk memastikan bahwa perangkat lunak yang dikembangkan dapat berfungsi sesuai dengan tujuan yang diharapkan. Kebutuhan itu dibagi ke dalam dua kategori utama, yaitu perangkat keras (*hardware*) yang mendukung kinerja komputasi dan perangkat lunak (*software*) sebagai dasar pengembangan sistem.

4.1.1 Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) yang digunakan untuk membuat website sistem pakar yaitu laptop.

4.1.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan dalam pengembangan sistem pakar berbasis web ini yaitu ;

- a. Sistem Operasi Windows 11 digunakan sebagai dasar utama sistem operasi pada perangkat keras. Pemilihan sistem operasi ini ditentukan oleh stabilitas serta dukungan yang luas untuk berbagai perangkat lunak pengembangan Python yang lainnya.

- b. Python digunakan sebagai bahasa pemrograman utama yang diterapkan dalam pengembangan ini. Python dipilih karena memiliki sintaks yang mudah dan efisien, serta didukung oleh berbagai pustaka (*library*) khusus yang mempermudah penerapan perhitungan matematis pada metode *Certainty Factor*.
- c. Flask bertindak sebagai *framework* sifatnya yang *micro-framework*, ringan, dan memberikan kontrol penuh terhadap routing URL serta integrasi *frontend* menggunakan HTML dan CSS. Hal ini memudahkan dalam merancang antarmuka yang lebih kustom sesuai dengan kebutuhan teknisi di lapangan.
- d. Visual Studio Code (VS Code) digunakan sebagai *Integrated Development Environment* (IDE) atau editor kode utama. Perangkat lunak ini digunakan untuk menulis, mengedit, dan melakukan debugging pada kode program Python agar berfungsi sesuai dengan logika yang telah dibuat.
- e. XAMPP memiliki peranan krusial dalam mengatur koneksi server lokal dan menjalankan layanan database MySQL, sehingga pengembang dapat membangun serta menguji sistem di lingkungan localhost sebelum diterapkan secara luas.
- f. MySQL digunakan sebagai sistem manajemen basis data relasional (*Relational Database Management System*) untuk menyimpan seluruh data pengetahuan sistem pakar.

- g. *Phpmyadmin* digunakan sebagai perangkat lunak berbasis web yang berfungsi untuk mengelola database MySQL secara visual (Antarmuka Pengguna Grafis)

4.1.3 Kebutuhan Data

Data yang dipakai dalam pembuatan sistem pakar ini adalah representasi pengetahuan yang didapat dari pakar (teknisi AC). Ringkasan dari kebutuhan data tersebut dapat dilihat pada Tabel 4.1 di bawah ini:

Tabel 4. 1 Kebutuhan data

No	Jenis Data	Kegunaan
1.	Data Kerusakan	Dipakai sebagai data hasil deteksi (<i>output</i>) yang akan ditunjukkan sistem setelah melaksanakan proses perhitungan
2.	Data Gejala	Dipakai sebagai parameter masukan (<i>input</i>) bagi pengguna untuk menentukan kondisi fisik atau tanda-tanda ketidaknormalan yang terjadi pada unit AC.
3.	Data Nilai CF	Digunakan sebagai nilai derajat keyakinan pakar terhadap suatu gejala, yang menjadi dasar utama dalam perhitungan algoritma <i>Certainty Factor</i> .
4.	Data Aturan	Digunakan untuk menghubungkan antara sekumpulan gejala dengan jenis gangguan tertentu (logika <i>If-Then</i>) agar sistem dapat melakukan penalaran.

4.2 Implementasi Sistem

Tahap implementasi adalah lanjutan dari hasil analisis dan desain yang telah dikerjakan sebelumnya. Pada tahap ini, desain sistem direalisasikan ke dalam bentuk nyata yang mencakup pengaturan basis data dan penulisan kode program

4.2.1 Implementasi Basis Data

Implementasi basis data dilakukan menggunakan RDBMS MySQL melalui antarmuka phpMyAdmin. Basis data ini berperan sebagai wadah penyimpanan pengetahuan (*knowledge base*) yang akan diproses oleh algoritma *Certainty Factor*. Berikut adalah detail tabel yang telah diterapkan:

a. Tabel Kerusakan (tb_kerusakan)

Tabel ini berfungsi untuk menyimpan data master kerusakan AC yang dapat dideteksi oleh sistem.

Tabel 4. 2 Tampilan tb_kerusakan

No	Nama Kolom	Jenis Data	Atribut
1.	id_kerusakan	int(11)	Primary Key
2.	kode_kerusakan	varchar(5)	-
3.	nama_kerusakan	varchar(100)	-
4.	deskripsi	text	-
5.	solusi	text	-

b. Tabel Gejala (tb_gejala)

Tabel ini berfungsi untuk menyimpan seluruh rincian gejala yang mungkin dialami oleh unit AC. Setiap gejala dilengkapi dengan kode unik (*Primary Key*) guna mempermudah hubungan data

Tabel 4. 3 Tampilan tb_gejala

No	Nama Kolom	Jenis Data	Atribut
1.	id_gejala	int(11)	Primary Key
2.	kode_gejala	varchar(5)	-
3.	nama_gejala	varchar(255)	-

c. Tabel Aturan (tb_aturan)

Tabel ini adalah representasi dari basis pengetahuan yang mengaitkan gejala dengan kerusakan. Dalam tabel ini juga tercatat nilai CF Pakar yang berfungsi sebagai bobot keyakinan seorang pakar mengenai hubungan antara gejala dan kerusakan tersebut.

Tabel 4. 4 Tampilan tb_aturan

No	Nama Kolom	Jenis Data	Atribut
1.	id_role	int(11)	Primary Key
2.	id_kerusakan	int(11)	index
3.	id_gejala	int(11)	index
4.	mb	float	-
5.	md	float	-

d. Tabel hasil (tb_riwayat hasil)

Tabel ini berfungsi untuk menyimpan hasil akhir dari proses perhitungan metode *Certainty Factor* yang sudah dilakukan oleh pengguna (user).

Tabel 4. 5 Tampilan tb_riwayat hasil

1.	id_hasil	int(11)	Primary Key
2.	id_user	int(11)	index
3.	id_kerusakan	int(11)	index
4.	cf_total	decimmal(5.4)	-
5.	tanggal	decimal	-
5	tanggal	decimal	-

e. Tabel user (tb_user)

Tabel ini merupakan akses awal ke sistem. Tanpa tabel ini, fitur login tidak dapat berfungsi untuk menyimpan data kredensial (hak akses) pengguna yang berwenang masuk ke sistem.

Tabel 4. 6 Tampilan tb_user

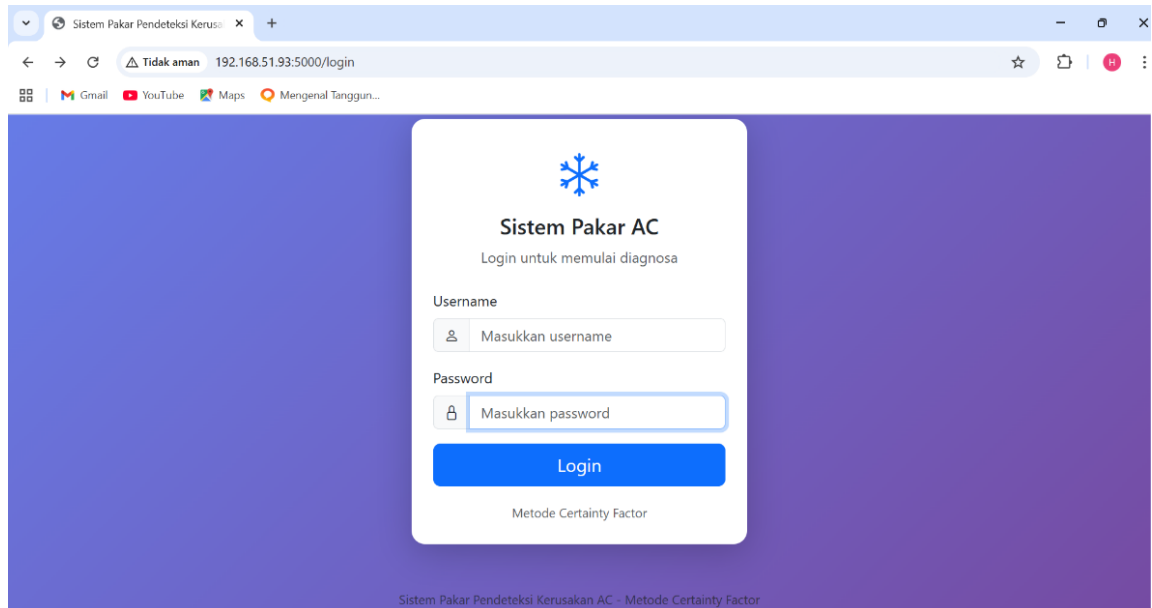
No	Nama Kolom	Jenis Data	Atribut
1.	id_user	int(11)	Primary Key
2.	nama	varchar(100)	-
3.	username	varchar(50)	index/unique
4.	password	varchar(255)	-
5.	role	enum(admin,pakar,user)	-
6.	created_at	timestamp	-

4.3 Hasil Pengujian Sistem

Pada bagian implementasi antarmuka berisi pemaparan setiap tampilan web sistem pakar pendeteksi AC yang dibangun beserta layout dari masing- masing antarmuka.

a. Halaman Login

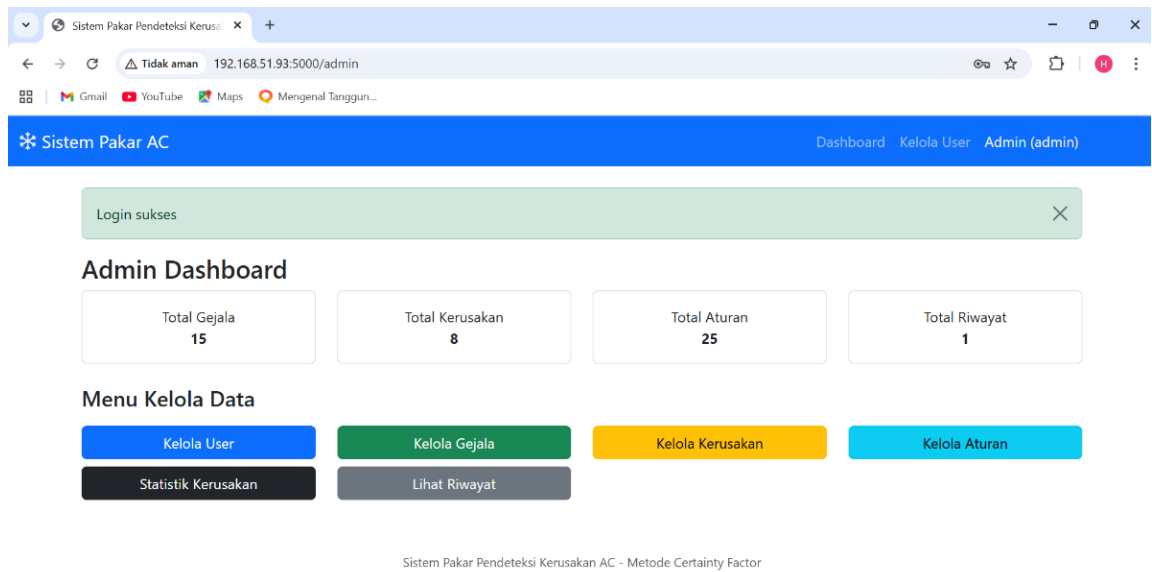
Halaman login merupakan antarmuka pertama yang diakses oleh pengguna. Fungsi utamanya adalah untuk memverifikasi identitas pengguna menggunakan username dan password.



Gambar 4. 1Tampilan Halaman login

b. Halaman Dashboard Admin

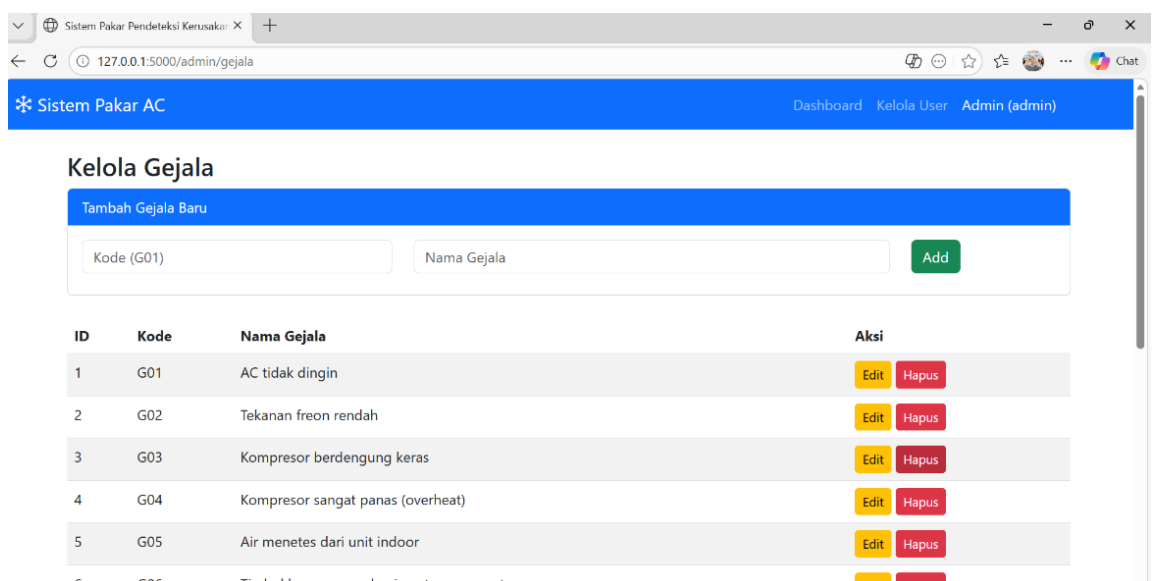
Halaman dashboard admin berfungsi untuk memberikan ringkasan data secara cepat (*overview*). Di sini ditampilkan total jumlah gejala, total kerusakan, dan jumlah aturan yang tersimpan dalam database.



Gambar 4. 2 Tampilan Dashboard Admin

c. Halaman Kelola Data Gejala

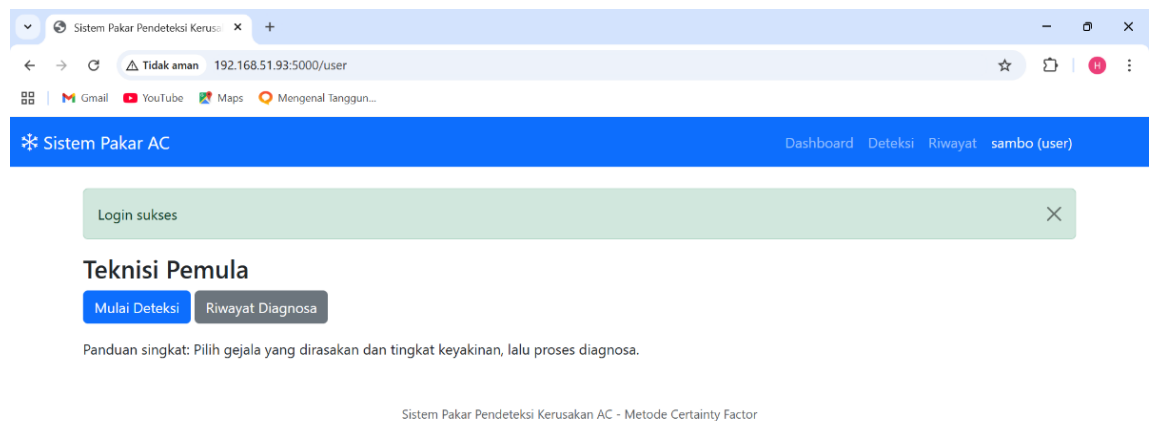
Halaman kelola data gejala digunakan oleh admin untuk melakukan manipulasi data (*CRUD: Create, Read, Update, Delete*). Admin dapat menambah gejala baru, mengubah deskripsi gejala, atau menghapus gejala yang sudah tidak relevan.



Gambar 4. 3 Tampilan Kelola Data Gejala

d. Halaman Dashboard User

Dashboard user dirancang khusus untuk teknisi AC pemula. Tampilan ini lebih sederhana dengan fokus pada dua menu utama: Mulai Deteksi untuk memulai proses diagnosa baru dan Riwayat Deteksi untuk melihat hasil diagnosa yang pernah dilakukan sebelumnya.



Gambar 4. 4 Tampilan Dashboard User

e. Tampilan Deteksi Gejala

Halaman deteksi gejala adalah tempat teknisi memasukkan observasi lapangan. Teknisi akan memilih tingkat keyakinan (misalnya: 'Sangat Yakin', 'Mungkin', atau 'Tidak Tahu') untuk setiap gejala yang muncul pada AC.

Sistem Pakar AC

Dashboard Deteksi Riwayat sambo (user)

Deteksi Gejala AC

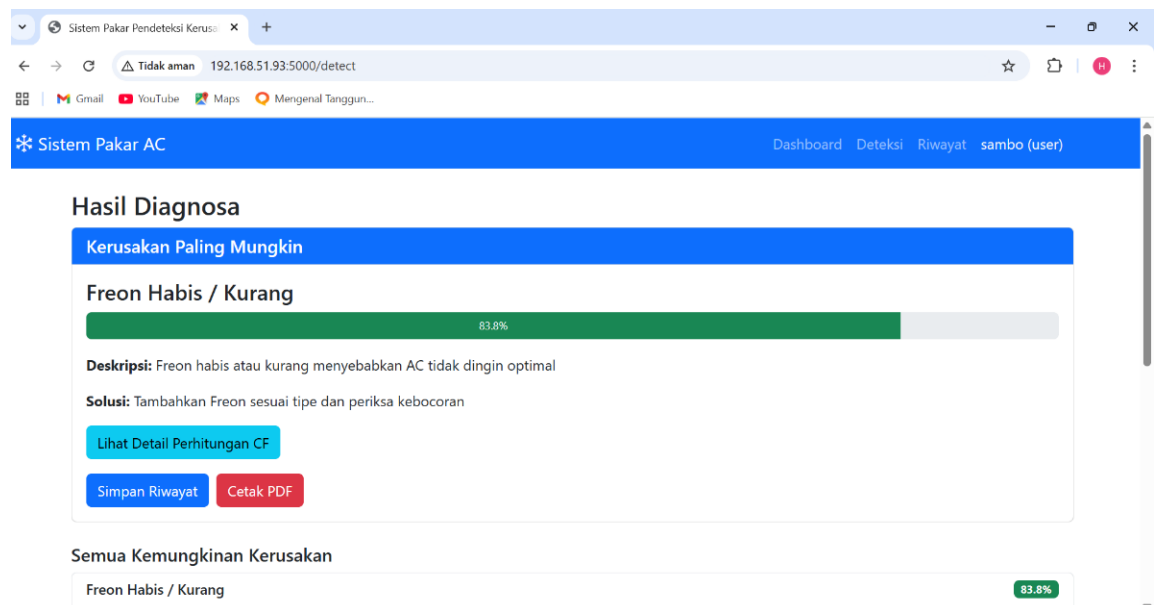
Pilih gejala yang dirasakan dan tingkat keyakinan Anda:

- G01 - AC tidak dingin**
Tingkat Keyakinan: Tidak / Tidak Ada (0)
- G02 - Tekanan freon rendah**
Tingkat Keyakinan: Tidak / Tidak Ada (0)
- G03 - Kompresor berdengung keras**
Tingkat Keyakinan: Tidak / Tidak Ada (0)
- G04 - Kompresor sangat panas (overheat)**
Tingkat Keyakinan: Tidak / Tidak Ada (0)
- G05 - Air menetes dari unit indoor**
Tingkat Keyakinan: Tidak / Tidak Ada (0)
- G06 - Timbul bunga es pada pipa atau evaporator**
Tingkat Keyakinan: Tidak / Tidak Ada (0)

Gambar 4. 5 Tampilan Deteksi Gejala

f. Tampilan Hasil Deteksi

Halaman hasil deteksi menampilkan deteksi akhir berupa jenis kerusakan yang paling dominan beserta persentase tingkat keyakinan.



Gambar 4. 6 Tampilan Hasil Deteksi

g. Tampilan Riwayat Deteksi

Pada halaman riwayat deteksi, teknisi dapat melihat kembali laporan perbaikan AC di masa lalu sebagai bahan referensi.



Sistem Pakar AC

Dashboard Deteksi Riwayat sambo (user)

Riwayat Diagnosa

No	Tanggal	Nama Teknisi	Kerusakan	Persentase	Aksi
1	2026-03-05 13:50:05	sambo	Freon Habis / Kurang	70.00%	Hapus

Sistem Pakar Pendeteksi Kerusakan AC - Metode Certainty Factor

Gambar 4. 7 Tampilan Riwayat Deteksi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian, perancangan, dan pengujian yang telah dilakukan pada Sistem Pakar Pendeteksi Kerusakan AC dengan metode *Certainty Factor*, maka dapat ditarik kesimpulan sebagai berikut:

1. Sebuah sistem pakar berbasis web telah berhasil dirancang dan diimplementasikan dengan menggunakan bahasa pemrograman Python dan framework Flask yang dapat mendeteksi jenis kerusakan pada unit AC berdasarkan gejala yang dipilih oleh pengguna.
2. Metode *Certainty Factor* (CF) terbukti efektif dalam mengatasi ketidakpastian saat mendeteksi kerusakan AC. Dengan mengintegrasikan bobot ahli dan bobot kepercayaan pengguna, sistem mampu menghasilkan output akhir berupa tingkat kepercayaan dalam persen mengenai kemungkinan terjadinya kerusakan.
3. Fitur riwayat dan hasil deteksi membuat sistem ini berguna bagi teknisi baru dalam mengenali kerusakan AC serta memberikan informasi solusi awal yang bisa dilakukan sebelum perbaikan lebih lanjut.
4. Penerapan basis data MySQL melalui phpMyAdmin mempermudah pengelolaan basis pengetahuan seperti informasi gejala dan kerusakan secara teratur, sehingga data bisa ditambahkan, diperbarui, atau dikelola oleh admin dengan lebih gampang di kemudian hari.

5. Sistem pakar yang dikembangkan dapat memberikan informasi hasil diagnosis dengan cepat dan teratur, sehingga bisa berfungsi sebagai alat bantu dalam proses pengambilan keputusan terkait identifikasi kerusakan AC tanpa selalu bergantung pada teknisi berpengalaman.
6. Melalui sistem ini, diharapkan proses identifikasi kerusakan AC dapat dilaksanakan dengan lebih efektif dan efisien, serta dapat menjadi wadah pembelajaran bagi teknisi pemula dalam memahami hubungan antara gejala dan jenis kerusakan pada AC.

5.2 Saran

Meskipun sistem ini telah berfungsi dengan baik, namun masih terdapat ruang untuk pengembangan lebih lanjut. Adapun saran yang dapat diberikan adalah:

1. Kedepannya, sistem ini dapat dikembangkan menjadi aplikasi berbasis Android agar lebih praktis digunakan oleh teknisi saat berada di lapangan (lokasi perbaikan).
2. Perlu dilakukan pembaruan (*update*) data gejala dan jenis kerusakan secara berkala mengikuti perkembangan teknologi unit AC terbaru (seperti teknologi Inverter atau *Smart AC*).
3. Penambahan fitur gambar atau video panduan perbaikan pada halaman hasil diagnosa akan sangat membantu teknisi dalam melakukan tindakan perbaikan setelah kerusakan diketahui.
4. Sistem pakar ini bisa diperluas dengan menambah data kerusakan dan gejala, agar hasil diagnosa yang dihasilkan menjadi lebih komprehensif dan tepat.

5. Dalam penelitian berikutnya, metode *Certainty Factor* bisa dipadukan dengan metode lain, seperti *Forward Chaining*, *Naïve Bayes*, atau metode kecerdasan buatan lainnya untuk membandingkan akurasi hasil deteksi sistem.
6. Pengembangan sistem berikutnya dapat menyertakan fitur manajemen pengguna yang lebih komprehensif, seperti pengelolaan akun teknisi atau admin, sehingga sistem bisa diakses oleh lebih banyak pengguna secara terorganisir.
7. Dalam pengembangan berikutnya, sistem ini dapat dibuat agar dapat diakses oleh masyarakat luas, sehingga tidak terbatas hanya untuk admin, ahli, atau teknisi saja. Dengan cara ini, pengguna biasa yang menghadapi masalah pada AC dapat melakukan deteksi awal melalui sistem sebelum menghubungi teknisi untuk melakukan perbaikan lebih lanjut.

DAFTAR PUSTAKA

- Afriansyah, D., & Mawarni, R. (2022). *Sistem Pakar Kerusakan Air Conditioner Berbasis Web Pada Jaya Service AC*. 2(1).
- Afwan, R., Astuti, I. F., Cahyadi, D., Teknik, F., & Mulawarman, U. (2022). *Implementasi Metode Certainty Factor Untuk Diagnosis Kerusakan Software Pada Smartphone Berbasis Web*. 17(2).
- Agustinus Budi Santoso, S.St., M. C. (n.d.). *Pemograman WEB PHP Dasar*. Penerbit Widina Bhakti Persada Bandung.
- Chandra, A. Y., & Setyaningsih, P. W. (2025). *BULLETIN OF COMPUTER SCIENCE RESEARCH Benchmarking Local Development Environments: Analyzing the Performance of XAMPP, MAMP, and Laragon*. 5(3).
<https://doi.org/10.47065/bulletincsr.v5i3.493>
- Firdaus, R., Martini, N., Utomo, G. P., & Avillah, S. H. (2025). *Mekanika : Jurnal Teknik Mesin ANALISA PERFORMANSI DAN KARAKTERISTIK MESIN PENDINGIN AC SPLIT R-32 DENGAN VARIASI ALAT EKSPANSI*. 11.
- Latumakulita, L. A. (2021). *SISTEM PAKAR PENDIAGNOSA PENYAKIT ANAK MENGGUNAKAN CERTAINTY FACTOR (CF) EXPERT SYSTEM FOR DIAGNOSING CHILD DISEASE*.
- Mechael, J., Lumbantoruan, P., & Purnomo, A. S. (2025). *Rancang Bangun Sistem Pakar Berbasis Website Menggunakan Metode Certainty Factor Untuk Mendiagnosa Gangguan Tidur*. 5(2).
- Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). *Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS) Universitas Nasional Karangturi*

- Semarang , Indonesia (deskripsi) dan perancangan sistem , khususnya pada pemrograman berorientasi objek (Nistrina. 3, 244–256.
- Nofrianis, D. (2022). *SISTEM PAKAR MENDETEKSI KERUSAKAN PADA AIR CONDITIONER (AC) DI CV . NETRAL SERVICE DENGAN*. 7(1).
- Nugraha, M., & Rosmeida, M. (2021). *Perancangan Sistem Informasi Beban Kerja Dosen Berbasis Web dengan UML*. 1, 141–150.
- Nurlita, I., Anggraini, R., Akuntansi, S. I., & Gunadarma, U. (2023). *Analysis and Design of Incoming and Outgoing Cash Accounting Information Systems at Kilometer 28 Laundry using the Pieces and Waterfall Methods with Unified Modeling Language (UML) Tools Analisis dan Perancangan Sistem Informasi Akuntansi Kas Masuk dan Keluar pada Laundry Kilometer 28 menggunakan Metode Pieces dan Waterfall dengan Unified Modelling Language (UML) Tools*. 2(6), 1065–1090.
- Pangestu, P. R., Voutama, A., Informasi, S., & Karawang, U. S. (2024). *PEMANFAATAN UML (UNIFIED MODELLING LANGUAGE) PADA SISTEM*. 8(6), 11846–11851.
- Ramadhan, P. S., Fatimah, U., & Pane, S. (2018). *Analisis Perbandingan Metode (Certainty Factor , Dempster Shafer dan Teorema Bayes) untuk Mendiagnosa Penyakit Inflamasi Dermatitis Imun pada Anak*. 17(2), 151–157.
- Roy, Yohanni Syahra, R. I. (2021). *Sistem pakar mendiagnosa penyakit infeksi saluran kemih dengan menggunakan metode teorema bayes*. 4(5), 1–17.
- Saputra, T. A. (2025). *Mengenal AC Split: Jenis, Cara Kerja & Perbedaannya dengan AC Inverte*. *AQUA ELEKTRONIK INDONESIA*.
<https://aquaelektronik.com/article/detail/771/mengenal-ac-split>
- Sinlae, F., Maulana, I., Setiyansyah, F., & Ihsan, M. (2024). *Pengenalan Pemrograman Web : Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL*. 2(2), 68–82.
- Suryaningrat, S.Kom., M. K. (2024). *Python Fundamental*. GANESHA KREASI

SEMESTA.

- Syarifuddin, M. Y., & Suryadi, A. (2025). *SISTEM PAKAR DIAGNOSIS KERUSAKAN MOTOR MATIC HONDA MENGGUNAKAN METODE CERTAINTY FACTOR*. 9(4), 1536–1546. <https://doi.org/10.52362/jisamar.v9i4.2118>
- Thoriq Achmad Alawi, M Ghofar Rohman, M. (2025). *Implementasi Sistem Pakar Diagnosa Kerusakan Televisi Berbasis Web Menggunakan Metode Certainty Factor*. 18(2), 2580–2582.
- Utomo, Y. B. (2021). *APLIKASI SISTEM PAKAR DALAM MENDETEKSI KERUSAKAN AC RUMAH BERBASIS ANDROID DENGAN MENGIMPLEMENTASIKAN*. 4, 175–182.
- Wayahdi, R., & Ruziq, F. (2023). *Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language (UML) (Studi Kasus : Programmer Association of Battuta)*. 12, 1514–1521.
- Yogi Ainur Rofiq Anggara, Suryo Adi Wibowo, Y. A. P. (2025). *Sistem Pakar Diagnosa Kerusakan Laptop Berbasis Forward Chaining dan Certainty Faktor*.
- Yusuf, I. (2021). *Arsitektur Sistem Pakar: Konsep Sistem Pkar*. (Cetakan ke). PT. LAMINTANG. <https://doi.org/10.31219/osf.io/h7t3r>

LAMPIRAN

1. Source code app.py

```

from flask import Flask, render_template, request, redirect, url_for, session, flash, send_file
from Werkzeug.security import check_password_hash, generate_password_hash
from config import SECRET_KEY
from db import query, execute
from cf import cf_single, combine_list
import math

app = Flask(__name__)
app.secret_key = SECRET_KEY

# Session timeout - 30 minutes
app.config['PERMANENT_SESSION_LIFETIME'] = 30 * 60

# Login required decorator
def login_required(role=None):
    def decorator(f):
        from functools import wraps
        @wraps(f)
        def decorated_function(*args, **kwargs):
            if 'user' not in session:
                flash('Silakan login terlebih dahulu', 'warning')
                return redirect(url_for('login'))
            if role and session['user']['role'] != role:
                flash('Akses ditolak', 'danger')
                return redirect(url_for('index'))
            return f(*args, **kwargs)
        return decorated_function
    return decorator

# --- Auth ---
def get_user_by_username(username):
    rows = query('SELECT * FROM tb_user WHERE username=%s', (username,))
    return rows[0] if rows else None

@app.route('/')
def index():
    if 'user' in session:
        role = session['user']['role']
        if role == 'admin':
            return redirect(url_for('admin_dashboard'))
        elif role == 'pakar':
            return redirect(url_for('pakar_dashboard'))
        else:
            return redirect(url_for('user_dashboard'))
    return redirect(url_for('login'))

```

```

@app.route('/login', methods=['GET','POST'])
def login:
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        user = get_user_by_username(username)
        if user and check_password_hash(user['password'], password):
            session['user'] = {'id_user': user['id_user'], 'username': user['username'], 'role': user['role'],
'nama': user['nama']}
            flash('Login sukses', 'success')
            return redirect(url_for('index'))
            flash('Username atau password salah', 'danger')
            return render_template('login.html')

@app.route('/logout')
def logout:
    session.pop('user', None)
    flash('Logout berhasil', 'info')
    return redirect(url_for('login'))

# --- Dashboards ---
@app.route('/admin')
def admin_dashboard:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    total_gejala = query('SELECT COUNT(*) AS c FROM tb_gejala')[0]['c']
    total_kerusakan = query('SELECT COUNT(*) AS c FROM tb_kerusakan')[0]['c']
    total_aturan = query('SELECT COUNT(*) AS c FROM tb_aturan')[0]['c']
    total_riwayat = query('SELECT COUNT(*) AS c FROM tb_riwayat')[0]['c']
    return render_template('admin_dashboard.html', total_gejala=total_gejala,
total_kerusakan=total_kerusakan, total_aturan=total_aturan, total_riwayat=total_riwayat)

@app.route('/pakar')
def pakar_dashboard:
    if 'user' not in session or session['user']['role'] != 'pakar':
        return redirect(url_for('login'))
    # list rules
    rules = query('SELECT r.*, k.nama_kerusakan, g.nama_gejala FROM tb_aturan r JOIN
tb_kerusakan k ON r.id_kerusakan=k.id_kerusakan JOIN tb_gejala g ON r.id_gejala=g.id_gejala')
    return render_template('pakar_dashboard.html', rules=rules)

@app.route('/user')
def user_dashboard:
    if 'user' not in session or session['user']['role'] != 'user':
        return redirect(url_for('login'))
    return render_template('user_dashboard.html')

# --- Detection ---
CF_USER_OPTIONS = {
    '0': 0.0,
    '0.2': 0.2,
    '0.4': 0.4,
    '0.6': 0.6,
    '0.8': 0.8,
    '1': 1.0
}

```

```

}

@app.route('/detect', methods=['GET','POST'])
def detect:
    if 'user' not in session:
        return redirect(url_for('login'))
    gejala = query('SELECT * FROM tb_gejala')
    if request.method == 'POST':
        selected = []
        for g in gejala:
            key = f'cf_{g["id_gejala"]}'
            if key in request.form:
                cf_user = float(request.form[key])
                if cf_user > 0: # Only adding it if the user expressed some presence (cf > 0)
                    selected.append({'id_gejala': g['id_gejala'], 'cf_user': cf_user, 'nama_gejala':
g['nama_gejala']})

        if not selected:
            # If nothing > 0 is selected, the results list will be naturally empty and result.html will show
            # the "Tidak ada hasil" warning as intended.
            return render_template('result.html', results=[], selected=[])

        # Fetch rules and compute CFs per kerusakan
        kerusakans = query('SELECT * FROM tb_kerusakan')
        results = []
        for k in kerusakans:
            # get rules for this kerusakan
            rules = query('SELECT * FROM tb_aturan WHERE id_kerusakan=%s', (k['id_kerusakan'],))
            cfs = []
            details = []
            for s in selected:
                # check if there is a rule linking this symptom to this kerusakan
                matched_rule = next((r for r in rules if r['id_gejala'] == s['id_gejala']), None)

                if matched_rule:
                    cf_rule = matched_rule['mb'] - matched_rule['md']
                    cf_h_e = cf_single(cf_rule, s['cf_user'])
                    cfs.append(cf_h_e)
                    details.append({'gejala': s['nama_gejala'], 'mb': matched_rule['mb'], 'md':
matched_rule['md'], 'cf_rule': cf_rule, 'cf_user': s['cf_user'], 'cf_h_e': cf_h_e})
                else:
                    # Symptom picked by user but not related to this kerusakan
                    details.append({'gejala': s['nama_gejala'], 'mb': 0.0, 'md': 0.0, 'cf_rule': 0.0, 'cf_user':
s['cf_user'], 'cf_h_e': 0.0})

            cf_total = combine_list(cfs)
            if cf_total > 0:
                results.append({'kerusakan': k, 'cf_total': cf_total, 'details': details})

        # choose highest
        results_sorted = sorted(results, key=lambda x: x['cf_total'], reverse=True)
        # convert to percentage
        for r in results_sorted:
            r['cf_percent'] = float(r['cf_total']) * 100
        return render_template('result.html', results=results_sorted, selected=selected)

```

```

return render_template('detect.html', gejala=gejalas, cf_options=CF_USER_OPTIONS)

@app.route('/save_result', methods=['POST'])
def save_result:
    if 'user' not in session:
        return redirect(url_for('login'))
    id_user = session['user']['id_user']
    id_kerusakan = request.form['id_kerusakan']
    cf_total = float(request.form['cf_total'])
    execute('INSERT INTO tb_riwayat (id_user, id_kerusakan, cf_total) VALUES (%s,%s,%s)',
(id_user, id_kerusakan, cf_total))
    flash('Hasil diagnosa disimpan', 'success')
    return redirect(url_for('user_dashboard'))

@app.route('/history')
def history:
    if 'user' not in session:
        return redirect(url_for('login'))
    if session['user']['role'] == 'admin':
        rows = query('SELECT r.*, u.nama, k.nama_kerusakan FROM tb_riwayat r JOIN tb_user u
ON r.id_user=u.id_user JOIN tb_kerusakan k ON r.id_kerusakan=k.id_kerusakan ORDER BY
r.tanggal DESC')
    else:
        rows = query('SELECT r.*, u.nama, k.nama_kerusakan FROM tb_riwayat r JOIN tb_user u
ON r.id_user=u.id_user JOIN tb_kerusakan k ON r.id_kerusakan=k.id_kerusakan WHERE
r.id_user=%s ORDER BY r.tanggal DESC', (session['user']['id_user'],))
    return render_template('history.html', rows=rows)

# ===== ADMIN CRUD ROUTES =====

# --- User Management ---
@app.route('/admin/users')
def admin_users:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    users = query('SELECT * FROM tb_user ORDER BY created_at DESC')
    return render_template('admin_users.html', users=users)

@app.route('/admin/user/add', methods=['POST'])
def admin_user_add:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    nama = request.form['nama']
    username = request.form['username']
    password = generate_password_hash(request.form['password'])
    role = request.form['role']
    try:
        execute('INSERT INTO tb_user (nama, username, password, role) VALUES (%s,%s,%s,%s)',
(nama, username, password, role))
        flash('User berhasil ditambahkan', 'success')
    except Exception as e:
        flash(f'Gagal menambahkan user: username mungkin sudah ada.', 'danger')
    return redirect(url_for('admin_users'))

```

```

@app.route('/admin/user/delete/<int:id>')
def admin_user_delete(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    if id == session['user']['id_user']:
        flash('Tidak dapat menghapus akun sendiri', 'danger')
        return redirect(url_for('admin_users'))
    execute('DELETE FROM tb_user WHERE id_user=%s', (id,))
    flash('User berhasil dihapus', 'success')
    return redirect(url_for('admin_users'))

# --- Gejala Management ---
@app.route('/admin/gejala')
def admin_gejala:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    gejalas = query('SELECT * FROM tb_gejala ORDER BY id_gejala')
    return render_template('admin_gejala.html', gejalas=gejalas)

@app.route('/admin/gejala/add', methods=['POST'])
def admin_gejala_add:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    kode_gejala = request.form['kode_gejala']
    nama_gejala = request.form['nama_gejala']
    execute('INSERT INTO tb_gejala (kode_gejala, nama_gejala) VALUES (%s,%s)', (kode_gejala,
nama_gejala))
    flash('Gejala berhasil ditambahkan', 'success')
    return redirect(url_for('admin_gejala'))

@app.route('/admin/gejala/edit/<int:id>', methods=['POST'])
def admin_gejala_edit(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    kode_gejala = request.form['kode_gejala']
    nama_gejala = request.form['nama_gejala']
    execute('UPDATE tb_gejala SET kode_gejala=%s, nama_gejala=%s WHERE id_gejala=%s',
(kode_gejala, nama_gejala, id))
    flash('Gejala berhasil diupdate', 'success')
    return redirect(url_for('admin_gejala'))

@app.route('/admin/gejala/delete/<int:id>')
def admin_gejala_delete(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    execute('DELETE FROM tb_gejala WHERE id_gejala=%s', (id,))
    flash('Gejala berhasil dihapus', 'success')
    return redirect(url_for('admin_gejala'))

# --- Kerusakan Management ---
@app.route('/admin/kerusakan')
def admin_kerusakan:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    kerusakans = query('SELECT * FROM tb_kerusakan ORDER BY id_kerusakan')

```

```

return render_template('admin_kerusakan.html', kerusakans=kerusakans)

@app.route('/admin/kerusakan/add', methods=['POST'])
def admin_kerusakan_add:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    kode_kerusakan = request.form['kode_kerusakan']
    nama_kerusakan = request.form['nama_kerusakan']
    deskripsi = request.form['deskripsi']
    solusi = request.form['solusi']
    execute('INSERT INTO tb_kerusakan (kode_kerusakan, nama_kerusakan, deskripsi, solusi)
VALUES (%s,%s,%s,%s)', (kode_kerusakan, nama_kerusakan, deskripsi, solusi))
    flash('Kerusakan berhasil ditambahkan', 'success')
    return redirect(url_for('admin_kerusakan'))

@app.route('/admin/kerusakan/edit/<int:id>', methods=['POST'])
def admin_kerusakan_edit(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    kode_kerusakan = request.form['kode_kerusakan']
    nama_kerusakan = request.form['nama_kerusakan']
    deskripsi = request.form['deskripsi']
    solusi = request.form['solusi']
    execute('UPDATE tb_kerusakan SET kode_kerusakan=%s, nama_kerusakan=%s, deskripsi=%s,
solusi=%s WHERE id_kerusakan=%s', (kode_kerusakan, nama_kerusakan, deskripsi, solusi, id))
    flash('Kerusakan berhasil diupdate', 'success')
    return redirect(url_for('admin_kerusakan'))

@app.route('/admin/kerusakan/delete/<int:id>')
def admin_kerusakan_delete(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    execute('DELETE FROM tb_kerusakan WHERE id_kerusakan=%s', (id,))
    flash('Kerusakan berhasil dihapus', 'success')
    return redirect(url_for('admin_kerusakan'))

# --- Aturan Management ---
@app.route('/admin/aturan')
def admin_aturan:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    aturans = query("""
SELECT a.*, k.nama_kerusakan, g.nama_gejala
FROM tb_aturan a
JOIN tb_kerusakan k ON a.id_kerusakan=k.id_kerusakan
JOIN tb_gejala g ON a.id_gejala=g.id_gejala
ORDER BY a.id_rule
""")
    kerusakans = query('SELECT * FROM tb_kerusakan')
    gejala = query('SELECT * FROM tb_gejala')
    return render_template('admin_aturan.html', aturans=aturans, kerusakans=kerusakans,
gejalas=gejalas)

@app.route('/admin/aturan/add', methods=['POST'])
def admin_aturan_add:

```

```

if 'user' not in session or session['user']['role'] != 'admin':
    return redirect(url_for('login'))
id_kerusakan = request.form['id_kerusakan']
id_gejala = request.form['id_gejala']
mb = float(request.form['mb'])
md = float(request.form['md'])
execute('INSERT INTO tb_aturan (id_kerusakan, id_gejala, mb, md) VALUES (%s,%s,%s,%s)',
(id_kerusakan, id_gejala, mb, md))
flash('Aturan berhasil ditambahkan', 'success')
return redirect(url_for('admin_aturan'))

@app.route('/admin/aturan/edit/<int:id>', methods=['POST'])
def admin_aturan_edit(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    id_kerusakan = request.form['id_kerusakan']
    id_gejala = request.form['id_gejala']
    mb = float(request.form['mb'])
    md = float(request.form['md'])
    execute('UPDATE tb_aturan SET id_kerusakan=%s, id_gejala=%s, mb=%s, md=%s WHERE
id_rule=%s', (id_kerusakan, id_gejala, mb, md, id))
    flash('Aturan berhasil diupdate', 'success')
    return redirect(url_for('admin_aturan'))

@app.route('/admin/aturan/delete/<int:id>')
def admin_aturan_delete(id):
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    execute('DELETE FROM tb_aturan WHERE id_rule=%s', (id,))
    flash('Aturan berhasil dihapus', 'success')
    return redirect(url_for('admin_aturan'))

# --- History Management ---
@app.route('/history/delete/<int:id>')
def history_delete(id):
    if 'user' not in session:
        return redirect(url_for('login'))
    execute('DELETE FROM tb_riwayat WHERE id_hasil=%s', (id,))
    flash('Riwayat berhasil dihapus', 'success')
    return redirect(url_for('history'))

# --- Statistics ---
@app.route('/admin/statistik')
def admin_statistik:
    if 'user' not in session or session['user']['role'] != 'admin':
        return redirect(url_for('login'))
    stats = query("""
        SELECT k.nama_kerusakan, COUNT(r.id_hasil) as total,
            ROUND(AVG(r.cf_total) * 100, 2) as avg_cf
        FROM tb_riwayat r
        JOIN tb_kerusakan k ON r.id_kerusakan = k.id_kerusakan
        GROUP BY r.id_kerusakan, k.nama_kerusakan
        ORDER BY total DESC
    """)
    # Convert Decimal to float for JSON serialization

```

```

for s in stats:
    s['avg_cf'] = float(s['avg_cf']) if s['avg_cf'] is not None else 0.0
    s['total'] = int(s['total'])
return render_template('admin_statistik.html', stats=stats)

# --- Pakar: Edit MB/MD ---
@app.route('/pakar/aturan/edit/<int:id>', methods=['POST'])
def pakar_aturan_edit(id):
    if 'user' not in session or session['user']['role'] != 'pakar':
        return redirect(url_for('login'))
    mb = float(request.form['mb'])
    md = float(request.form['md'])
    execute('UPDATE tb_aturan SET mb=%s, md=%s WHERE id_rule=%s', (mb, md, id))
    flash('Nilai MB & MD berhasil diupdate', 'success')
    return redirect(url_for('pakar_dashboard'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```

2. Source code cf.py

```

def cf_single(cf_rule, cf_user):
    # CF(H,E) = CF_rule * CF_user
    return cf_rule * cf_user

def combine_two(cf1, cf2):
    # CFcombine = CF1 + CF2 * (1 - CF1)
    return cf1 + cf2 * (1 - cf1)

def combine_list(cfs):
    if not cfs:
        return 0.0
    result = cfs[0]
    for cf in cfs[1:]:
        result = combine_two(result, cf)
    return result

```

3. Source code config.py

```

from dotenv import load_dotenv
import os

load_dotenv

DB_CONFIG = {
    'host': os.getenv('DB_HOST', 'localhost'),
    'user': os.getenv('DB_USER', 'root'),
    'password': os.getenv('DB_PASSWORD', ''),
    'db': os.getenv('DB_NAME', 'db_pakar_ac'),
    'charset': 'utf8mb4'
}

SECRET_KEY = os.getenv('SECRET_KEY', 'dev-secret')

```

4. Source code db.py

```

import pymysql
from config import DB_CONFIG

def get_conn:
    return pymysql.connect(
        host=DB_CONFIG['host'],
        user=DB_CONFIG['user'],
        password=DB_CONFIG['password'],
        db=DB_CONFIG['db'],
        charset=DB_CONFIG['charset'],
        cursorclass=pymysql.cursors.DictCursor,
        autocommit=True
    )

def query(sql, params=None):
    conn = get_conn
    try:
        with conn.cursor as cur:
            cur.execute(sql, params or )
            return cur.fetchall
    finally:
        conn.close

def execute(sql, params=None):
    conn = get_conn
    try:
        with conn.cursor as cur:
            cur.execute(sql, params or )
            return cur.lastrowid
    finally:
        conn.close

```

5. Source code populate_db.py

```

import pymysql
from db import get_conn

def setup_db:
    conn = get_conn
    cur = conn.cursor

    # 1. Clear existing data
    print("Clearing data...")
    cur.execute('DELETE FROM tb_riwayat')
    cur.execute('DELETE FROM tb_aturan')
    cur.execute('DELETE FROM tb_gejala')
    cur.execute('DELETE FROM tb_kerusakan')

    # Reset auto increment
    cur.execute('ALTER TABLE tb_riwayat AUTO_INCREMENT = 1')
    cur.execute('ALTER TABLE tb_aturan AUTO_INCREMENT = 1')
    cur.execute('ALTER TABLE tb_gejala AUTO_INCREMENT = 1')
    cur.execute('ALTER TABLE tb_kerusakan AUTO_INCREMENT = 1')

```

```

# 2. Insert Kerusakan
print("Inserting Kerusakan...")
kerusakans = [
    ('K01','Freon Habis / Kurang','Freon habis atau kurang menyebabkan AC tidak dingin optimal','Tambahkan Freon sesuai tipe dan periksa kebocoran'),
    ('K02','Kompresor Rusak / Lemah','Kompresor tidak bekerja optimal atau rusak','Periksa dan ganti kompresor jika perlu'),
    ('K03','Kondensor Kotor','Kondensor kotor menyebabkan panas tidak terbuang dengan baik','Bersihkan kondensor dengan air atau sikat'),
    ('K04','Evaporator Kotor','Evaporator kotor menyebabkan aliran udara tersumbat','Bersihkan evaporator dengan vacuum atau sikat lembut'),
    ('K05','Kapasitor Rusak','Kapasitor rusak menyebabkan motor tidak dapat bekerja dengan normal','Periksa dan ganti kapasitor yang rusak'),
    ('K06','Thermistor Rusak','Thermistor rusak menyebabkan sensor suhu tidak akurat','Periksa dan ganti thermistor'),
    ('K07','Modul PCB Bermasalah','Modul PCB bermasalah menyebabkan kontrol sistem gagal','Periksa dan ganti modul PCB jika rusak'),
    ('K08','Pipa Kapiler Tersumbat','Pipa kapiler tersumbat menyebabkan aliran freon terganggu','Periksa dan bersihkan atau ganti pipa kapiler')
]
cur.executemany("INSERT INTO tb_kerusakan (kode_kerusakan, nama_kerusakan, deskripsi, solusi) VALUES (%s,%s,%s,%s)", kerusakans)

# 3. Insert Gejala
print("Inserting Gejala...")
gejalas = [
    ('G01','AC tidak dingin'), # 1
    ('G02','Tekanan freon rendah'), # 2
    ('G03','Kompresor berdengung keras'), # 3
    ('G04','Kompresor sangat panas (overheat)'), # 4
    ('G05','Air menetes dari unit indoor'), # 5
    ('G06','Timbul bunga es pada pipa atau evaporator'), # 6
    ('G07','Angin keluar lemah dari indoor'), # 7
    ('G08','AC tidak mau menyala'), # 8
    ('G09','Lampu indikator berkedip (error)'), # 9
    ('G10','AC tidak mau cut-off (terus menyala)'), # 10
    ('G11','Outdoor fan tidak berputar'), # 11
    ('G12','Bau tidak sedap dari indoor'), # 12
    ('G13','AC sering mati sendiri'), # 13
    ('G14','Pipa kecil membeku'), # 14
    ('G15','Arus listrik kompresor tinggi (overload)') # 15
]
cur.executemany("INSERT INTO tb_gejala (kode_gejala, nama_gejala) VALUES (%s,%s)",
gejalas)

# 4. Insert Comprehensive Rules
print("Inserting Rules...")
# Map (id_kerusakan, id_gejala, mb, md)
rules = [
    # K01: Freon Habis / Kurang
    (1, 1, 0.8, 0.1), # AC tidak dingin
    (1, 2, 0.9, 0.0), # Tekanan freon rendah
    (1, 6, 0.7, 0.1), # Timbul bunga es

    # K02: Kompresor Rusak / Lemah

```

```

(2, 1, 0.6, 0.2), # AC tidak dingin
(2, 3, 0.8, 0.1), # Kompresor berdengung
(2, 4, 0.9, 0.0), # Kompresor panas
(2, 15, 0.85, 0.05), # Arus kompresor tinggi

# K03: Kondensator Kotor
(3, 1, 0.5, 0.2), # AC tidak dingin
(3, 4, 0.6, 0.3), # Kompresor panas (secondary effect)

# K04: Evaporator Kotor
(4, 1, 0.6, 0.1), # AC tidak dingin
(4, 5, 0.8, 0.1), # Air menetes
(4, 7, 0.9, 0.05), # Angin lemah
(4, 12, 0.7, 0.1), # Bau tidak sedap

# K05: Kapasitor Rusak
(5, 3, 0.7, 0.2), # Kompresor berdengung
(5, 11, 0.9, 0.05), # Outdoor fan tidak berputar
(5, 8, 0.5, 0.4), # AC tidak menyala (kadang-kadang efeknya)

# K06: Thermistor Rusak
(6, 1, 0.4, 0.4), # AC tidak dingin (kadang)
(6, 10, 0.9, 0.05), # AC tidak cut-off
(6, 13, 0.8, 0.1), # AC sering mati sendiri

# K07: Modul PCB Bermasalah
(7, 8, 0.8, 0.1), # AC tidak mau menyala
(7, 9, 0.9, 0.05), # Lampu kedip error
(7, 13, 0.6, 0.3), # Sering mati

# K08: Pipa Kapiler Tersumbat
(8, 1, 0.7, 0.1), # AC tidak dingin
(8, 14, 0.9, 0.05), # Pipa kecil membeku
(8, 4, 0.6, 0.2) # Kompresor re-heat
]
cur.executemany("INSERT INTO tb_aturan (id_kerusakan, id_gejala, mb, md) VALUES
(%s,%s,%s,%s)", rules)

conn.commit
conn.close
print("Database successfully populated with comprehensive rules!")

if __name__ == '__main__':
    setup_db

```



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi Unggul Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 174/SK/BDAN-PT/Ak/Pg/PT/10/2024
 Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp (061) 6622400 - 66224567 Fax (061) 6625474 - 6631003

Website: www.umsumedan.ac.id Email: info@umsumedan.ac.id Facebook: [umsumedan](https://www.facebook.com/umsumedan) Instagram: [umsumedan](https://www.instagram.com/umsumedan) YouTube: [umsumedan](https://www.youtube.com/umsumedan)

FORMULIR PERBAIKAN UJIAN SKRIPSI

Pada hari ini, Jumat 10 April 2026 telah dilaksanakan Ujian Skripsi bagi mahasiswa Fakultas Ilmu Komputer Dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara. Sbb:

Nama Mahasiswa : Helena Rayani Saragih
 NPM : 2209010169
 Program Studi : Sistem Informasi
 Judul Proposal : Sistem Pakar Pendeteksi Kerusakan Ac Menggunakan Metode Certainty Faktor Untuk Mendukung Teknisi Pemula Berbasis Web

Materi Point yang Diperbaiki :

	Paraf
Yohanni Syahra, S.Si.,M.Kom.	
Dr. Al-Khowarizmi, M Kom	 - Hapus Screenshot diubah menjadi label - Latar belakang
Rizaldy Khair, M Kom	 - Di Lambai dibagian batasan masalah, bagian kata Misalnya Deletfs nya enggak keluar-mengjadi keturangan - Di bagian bab 3 soal Data Gejala, kemufakan, toki aluran di sebutkan sumbernya dari mana

Berita acara ini ditandatangani setelah skripsi diperbaiki sesuai petunjuk/arahan dari Pembimbing dan Penguji Pembahas.

SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FAKTOR UNTUK MENDUKUNG TEKNISI PEMULA BERBASIS WEB

ORIGINALITY REPORT

12%	11%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.umsu.ac.id Internet Source	3%
2	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	2%
3	repository.bunghatta.ac.id Internet Source	1%
4	smart.stmikplk.ac.id Internet Source	1%
5	Submitted to Universitas Bengkulu Student Paper	<1%
6	elib.pnc.ac.id Internet Source	<1%
7	repo.palcomtech.ac.id Internet Source	<1%
8	Submitted to Sultan Agung Islamic University Student Paper	<1%

9	repositori.unimma.ac.id Internet Source	<1 %
10	repository.bakrie.ac.id Internet Source	<1 %
11	Submitted to Universitas Nahdlatul Ulama Lampung Student Paper	<1 %
12	Submitted to STKIP Sumatera Barat Student Paper	<1 %
13	Submitted to Universitas Islam Riau Student Paper	<1 %
14	tajdid.id Internet Source	<1 %
15	docplayer.info Internet Source	<1 %
16	kc.umn.ac.id Internet Source	<1 %
17	Submitted to Universitas Pamulang Student Paper	<1 %
18	digilib.uin-suka.ac.id Internet Source	<1 %
19	journal.bengkuluinstitute.com Internet Source	<1 %
	repository.uin-suska.ac.id	

20	Internet Source	<1 %
21	library.stmikgici.ac.id Internet Source	<1 %
22	Submitted to Universidad Tecnológica de Bolívar, UTB Student Paper	<1 %
23	repository.unpar.ac.id Internet Source	<1 %
24	www.lsis.org Internet Source	<1 %
25	Submitted to Universitas Muhammadiyah Surakarta Student Paper	<1 %
26	Submitted to Universitas Pelita Harapan Student Paper	<1 %
27	Submitted to Universiti Tun Hussein Onn Malaysia Student Paper	<1 %
28	etheses.uin-malang.ac.id Internet Source	<1 %
29	text-id.123dok.com Internet Source	<1 %
30	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	<1 %

31	jurnal.pancabudi.ac.id Internet Source	<1 %
32	lib.unnes.ac.id Internet Source	<1 %
33	library.binus.ac.id Internet Source	<1 %
34	repository.its.ac.id Internet Source	<1 %
35	repository.unpkediri.ac.id Internet Source	<1 %
36	anzdoc.com Internet Source	<1 %
37	eprints.umm.ac.id Internet Source	<1 %
38	jurusan.tik.pnj.ac.id Internet Source	<1 %
39	repositori.usu.ac.id Internet Source	<1 %
40	repository.teknokrat.ac.id Internet Source	<1 %
41	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	<1 %

42	<p>I Kadek Bayu Danu Artha, I Komang Arya Adi Kusuma, Putu Gede Dimas Witjaksana, Dewa Gede Bhaskara Pramudya et al. "Development of Expert System for Diagnosing Pests and Diseases in Strawberry Plants Using the Certainty Factor", International Journal of Education, Management, and Technology, 2025 Publication</p>	<1 %
43	<p>adoc.pub Internet Source</p>	<1 %
44	<p>eprints.walisongo.ac.id Internet Source</p>	<1 %
45	<p>id.123dok.com Internet Source</p>	<1 %
46	<p>pyungshera.blogspot.com Internet Source</p>	<1 %
47	<p>repository.stei.ac.id Internet Source</p>	<1 %
48	<p>repository.uml.ac.id Internet Source</p>	<1 %
49	<p>repository.upbatam.ac.id Internet Source</p>	<1 %
50	<p>repository.upnjatim.ac.id Internet Source</p>	<1 %

51 van der Gaag, L.C.. "Different notions of uncertainty in quasi-probabilistic models", International Journal of Man-Machine Studies, 199011
Publication <1%

52 www.coursehero.com
Internet Source <1%

53 123dok.com
Internet Source <1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

Kohesi: Jurnal Sains dan Teknologi

← Back to Submissions

9761 / Helena Rayani Saragih et al. / SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FACTOR UNTU [Library](#)

Workflow **Publication**

Submission **Review** Copyediting Production

Submission Files [Search](#)

27847	090526_SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FACTOR UNTUK Mendukung Teknisi Pemula Berbasis Web.pdf	May 9, 2026	Article Text
-------	---	-------------	--------------

[Download All Files](#)

Pre-Review Discussions [Add discussion](#)

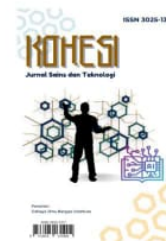
Name	From	Last Reply	Replies	Closed
No Items				

KOHESI

JURNAL SAINS DAN TEKNOLOGI

CAHAYA ILMU BANGSA INSTITUTE

Biro Penelitian, Publikasi, dan Pengabdian Kepada Masyarakat
KEMENKUMHAM AHU-0018912-AH.01.14
Perum Puri Kartika Asri Blok 2 A2 Malang
e-mail: admin@cahayailmubangsa.institute



LETTER OF ACCEPTANCE

NO: 2026/CIB013/LOA592

Assalamualaikum Wr. Wb.

Bersama surat ini, kami menerangkan bahwa artikel dengan keterangan naskah berikut

Judul	SISTEM PAKAR PENDETEKSI KERUSAKAN AC MENGGUNAKAN METODE CERTAINTY FACTOR UNTUK Mendukung Teknisi Pemula Berbasis Web
Author	Helena Rayani Saragih, Yohanni Syahra
Instansi	Universitas Muhammadiyah Sumatera Utara
Korespondensi	helenarayani1@gmail.com
Jurnal	Kohesi: Jurnal Sains dan Teknologi
Volume	Vol. 10 No. 12 (2026)
Link Terbitan	https://cibangsa.com/index.php/kohesi/article/view/9761

Berstatus ACCEPTED untuk dipublish. Keputusan ini dibuat sebagai tanda bahwa naskah yang bersangkutan telah lolos plagiarism checker. Dan LoA ini dibuat sebagai bukti bahwa author telah menyelesaikan APC yang telah ditetapkan oleh pengelola jurnal. LOA Berlaku jika dilengkapi link dan pdf publish. Hubungi kami di admin_jurnal@cahayailmubangsa.institute jika ada pertanyaan lebih lanjut, terima kasih.

Malang, 09 May 2026



Dr. Umam Rofiq, M.Pd., Ph.D
Director



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/2019
 Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
 http://www.umsumed.ac.id | email: fak@umsumed.ac.id | umsumedan | umsumedan | umsumedan | umsumedan

Berita Acara Pembimbingan Skripsi

Nama Mahasiswa : Helena Rayani Saragih
 Program Studi : Sistem Informasi
 NPM : 2209010169
 Judul Penelitian : Sistem Pakar Pendeteksi Kerusakan AC Menggunakan Metode Certainty Faktor Untuk Mendukung Teknisi Pemula Berbasis Web
 Nama Dosen Pembimbing : Yohanni Syahra, S.Si.,M.Kom.

Tanggal Bimbingan	Hasil Evaluasi	Paraf Dosen
08/01/26	Ngasih Data	ya
13/01/26	Bab 1 dan bab 2	ya
22/01/26	Merapihin Bab 1- bab 3	ya
23/01/26	Bab 1, Bab 2, Bab 3 Di Acc Seminar Proposal	ya
06/03/26	Bab 4	ya
09/03/26	Bab 5	ya
10/03/26	Cover, kata pengantar	ya
12/03/26	Acc all Bab dan Acc Sidang	ya

Medan, 13 - Januari 2026

Diketahui oleh :

Ketua Program Studi

Mahardika Abdi Prawira Tanjung, M.Kom

Disetujui oleh :

Dosen Pembimbing

Yohanni Syahra, S.Si.,M.Kom

