Jurnal Cakrawala Akademika (JCA)
Vol. 3 No. 2 Tahun 2026
Licenced CC BY-SA, E-ISSN: 3062-9942, Hal 623-634
DOI: https://doi.org/10.71417/jca.v3i2.2439
https://e-journal.lppmpustakacendekia.or.id/index.php/JCA/index

Sistem Pakar Deteksi Dini Kerusakan Laptop dengan Penentuan Tingkat Urgensi Penanganan Menggunakan *Forward Chaining*

Virgian Ramzy Arindra[1], Firza Ma'Rifatus Sapangat[2], Eliza Rachmayanti[3], Rifal Sabilla Rosad[4], Rofif Najmi Efendi[5]

[1,2,3,4,5] Prodi Teknologi Informasi, Fakultas Teknik dan Informatika,
Universitas Bina Sarana Informatika, Jakarta, Indonesia

Alamat Universitas: Jl. Kramat Raya No.98, Jakarta Pusat., Indonesia
E-mail seluruh Penulis : virgianramzy06@gmail.com[1], firzasapangat@gmail.com[2], elizaaryn@gmail.com[3], rifaldjokam354@gmail.com[4], rofifnajmiefendi@gmail.com[5]

Abstract : *Laptops are essential devices for studying, working, and doing business, so any damage to them can disrupt their users' productivity. Many users do not understand the early symptoms of damage and find it difficult to decide whether a problem can be handled on their own or must be taken to a technician immediately. As a result, handling is often delayed and repair costs increase. This study aims to build an expert system capable of detecting laptop damage at an early stage while determining the urgency level of its handling. The method used is forward chaining, a data-driven reasoning approach that starts from the symptoms selected by the user and matches them against an IF-THEN rule base to conclude the type of damage. The knowledge base is compiled from interviews with laptop technicians, observation, and literature study. The diagnosis results are classified into low, medium, and high urgency levels, accompanied by handling recommendations. The system is developed using the Waterfall method and tested using black box testing and accuracy testing by comparing the system's results with an expert's diagnosis. The black-box test results across 12 scenarios indicate that all tested features operated as expected. The system generates a diagnosis when the corresponding rule conditions are met and displays the urgency level and recommended action. However, diagnostic accuracy against expert assessments cannot yet be concluded from the manuscript because the comparison details for the 10 test cases have not been reported. Therefore, the system should be regarded as a preliminary diagnostic aid rather than a replacement for a technician's examination.*

Keywords*: Expert System, Forward chaining, Early Detection, Laptop Damage, Urgency Level.*

Abstrak : Laptop merupakan perangkat yang sangat dibutuhkan untuk belajar, bekerja, dan berbisnis, sehingga kerusakan pada laptop dapat mengganggu produktivitas penggunanya. Banyak pengguna belum memahami gejala awal kerusakan dan kesulitan menentukan apakah masalah dapat ditangani sendiri atau harus segera dibawa ke teknisi. Akibatnya, penanganan sering terlambat dan biaya perbaikan menjadi lebih besar. Penelitian ini bertujuan membangun sistem pakar yang mampu mendeteksi dini kerusakan laptop sekaligus menentukan tingkat urgensi penanganannya. Metode yang digunakan adalah *forward chaining*, yaitu penalaran maju yang dimulai dari gejala-gejala yang dipilih pengguna, kemudian dicocokkan dengan basis aturan berbentuk IF-THEN untuk menyimpulkan jenis kerusakan. Basis pengetahuan disusun dari hasil wawancara dengan teknisi laptop, observasi, dan studi pustaka. Hasil diagnosis diklasifikasikan ke dalam tingkat urgensi ringan, sedang, dan berat, disertai rekomendasi penanganan. Sistem dikembangkan dengan metode Waterfall dan diuji menggunakan pengujian black box serta pengujian akurasi dengan membandingkan hasil sistem terhadap diagnosis pakar. Hasil pengujian black box pada 12 skenario menunjukkan bahwa seluruh fitur yang diuji berjalan sesuai dengan fungsi yang diharapkan. Sistem menghasilkan diagnosis berdasarkan aturan yang terpenuhi dan menampilkan tingkat urgensi serta rekomendasi penanganan. Namun, nilai akurasi diagnosis terhadap pakar belum dapat disimpulkan dari naskah ini karena rincian perbandingan 10 kasus uji belum disajikan. Dengan demikian, sistem dapat diposisikan sebagai alat bantu diagnosis awal, bukan pengganti pemeriksaan teknisi.

Kata kunci: Sistem Pakar, *Forward Chaining*, Deteksi Dini, Kerusakan Laptop, Tingkat Urgensi.

1. PENDAHULUAN

Laptop banyak diminati karena harganya semakin terjangkau dan mudah dibawa (Delimayanti et al., 2022). Namun, komponen hardware-nya akan mengalami kerusakan seiring pemakaian. Minimnya pemahaman ini membuat pengguna mengeluarkan biaya besar kepada teknisi (Hermanto et al., 2024).

Received: November 09, 2026; Revised: November 12, 2026; Accepted: November 13, 2026;
*Virgian Ramzy Arindra, virgianramzy06@gmail.com

Di sisi lain, diagnosis manual di tempat servis memakan waktu lama dan bergantung pada pengalaman teknisi (Permadi et al., 2026). Penelitian mutakhir juga menunjukkan penerapan *forward chaining* untuk membantu deteksi dini berbagai kerusakan laptop berdasarkan gejala pengguna (Chaniago et al., 2025).

Sistem pakar dapat memindahkan keahlian teknisi ke dalam aplikasi sehingga pengguna non-pakar dapat mendiagnosis kerusakan secara mandiri (Hermanto et al., 2024; Chaniago et al., 2025). Salah satu metode inferensinya adalah *forward chaining*, yaitu penalaran yang dimulai dari gejala masukan lalu dicocokkan dengan aturan IF–THEN hingga diperoleh kesimpulan (Hermanto et al., 2024). Metode ini sesuai karena pengguna umumnya hanya mengetahui gejala, bukan jenis kerusakannya, dan implementasinya menggunakan pencocokan fakta dengan premis aturan untuk menghasilkan kesimpulan (Hasnan & Willy, 2022; Chaniago et al., 2025).

Penelitian terdahulu telah menerapkan *forward chaining* untuk kerusakan laptop, baik berbasis Android maupun web dengan 15 jenis kerusakan dan 40 gejala (Delimayanti et al., 2022; Hasnan & Willy, 2022; Taufiq & Sandi, 2021). Hermanto et al. (2024) menambahkan saran perawatan, Permadi et al. (2026) menerapkannya pada layanan servis laptop, sedangkan Chaniago et al. (2025) menguji diagnosis pada 50 unit laptop. Supu et al. (2026) juga menerapkan *forward chaining* untuk troubleshooting perangkat keras komputer. Meski demikian, keluaran sistem-sistem tersebut berfokus pada jenis kerusakan dan solusi, belum secara eksplisit menyajikan tingkat urgensi penanganan dalam tiga kategori sebagaimana diusulkan penelitian ini. Padahal tingkat kedaruratannya berbeda: kerusakan mainboard dan chipset perlu segera dibawa ke pusat servis, sedangkan gangguan RAM dapat dicoba diatasi lebih dahulu dengan membersihkan modulnya (Permadi et al., 2026).

Untuk mengisi celah tersebut, penelitian ini mengusulkan sistem pakar deteksi dini kerusakan laptop berbasis *forward chaining* yang menyimpulkan jenis kerusakan sekaligus tingkat urgensi penanganannya (ringan, sedang, atau berat) beserta rekomendasi tindakan. Nilai barunya terletak pada keluaran berlapis ini, yang membantu pengguna memutuskan apakah kerusakan dapat ditangani sendiri atau perlu segera ke teknisi. Tujuan penelitian ini adalah merancang dan membangun sistem tersebut serta menguji kesesuaian hasil diagnosisnya dengan penilaian pakar.

2. METODE PENELITIAN

Tahapan Penelitian

Penelitian ini dilakukan melalui tujuh tahap, yaitu studi pustaka, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan penarikan kesimpulan, seperti ditunjukkan pada Gambar 1. Urutan tahap ini mengacu pada alur pengembangan sistem bertahap yang digunakan dalam pengembangan aplikasi berbasis Waterfall (Fajriati & Budiman, 2022; Permadi et al., 2026).



Gambar 1. Alur tahapan penelitian

Teknik Pengumpulan Data

Data dikumpulkan melalui studi pustaka (library research), karena pengetahuan untuk sistem pakar dapat bersumber dari pakar maupun dari buku, jurnal, dan dokumentasi lainnya (Hermanto et al., 2024; Chaniago et al., 2025). Proses pengumpulan dilakukan dalam empat langkah.

Langkah pertama adalah seleksi literatur. Sumber dipilih dari artikel jurnal dan prosiding yang membahas sistem pakar diagnosis kerusakan hardware laptop dan memuat tabel gejala, jenis kerusakan, serta solusi penanganan secara eksplisit. Langkah kedua adalah ekstraksi data. Dari setiap sumber diambil daftar gejala, jenis kerusakan, hubungan antara keduanya, dan solusi yang disarankan. Sumber utamanya adalah penelitian Delimayanti et al. (2022) yang memuat 15 jenis kerusakan dan 40 gejala, Permadi et al. (2026) dengan 10 kerusakan dan 39 gejala, Rahman et al. (2024) dengan 12 kerusakan dan 36 gejala, Hermanto et al. (2024) dengan 8 kerusakan dan 31 gejala, serta Suryawan et al. (2024) dengan 7 kerusakan dan 21 gejala. Literatur sejenis juga digunakan dalam pengembangan sistem diagnosis laptop berbasis *forward chaining* (Hasnan & Willy, 2022; Chaniago et al., 2025).

Langkah ketiga adalah konsolidasi. Data dari berbagai sumber digabungkan, gejala dan kerusakan yang sama disatukan penamaannya, lalu dipilih yang paling sering muncul di berbagai sumber agar basis pengetahuan ringkas namun mewakili kerusakan yang umum. Konsolidasi basis pengetahuan dan perumusan aturan gejala-ke-kerusakan juga digunakan pada penelitian sistem pakar laptop terdahulu (Hasnan & Willy, 2022; Chaniago et al., 2025). Hasilnya diberi kode gejala (G) dan kode kerusakan (K) serta disusun menjadi aturan IF–THEN. Langkah keempat adalah penentuan tingkat urgensi. Setiap kerusakan dikategorikan berdasarkan isi solusi pada literatur, misalnya kerusakan yang menurut sumber tidak dapat ditangani secara manual dan perlu dibawa ke pusat servis (Permadi et al., 2026) dikelompokkan sebagai urgensi berat. Kategorisasi mengikuti kriteria pada Tabel 1.

Metode *Forward chaining*

Forward chaining adalah metode penalaran maju yang bersifat data-driven, yaitu proses inferensi dimulai dari fakta atau gejala masukan, dicocokkan dengan bagian IF pada aturan, lalu menghasilkan kesimpulan pada bagian THEN (Hermanto et al., 2024; Chaniago et al., 2025). Basis pengetahuan direpresentasikan dalam bentuk kaidah produksi sebagai berikut:

IF Gx AND Gy AND Gz THEN Kn

dengan G sebagai kode gejala dan K sebagai kode kerusakan. Alur inferensi pada sistem ini adalah: (1) pengguna memilih gejala yang dialami laptop, (2) mesin inferensi mencocokkan gejala tersebut dengan seluruh aturan pada basis pengetahuan, (3) aturan yang seluruh premisnya terpenuhi dieksekusi sehingga menghasilkan jenis kerusakan, dan (4) sistem menampilkan jenis kerusakan beserta tingkat urgensi dan rekomendasi penanganannya. Apabila tidak ada aturan yang cocok, sistem menyampaikan bahwa diagnosis tidak ditemukan.

Penentuan Tingkat Urgensi Penanganan

Pembeda utama sistem ini dari penelitian terdahulu adalah penentuan tingkat urgensi. Setiap jenis kerusakan dipetakan ke salah satu dari tiga kategori pada Tabel 1. Pemetaan dilakukan berdasarkan hasil wawancara dengan pakar, sehingga setiap kerusakan sudah memiliki tingkat urgensi tetap pada basis pengetahuan dan tidak dihitung ulang oleh sistem.

Tabel 1. Kategori tingkat urgensi penanganan

Tingkat urgensi	Kriteria	Rekomendasi tindakan
Ringan	Laptop masih dapat digunakan, risiko kerusakan lanjutan kecil, dan dapat ditangani tanpa alat khusus	Perbaikan mandiri sesuai panduan
Sedang	Laptop masih dapat digunakan terbatas, ada risiko kerusakan lanjutan atau kehilangan data jika dibiarkan	Penanganan awal mandiri, lalu diperiksakan ke teknisi jika masalah berlanjut
Berat	Laptop tidak dapat digunakan atau penanganannya memerlukan keahlian dan alat khusus	Segera dibawa ke teknisi atau pusat servis

Metode Pengujian

Pengujian akurasi dilakukan dengan membandingkan hasil diagnosis sistem terhadap diagnosis pakar pada 10 kasus uji. Setiap kasus h arus memuat gejala yang diberikan, diagnosis jenis kerusakan menurut pakar, tingkat urgensi menurut pakar, diagnosis sistem, dan tingkat urgensi yang dihasilkan sistem. Penilaian dilakukan secara terpisah untuk jenis kerusakan dan tingkat urgensi. Akurasi masing-masing keluaran dihitung dengan membagi jumlah kasus yang hasil sistemnya sama dengan pakar dengan jumlah seluruh kasus uji, kemudian dikalikan 100%. Pemisahan kedua ukuran ini diperlukan agar kesesuaian jenis kerusakan tidak menutupi ketidaksesuaian tingkat urgensi (Delimayanti et al., 2022; Farhan & Perkasa, 2026).

Akurasi = (Jumlah hasil sistem yang sesuai dengan pakar / Jumlah kasus uji) × 100%

3. HASIL DAN PEMBAHASAN

Basis Pengetahuan

Basis pengetahuan merupakan inti sistem pakar yang memuat fakta dan aturan untuk proses inferensi. Basis pengetahuan penelitian ini disusun dari konsolidasi literatur (Hermanto et al., 2024), dengan tiga komponen: data kerusakan, data gejala, dan aturan. Hasil konsolidasi berupa 10 jenis kerusakan dan 29 gejala.

Tabel 2. Data kerusakan, solusi, dan tingkat urgensi

Kode	Kerusakan	Solusi penanganan	Urgensi
K01	*Mainboard*	Periksa area *mainboard*, terutama kapasitor yang mengembung; segera bawa ke teknisi	Berat
K02	*Chipset*/VGA	Tidak dapat ditangani manual; segera bawa ke pusat servis	Berat
K03	*Hardisk*	Periksa sambungan *hardisk* dengan soketnya; segera *backup* data sebelum rusak total	Sedang
K04	RAM	Lepas RAM, bersihkan pin dengan penghapus karet, pasang kembali; jika tetap gagal, ganti RAM	Sedang
K05	LCD	*Hard reset* (lepas baterai dan adaptor, tekan tombol power beberapa detik); jika berlanjut, periksa kabel fleksibel dan konektor LCD	Sedang
K06	Kipas pendingin	Bersihkan kipas dari debu, beri pelumas, ganti pasta termal jika kering; hindari memakai laptop di atas kasur	Sedang
K07	*Keyboard*	Matikan *Num Lock*/*Fn Lock*, bersihkan *keyboard* dari debu; jika belum teratasi, ganti *keyboard*	Ringan
K08	Baterai	Nyalakan laptop hanya dengan adaptor; jika normal tetapi baterai tidak mengisi, ganti baterai	Ringan
K09	*Charger*/adaptor	Uji adaptor pada laptop lain bermerek sama; jika gejala tetap, ganti adaptor	Ringan
K10	Sistem operasi	Perbaiki melalui *repair mode*, instal ulang jika perlu, bersihkan virus dengan antivirus	Ringan

Tingkat urgensi pada Tabel 2 ditentukan dari isi solusi pada literatur dengan mengacu pada kriteria Tabel 1. Kerusakan yang menurut literatur tidak dapat ditangani secara manual dan perlu dibawa ke teknisi atau pusat servis, yaitu *mainboard* dan *chipset*/VGA (Permadi et al., 2026), dikategorikan berat. Kerusakan yang dapat ditangani awal secara mandiri tetapi berisiko menimbulkan kehilangan data atau kerusakan lanjutan, seperti *hardisk*, RAM, LCD, dan kipas pendingin, dikategorikan sedang. Kerusakan yang solusinya berupa pemeriksaan sederhana atau penggantian komponen luar, yaitu *keyboard*, baterai, adaptor, dan perbaikan sistem operasi, dikategorikan ringan.

Tabel 3. Data gejala kerusakan

Kode	Gejala	Kerusakan
G01	Laptop tidak hidup meskipun adaptor dan baterai normal	K01
G02	Semua lampu indikator LED tidak menyala	K01
G03	Laptop sering mati atau *restart* sendiri	K01, K06
G04	Layar kosong (*blank*) padahal laptop menyala	K02
G05	Muncul garis-garis warna tidak beraturan pada layar	K02
G06	Gambar tidak tampil pada monitor eksternal	K02

G07	*Hardisk* tidak terdeteksi pada BIOS	K03
G08	Terdengar suara aneh dari *hardisk*	K03
G09	Muncul pesan *unknown disk boot error* atau *retry boot disk*	K03
G10	Muncul *blue screen*	K04
G11	Laptop gagal *booting*	K04
G12	Terdengar bunyi *beep* saat *booting*	K04
G13	Layar tampak redup atau gelap	K05
G14	Muncul garis vertikal atau horizontal pada layar	K05
G15	Gambar tampil normal pada monitor eksternal	K05
G16	Laptop panas berlebih	K06
G17	Bunyi kipas lebih kencang dari biasanya	K06
G18	Sebagian atau seluruh tombol *keyboard* tidak berfungsi	K07
G19	Huruf atau angka yang muncul tidak sesuai saat mengetik	K07
G20	Tombol *Fn* tidak berfungsi	K07
G21	Baterai tidak dapat terisi penuh	K08
G22	Terdapat tanda silang pada indikator baterai	K08
G23	Laptop hanya dapat menyala jika tersambung adaptor	K08
G24	Indikator pengisian tidak menyala saat adaptor dipasang	K09
G25	Lampu adaptor berkedip berulang kali	K09
G26	Laptop hidup lalu mati saat di-*charge*	K09
G27	Laptop sangat lambat	K10
G28	Instalasi atau pembaruan *Windows* gagal	K10
G29	Laptop terserang virus	K10

Gejala G06 dan G15 berfungsi sebagai pembeda antara kerusakan chipset/VGA dan LCD. Jika gambar tidak tampil pada monitor eksternal, gangguan diduga berada pada pengolah grafis, sedangkan jika gambar tampil normal, gangguan diduga berada pada panel LCD atau kabelnya. Penggunaan gejala sebagai fakta pembeda untuk mengarahkan diagnosis juga sejalan dengan pendekatan sistem pakar diagnosis hardware laptop berbasis *forward chaining* (Hasnan & Willy, 2022; Suryawan et al., 2024).

Tabel 4. Aturan (*rule*) IF–THEN

Rule	Aturan	Kerusakan	Urgensi
R01	IF G01 AND G02 AND G03	K01 *Mainboard*	Berat
R02	IF G04 AND G05 AND G06	K02 *Chipset*/VGA	Berat
R03	IF G07 AND G08 AND G09	K03 *Hardisk*	Sedang
R04	IF G10 AND G11 AND G12	K04 RAM	Sedang
R05	IF G13 AND G14 AND G15	K05 LCD	Sedang
R06	IF G03 AND G16 AND G17	K06 Kipas pendingin	Sedang
R07	IF G18 AND G19 AND G20	K07 *Keyboard*	Ringan
R08	IF G21 AND G22 AND G23	K08 Baterai	Ringan
R09	IF G24 AND G25 AND G26	K09 *Charger*/adaptor	Ringan
R10	IF G27 AND G28 AND G29	K10 Sistem operasi	Ringan

Setiap aturan pada Tabel 4 aktif apabila seluruh gejala pada bagian IF dipilih pengguna, lalu menghasilkan jenis kerusakan dan tingkat urgensinya pada bagian THEN. Gabungan gejala pada R04 (G10, G11, G12) merujuk pada aturan RAM yang dirumuskan (Permadi et al., 2026), sedangkan aturan lainnya disusun dari konsolidasi gejala pada sumber literatur yang sama.

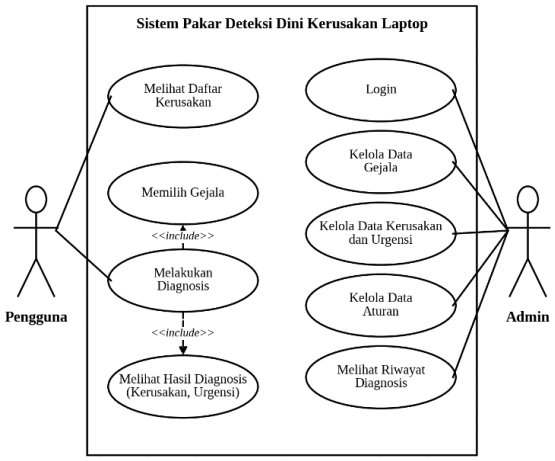
Perancangan Sistem

Perancangan sistem dimodelkan dengan Unified Modeling Language (UML) yang terdiri dari use case diagram, activity diagram, dan diagram alir proses inferensi beserta penentuan tingkat urgensi. Sistem dirancang untuk dua aktor, yaitu pengguna dan admin, sebagaimana pola perancangan sistem pakar kerusakan laptop berbasis web yang juga menggunakan pemodelan sistem (Permadi et al., 2026; Taufiq & Sandi, 2021).

Use Case Diagram

Use Case Diagram pada Gambar 2 menggambarkan fungsi yang dapat diakses setiap aktor. Pengguna dapat melihat daftar kerusakan dan melakukan diagnosis. Proses diagnosis mencakup pemilihan gejala dan penampilan hasil diagnosis berupa jenis kerusakan, tingkat urgensi, dan

rekomendasi tindakan. Pengguna tidak perlu melakukan login, sehingga sistem dapat langsung dipakai untuk diagnosis awal (Delimayanti et al., 2022). Admin harus melakukan login terlebih dahulu, kemudian dapat mengelola data gejala, data kerusakan beserta tingkat urgensinya, dan data aturan, serta melihat riwayat diagnosis. Pengelolaan data oleh admin memungkinkan basis pengetahuan diperbarui seiring perkembangan jenis kerusakan laptop tanpa mengubah program.

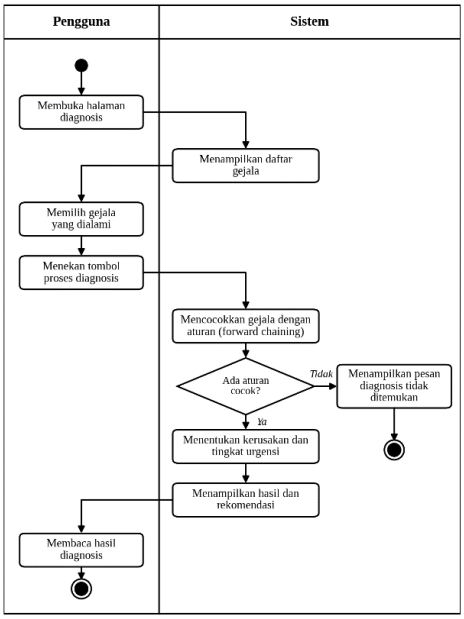


Sumber: Hasil Penelitian (2026)

Gambar 2. *Use Case Diagram* sistem pakar deteksi dini kerusakan laptop

Activity Diagram

Activity diagram pada Gambar 3 menggambarkan alur proses diagnosis antara pengguna dan sistem. Proses diawali ketika pengguna membuka halaman diagnosis, lalu sistem menampilkan daftar gejala. Pengguna memilih gejala yang dialami laptop dan menekan tombol proses diagnosis. Sistem mencocokkan gejala yang dipilih dengan aturan pada basis pengetahuan menggunakan *forward chaining*. Jika terdapat aturan yang cocok, sistem menentukan jenis kerusakan beserta tingkat urgensinya, kemudian menampilkan hasil dan rekomendasi yang dibaca pengguna. Jika tidak ada aturan yang cocok, sistem menampilkan pesan bahwa diagnosis tidak ditemukan.



Sumber: Hasil Penelitian (2026)

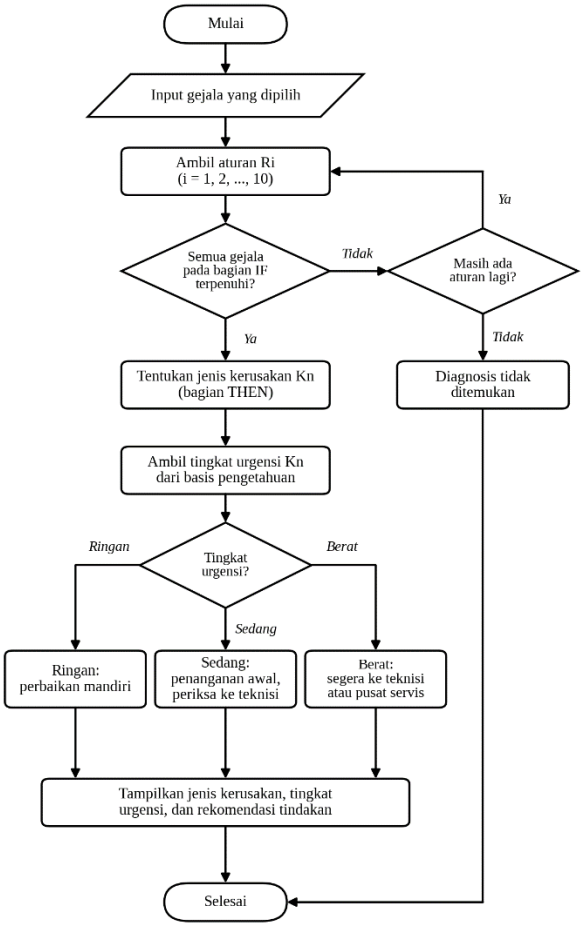
Gambar 3. Activity diagram proses diagnosis kerusakan laptop

Alur Inferensi dan Penentuan Tingkat Urgensi

Alur proses inti sistem ditunjukkan pada Gambar 4. Setelah pengguna memasukkan gejala yang dipilih, mesin inferensi mengambil aturan R01 sampai R10 pada Tabel 4 secara berurutan. Setiap aturan

diperiksa apakah seluruh gejala pada bagian IF terpenuhi. Jika terpenuhi, bagian THEN menghasilkan jenis kerusakan, lalu sistem mengambil tingkat urgensi kerusakan tersebut dari basis pengetahuan (Tabel 2). Jika tidak terpenuhi, sistem memeriksa aturan berikutnya sampai seluruh aturan habis diperiksa. Bila tidak ada satu pun aturan yang cocok, sistem menyatakan diagnosis tidak ditemukan. Alur pencocokan fakta terhadap aturan produksi tersebut merupakan karakteristik inferensi *forward chaining* (Chaniago et al., 2025; Hasnan & Willy, 2022).

Penentuan tingkat urgensi dilakukan dengan mengambil nilai yang telah dipetakan pada setiap jenis kerusakan, sehingga sistem tidak menghitung urgensi secara terpisah. Rekomendasi yang ditampilkan mengikuti kriteria pada Tabel 1: urgensi ringan menghasilkan anjuran perbaikan mandiri, urgensi sedang menghasilkan anjuran penanganan awal dan pemeriksaan ke teknisi jika masalah berlanjut, dan urgensi berat menghasilkan anjuran untuk segera dibawa ke teknisi atau pusat servis. Dengan alur ini, pengguna memperoleh tiga keluaran sekaligus, yaitu jenis kerusakan, tingkat urgensi, dan tindakan yang disarankan.



Sumber: Hasil Penelitian (2026)

Gambar 4. Diagram alir proses inferensi dan penentuan tingkat urgensi

Implementasi *Forward chaining*

Proses inferensi mengikuti alur pada Gambar 4 dan dirumuskan dalam Algoritma 1. Mesin inferensi bekerja secara data-driven: penalaran dimulai dari himpunan gejala yang dipilih pengguna, lalu aturan pada basis pengetahuan diperiksa satu per satu dari R01 sampai R10. Sebuah aturan aktif apabila seluruh gejala pada bagian IF terpenuhi oleh gejala yang dipilih (Hermanto et al., 2024). Setelah aturan aktif, bagian THEN menghasilkan jenis kerusakan, dan tingkat urgensi diambil dari data kerusakan pada Tabel 2.

Algoritma 1. *Forward chaining* dengan penentuan tingkat urgensi
Input : G = himpunan gejala yang dipilih pengguna
Output : jenis kerusakan, tingkat urgensi, rekomendasi tindakan

```
1  for i ← 1 to 10 do
2     if semua gejala pada bagian IF aturan Ri ∈ G then
3        Kn ← bagian THEN aturan Ri
4        urgensi ← tingkat urgensi Kn pada basis pengetahuan
5        tampilkan Kn, urgensi, dan rekomendasi sesuai urgensi
6        selesai
7  tampilkan "Diagnosis tidak ditemukan"
```

Penerapan algoritma diperagakan pada lima contoh kasus di Tabel 5. Format penyajiannya mengikuti contoh kasus pada , yaitu gejala yang dipilih, aturan yang cocok, hasil(Permadi et al., 2026), dan solusi, dengan tambahan kolom tingkat urgensi.

Tabel 5. Contoh kasus penerapan *Forward Chaining*

Kasus	Gejala yang dipilih	Aturan cocok	Kerusakan	Urgensi	Rekomendasi
1	G07, G08, G09	R03	K03 *Hardisk*	Sedang	Periksa sambungan *hardisk* dengan soketnya dan segera *backup* data sebelum rusak total
2	G24, G25, G26	R09	K09 *Charger*/adaptor	Ringan	Uji adaptor pada laptop lain bermerek sama; jika gejala tetap, ganti adaptor
3	G04, G05, G06	R02	K02 *Chipset*/VGA	Berat	Tidak dapat ditangani manual; segera bawa ke pusat servis
4	G03, G16, G17	R06	K06 Kipas pendingin	Sedang	Bersihkan kipas dari debu, beri pelumas, ganti pasta termal jika kering
5	G07, G08	Tidak ada	Tidak ditemukan	–	Pesan “diagnosis tidak ditemukan”

Pada Kasus 1, pengguna memilih G07 (hardisk tidak terdeteksi pada BIOS), G08 (suara aneh dari hardisk), dan G09 (pesan unknown disk boot error). Mesin inferensi memeriksa R01 yang membutuhkan G01, G02, dan G03 sehingga tidak terpenuhi. R02 yang membutuhkan G04, G05, dan G06 juga tidak terpenuhi. R03 membutuhkan G07, G08, dan G09, seluruhnya terpenuhi, sehingga aturan ini aktif dan menyimpulkan kerusakan hardisk dengan urgensi sedang. Pengguna diarahkan untuk mencadangkan data segera sebelum hardisk rusak total, sedangkan penanganan lanjutan perlu diperiksakan ke teknisi.

Kasus 4 memperlihatkan peran gejala yang dipakai bersama. G03 (laptop sering mati atau restart sendiri) ada pada R01 dan R06. Karena G01 dan G02 tidak dipilih, R01 tidak terpenuhi, dan karena G16 dan G17 dipilih, R06 aktif sehingga disimpulkan kerusakan kipas pendingin. Kasus 5 memperlihatkan keterbatasan aturan yang menuntut seluruh gejala terpenuhi. Pengguna hanya memilih G07 dan G08, sedangkan R03 juga membutuhkan G09, sehingga tidak ada aturan yang cocok dan sistem menampilkan pesan bahwa diagnosis tidak ditemukan. Keterbatasan ini merupakan konsekuensi dari aturan deterministik yang mensyaratkan seluruh premis terpenuhi. Farhan dan Perkasa (2026) melaporkan bahwa kombinasi Forward Chaining–Certainty Factor pada data uji mereka dapat menangani masukan gejala yang tidak lengkap atau tidak pasti dengan lebih baik dibandingkan Forward Chaining saja. Temuan tersebut mendukung peluang pengembangan sistem ini, tetapi angka akurasinya tidak dapat langsung dibandingkan karena jumlah kasus, basis pengetahuan, dan prosedur pengujiannya berbeda.

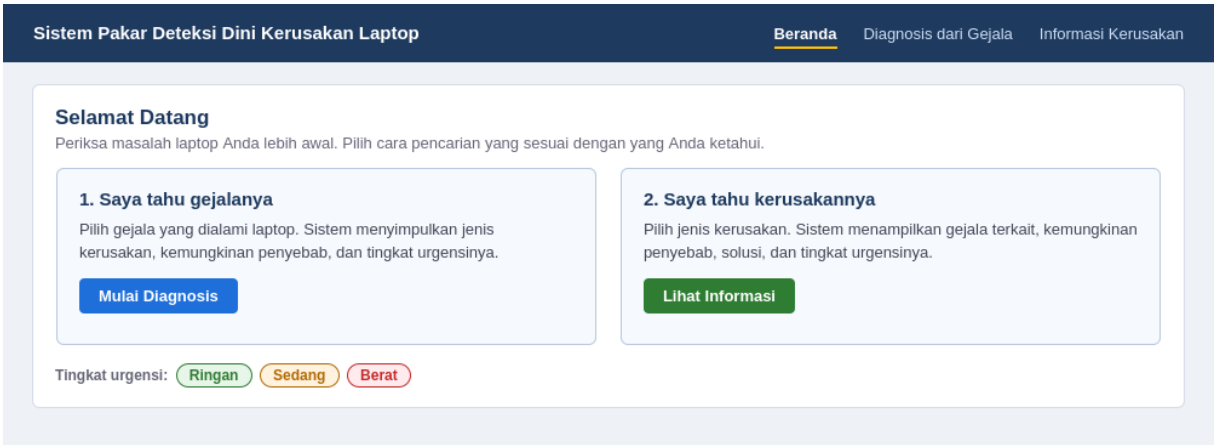
Perbandingan Kasus 2 dan Kasus 3 menunjukkan manfaat penentuan tingkat urgensi. Kedua kasus sama-sama menghasilkan jenis kerusakan, tetapi tindakan yang disarankan sangat berbeda: kerusakan adaptor pada Kasus 2 dapat diuji dan diganti sendiri oleh pengguna, sedangkan kerusakan chipset/VGA pada Kasus 3 harus segera dibawa ke pusat servis. Pada sistem yang hanya menampilkan jenis kerusakan, perbedaan ini harus ditafsirkan sendiri oleh pengguna. Pada sistem ini, perbedaan tersebut langsung dinyatakan dalam keluaran.

Implementasi Antarmuka

Antarmuka sistem dikembangkan berbasis web dan dirancang sederhana agar pengguna awam dapat langsung memakainya tanpa login. Sistem menyediakan dua cara pencarian sesuai informasi yang diketahui pengguna. Jika pengguna hanya mengetahui gejala, sistem menyimpulkan jenis kerusakannya

melalui proses *forward chaining*. Jika pengguna sudah mengetahui jenis kerusakannya, sistem menampilkan informasi kerusakan tersebut dari basis pengetahuan.

Halaman utama

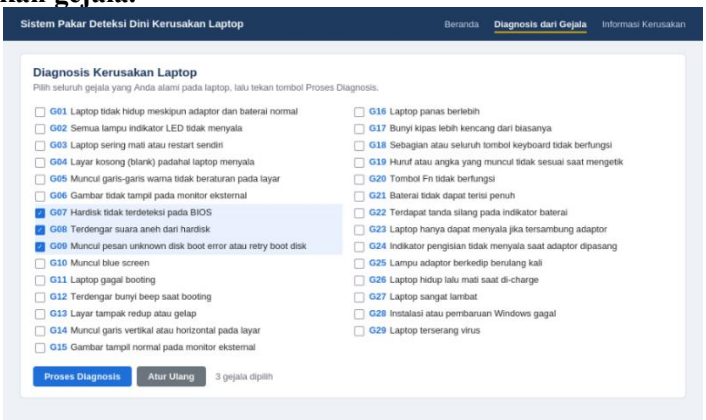


Sumber: Hasil Penelitian (2026)

Gambar 5. Tampilan antar muka halaman utama

memuat dua pilihan, yaitu “Saya tahu gejalanya” dan “Saya tahu kerusakannya”. Halaman ini juga menjelaskan arti label tingkat urgensi yang berwarna hijau untuk ringan, oranye untuk sedang, dan merah untuk berat, sehingga pengguna dapat mengenali tingkat kedaruratan dalam sekali lihat.

Pencarian berdasarkan gejala.

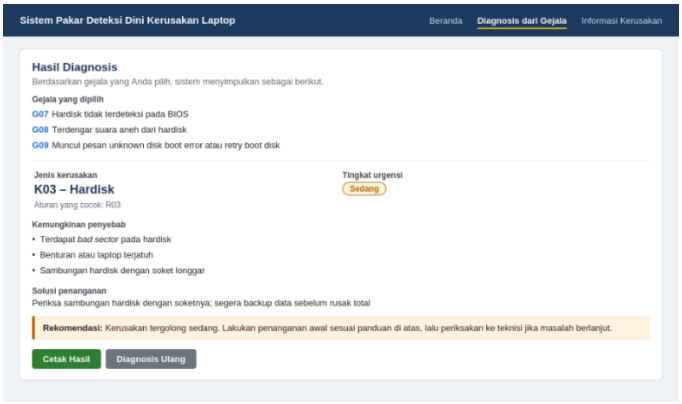


Sumber: Hasil Penelitian (2026)

Gambar 6. Tampilan antar muka halaman diagnosis

menampilkan seluruh 29 gejala pada Tabel 3 dalam dua kolom dengan kotak centang. Pengguna memilih gejala yang dialami laptopnya, kemudian menekan tombol Proses Diagnosis. Gejala yang dipilih ditandai dengan warna latar berbeda dan sistem menampilkan jumlah gejala yang telah dipilih. Tombol Atur Ulang menghapus seluruh pilihan.

Halaman hasil diagnosis

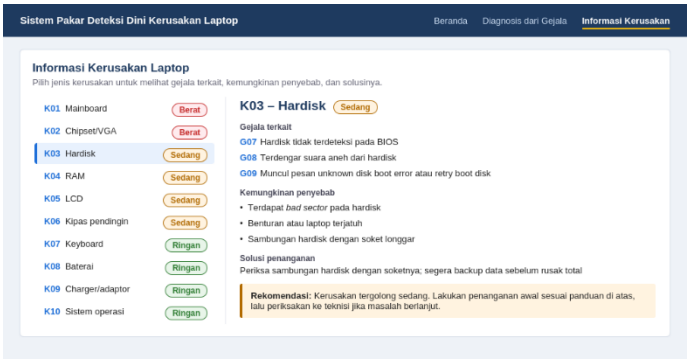


Sumber: Hasil Penelitian (2026)

Gambar 7. Tampilan antar muka halaman hasil diagnosis

menampilkan gejala yang dipilih, jenis kerusakan beserta aturan yang cocok, tingkat urgensi, kemungkinan penyebab, solusi penanganan, dan rekomendasi tindakan. Gambar 7 memperlihatkan hasil untuk Kasus 1 pada Tabel 5, yaitu gejala G07, G08, dan G09 yang menghasilkan kerusakan hardisk (K03) dengan urgensi sedang. Kotak rekomendasi menyesuaikan tingkat urgensi sehingga pengguna langsung mengetahui apakah kerusakan dapat ditangani sendiri atau perlu dibawa ke teknisi.

Pencarian berdasarkan kerusakan.



Sumber: Hasil Penelitian (2026)

Gambar 8. Tampilan antar muka halaman informasi kerusakan

menampilkan daftar 10 jenis kerusakan beserta label urgensinya. Ketika pengguna memilih salah satu kerusakan, sistem menampilkan gejala terkait, kemungkinan penyebab, solusi penanganan, dan rekomendasi tindakan. Gambar 8 memperlihatkan informasi kerusakan hardisk (K03). Pada cara ini sistem hanya mengambil data dari basis pengetahuan dan tidak menjalankan inferensi, sehingga hasilnya tetap konsisten dengan hasil pencarian berdasarkan gejala.

Pengujian Sistem

Pengujian dilakukan dalam dua tahap sesuai metode pengujian yang telah ditetapkan, yaitu black box testing untuk memeriksa fungsi setiap fitur dan pengujian akurasi untuk membandingkan hasil diagnosis sistem dengan diagnosis pakar. Penggunaan black box testing untuk mengevaluasi fungsi aplikasi web juga diterapkan dalam penelitian pengujian perangkat lunak terkini (Putri et al., 2026).

Hasil Black Box Testing

Black box testing dilakukan dengan memberikan masukan pada setiap fitur lalu mengamati apakah keluaran sistem sesuai dengan yang diharapkan, tanpa memeriksa struktur kode program. Pendekatan pengujian fungsional dari sisi input dan output ini digunakan pula pada evaluasi aplikasi berbasis web (Putri et al., 2026). Hasil pengujian ditunjukkan pada Tabel 6.

Tabel 6. Hasil *Black Box Testing*

No	Fitur yang diuji	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian	Status
1	Halaman utama	Membuka halaman utama sistem	Tampil pilihan “Saya tahu gejalanya” dan “Saya tahu kerusakannya” beserta keterangan label urgensi	Sesuai	Valid
2	Pemilihan gejala	Mencentang beberapa gejala pada halaman diagnosis	Gejala yang dipilih berubah warna latar dan jumlah gejala terpilih bertambah	Sesuai	Valid
3	Proses diagnosis (gejala lengkap)	Memilih G07, G08, G09 lalu menekan Proses Diagnosis	Tampil kerusakan *hardisk* (K03), urgensi sedang, dan rekomendasi tindakan	Sesuai	Valid
4	Proses diagnosis (gejala tidak lengkap)	Memilih G07 dan G08 saja lalu menekan Proses Diagnosis	Tampil pesan “diagnosis tidak ditemukan”	Sesuai	Valid
5	Gejala yang dipakai bersama	Memilih G03, G16, G17	Tampil kerusakan kipas pendingin (K06), bukan *mainboard*	Sesuai	Valid
6	Tombol Atur Ulang	Menekan tombol Atur Ulang setelah memilih gejala	Seluruh pilihan gejala terhapus	Sesuai	Valid

7	Label urgensi	Menjalankan diagnosis pada kerusakan ringan, sedang, dan berat	Label tampil hijau (ringan), oranye (sedang), dan merah (berat)	Sesuai	Valid
8	Rekomendasi sesuai urgensi	Membandingkan hasil kerusakan adaptor dan *chipset*/VGA	Adaptor menghasilkan anjuran perbaikan mandiri, *chipset*/VGA menghasilkan anjuran segera ke pusat servis	Sesuai	Valid
9	Pencarian berdasarkan kerusakan	Memilih salah satu kerusakan pada daftar	Tampil gejala terkait, kemungkinan penyebab, solusi, dan rekomendasi	Sesuai	Valid
10	Login admin	Login dengan akun yang benar dan akun yang salah	Akun benar masuk ke halaman admin, akun salah ditolak	Sesuai	Valid
11	Kelola data admin	Menambah, mengubah, dan menghapus data gejala, kerusakan, dan aturan	Data tersimpan dan perubahan tampil pada sistem	Sesuai	Valid
12	Riwayat diagnosis	Membuka menu riwayat diagnosis pada akun admin	Tampil daftar diagnosis yang pernah dilakukan pengguna	Sesuai	Valid

4. KESIMPULAN, KETERBATASAN DAN SARAN

Penelitian ini berhasil merancang dan membangun sistem pakar deteksi dini kerusakan laptop berbasis web menggunakan metode Forward Chaining dengan basis pengetahuan berupa 10 jenis kerusakan, 29 gejala, dan 10 aturan IF–THEN. Sistem mampu mengidentifikasi jenis kerusakan, menentukan tingkat urgensi (ringan, sedang, dan berat), serta memberikan rekomendasi penanganan sesuai hasil diagnosis. Pengujian menunjukkan bahwa seluruh fitur berfungsi sesuai skenario dan hasil diagnosis sesuai dengan penilaian pakar pada kasus yang diuji. Dengan demikian, sistem dapat membantu pengguna melakukan diagnosis awal dan menentukan tindakan penanganan kerusakan laptop secara tepat.

Penelitian ini memiliki keterbatasan pada cakupan basis pengetahuan yang masih terbatas, serta proses diagnosis yang mensyaratkan seluruh gejala dalam suatu aturan terpenuhi. Selain itu, pengujian masih terbatas pada 10 kasus sehingga belum mencakup seluruh variasi kerusakan laptop. Oleh karena itu, penelitian selanjutnya disarankan untuk menambah jenis kerusakan dan gejala, mengintegrasikan metode Certainty Factor untuk mengatasi ketidakpastian diagnosis, memperluas pengujian dengan lebih banyak kasus, serta melakukan pengujian kemudahan penggunaan. Pengembangan aplikasi berbasis mobile juga dapat dilakukan untuk meningkatkan aksesibilitas sistem bagi pengguna.

DAFTAR PUSTAKA

Chaniago, J., Rohayani, H., & Halim, A. (2025). Implementation of forward chaining method in laptop damage detection expert system. Journal of Applied Research in Computer Science and Information Systems, 3(1), 232–239. https://doi.org/10.61098/jarcis.v3i1.246

Delimayanti, M. K., Yusfendri, P. R., Laya, M., & Naryanto, R. F. (2022). Sistem pakar diagnosa kerusakan hardware laptop dengan metode forward chaining berbasis web. Seminar Nasional Inovasi Vokasi, 1, 1–11. https://prosiding.pnj.ac.id/index.php/sniv/article/view/269

Fajriati, N., & Budiman, K. (2022). Web-based employee attendance system development using waterfall method. Journal of Advances in Information Systems and Technology, 3(2), 8–20. https://doi.org/10.15294/jaist.v3i2.52942

Farhan, M., & Perkasa, E. B. (2026). Accuracy testing of laptop damage diagnosis using forward chaining and certainty factor algorithms. International Journal Software Engineering and Computer Science, 6(2), 728–738. https://doi.org/10.35870/ijsecs.v6i2.7726

Hasnan, A., & Willy, A. (2022). Sistem pakar diagnosa kerusakan hardware laptop menggunakan meode forward chaining berbasis web. Jurnal Surya Informatika, 12(2), 1–7. https://doi.org/10.48144/suryainformatika.v12i2.1364

Hermanto, H., Hendi, A., Rohman, T., & Pamuji, F. S. (2024). Pembuatan sistem pakar diagnosa kerusakan perangkat keras laptop dengan metode forward chaining. JATI (Jurnal Mahasiswa Teknik Informatika), 8(3), 3795–3801. https://doi.org/10.36040/jati.v8i3.9788

Permadi, D., Handayani, D., Kustanto, P., Yasir, M., Noeman, A., & Hidayat, A. (2026). Implementasi forward chaining pada sistem pakar diagnosa kerusakan laptop berbasis web: Studi kasus layanan servis laptop. Journal of Informatic and Information Security, 6(2), 115–128. https://doi.org/10.31599/paw2f574

Putri, N. A., Rahma, D. N., Zahir, R. H., Thoriq, M. F., Ramdhana, M. F., Wicaksono, A., & Nasir, M. (2026). Pengujian black box pada sistem website Bio Futurindo menggunakan teknik decision table. Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM), 7(1), 1–12. https://doi.org/10.31102/jatim.v7i1.3115

Rahman, A. D., Juhriah, E., & Erlangga, F. (2024). Sistem pakar diagnosa kerusakan laptop pada PT. Primalayan Citra Mandiri menggunakan forward chaining berbasis Java. Jurnal Rekayasa Komputasi Terapan, 4(2), 132–139. https://doi.org/10.30998/jrkt.v4i02.10238

Supu, F. T., Thayf, S. S., & Marlina. (2026). Sistem pakar troubleshooting kerusakan hardware komputer berbasis web dengan metode forward chaining pada laboratorium STMIK Kharisma Makassar. JTRISTE, 13(1), 23–33. https://doi.org/10.55645/jtriste.v13i1.710

Suryawan, S. B., Wibawanti, Y., & Irawan, A. (2024). Sistem pakar diagnosa kerusakan hardware laptop menggunakan metode forward chaining berbasis Java. Jurnal Rekayasa Komputasi Terapan, 4(4), 322–329. https://doi.org/10.30998/jrkt.v4i04.11797

Taufiq, R., & Sandi, A. P. (2021). Perancangan sistem pakar diagnosa kerusakan laptop dengan penerapan metode forward chaining. JIKA (Jurnal Informatika), 5(2), 260–263. https://doi.org/10.31000/jika.v5i2.4598