

Rancang Bangun Aplikasi Diagnosa Kerusakan Komputer dengan Sistem Pakar Metode Forward Chaining

Susi Widayati¹, Yudi Irawan Chandra², Raudhatul Zanna Fitriyani³

^{1,2}STMIK Jakarta STI&K, Jakarta Selatan, Indonesia

³Universitas Gunadarma, Jawa Barat, Indonesia

E-mail: ¹widayatisusi@gmail.com, ²yirawanc@gmail.com,

³rz.fitriyani@gmail.com

* Corresponding Author

Abstract

Computers can incur a wide range of damage, varying in severity from minor issues that are easily resolved to more complex problems that might potentially render the computer inoperable. This damage exhibits nearly identical symptoms across many computer components. Unfortunately, computer owners typically overlook these signs, leading to a worsening of the damage and subsequent disruption for computer users. Technicians are highly skilled professionals in the domain of computers. His expertise can be used to a system known as an expert system. The advancement of information technology necessitates innovation in the realm of computer maintenance and identification of hardware malfunctions. The objective of this study is to develop an application that utilizes an Expert System with the Forward Chaining technique to diagnose computer malfunctions. This strategy enables the system to generate diagnostic procedures by utilizing factual information obtained through an interactive exchange with the user. This application is specifically developed to aid users, particularly those with less technical expertise, in discovering and fixing computer issues. The expert system employs artificial intelligence to examine the symptoms provided by the user. It then employs a forward chaining procedure to ascertain the potential harm and the appropriate remedy. The key benefit of this application is its capacity to swiftly and precisely deliver solutions by leveraging the integrated knowledge within the system. The deployment of this application is expected to enhance efficiency in identifying and resolving computer malfunctions, while also offering a superior user experience in managing technical issues on their devices..

Keywords: diagnosis application, computer malfunction, expert system, forward chaining, knowledge integration

Abstrak

Banyak jenis kerusakan komputer yang dialami oleh komputer, mulai dari yang mudah ditangani hingga yang sulit ditangani hingga menyebabkan kerusakan yang fatal pada komputer. Kerusakan ini memiliki gejala yang hampir sama pada setiap komponen komputer, sering sekali pemilik komputer mengabaikan gejala yang ditimbulkan, sehingga pada saat kerusakan semakin parah dan mengganggu pengguna komputer. Teknisi merupakan pakar dalam bidang komputer. Pengetahuannya dapat diimplementasikan menjadi sebuah sistem yang disebut sistem pakar. Perkembangan teknologi informasi menuntut adanya inovasi dalam bidang perawatan dan diagnosa kerusakan komputer. Penelitian ini bertujuan untuk merancang sebuah aplikasi diagnosa kerusakan komputer menggunakan Sistem Pakar dengan metode Forward Chaining. Metode ini memungkinkan sistem untuk menyusun langkah-langkah diagnosis berdasarkan fakta-fakta yang ditemukan selama proses interaktif dengan pengguna. Aplikasi ini dirancang untuk membantu pengguna, terutama yang memiliki keterbatasan pengetahuan teknis, dalam mengidentifikasi dan mengatasi masalah yang mungkin terjadi

pada komputer. Dengan memanfaatkan kecerdasan buatan, sistem pakar akan menganalisis gejala-gejala yang dilaporkan oleh pengguna, kemudian melakukan proses forward chaining untuk menentukan kemungkinan kerusakan dan solusi yang sesuai. Keunggulan aplikasi ini terletak pada kemampuan untuk memberikan solusi secara cepat dan akurat berdasarkan pengetahuan yang terintegrasi dalam sistem. Diharapkan bahwa implementasi aplikasi ini dapat meningkatkan efisiensi dalam mendeteksi dan memperbaiki kerusakan komputer, serta memberikan pengalaman pengguna yang lebih baik dalam mengelola permasalahan teknis pada perangkat mereka.

Kata kunci: *aplikasi diagnosa, kerusakan komputer, sistem pakar, forward chaining, integrasi pengetahuan*

1. Pendahuluan

Perkembangan teknologi dan komputer pada masa kini telah berkembang pesat sehingga dapat membantu manusia mempermudah pekerjaannya. Sebagai contoh kemajuan teknologi dan komputer adalah sistem pakar. Sistem pakar sendiri merupakan suatu program komputer cerdas yang memiliki pengetahuan untuk menggunakan penalaran inferensi menyerupai seorang ahli dalam menyelesaikan masalah tertentu. Implementasi sistem pakar dapat diterapkan dalam dunia teknologi komputer selain sebagai media informasi bagi masyarakat terutama untuk mengetahui jenis kerusakan yang dialami komputer sebagai diagnosa awal, juga digunakan alat bantu bagi teknisi agar dapat mengambil keputusan secara cepat dan akurat. Pengetahuan yang disimpan dalam sistem pakar umumnya diambil dari manusia yang ahli dalam masalah tersebut dan sistem pakar itu sendiri meniru metodologi dan kinerjanya [1], [2].

Diagnosa yang ditampilkan akan membantu pengguna memahami bahwa kerusakan komputer bukan hanya hang atau crash, namun ada beberapa jenis kerusakan yang lainnya yang mungkin muncul jarang diketahui karena minimnya informasi yang biasanya diketahui oleh teknisi. Dengan gejala awal yang pada umumnya hampir sama dengan kerusakan sistem komputer lainnya membuat diagnosa jenis kerusakan komputer ini lama diketahui.

Dengan permasalahan tersebut, maka dibuatlah suatu aplikasi sistem pakar diagnosa kerusakan sistem komputer. Aplikasi ini menggunakan metode sistem pakar dimana saat pengguna menjawab beberapa pertanyaan yang diajukan sistem, sistem akan menunjukkan kerusakan komputer yang sedang dialami oleh komputer tersebut [3], [4].

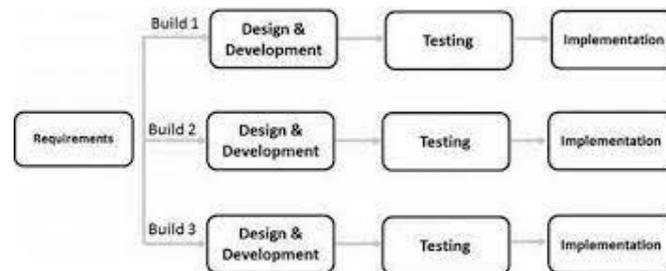
Pada sistem pakar ini difokuskan untuk memberikan informasi tentang berbagai jenis kerusakan komputer, mendiagnosa 10 jenis kerusakan komputer, dengan 25 gejala yang dialami. Sistem pakar ini menggunakan metode forward chaining. Adapun 10 jenis kerusakan terbagi menjadi 8 jenis kerusakan pada mouse, 1 kerusakan memori dan 1 kerusakan pada perangkat lunak.

Tujuan dari penulisan ini adalah sebagai pembuatan aplikasi sistem pakar diagnosa awal jenis kerusakan komputer serta memberikan pengetahuan mengenai berbagai macam gejala dan kerusakan yang dialami komputer.

2. Metodologi Penelitian

Metode iteratif adalah suatu pendekatan pengembangan atau penyelesaian masalah yang melibatkan serangkaian langkah-langkah yang diulang-ulang secara berulang [5], [6]. Dalam konteks pengembangan perangkat lunak atau proyek-proyek kompleks, metode iteratif melibatkan siklus pengembangan yang terdiri dari serangkaian tahap, dan setiap tahap dilakukan secara berulang sampai mencapai hasil yang diinginkan atau optimal [7], [8]. Beberapa karakteristik umum dari metode iteratif meliputi [9]:

- 1 Siklus Berulang: Proses pengembangan terdiri dari siklus atau iterasi yang diulang secara teratur. Setiap iterasi dapat mencakup tahap perencanaan, analisis, desain, implementasi, evaluasi, dan penyesuaian.
- 2 Pengembangan Bertahap: Pengembangan dilakukan secara bertahap, memungkinkan penambahan fungsionalitas atau perubahan desain seiring berjalannya waktu. Setiap tahap dapat menghasilkan prototipe atau versi yang dapat diuji.
- 3 Feedback dan Evaluasi: Pengguna dan pemangku kepentingan terlibat secara aktif dalam memberikan umpan balik pada setiap iterasi. Hal ini memungkinkan perbaikan dan penyesuaian berdasarkan respons yang diterima.
- 4 Fleksibilitas dan Adaptabilitas: Metode ini memberikan fleksibilitas dalam menanggapi perubahan kebutuhan atau persyaratan proyek, sehingga memudahkan adaptasi terhadap dinamika yang mungkin terjadi selama pengembangan.
- 5 Optimasi Melalui Iterasi: Setiap iterasi bertujuan untuk meningkatkan kualitas dan performa produk atau sistem, sehingga secara akhir menghasilkan solusi yang optimal.



Gambar 1. Model Iteratif

Kelebihan Metode Iteratif [10]:

- 1 Fleksibilitas: Metode iteratif memungkinkan fleksibilitas dalam menanggapi perubahan kebutuhan atau persyaratan proyek. Tim pengembang dapat dengan mudah menyesuaikan perangkat lunak dengan perubahan yang mungkin terjadi selama pengembangan.
- 2 Umpan Balik Berkala: Keterlibatan pengguna dan pemangku kepentingan dalam setiap iterasi memberikan umpan balik yang berkala. Hal ini memungkinkan untuk perbaikan cepat dan penyesuaian sesuai dengan kebutuhan pengguna.
- 3 Pengendalian Risiko: Dengan siklus pengembangan yang terulang, risiko proyek dapat diidentifikasi dan dikendalikan lebih efektif. Hal ini membantu dalam meminimalkan risiko yang terkait dengan perubahan kebutuhan atau ketidakpastian proyek.
- 4 Peningkatan Kualitas: Kualitas perangkat lunak dapat ditingkatkan secara bertahap melalui setiap iterasi. Pengujian yang berulang memungkinkan pengembang untuk mendeteksi dan memperbaiki bug atau kelemahan pada tahap awal pengembangan.
- 5 Evolusi Berkelanjutan: Metode ini mendukung evolusi berkelanjutan produk atau sistem seiring waktu. Setiap iterasi dapat menambahkan fungsionalitas baru atau memperbaiki yang sudah ada, menciptakan produk yang lebih matang dan sesuai dengan kebutuhan pengguna.

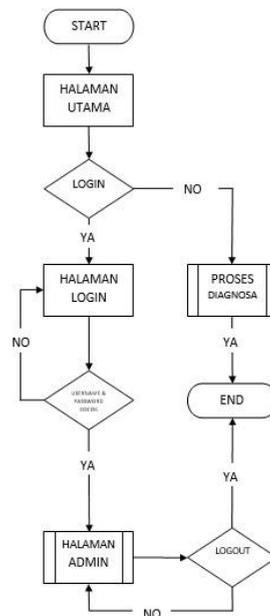
Kelemahan Metode Iteratif [11]:

- 1 Kebutuhan Sumber Daya yang Lebih Besar: Proses iteratif membutuhkan komitmen sumber daya yang lebih besar, terutama dalam hal waktu dan anggaran, karena melibatkan siklus pengembangan berulang.
- 2 Tidak Cocok untuk Proyek Sederhana: Metode ini mungkin terlalu kompleks untuk proyek-proyek yang sederhana dan tidak memerlukan banyak perubahan atau penyesuaian.

- 3 Kemungkinan Lingkaran Iterasi yang Tak Terbatas: Jika tidak dikelola dengan baik, metode iteratif dapat mengalami masalah dengan lingkaran iterasi yang tak terbatas, di mana pengembang terus melakukan perubahan tanpa mencapai tujuan yang jelas.
- 4 Keterbatasan pada Proyek dengan Batasan Waktu yang Ketat: Jika proyek memiliki batasan waktu yang ketat, metode iteratif mungkin tidak sesuai karena memerlukan waktu yang lebih lama untuk mencapai hasil yang optimal.
- 5 Memerlukan Komunikasi yang Efektif: Keterlibatan aktif pengguna dan pemangku kepentingan memerlukan komunikasi yang efektif dan kontinyu. Jika komunikasi terhambat, hal ini dapat menyulitkan proses iteratif.

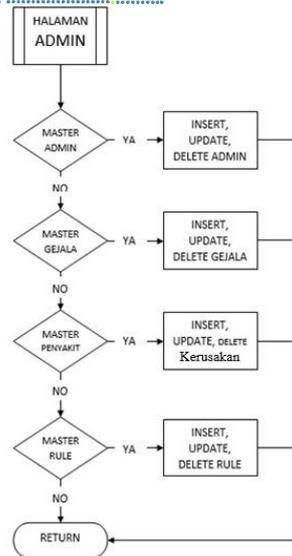
Metode iteratif sering digunakan dalam pengembangan perangkat lunak, desain produk, dan proyek-proyek kompleks lainnya untuk mengatasi ketidakpastian, meminimalkan risiko, dan memberikan hasil yang lebih baik melalui evolusi berkelanjutan. Beberapa contoh metode iteratif dalam pengembangan perangkat lunak termasuk Scrum, Extreme Programming (XP), dan Metode Spiral [12].

Pada aplikasi ini pembuatan *flowchart* bertujuan untuk memudahkan perancang membuat aplikasi secara terstruktur dan jelas. Sehingga aplikasi tersebut dapat berjalan lancar sesuai dengan alur kerja yang telah ditentukan perancang pada tahap pembuatan *flowchart*, terlihat pada Gambar 2.



Gambar 2. *Flowchart Aplikasi*

Flowchart halaman admin dibuat untuk menjelaskan alur kerja yang terjadi pada halaman admin. Selain itu dapat terlihat kegiatan yang terjadi pada halaman admin, seperti pada Gambar 3.



Gambar 3. Flowchart pada Halaman Admin

3. Hasil dan Pembahasan

Berikut ini adalah tahapan pembangunan aplikasi berdasarkan model iterative:

3.1. Requirement

Dalam penerapan penggunaan sistem yang dibuat, digunakan perangkat keras dan perangkat lunak antara lain :

1. Perangkat Keras

Hardware merupakan sarana fisik untuk menghasilkan data, program dan keluaran. Spesifikasi perangkat keras yang digunakan sebagai pendukung server dalam menjalankan aplikasi ini yaitu :

- a. Processor: Intel Core i7
- b. Memori : 16.0GB Dual-Channel
- c. HardDisk: 2794GB Seagate dan 931GB Seagate
- d. Monitor : 1920x1080 pixels
- e. Keyboard : 108 keys
- f. Mouse : Optic PS/2

2. Perangkat lunak

Dalam pengembangan aplikasi ini penulis menggunakan perangkat lunak dengan spesifikasi sebagai berikut :

- a. Sistem Operasi: Windows 11
- b. Paket Program :
 - 1) Visual Studio 2010
 - 2) Microsoft Access

3.2. Design and Development

Dalam aplikasi ini terdapat menu Informasi, menu ini berisikan tentang informasi seputar kerusakan komputer, penjelasan singkat tentang komputer, dan tahap pemulihan komputer dari kerusakan tersebut. Menu selanjutnya yaitu menu konsultasi, menu ini merupakan sistem pakar yang akan digunakan pengguna untuk mendiagnosa kerusakan komputer. sistem pakar yang berisikan 10 jenis kerusakan dan 25 gejala. Menu *login* merupakan menu yang dikhususkan hanya untuk admin, dimana pada menu login ini nantinya admin dapat merubah isi database yang tersedia. Menu bantuan, pada menu ini berisikan mengenai website sistem pakar itu sendiri, menu terakhir yaitu menu tentang berisi tentang penulis.

Dalam merancang dan membangun sistem pakar ini perlu menganalisa dan mengumpulkan seluruh kebutuhan dan informasi yang dibutuhkan pada sistem pakar ini. Dalam membuat sistem pakar ini pertama kali dilakukan adalah menentukan metode representasi pengetahuan yaitu menggunakan kaidah produksi. Pada umumnya, kaidah produksi dituliskan dalam bentuk JIKA-MAKA (IF-THEN). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian yaitu premis (jika) dan bagian konklusi (maka). Apabila premis bernilai benar, maka bagian konklusi juga akan bernilai benar. Kaidah bisa terdiri atas beberapa premis dan lebih dari satu konklusi yang dihubungkan “dan” atau “atau”. Pada perancangan sistem untuk mendiagnosa kerusakan komputer, gejala-gejala yang dialami komputer disebut premis dan jenis-jenis kerusakan disebut sebagai konklusi. Sistem pakar yang mendiagnosa kerusakan komputer akan disajikan dalam bentuk *rule-rule* kaidah produksi.

3.3. Mesin Inferensi

Mesin inferensi adalah program komputer yang dibuat untuk memberikan metodologi untuk panalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace* untuk memformulasikan kesimpulan. Untuk membuat mesin inferensi dimulai dengan membuat pohon dari tabel keputusan yang telah ditentukan berdasarkan pengetahuan dari pakarnya. Tabel keputusan jenis kerusakan dan gejala kerusakan komputer ditunjukkan pada Tabel 1.

Tabel 1. Jenis dan Gejala Kerusakan Komputer

No	Gejala	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
1	Mouse tidak memberikan respon	X	X	X	X						
2	Hasil pemeriksaan tipe mouse benar			X	X						
3	Hasil pemeriksaan tipe mouse salah	X									
4	Mouse memberikan respon setelah dibersihkan		X			X					
5	Mouse tidak memberikan respon setelah dibersihkan			X	X			X			
6	Hasil pemeriksaan port tidak ada kesalahan dan mouse memberikan respon			X							
7	Hasil pemeriksaan port tidak ada kesalahan dan mouse tidak memberikan respon				X						
8	Mouse merespon setelah dilakukan pemasangan kembali dengan benar				X						
9	Mouse berespon ketika dipasang di computer lain							X			
10	Mouse tidak berespon ketika di pasang di komputer lain				X						
11	Mouse macet atau gerakannya erratic					X	X	X			
12	Pemeriksaan plug telah kencang				X	X		X			
13	Pemeriksaan plug kendur						X				
14	Komputer bisa di boot normal tapi kadang- kadang macet									X	X
15	Komputer bisa di boot normal tetapi kadang- kadang me reset sendiri								X		
16	komputer bisa di boot tidak me reset sendiri										
17	Hasil pemeriksaan chip memory tidak ditemukan memory yang buruk										

18	Komputer tidak bekerja walaupun sudah diganti pencatu daya										X	X
19	Komputer bekerja setelah diganti <i>motherboard</i>											X
20	Ada <i>Chip RAM</i> yang gagal tes								X			
21	Tidak ada <i>Chip RAM</i> yang gagal tes									X		

No	Gejala	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
22	Komputer bekerja normal setelah memasang UPS										
23	Komputer tidak bekerja walaupun setelah dipasang UPS										X
24	<i>Undefined</i> / tidak ada kerusakan awal										
25	Masalah IWF								X	X	X

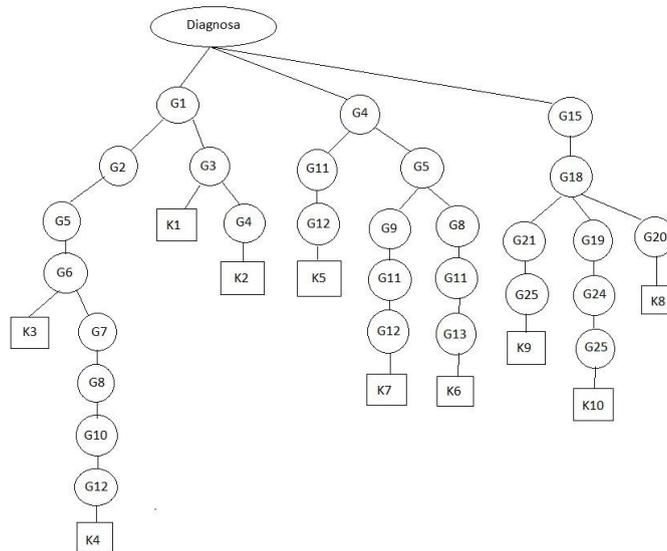
Keterangan :

- K1 : tipe *mouse* salah
- K2 : *mouse* kotor
- K3 : ada masalah pada *port serial*
- K4 : *mouse* rusak
- K5 : *mouse* kotor
- K6 : *plug* kendur
- K7 : *socket* PC rusak
- K8 : memori rusak
- K9 : perangkat lunak tidak lengkap atau BIOS yang kadaluwarsa
- K10 : *motherboard* rusak

Tabel Jenis kerusakan komputer dan gejala merupakan tabel yang digunakan untuk menentukan kerusakan yang dialami oleh komputer dimana pada tabel ini terdapat 25 gejala dan 10 jenis kerusakan komputer, setiap gejala akan mengidentifikasi kerusakan yang dialami komputer. Terdapat gejala umum yang dapat diidentifikasi ke berbagai jenis kerusakan perangkat komputer tetapi terdapat juga gejala khusus yang merupakan ciri dari kerusakan tertentu, seperti halnya *mouse* tidak memberikan respon gejala ini muncul pada setiap Kerusakan yang terjadi pada *mouse*.

3.4. Pohon Keputusan

Pohon keputusan merupakan Gambaran mengenai tabel jenis kerusakan dan gejala kerusakan komputer, dimana keputusan ini memberikan keterangan mengenai gejala yang dialami oleh komputer. G1, G4 dan G15 merupakan gejala umum yang ada pada setiap kerusakan komputer, dari gejala awal kemudian turun ke gejala khusus dimana hanya kerusakan-kerusakan tertentu yang mengalami gejala tersebut, hubungan jenis kerusakan komputer dan gejalanya ditunjukkan oleh tabel hubungan kerusakan komputer dengan gejala-gejalanya, seperti terlihat pada Gambar 4.



Gambar 4. *Pohon Keputusan Kerusakan Komputer*

Berikut adalah hubungan kerusakan komputer dengan gejala dapat dilihat pada Tabel 2:

Tabel 2. *Hubungan Kerusakan Komputer Dengan Gejala*

Jenis kerusakan	Gejala	Keterangan
Type mouse salah	1,2,3	K1
Mouse kotor	1,2,3,4	K2
Ada masalah pada port serial	1,2,5,6	K3
Mouse rusak	1,2,5,6,7,8,10,12	K4
Mouse kotor	4,11,12	K5
Plug kendor	4,5,8,11,13	K6
Socket PC rusak	4,5,9,11,12	K7
Memori rusak	15,18,20	K8
Perangkat lunak tidak lengkap atau BIOS yang kadaluwarsa	15,18,21,25	K9
Motherboard rusak	15,18,19,24,25	K10

3.5. Logika Deduktif

Logika deduktif dari sistem pakar untuk mendiagnosis kerusakan komputer ini diambil dari premis yang bersifat umum. Pengambilan kesimpulannya dengan menggunakan satu premis dan beberapa premis.

- 1) *Rule A (Tipe Mouse Salah)*
 Pada aturan A atau *rule A* berdasarkan tabel 3.1, dapat ditentukan aturan yang digunakan untuk mengetahui tipe mouse salah .
 JIKA *mouse* tidak memberikan respon
 DAN hasil pemeriksaan tipe *mouse* salah
- 2) *Rule B (Mouse Kotor)*
 JIKA *mouse* tidak memberikan respon
 DAN *mouse* memberikan respon setelah dibersihkan
- 3) *Rule C (Ada Masalah Pada Port Serial)* JIKA *mouse* tidak memberikan respon
 DAN hasil pemeriksaan tipe *mouse* benar
 DAN *mouse* tidak memberikan respon setelah dibersihkan
 DAN hasil pemeriksaan *port* tidak ada konflik

- 4) *Rule D (Mouse Rusak)*
JIKA *mouse* tidak memberikan respon
DAN hasil pemeriksaan tipe *mouse* benar
DAN *mouse* tidak memberikan respon setelah dibersihkan
DAN hasil pemeriksaan *port* tidak ada konflik
DAN pemeriksaan *plug* kencang
- 5) *Rule E (Mouse Kotor)*
JIKA *mouse* memberikan respon setelah dibersihkan
DAN *mouse* macet
DAN pemeriksaan *plug* kencang
- 6) *Rule F (Plug Kendor)* JIKA *mouse* macet
DAN pemeriksaan *plug* kendor
- 7) *Rule G (Socket PC Rusak)*
JIKA *mouse* tidak memberikan respon setelah dibersihkan DAN *mouse* berespon
ketika dicoba pada Komputer lain
DAN *mouse* macet
DAN pemeriksaan *plug* kencang
- 8) *Rule H (Memori Rusak)*
JIKA *Intermittent Weird Failure*
DAN komputer bisa di *boot* normal tetapi kadang-kadang *mereset*
sendiri
DAN terdapat *chip RAM* yang gagal test
- 9) *Rule I (Perangkat Lunak yang Tidak lengkap atau BIOS Kadaluwarsa)* JIKA
Intermittent Weird Failure
DAN komputer bisa *diboot* normal tetapi kadang-kadang macet
DAN tidak ada *chip RAM* yang gagal test
DAN komputer tidak berkerja walaupun anda telah mengganti pencatu daya
- 10) *Rule J (Motherboard Rusak)* JIKA *Intermittent Weird Failure*
DAN Komputer bisa *diboot* normal tetapi kadang-kadang *mereset* sendiri
DAN Komputer tidak berkerja walaupun anda telah mengganti pencatu daya
DAN Komputer tidak berkerja walaupun telah dipasang UPS
DAN Komputer berkerja setelah anda mengganti *motherboard* yang baru

3.6. Pelacakan Ke Depan (*Forward Chaining*)

Dalam pendekatan ini pelacakan dimulai dari informasi yang diberikan oleh *user* selanjutnya sistem pakar akan mengGambarkan kesimpulannya. Pelacakan kedepan ini akan mencari fakta yang sesuai dengan bagian aturan *IF- THEN*. Penalaran dimulai dari kesimpulan data menuju pada kesimpulan. Sistem pakar akan mengevaluasi dari kondisi yang ada. Jika kondisi benar, maka aturan dilanjutkan dan aturan berikutnya diuji. Jika kondisi salah, maka aturan tidak diteruskan dan aturan berikutnya diuji.

3.7. Cara Kerja Sistem Pakar

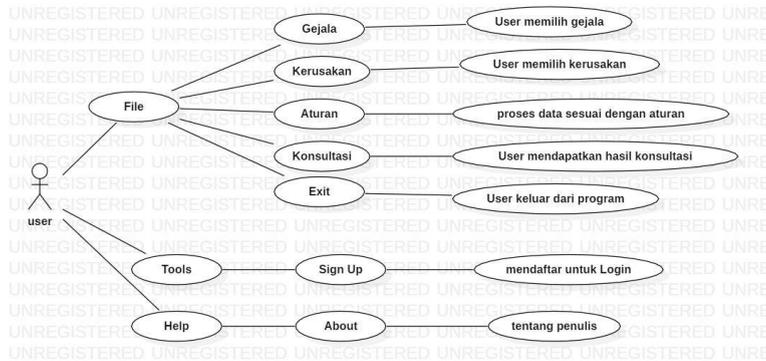
Cara kerja sistem untuk mendiagnosa kerusakan komputer ini adalah pengguna/*user* memilih gejala yang diberikan mengenai gejala yang dialami. Penulis menyiapkan 25 gejala, kemudian berdasarkan gejala tersebut akan disimpulkan jenis kerusakan apa yang dialami dengan mengikuti *rule* yang ada.

3.8. Perancangan UML

Perancangan sistem aplikasi ini dibuat dalam empat bentuk model proses yaitu, *Use Case Diagram*, *Sequence Diagram* dan *Class Diagram*. Perancangan model ini dibuat untuk menjelaskan interaksi antara *user* dan sistem aplikasi.

3.9. Use Case Diagram

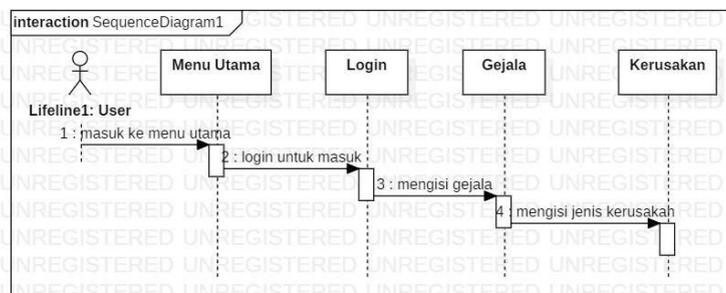
Proses pengembangan aplikasi ini, melibatkan interaksi antara pengguna dan sistem. Pada *Use Case Diagram* ini menggambarkan interaksi pemain antara sistem ada 3 *case* utama yaitu *File*, *Tools* dan *Help*. Pada *Use case File* terdapat *form* konsultasi pengguna akan diminta memasukkan gejala yang dialami, kemudian sistem akan menampilkan diagnosa kerusakan kepada user. *Use case Help* terdapat *menu about* yang berisi profil penulis dan dosen pembimbing yang membimbing dalam pembuatan penulisan ini. *Use Case Tools* terdapat *menu Sign Up* dan *Login* untuk masuk ke dalam aplikasi. *Use Case diagram* aplikasi sistem pakar untuk mendiagnosa kerusakan Komputer ditunjukkan oleh Gambar 5.



Gambar 5. Use Case Diagram User

3.9. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek dengan menekankan urutan waktu penyampaian pesan. Pada *sequence diagram* aplikasi sistem ini, penulis memisahkan interaksi diagram antara objek pada *menu* konsultasi dengan *menu* lainnya. Pada *sequence diagram* interaksi pengguna atau aktor dengan objek konsultasi adalah pengguna mengisi gejala yang dirasakan pada objek konsultasi. Kemudian sistem akan membaca *rule* dari gejala yang diberikan pengguna, sistem pakar memberikan kesimpulan dari kerusakan yang dialami dan memberikan diagnosa kerusakan yang dialami komputer. Penulis juga memisahkan penggambaran *sequence diagram* pengguna dengan admin. *Sequence diagram* aplikasi sistem pakar untuk mendiagnosa kerusakan komputer ditunjukkan pada Gambar 6 dan 7



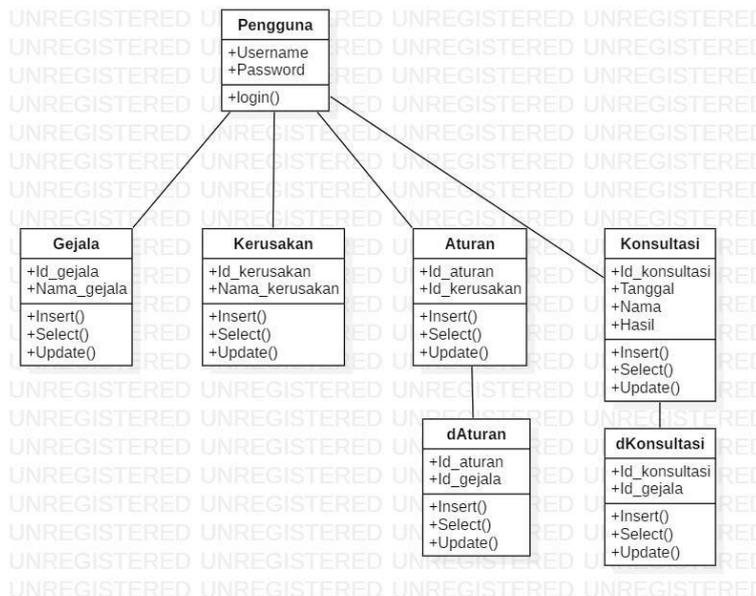
Gambar 6. Sequence Diagram User



Gambar 7. Sequence Diagram User dengan Menu Konsultasi

3.10. Class Diagram

Class diagram dapat membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang sering digunakan. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model suatu design dari suatu sistem. *Class diagram* sistem pakar mendiagnosa kerusakan komputer ditunjukkan oleh Gambar 8



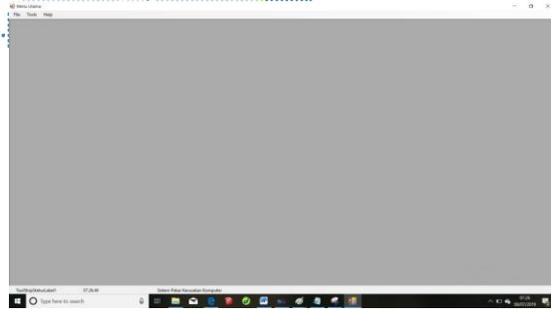
Gambar 8. Class Diagram

3.11. Implementation

Tahap berikutnya adalah menerapkan sistem dan perancangan kedalam kode dengan menggunakan VB.Net dan Microsoft Access. Hasil dari penerapan sistem berupa menu yang terbagi dari File, Tools dan Help. Pada menu File berisikan Form pengguna, aturan, kerusakan, gejala dan konsultasi. Pada menu Tools berisikan form sign up untuk mendaftar terlebih dahulu agar bisa melakukan login. Pada menu Help berisikan about.

a) Tampilan Halaman Menu Utama

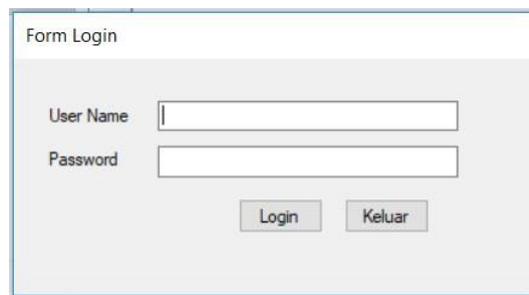
Menu utama berisi *form login* yang digunakan untuk masuk kedalam aplikasi. Menu utama merupakan tampilan pertama yang akan muncul ketika seorang *user* menggunakan aplikasi ini, terlihat pada Gambar 9.



Gambar 9. *Tampilan Halaman Menu Utama*

b) Tampilan Halaman Login

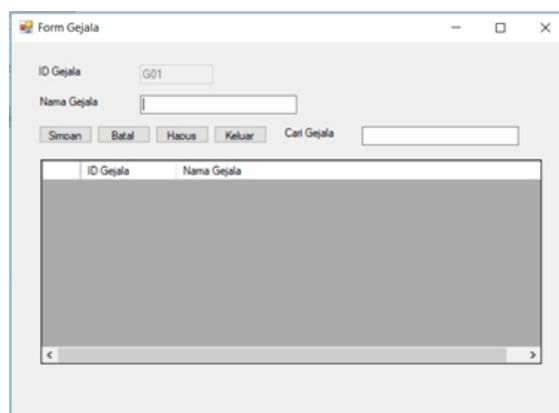
Tampilan login adalah tampilan dimana admin dapat mengubah sistem pakar jika akan digunakan admin perlu memasukkan username dan password dengan benar sesuai dengan username dan password yang tersimpan pada database sistem, terlihat pada Gambar 10.



Gambar 10. *Tampilan Halaman Login*

c) Tampilan Halaman Gejala

Tampilan gejala menampilkan data mengenai gejala yang dialami oleh komputer. Halaman ini diatur oleh *admin*, dimana admin dapat menampilkan data gejala, menambahkan, mengedit dan menghapus data gejala, terlihat pada Gambar 11.

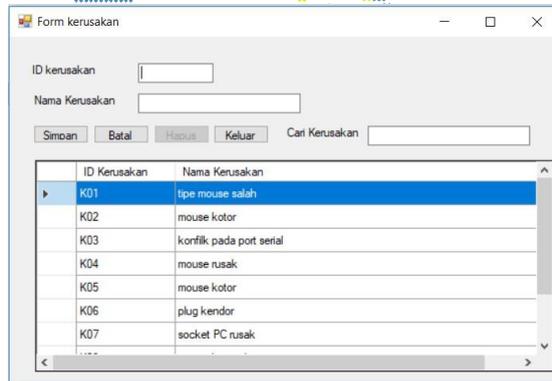


Gambar 11. *Tampilan Halaman Gejala*

d) Tampilan Halaman Kerusakan

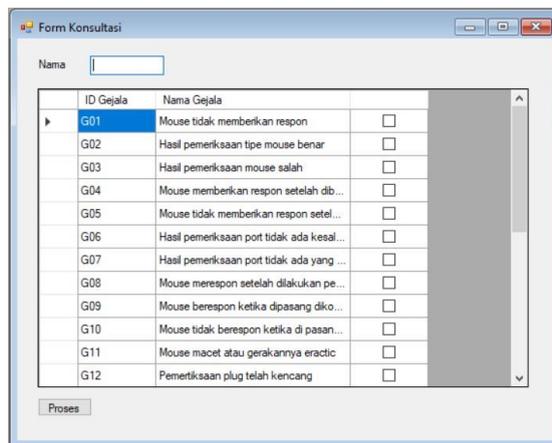
Tampilan kerusakan menampilkan data mengenai jenis kerusakan yang dialami oleh komputer. Halaman ini diatur oleh *admin*, dimana admin dapat menampilkan data

kerusakan, menambahkan, mengedit dan menghapus data kerusakan, terlihat pada Gambar 12.



Gambar 12. Tampilan Halaman Kerusakan

- e) Tampilan Halaman Konsultasi
 Tampilan konsultasi menampilkan data mengenai konsultasi yang dilakukan oleh user. Halaman ini diatur oleh admin, dimana admin dapat menampilkan data kerusakan dan gejala serta memilih gejala yang dialami oleh komputer, terlihat Gambar 13.



Gambar 13. Tampilan Halaman Konsultasi

- f) Tampilan Halaman Informasi Kerusakan
 Tampilan informasi kerusakan menampilkan data yang berisikan cara penanganan kerusakan yang ada pada komputer . berikut ini adalah tampilan informasi kerusakan komputer, terlihat pada Gambar 14.



Gambar 14. Tampilan Halaman Informasi Kerusakan

3.12. Testing

Black-box testing, juga dikenal sebagai pengujian kotak hitam, adalah suatu metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memperhatikan struktur internal atau logika dari sistem yang diuji. Dalam pendekatan ini, pengujian fokus pada input dan output yang dihasilkan oleh perangkat lunak tanpa pengetahuan terperinci tentang bagaimana proses internalnya bekerja. Analisis atau pengujian hanya memiliki akses ke spesifikasi fungsional perangkat lunak, dan tujuan utama adalah memastikan bahwa perangkat lunak berperilaku sesuai dengan yang diharapkan. Tabel 2 menunjukkan hasil pengujian dari aplikasi yang dibuat.

Tabel 3. Hasil Uji Coba Black Box

Butir Uji	Kelas Uji	Kondisi Awal	Skenario Uji	Hasil yang diharapkan	Hasil Uji	
Pengujian Fungsi user	a. Fungsi Menu Utama	Form Menu Utama	Klik Menu Utama	Membuka Menu Utama	Berhasil	
	b. Fungsi Konsultasi	Form Konsultasi	Klik Menu Konsultasi	Membuka Form Konsultasi	Berhasil	
	c. Fungsi About	Form About	Klik Menu About	Membuka Form About	Berhasil	
Pengujian Fungsi Admin	a. Fungsi Menu Utama	Form Menu Utama	Klik Menu Utama	Membuka Menu Utama	Berhasil	
	b. Fungsi Login	Form Login	Masukan Username dan Password	Membuka Menu Utama Admin	Berhasil	
	c. Fungsi Form Kerusakan			klik menu kerusakan	menampilkan data kerusakan	Berhasil
				Klik Tambah	Data kerusakan yang ditambah disimpan kedalam <i>database</i> .	Berhasil
				Klik Simpan	Menyimpan data kerusakan ke dalam <i>database</i> .	Berhasil
				Klik Hapus	Menghapus data yang telah disimpan ke <i>database</i> .	Berhasil
				Klik keluar	Keluar dari Form kerusakan	Berhasil
			Klik cari	Mencari data kerusakan yang telah disimpan ke <i>database</i> .	Berhasil	
d. Fungsi Form Gejala	Form gejala		Klik menu gejala	Menampilkan data gejala	Berhasil	
			Klik Tambah	Data gejala yang ditambah disimpan kedalam <i>database</i> .	Berhasil	
			Klik Simpan	Menyimpan data gejala ke dalam <i>database</i> .	Berhasil	
			Klik Hapus	Menghapus data yang telah disimpan ke <i>database</i> .	Berhasil	
			Klik keluar	Keluar dari Form gejala.	Berhasil	
			Klik cari	Mencari data kerusakan yang telah disimpan ke <i>database</i> .	Berhasil	
e. Fungsi Form Konsultasi	Form konsultasi	Klik menu konsultasi	Menampilkan data konsultasi	Berhasil		

Dari tabel hasil uji di, terlihat bahwa aplikasi berjalan dengan baik sesuai dengan hasil yang diharapkan.

4. Kesimpulan

Dalam perancangan aplikasi diagnosa kerusakan komputer menggunakan Sistem Pakar dengan metode Forward Chaining, penelitian ini memiliki tujuan utama untuk memberikan solusi efisien dan akurat dalam mendeteksi serta mengatasi masalah yang mungkin timbul pada perangkat komputer. Melalui penerapan kecerdasan buatan dan pendekatan sistem pakar, aplikasi ini dirancang untuk menghadirkan pengalaman pengguna yang lebih baik dalam mengelola permasalahan teknis pada komputer mereka. Metode Forward Chaining yang digunakan dalam aplikasi ini memungkinkan sistem untuk menyusun langkah-langkah diagnosis secara sistematis berdasarkan informasi yang diberikan oleh pengguna. Proses ini memastikan bahwa solusi yang dihasilkan sesuai dengan gejala yang dilaporkan, meningkatkan efisiensi dalam penanganan masalah. Integrasi pengetahuan yang mendalam tentang kerusakan komputer dalam sistem pakar

menjadi landasan untuk keputusan otomatis, mempercepat waktu respons dan mengurangi ketergantungan pada pengetahuan teknis pengguna. Dengan demikian, perancangan aplikasi ini diharapkan dapat membantu pengguna, terutama yang tidak memiliki pengetahuan teknis yang mendalam, dalam mendeteksi, mendiagnosis, dan mengatasi kerusakan komputer dengan solusi yang tepat dan terukur. Implementasi aplikasi ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan produktivitas pengguna, mengoptimalkan waktu perbaikan, dan mengurangi dampak negatif dari gangguan teknis pada kinerja komputer.

Dalam merumuskan saran untuk perancangan aplikasi diagnosa kerusakan komputer menggunakan Sistem Pakar dengan metode Forward Chaining, beberapa aspek penting perlu diperhatikan. Pertama, disarankan untuk mengembangkan antarmuka pengguna yang intuitif dan interaktif guna memastikan pengguna dapat dengan mudah berinteraksi dengan sistem pakar. Selanjutnya, perlu dilakukan pengoptimalan algoritma Forward Chaining agar aplikasi dapat menangani dataset besar dengan efisien. Peningkatan terus-menerus pada basis pengetahuan aplikasi, dengan memasukkan informasi terkini tentang perkembangan teknologi dan kerusakan komputer, juga sangat disarankan. Integrasi dengan sumber daya eksternal seperti basis data pengetahuan online atau komunitas teknologi dapat memperluas cakupan pengetahuan aplikasi. Untuk menjaga keamanan dan privasi, perlunya perhatian khusus terhadap aspek tersebut dalam pengembangan aplikasi. Selain itu, implementasi modul pelatihan dan mekanisme umpan balik pengguna dapat meningkatkan pemahaman dan penerimaan pengguna terhadap sistem pakar. Kolaborasi dengan ahli teknis dalam pengembangan dan evaluasi aplikasi juga menjadi sarana penting untuk memastikan keakuratan dan keandalan diagnosa yang dihasilkan. Dengan mempertimbangkan saran-saran ini, aplikasi diagnosa kerusakan komputer dapat lebih optimal dan responsif terhadap kebutuhan pengguna.

Daftar Pustaka

- [1] B. H. Hayadi, *Sistem Pakar*. Deepublish, 2018.
- [2] H. Listiyono, "Merancang Dan Membuat Sistem Pakar," *Dinamik*, Vol. 13, No. 2, Art. No. 2, 2008, Doi: 10.35315/Dinamik.V13i2.76.
- [3] R. Rosnelly And U. P. Utama, *Sistem Pakar: Konsep Dan Teori*. Penerbit Andi.
- [4] P. S. R. M.Kom And U. F. S. P. M.Kom, *Mengenal Metode Sistem Pakar*. Uwais Inspirasi Indonesia.
- [5] M. Ipan And T. Oktarina, "Application Of The Iterative Model In Designing An Academic E-Counseling System At Bina Darma University," *J. Tek. Inform. Jutif*, Vol. 4, No. 1, Art. No. 1, Feb. 2023, Doi: 10.52436/1.Jutif.2023.4.1.838.
- [6] N. Dwivedi, D. Katiyar, And G. Goel, "A Comparative Study Of Various Software Development Life Cycle (Sdlc) Models," *Int. J. Res. Eng. Sci. Manag.*, Vol. 5, No. 3, Art. No. 3, Mar. 2022.
- [7] K. Rokoyah, Y. I. Chandra, And S. Lukman, "Penerapan Model Incremental Dalam Merancang Aplikasi Pengenalan Bentuk Dan Fungsi Gigi Pada Manusia Berbasis Web," *J. Sikomtek*, Vol. 12, No. 2, Art. No. 2, Jul. 2022.
- [8] N. P. Ayudhana, L. Andrawina, And A. Musnansyah, "Perancangan Dan Pembangunan Knowledge Management System Pada Modul Pengabdian Masyarakat Dan Penunjang Menggunakan Framework Codeigniter Dengan Metode Iterative Incremental," *Eproceedings Eng.*, Vol. 2, No. 2, Art. No. 2, Aug. 2015, Doi: 10.34818/Eoe.V2i2.2192.
- [9] P. Abrahamsson, N. Oza, And M. T. Siponen, "Agile Software Development Methods: A Comparative Review1," In *Agile Software Development: Current Research And Future Directions*, T. Dingsøyr, T. Dybå, And N. B. Moe, Eds., Berlin, Heidelberg: Springer, 2010, Pp. 31–59. Doi: 10.1007/978-3-642-12575-1_3.

- [10] M. Riesener, C. Doelle, S. Perau, P. Lossie, And G. Schuh, “Methodology For Iterative System Modeling In Agile Product Development,” *Procedia Cirp*, Vol. 100, Pp. 439–444, Jan. 2021, Doi: 10.1016/J.Procir.2021.05.101.
- [11] J. Lin, K. H. Chai, Y. S. Wong, And A. C. Brombacher, “A Dynamic Model For Managing Overlapped Iterative Product Development,” *Eur. J. Oper. Res.*, Vol. 185, No. 1, Pp. 378–392, Feb. 2008, Doi: 10.1016/J.Ejor.2006.12.022.
- [12] B. Blocken And C. Gualtieri, “Ten Iterative Steps For Model Development And Evaluation Applied To Computational Fluid Dynamics For Environmental Fluid Mechanics,” *Environ. Model. Softw.*, Vol. 33, Pp. 1–22, Jul. 2012, Doi: 10.1016/J.Envsoft.2012.02.001.