

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi yang berkembang begitu pesat menuntut semua pihak yang aktivitas hidupnya terikat dengan perangkat tersebut untuk segera menyesuaikan diri baik dalam kebutuhan hardware maupun skill. Hal tersebut bukanlah mengikuti tren semata, namun lebih berorientasi pada penjunjung kelancaran kerja ataupun aktivitas yang dilakukan sehari-hari, tentu saja dengan adanya teknologi baru baik hardware maupun software akan sangat mendukung terhadap kecepatan dan kekuatan hasil kinerja di segala bidang.

Komputer merupakan salah satu hasil produk teknologi yang banyak dipakai dalam menunjang aktivitas kerja sebuah lembaga maupun perorangan. Karena dengan kebutuhan perangkat tersebut mampu mempercepat dan kesempurnaan aktivitas kerja. Namun di sisi lain, tingginya ketergantungan aktivitas kerja pada komputer akan berdampak secara langsung pada kemampuan kinerja perangkat itu sendiri.

Berdasarkan data dari Kompas.com, jumlah pengguna internet tahun 2019 telah mencapai 171,17 juta jiwa dari 264 juta jiwa penduduk Indonesia . Meningkatnya pengguna internet, otomatis penggunaan perangkat komputer juga meningkat dan menurut Widiastono, A., Banyal, N. A., & Angriani, L. (2016), banyaknya pengguna perangkat tersebut yang tidak mampu menangani beberapa masalah terkait dengan kerusakan sistem software terutama sistem operasi yang digunakan, rata-rata pengguna hanya dapat menggunakan perangkat tersebut namun akan kesulitan saat terjadi kerusakan dan rata-rata pengguna akan membawa perangkat tersebut ke toko komputer untuk memperbaikinya.

Berdasarkan wawancara dengan teknisi komputer Berkah Com yaitu meningkatnya pengguna komputer untuk memperbaiki kerusakan ke toko komputer, otomatis akan menambah pekerjaan teknisi komputer menjadi semakin banyak, dengan meningkatnya jumlah pekerjaan teknisi sehingga membutuhkan waktu yang cukup lama untuk menyelesaikan pekerjaannya, untuk membantu hal tersebut maka dibutuhkan media yang dapat membantu teknisi komputer dalam menyelesaikan pekerjaannya.

Saragih Sehatman, A., Cristina Sherly, & Elsyawina Tiara (2018), menyatakan kemampuan seorang teknisi yang dapat mendeteksi kerusakan komputer dapat diimplementasikan ke dalam sebuah sistem aplikasi. Sistem aplikasi yang dapat bekerja sebagaimana manusia atau seorang pakar bekerja dikenal dengan sistem pakar, sistem tersebut dapat menampung kemampuan atau keahlian seorang pakar agar proses menganalisa suatu masalah dapat dikerjakan secara otomatis oleh aplikasi yang dirancang dan dapat mewakili pakarnya dalam menganalisa suatu permasalahan. Hal ini memungkinkan teknisi melakukan penanganan kerusakan komputer dengan efektif dan efisien.

Widyawati, A. C. (2018), menyimpulkan dalam salah satu metode dalam sistem pakar yaitu metode Backward Chaining, metode ini melakukan proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Menurut penelitian terdahulu Minarni, & Hidayat, R. (2013), metode backward chaining dapat menyelesaikan masalah terkait kerusakan komputer .

Berdasarkan uraian tersebut maka diusulkan pembuatan **“SISTEM PAKAR PENENTU KERUSAKAN SOFTWARE KOMPUTER BERBASIS WINDOWS MENGGUNAKAN BACKWARD CHAINING”**, sehingga dapat membantu teknisi komputer dalam menyelesaikan pekerjaannya.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah diatas, dapat di identifikasikan beberapa masalah yang akan dijadikan bahan penelitian selanjutnya.

1. Pada saat teknisi komputer melakukan diagnosa membutuhkan waktu yang lebih lama untuk mendapatkan solusi.
2. Pengguna komputer membutuhkan waktu yang lebih cepat untuk mendapatkan hasil dari kerusakan komputer.

1.3 Rumusan Masalah

1. Bagaimana membuat suatu sistem pakar yang dapat mendiagnosa kerusakan komputer berbasis windows 10 ?
2. Bagaimana mengimplementasikan sistem ini sebagai upaya untuk membantu teknisi komputer dalam mendiagnosa dan menemukan solusi kerusakan komputer berbasis windows 10 ?

1.4. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah :

1. Penelitian ini dilakukan untuk mendiagnosa kerusakan *Software Operating System Windows 10*.
2. Tempat pada penelitian ini adalah Toko Komputer Berkah Com.
3. Penelitian ini menggunakan metode *Backward Chaining* untuk penelusuran kerusakan.
4. Penelitian ini menggunakan variable yaitu jenis kerusakan, penyebab kerusakan dan solusi.

1.5 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Membuat sistem yang bertujuan untuk membantu teknisi komputer dalam menyelesaikan pekerjaannya berbasis website.
2. Mempercepat pengguna komputer dalam mendapatkan hasil dari kerusakan komputer yang diperbaiki oleh teknisi komputer.

3. Membantu teknisi komputer dalam menyelesaikan pekerjaannya dengan lebih cepat.

1.6 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Menghasilkan sistem berbasis website yang dapat membantu teknisi komputer dalam menyelesaikan pekerjaannya.
2. Memudahkan pengguna komputer untuk mendapatkan hasil kerusakan dengan lebih cepat.
3. Memudahkan teknisi komputer dalam menyelesaikan pekerjaannya dengan lebih cepat.

1.7 Sistematika Penulisan

Uraian singkat mengenai sistematika penulisan pada masing-masing bab adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, ruang lingkup penelitian, tujuan dan manfaat penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini berisi tentang teori-teori yang mendukung penelitian yang akan dilakukan oleh penulis/peneliti.

BAB III : METODE PENELITIAN

Bab ini berisi tentang metode-metode pendekatan penyelesaian permasalahan yang dinyatakan dalam perumusan masalah pada penelitian yang dilakukan.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi tentang pemaparan hasil analisa persoalan yang dibahas dengan berpedoman pada teori-teori yang dikemukakan pada Bab II.

BAB V : SIMPULAN DAN SARAN

Bab ini merupakan penutup yang berisi kesimpulan serta saran yang dapat membantu pengembangan aplikasi ini di masa yang akan datang.

BAB II

LANDASAN TEORI

2.1 Software

Pressman (2012) menyatakan, “Perangkat Lunak (*Software*) adalah instruksi–instruksi (program komputer) yang ketika di jalankan menyediakan fitur- fitur, fungsi-fungsi dan kinerja-kinerja yang di kehendaki.”

Sukamto, et.al (2014),menyatakan “Perangkat lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).”

Ladjamudin (2013) menyatakan, “*Software* merupakan kumpulan dari perintah atau fungsi yang di tulis dengan dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.”

2.2 Sistem Operasi

Haryanto (2012) berpendapat bahwa Sistem operasi komputer adalah perangkat lunak komputer atau *software* yang bertugas untuk melakukan *control* dan manajemen perangkat keras dan juga operasi-operasi dasar sistem, termasuk menjalankan *software* aplikasi seperti program-program pengolah data yang bisa digunakan untuk mempermudah kegiatan manusia.

2.3 Windows 10

Sistem operasi terbaru sekaligus keluaran terakhir perusahaan *Microsoft* ini dirilis pada 29 Juli 2015 untuk mengatasi kekurangan pada sistem operasi sebelumnya dengan menambahkan berbagai fitur termasuk tampilan menu *start* yang ada pada *Windows 7*. Gayatrie, M. S., Kusyanti, A., & Saputra, M. C (2017) ,berpendapat bahwa sistem operasi ini juga memiliki fitur untuk melakukan pembaharuan terhadap sistem yang dilakukan secara otomatis dengan tampilan yang lebih baik dan *modern*.

Beberapa fitur yang ada pada *Windows 10* menurut Gayatrie, M. S., Kusyanti, A., & Saputra, M. C (2017) yaitu:

a. Windows Hello

Fitur ini menggantikan fungsi *password* dengan deteksi wajah atau sidik jari dengan otentikasi lebih aman yang mendukung seumur hidup.

b. Microsoft edge

Microsoft edge adalah *browser* yang dibangun *modern* dan lebih bersifat pribadi serta responsif.

c. Windows Ink

Fitur ini memungkinkan ide-ide didalam pikiran dituliskan dengan pena khusus untuk kemudian ditangkap kedalam gambar dan diolah.

d. Cortana

Fitur ini layaknya seperti asisten pribadi yang memungkinkan pengguna dapat menyelesaikan tugas, mengatur pengingat dan bekerja dengan perangkat secara lebih produktif.

e. Xbox Play Anywhere

Fitur ini memungkinkan pengguna terhubung dengan komunitas game dimanapun untuk bermain dengan performa yang baik.

f. Continuum

Fitur ini memungkinkan perangkat untuk dapat digunakan kedalam mode 2-in-1 yang berjalan sesuai mode yang diinginkan.

2.4 Sistem Pakar

Sistem pakar menurut Widyawati, A. C. (2018) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya.

Sistem pakar ini juga akan dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan. Dalam

penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu.

2.4.1 Struktur Sistem Pakar

Triwahyudi (2013) berpendapat bahwa sistem pakar memiliki dua bagian utama, yaitu :

- a. Lingkungan pengembangan (*development environment*), yaitu bagian yang digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar.
- b. Lingkungan konsultasi (*consultation environment*), yaitu bagian yang digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan. Sistem pakar memiliki 3 bagian, yaitu antar muka pengguna, mesin inferensi dan basis pengetahuan.

1. Antar Muka Pengguna

Antar muka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Antar muka pengguna memberikan fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sehingga ditemukan sebuah solusi.

2. Mesin Inferensi

Mesin Inferensi (*Inference Engine*), merupakan otak dari Sistem Pakar, juga dikenal sebagai penerjemah aturan (*rule interpreter*). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan. Kerja mesin inferensi meliputi:

- Menentukan aturan mana akan dipakai
 - Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
 - Menambahkan jawaban ke dalam memori Sistem Pakar.
 - Menyimpulkan fakta baru dari sebuah aturan.
 - Menambahkan fakta ke dalam memori.
3. Basis Pengetahuan Basis pengetahuan merupakan inti program sistem pakar. Pengetahuan ini merupakan representasi pengetahuan dari seorang pakar.

2.4.2 Keuntungan dan Kelemahan Sistem Pakar

Keuntungan pada sistem pakar adalah sebagai berikut :

- a. Memungkinkan orang awam dapat mengerjakan pekerjaan para ahli.
- b. Dapat melakukan proses secara berulang secara otomatis.
- c. Menyimpan pengetahuan dan keahlian para pakar.
- d. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
- e. Meningkatkan hasil dan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia.
- f. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
- g. Memiliki kemampuan untuk mengakses pengetahuan.
- h. Meningkatkan kapabilitas dalam penyelesaian masalah.
- i. Menghemat waktu dalam pengambilan keputusan.

Kelemahan pada sistem pakar adalah sebagai berikut :

- a. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkannya sangat mahal.
- b. Sulit dikembangkan, hal ini erat kaitannya dengan ketersediaan pakar di bidangnya.
- c. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.

- d. Kepakaran sangat sulit diekstrak dari manusia.
- e. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
- f. Sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah.
- g. Sistem pakar bekerja baik untuk suatu bidang yang sempit.
- h. Istilah yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.
- i. Transfer pengetahuan dapat bersifat subyektif dan biasa.

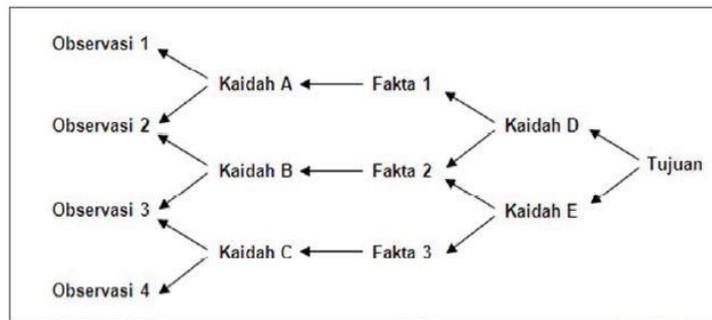
2.5 Backward Chaining

Metode Backward Chaining menurut Widyawati, A. C (2018), sesuai dengan namanya bekerja mundur dari asumsinya. Jika asumsinya q benar maka tak ada yang perlu dikerjakan selanjutnya. Selain itu algoritmanya akan mencari implikasi-implikasi di dalam basis pengetahuan atau *Knowledge Based (KB)* yang kesimpulannya q . Jika premis-premis dari salah satu implikasi tersebut bisa dibuktikan benar, maka q adalah benar.

Ciri- ciri dari Backward Chaining menurut Tarigan A. Feriani (2014), yaitu :

Menggunakan pendekatan *goal-driven*, dimulai dari harapan apa yang akan terjadi (hipotesis) dan kemudian mencari bukti yang mendukung (atau berlawanan) dengan harapan kita. Sering hal ini memerlukan perumusan dan pengujian hipotesis sementara.

Pada metode inferensi dengan backward chaining akan mencari aturan atau *rule* yang memiliki konsekuen (*Then klausa ..*) yang mengarah kepada tujuan yang di skenariokan / di inginkan.



Gambar 2.1 Diagram Backward Chaining

2.6 Tahapan Metode Backward Chaining

Zulfiansyah (2015), menyimpulkan di metode ini, proses dimulai dari kesimpulan. Kemudian kesimpulan dicocokkan dengan mengarah ke konklusi dari suatu aturan. Apabila ada aturan yang mempunyai konklusi yang sama dengan kesimpulan yang dicari, maka dilakukan pencarian fakta dengan cara bertanya ke pengguna untuk memastikan premis aturan dipenuhi.

Contoh: Aturan 1:

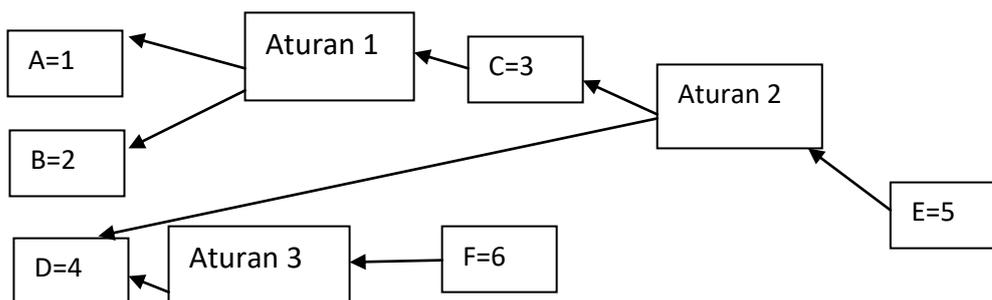
JIKA A=1 dan B=2 MAKA C=3

Aturan 2:

JIKA D=4 dan C=3 MAKA E=5

Aturan 3:

JIKA D=4 MAKA F=6



Gambar 2.2 Contoh Aturan Backward Chaining

Pada Gambar 2.2, contoh goal yang dipilih adalah E=5. Data dimulai dari E=5 merupakan goal atau kesimpulan, pada E=5 diperiksa aturan yang memiliki

kesimpulan tersebut dan ternyata ada aturan yang memiliki konklusi tersebut yaitu Aturan 2.

Pada Aturan 2 terdapat premis $C=3$ dan $D=4$. Untuk $D=4$ dapat langsung dibuktikan kebenarannya dikarenakan tidak ada aturan yang memiliki kesimpulan tersebut sedangkan untuk $C=3$ ada aturan yang memiliki kesimpulan tersebut yaitu Aturan 1.

Pada Aturan 1 terdapat premis $A=1$ dan $B=2$, disebabkan tidak ada aturan pada $A=1$ dan $B=2$ dapat langsung dibuktikan kebenarannya, pembuktian kebenaran umumnya dilakukan dengan mencari fakta dengan cara ditanyakan kepada pengguna, pemeriksaan dimulai dari goal menuju data yang berkaitan dengan aturan yang dijalankan, pada runut balik, sistem hanya menampilkan data yang berkaitan dengan aturan yang dijalankan.

2.7 Metode Pengembangan Perangkat Lunak

2.7.1 Prototype

Pressman (2012) berpendapat bahwa Metode *Prototype* adalah proses pembuatan model sederhana *software* yang memungkinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal.

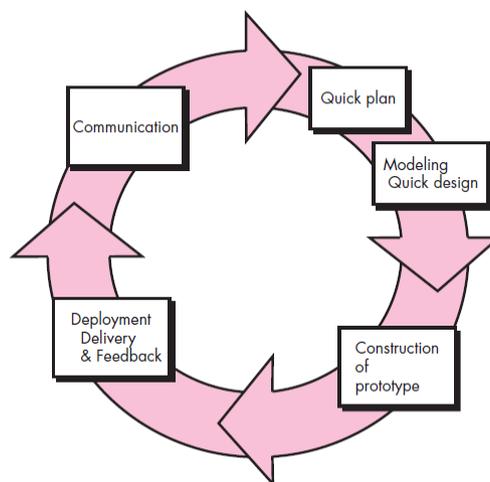
Prototype memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan di buat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali.

Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum di produksi secara benar.

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali, segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik.

Berikut adalah tahapan dalam metode *prototype* :

1. Komunikasi (*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan *user* untuk menentukan kebutuhan.
2. Perencanaan cepat (*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat (*Modeling Quick Design*), yaitu membuat rancangan desain program.
4. Pembentukan *prototype* (*Construction of prototype*), yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat
5. Penyerahan sistem dan umpan balik (*Development Delivery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.3 Diagram Prototype

Penjelasan dari gambar 2.3 diatas adalah sebagai berikut :

Tahap pertama adalah *communication* dan pengumpulan data awal yaitu tahap suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya.

Tahap kedua adalah *quick plan* yaitu analisis terhadap kebutuhan pengguna.

Tahap ketiga adalah *modelling quick design* yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.

Tahap keempat adalah *construction of prototype* adalah pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan.

Tahap kelima adalah *deployment, delivery, and feedback* adalah tahap penyerahan sistem ke pengguna dan umpan balik.

2.8 Pengujian *Black-Box Testing*

Menurut Tamagola & Wintoro (2017), metode pengujian pada aplikasi ini menggunakan *Black Box Testing* yaitu yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.9 Unified Modeling Language (UML)

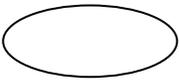
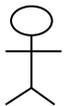
Menurut Rosa A.S dan Salahudin(2015), *unified modeling language* (UML) adalah salah satu *standar* bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat aplikasi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

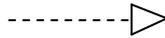
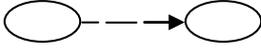
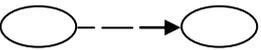
Dengan demikian, penulis dapat mengutarakan bahwa metode *UML* (*Unified Modeling Language*) merupakan sebuah metode atau sebuah bahasa yang digunakan dalam menterjemahkan, menjelaskan, memodelkan, mendefinisikan suatu sistem dengan bentuk simbol-simbol tertentu yang bertujuan untuk memberikan penjelasan-penjelasan detail dari sebuah sistem.

2.9.1 Use Case Diagram

Rosa dan Salahudin (2015), menguraikan *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.1 Simbol *Use Case Diagram*

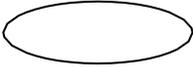
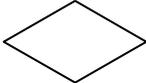
| Simbol | Keterangan |
|--|---|
| <i>Use Case</i>  | Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. |
| Aktor  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| Asosiasi  | Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki |

| | |
|---|---|
| Generalisasi  | Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> . |
| <<Include>>  | <i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang |
| <<Extend>>  | <i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas |

2.9.2 ERD (*Entity Relationship Diagram*)

Menurut Fathansyah (2012), *ERD* adalah diagram yang memperlihatkan entitas-entitas yang terlibat dalam suatu sistem serta hubungan-hubungan atau relasi antar entitas tersebut. *Model Entity-Relationship* yang berisi komponen-komponen himpunan entitas dan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari “Dunia Nyata” yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram *Entity Relationship*.

Tabel 2.2 Simbol *ERD* Diagram

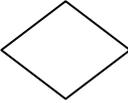
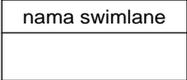
| Simbol | Komponen | Keterangan |
|---|----------|--|
|  | Entitas | Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain |
|  | Atribut | Mendesripsikan karakter entitas. Misalnya atribut nama pekerja dari entitas pekerja. Setiap entitas bisa terdapat lebih dari satu atribut |
|  | Relasi | Menunjukkan hubungan diantara sejumlah entitas yang berbeda. |

| | | |
|---|------------|--|
|  | Penghubung | Menggambarkan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya |
|---|------------|--|

2.9.3 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh *system*.

Tabel 2.3 Simbol *Activity Diagram*

| Simbol | Keterangan |
|--|--|
| Status Awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan  | Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu. |
| Penggabungan  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas. |

2.10 Penelitian Terdahulu

Tabel 2.4 berikut merupakan penelitian terdahulu yang telah dilakukan terkait dengan Sistem Pakar Penentu Kerusakan Software Komputer Berbasis Windows Menggunakan Backward Chaining.

Tabel 2.4 Jurnal Terdahulu

| No | Nama | Judul | Tahun | Analisa |
|----|-------------------------------------|--|-------|---|
| 1 | Apri Candra Widiawati | Diagnosa Penyakit Tanaman Kopi dengan Menggunakan Backward Chaining Pada Kabupaten Tanggamus | 2018 | Pada penelitian ini membahas tentang pembuktian dugaan atau kesimpulan dari penyakit tanaman kopi menggunakan teknik backward chaining dengan menggunakan 5 variable dan 5 rule untuk mendapatkan kesimpulan dari fakta yang terjadi. |
| 2 | Anastasya Latubess dan Ahmad Jazuli | Analisis Model Penelusuran Backward Chaining dalam Mendeteksi Tingkat Kecanduan Game Pada Anak | 2017 | Pada penelitian ini membahas tingkat kecanduan game pada anak menggunakan metode backward chaining dengan menggunakan 6 variable dan 7 rule untuk mendapatkan kesimpulan. |

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

3.1.1 Metode Pengumpulan data

Dalam tahapan ini yang dilakukan dalam pengumpulan data ini meliputi sebagai berikut:

3.1.1.1 Wawancara

Metode wawancara ini dilakukan secara langsung dengan cara mengajukan sejumlah pertanyaan kepada teknisi komputer Berkah Com. Pertanyaan yang diajukan meliputi bagaimana cara diagnosa kerusakan *software windows 10* dan bagaimana cara penyelesaiannya.

3.1.1.2 Observasi

Dalam metode observasi ini penulis diberikan kesempatan untuk melakukan pengumpulan data dengan cara mengamati langsung proses *service* komputer yang dilakukan oleh teknisi komputer Berkah Com.

3.1.1.3 Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur yang bersumber dari jurnal, buku, pakar ataupun dari hasil penelitian orang lain yang berkaitan dengan penelitian yang akan dilakukan.

3.2 Metode Pengembangan Perangkat Lunak

Pada tahapan pengumpulan perangkat lunak, penelitian ini dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu metode pengembangan sistem *Prototype*.

3.2.1 Komunikasi

Komunikasi harus dilakukan dengan yang tepat. Data objektif dan relevan dengan pokok pembahasan menjadi indikator keberhasilan suatu penelitian.

Komunikasi dilakukan dengan cara mengadakan interaksi dengan admin dan user sehingga kebutuhan perangkat lunak dapat terpenuhi.

3.2.2 Perencanaan Cepat

Quick Plan (rencana cepat) merupakan tahapan dengan melakukan analisis dan perencanaan setelah kita mendapatkan data-data dari tahapan komunikasi. Analisa kebutuhan *non fungsional* adalah sebuah langkah dimana seseorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang dibangun. Analisis kebutuhan *non fungsional* tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak agar aplikasi dapat berjalan dengan baik. Analisis *non fungsional* yang dilakukan dibagi dalam tiga tahapan, yaitu :

3.2.2.1 Analisis Kebutuhan Pengguna (*Admin Dan User*)

Aplikasi untuk menentukan konsentrasi skripsi dan rekomendasi bahasa pemrograman ini akan digunakan oleh *user* dan *admin* dengan ketentuan sebagai berikut:

- a. Dapat menggunakan aplikasi yang ada di *system* operasi *Windows*.
- b. Dapat menggunakan media pencarian seperti *Mozilla Firefox*, *Google Chrome*, atau *browser* lain.

3.2.2.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Analisis kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi berbasis web adalah sebagai berikut :

- a. Sistem operasi *Microsoft Windows*.
- b. *Xampp*.
- c. *Browser Internet Mozilla Firefox*, *Google Chrome*, atau *Browser* lain .
- d. *Visual Code*.

3.2.2.3 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Analisis kebutuhan perangkat keras yang digunakan untuk membangun sebuah sistem adalah sebagai berikut :

Spesifikasi minimum untuk PC :

- a. *Processor AMD Ryzen 5 2500U*
- b. *Ram 8 GB.*
- c. *Hardisk 500 GB.*
- d. *Keyboard dan Mouse.*

Analisa tersebut bukanlah hal yang mutlak, namun merupakan pendapat peneliti tentang minimum penggunaan perangkat keras yang dipakai dalam pengembangan aplikasi.

3.3 Analisis Data

Pada tahap ini akan dijelaskan tentang sumber daya yang digunakan dan tahap sebelum pemrosesan data.

3.3.1 Sumber Data

Sumber data yang digunakan dalam penelitian ini diambil dari bagian teknis komputer Berkah Com.

3.3.2 Tahapan Algoritma Backward Chaining.

Pada penelitian ini akan dilakukan implementasi metode backward chaining untuk menentukan kerusakan *software windows 10*.

3.3.3 Penentuan Atribut yang akan digunakan

Pengumpulan data yang di gunakan pada sistem pakar ini didapatkan dari berbagai sumber diantaranya seorang pakar, buku, *internet* dan sumber lainnya yang sesuai. Dari sumber informasi tersebut maka dapat diketahui beberapa jenis kerusakan yang terjadi pada *operating system windows 10* meliputi jenis keusakan, gejala dan solusi untuk mengatasi kerusakan yang

terjadi pada system tersebut. Berikut ini adalah macam-macam kerusakan yang terjadi pada *operating system windows 10* :

a. Error Driver

Gejala :

- a. *Sound* tidak berfungsi.
- b. *Wifi* tidak berfungsi.
- c. Tampilan resolusi layar tidak sesuai VGA.
- d. *Pointer mouse* tidak berfungsi.
- e. *Bluetooth* tidak berfungsi.
- f. *Port LAN* tidak berfungsi.
- g. *USB* tidak berfungsi.

Solusi:

Install ulang driver yang bermasalah, dengan cara masuk ke menu *control panel > system and security > system > device manager*.

Kemudian *update driver* yang bermasalah lalu terdapat dua pilihan yaitu *search driver automatically for driver* dan *browse my computer for driver*.

Driver automatically for driver untuk menginstall *driver* secara otomatis menggunakan koneksi internet, *browse my computer for driver* untuk menginstal *driver* yang sudah ada dalam bentuk *file* yang sudah jadi di dalam *drive* komputer.

b. Blue Screen

Gejala :

- a. *Layar* dekstop menjadi biru dan ada symbol :(.
- b. Aplikasi yang digunakan tidak bisa beroperasi.
- c. Sistem operasi tidak dapat beroperasi.
- d. *Windows* akan otomatis masuk ke *system repair*.

Solusi:

Restart komputer, jika dapat mengakses desktop kemudian masuk ke menu *system and security > change setting* dibagian *computer name, domain* dan *workgroup settings >* setelah itu pilih *tab advance, settings* pada *startup and recovery*, kemudian hilangkan centang pada *automatically restart*.

Jika tidak dapat mengakses desktop akan otomatis masuk ke *system repair*, setelah otomatis masuk ke *system repair* akan muncul menu *choose and option >* pilih *troubleshoot > advance option > system restore*.

Dalam *system restore* pilih *restore point* yang memiliki tanggal sebelum terjadinya *blue screen*, setelah *system restore* lalu *restart* komputer kembali.

c. *Windows explorer not reponding*

Gejala :

- a. *Loading* lama saat proses membuka *drive*
- b. Gagal dalam proses *cut, copy, paste* dan *delete file*
- c. *Windows explorer* otomatis *end proses*

Solusi:

Buka *task manager* dengan cara tekan tombol *ctrl+alt+del*, kemudian pilih *processes > windows explorer > button end task* untuk mengakhiri operasi *windows explorer*.

Buka *control panel > search >* ketik "*file explorer option*" *> general* lalu klik *button clear* dan *restore defaults* lalu *apply* dan *ok*, kemudian *restart* komputer.

d. *Error Update Windows*

Gejala :

- a. *Wifi* tidak berfungsi.

- b. Beberapa aplikasi tidak berjalan.
- c. *Windows* akan otomatis masuk ke *system repair*.

Solusi:

Restart komputer, jika dapat mengakses *dekstop* bersihkan berbagai sampah *file* melalui *tool disk clean up*. Bersihkan *device driver packages, previous windows installations dan temporary windows installations*. Setelah itu cabut berbagai kabel *peripheral* lalu lakukan *update windows* kembali.

Jika tidak mau *update windows*, lakukan *nonaktif update windows* dengan cara melalui *task manager > open services > windows update > startup type* menjadi *disable > service status* menjadi *stopping > apply* lalu *oke*. Jika ada *driver* yang bermasalah *install* kembali *driver* melalui *device manager*.

Jika tidak dapat mengakses *dekstop* akan otomatis masuk ke *system repair*, setelah otomatis masuk ke *system repair* akan muncul menu *choose and option*, pilih *troubleshoot > advance option > system restore*.

Dalam *system restore* pilih *restore point* yang memiliki tanggal sebelum terjadinya *blue screen*, setelah *system restore* lalu *restart* komputer kembali.

e. *Windows Button Not Responding*

Gejala :

- a. Tombol *start menu* tidak berfungsi.
- b. Tidak dapat menjalankan aplikasi yang ada di dalam *start menu*.
- c. Tidak dapat melakukan pencarian aplikasi melalui *search menu*.
- d. Tidak dapat mengakses jam dan kalender.

Solusi:

Reset windows dengan cara *settings > windows recovery > recovery*.

Pencegahan agar tidak eror kembali yaitu dengan cara :

1. Jangan *install* 3 rd seperti *avast*, *avira* dan lain- lain.
2. Jangan gunakan tombol *tune up* seperti *tune up utility*, *asc*.
3. Matikan *windows update*.

f. *Error Application*

Gejala :

1. Aplikasi gagal saat proses *install*.
2. Aplikasi yang terinstall tidak dapat beroperasi.
3. Aplikasi otomatis *end proses* saat berjalan.

Solusi:

Install directx, *net framework*, dan *visual c++* terbaru yang dapat *support* oleh *windows 10*.

Resset data aplikasi : *settings* > *system* > *cari apps & features* > cari dan pilih aplikasi yang diinginkan > *advanced options* > *resset*.

Restart komputer dan coba *install* atau jalankan aplikasi kembali.

3.3.4 Basis Pengetahuan Sistem

Basis pengetahuan sistem diperoleh dari proses akusisi langsung pengetahuan dari pakar teknisi komputer Berkah Com melalui proses wawancara dan observasi.

Berdasarkan hasil akusisi pengetahuan disusun basis pengetahuan sistem yang terdiri dari fakta gejala (tabel 1) dan kerusakan (table 2) serta pengetahuan sistem yang berbentuk aturan (table 3).

Tabel 3.1 (Fakta Gejala)

| Kode Gejala | Nama Gejala |
|-------------|--|
| G001 | <i>Sound</i> tidak berfungsi |
| G002 | <i>Wifi</i> tidak berfungsi |
| G003 | Tampilan resolusi layar tidak sesuai VGA |
| G004 | <i>Pointer mouse</i> tidak berfungsi |
| G005 | <i>Bluetooth</i> tidak berfungsi |
| G006 | <i>Port LAN</i> tidak berfungsi |
| G007 | <i>USB</i> tidak berfungsi |
| G008 | Layar <i>desktop</i> menjadi biru dan ada <i>symbol</i> :(|
| G009 | Aplikasi yang digunakan tidak bisa beroperasi |
| G010 | Sistem operasi tidak dapat beroperasi |
| G011 | <i>Windows</i> akan otomatis masuk ke <i>system repair</i> |
| G012 | <i>Loading</i> lama saat proses membuka <i>drive</i> |
| G013 | Gagal dalam <i>proses cut, copy, paste</i> dan <i>delete file</i> |
| G014 | <i>Windows explorer</i> otomatis <i>end proses</i> |
| G015 | <i>Wifi</i> tidak berfungsi |
| G016 | Beberapa aplikasi tidak berjalan |
| G017 | <i>Windows</i> akan otomatis masuk ke <i>system repair</i> |
| G018 | Tombol <i>start menu</i> tidak berfungsi |
| G019 | Tidak dapat menjalankan aplikasi yang ada di dalam <i>start menu</i> |
| G020 | Tidak dapat melakukan pencarian aplikasi melalui <i>search menu</i> |
| G021 | Tidak dapat mengakses jam dan kalender |
| G022 | Aplikasi gagal saat proses <i>install</i> |
| G023 | Aplikasi yang terinstall tidak dapat beroperasi |
| G024 | Aplikasi otomatis <i>end proses</i> saat berjalan |

Tabel 3.2 Fakta Kerusakan

| Kode Kerusakan | Nama Kerusakan |
|----------------|--|
| K001 | <i>Error Driver</i> |
| K002 | <i>Blue screen</i> |
| K003 | <i>Windows explorer not responding</i> |
| K004 | <i>Error update windows</i> |
| K005 | <i>Windows button not responding</i> |
| K006 | <i>Error application</i> |

Referensi : <https://support.microsoft.com>

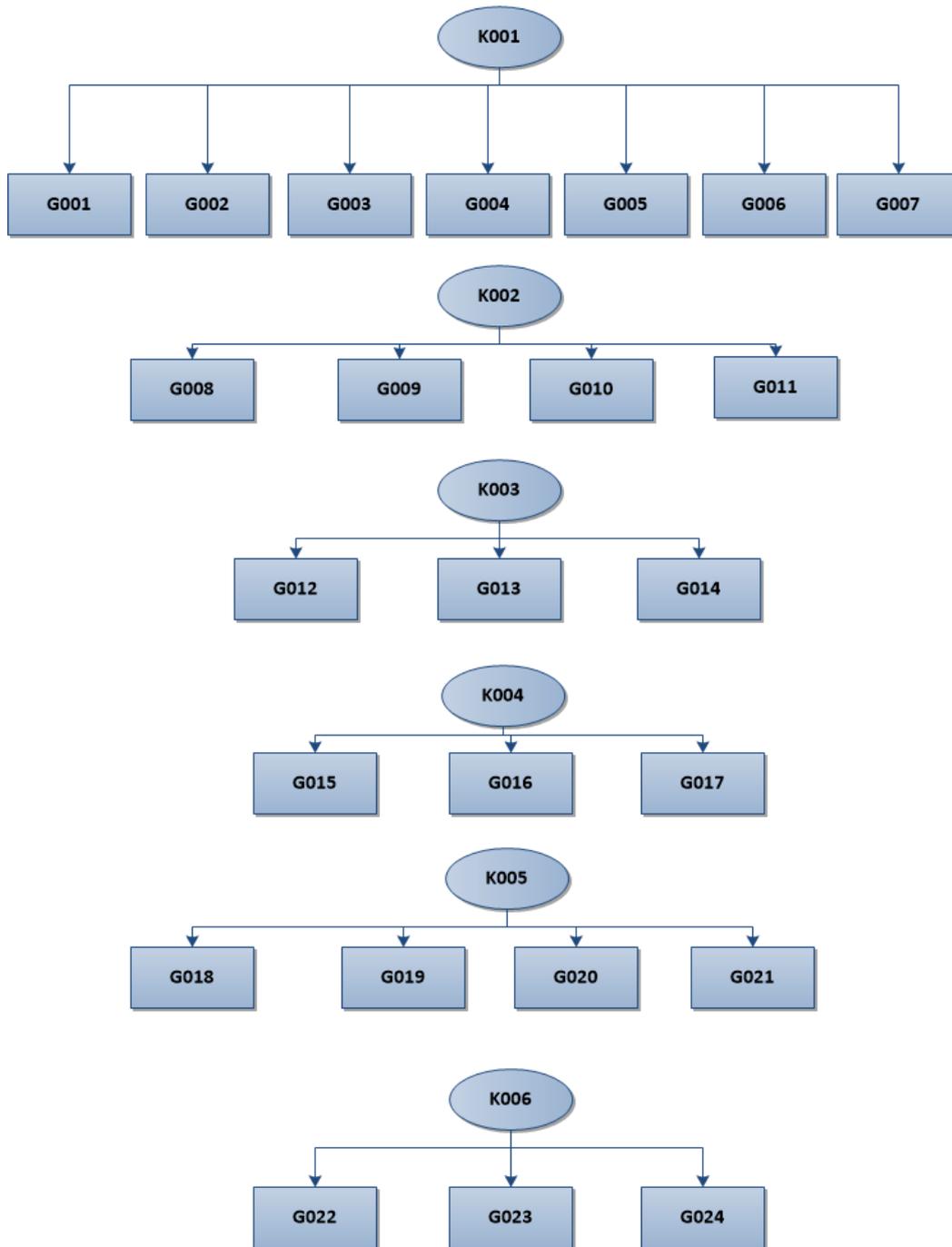
Tabel 3.3 Aturan Pada Sistem

| No | Aturan |
|----|---|
| 1 | <i>IF R1 AND G001 AND G002 AND G003 AND G004 AND G005 AND G006 AND G007 THEN K001</i> |
| 2 | <i>IF R2 AND G008 AND G009 AND G010 AND G011 THEN K002</i> |
| 3 | <i>IF R3 AND G012 AND G013 AND G014 THEN K003</i> |
| 4 | <i>IF R4 AND G015 AND G016 AND G017 THEN K004</i> |
| 5 | <i>IF R5 AND G018 AND G019 AND G020 AND G021 THEN K005</i> |
| 6 | <i>IF R6 AND A022 AND A023 AND A024 THEN K006</i> |

3.3.5 Penalaran Backward Chaining

Sistem diagnosa kerusakan *operating system windows 10* memiliki rantai inferensi seperti pada gambar 3.1, penelusuran runut balik dimulai dari aturan pertama hipotesis K001, selanjutnya sistem akan menampilkan aturan R1 yaitu berisi gejala G001 sampai G007, jika user menjawab salah satu pertanyaan gejala dan menjawab “ya” maka sistem akan menampilkan hasil yaitu solusi dari kerusakan yang dipilih, jika user menjawab “tidak” untuk semua pertanyaan gejala maka sistem akan menampilkan pesan bahwa hasil diagnosa kerusakan yang anda pilih tidak ada.

Begitupun pada aturan R2, R3, R4, R5 dan R6 jika salah satu pertanyaan gejala dijawab user dengan jawaban “ya” maka sistem akan menampilkan hasilnya, jika semua pertanyaan di jawab “tidak” maka sistem tidak akan menampilkan hasil solusi.



Gambar 3.1 Rantai Inferensi System

3.3.6 Analisis Kaidah Produksi

Kaidah produksi dituliskan dalam bentuk jika maka (*IF THEN*). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian, yaitu bagian *premise* (jika) dan bagian konklusi (maka). Apabila bagian *premise* dipenuhi maka bagian konklusi juga akan bernilai benar. Sebuah kaidah terdiri dari klausa – klausa. Sebuah klausa mirip sebuah kalimat subyek, kata kerja dan obyek yang menyatakan suatu fakta. Ada sebuah klausa *premise* dan klausa konklusi pada sebuah kaidah. Suatu kaidah juga terdapat terdiri atas beberapa *premise* dan lebih dari satu konklusi. Antara *premise* dan konklusi dapat berhubungan dengan “*OR*” atau “*AND*”. Berikut kaidah – kaidah produksi dalam menganalisis kerusakan *operating sytem windows 10* :

Rule 1 : IF Error Driver

THEN Sound tidak berfungsi
AND Wifi tidak berfungsi
AND Tampilan resolusi layar tidak sesuai VGA
AND Pointer mouse tidak berfungsi
AND Bluetooth tidak berfungsi
AND Port LAN tidak berfungsi
AND USB tidak berfungsi

Rule 2 : IF Blue Screen

THEN Layar desktop menjadi biru dan ada *symbol* :(
AND Aplikasi yang digunakan tidak bisa beroperasi
AND Sistem operasi tidak dapat beroperasi
AND Windows akan otomatis masuk ke *system repair*

Rule 3 : IF Windows Explorer Not Responding

THEN Loading lama saat proses membuka *drive*
AND Gagal dalam proses *cut, copy, paste, dan delete file*
AND Windows explorer otomatis *end proses*

Rule 4 : IF Error Update Windows

THEN Wifi tidak berfungsi

AND Beberapa aplikasi tidak berjalan

AND Windows akan otomatis masuk ke *system repair*

Rule 5 : IF Windows Button Not Responding

THEN Tombol *start menu* tidak berfungsi

AND Tidak dapat menjalankan aplikasi yang ada di dalam *start menu*

AND Tidak dapat melakukan pencarian aplikasi melalui *search menu*

AND Tidak dapat mengakses jam dan kalender

Rule 6 : IF Error Application

THEN Aplikasi gagal saat proses *install*

AND Aplikasi yang terinstall tidak dapat beroperasi

AND Aplikasi otomatis *end proses* saat berjalan

3.4 Permodelan Desain System

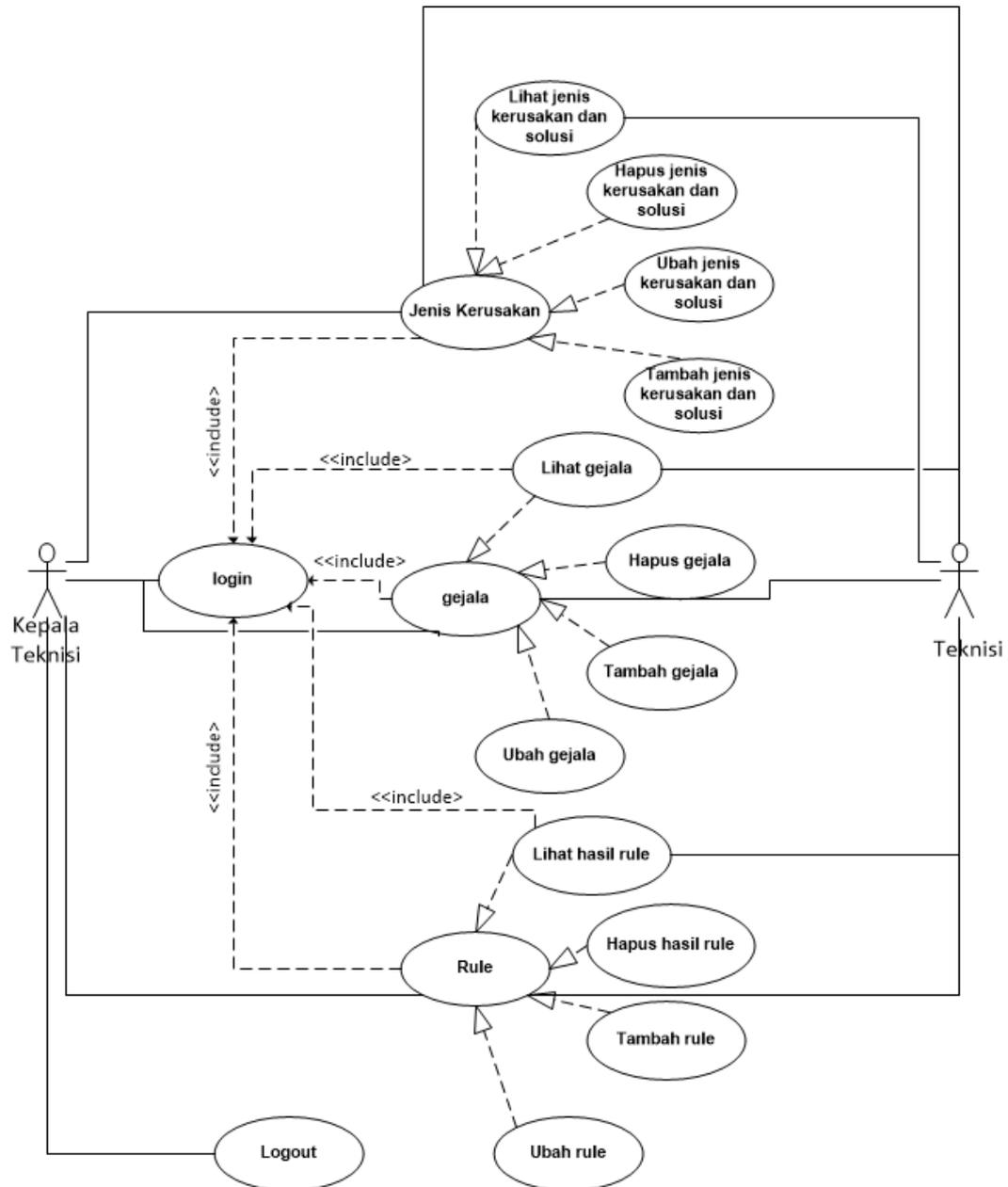
Analisis kebutuhan dari tahap sebelumnya akan dipelajari dalam *fase* ini dan desain sistem disiapkan. Desain Sistem membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3.4.1 Desain UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Pada Penelitian kali ini UML yang dipakai adalah *Use Case Diagram*, *Activity Diagram* dan *ERD Diagram*.

a. Rancangan *Use Case Diagram*

Rancangan *use case diagram* dapat dilihat pada gambar 3.2.



Gambar 3.2 Use case Diagram.

Berdasarkan *use case diagram* pada gambar 3.2 dapat dijelaskan secara singkat alur dari use case sebagai berikut :

Kepala teknisi dapat mengakses jenis kerusakan jika sudah melakukan *login* terlebih dahulu, setelah itu kepala teknisi dapat mengelola jenis kerusakan

seperti melihat jenis kerusakan dan solusi, ubah jenis kerusakan dan solusi, hapus jenis kerusakan dan solusi dan tambah jenis kerusakan dan solusi.

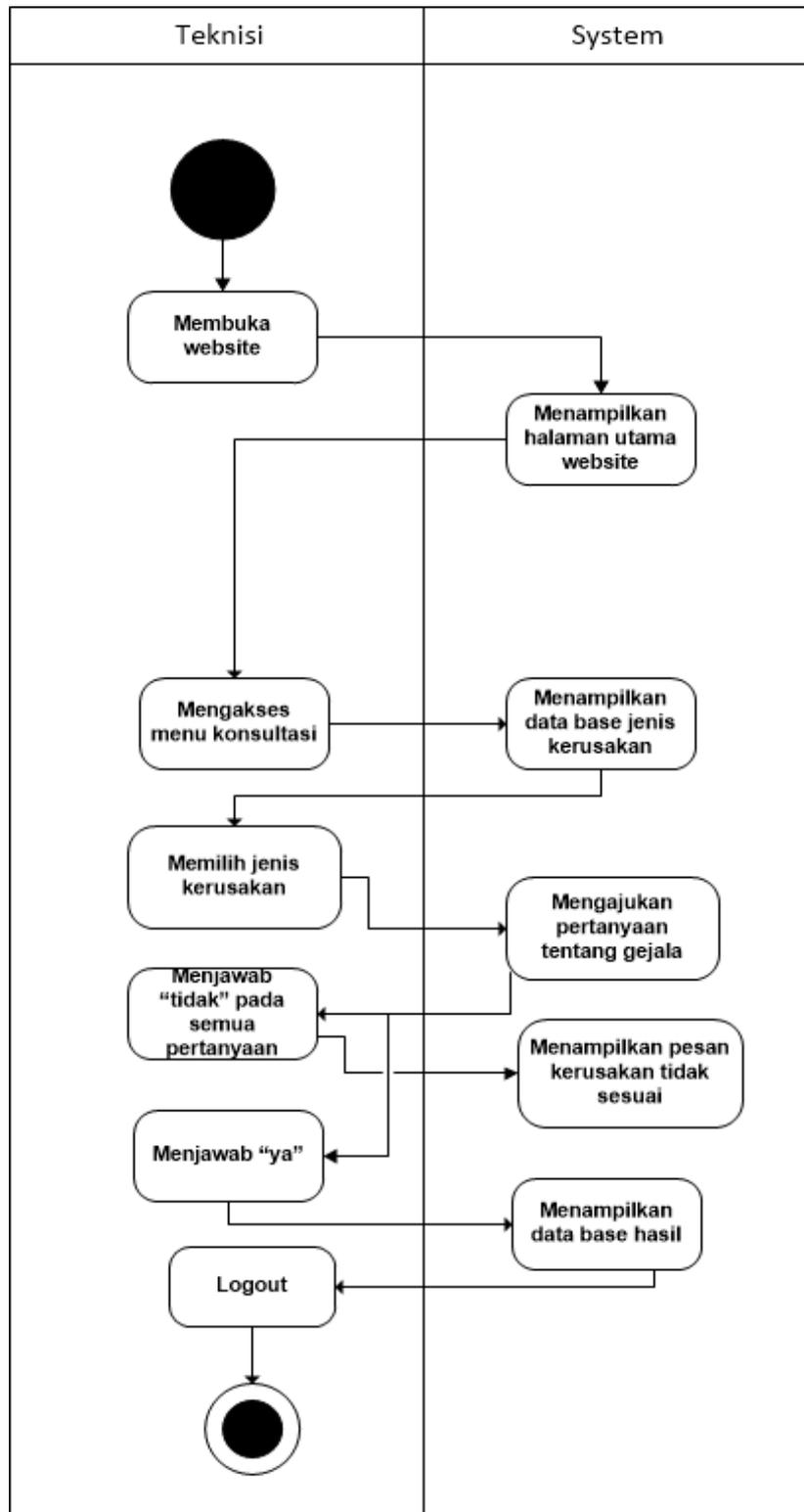
Kepala teknisi dapat mengakses gejala jika sudah melakukan *login* terlebih dahulu, setelah itu kepala teknisi dapat mengelola gejala seperti melihat gejala, tambah gejala, hapus gejala, ubah gejala.

Kepala teknisi dapat mengakses rule jika sudah melakukan *login* terlebih dahulu, setelah itu kepala teknisi dapat mengelola rule seperti melihat *rule*, tambah *rule*, hapus *rule*, ubah *rule*.

Teknisi dapat mengakses jenis kerusakan dan hanya dapat melihat jenis kerusakan dan solusi, kemudian teknisi dapat mengakses gejala dan hanya dapat melihat gejala kemudian teknisi dapat mengakses *rule* dan hanya dapat melihat hasil *rule*.

b. Rancangan *Activity Diagram* Teknisi

Rancangan *activity diagram* admin dapat dilihat pada gambar 3.3.



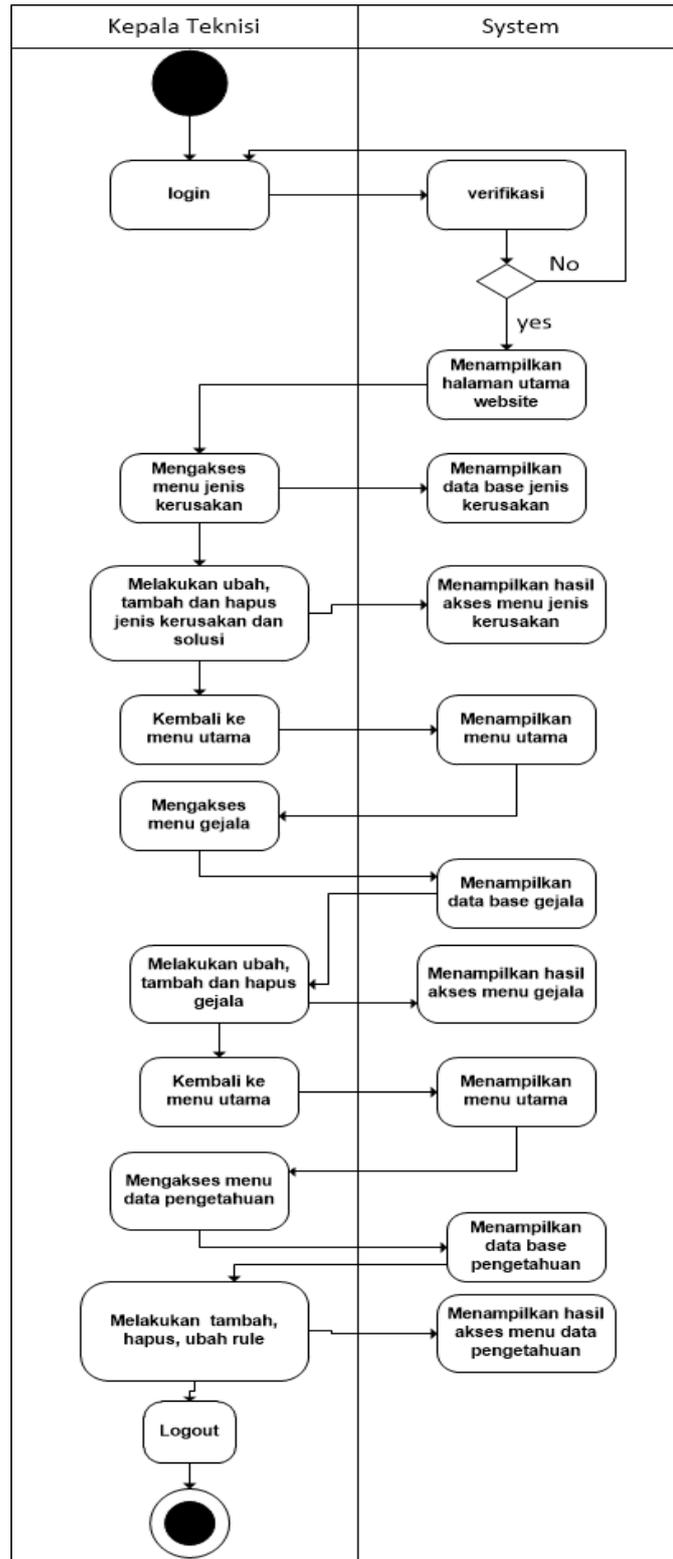
Gambar 3.3 Rancangan *Activity Diagram* Teknisi.

Berdasarkan *Activity Diagram* pada gambar 3.3 dapat dijelaskan secara singkat tentang alur *Activity Diagram* sebagai berikut :

1. Teknisi membuka *website*.
2. Teknisi memilih menu konsultasi kemudian *system* menampilkan halaman berisi data base jenis kerusakan.
3. Teknisi memilih jenis kerusakan kemudian *system* akan otomatis menampilkan gejala yang sesuai jenis kerusakan.
4. Teknisi menjawab pertanyaan yang di ajukan *system* sesuai dengan gejala yang tampil pada kerusakan, jika teknisi menjawab “ya” pada salah satu pertanyaan maka *system* akan menampilkan hasil diagnosa yaitu solusi dari kerusakan tersebut, jika teknisi menjawab “tidak” maka *system* akan memberi pesan kerusakan yang anda pilih tidak sesuai.
5. Teknisi melakukan *logout*.

c. Rancangan *Activity Diagram* Kepala Teknisi.

Rancangan *activity diagram* user dapat dilihat pada gambar 3.4.



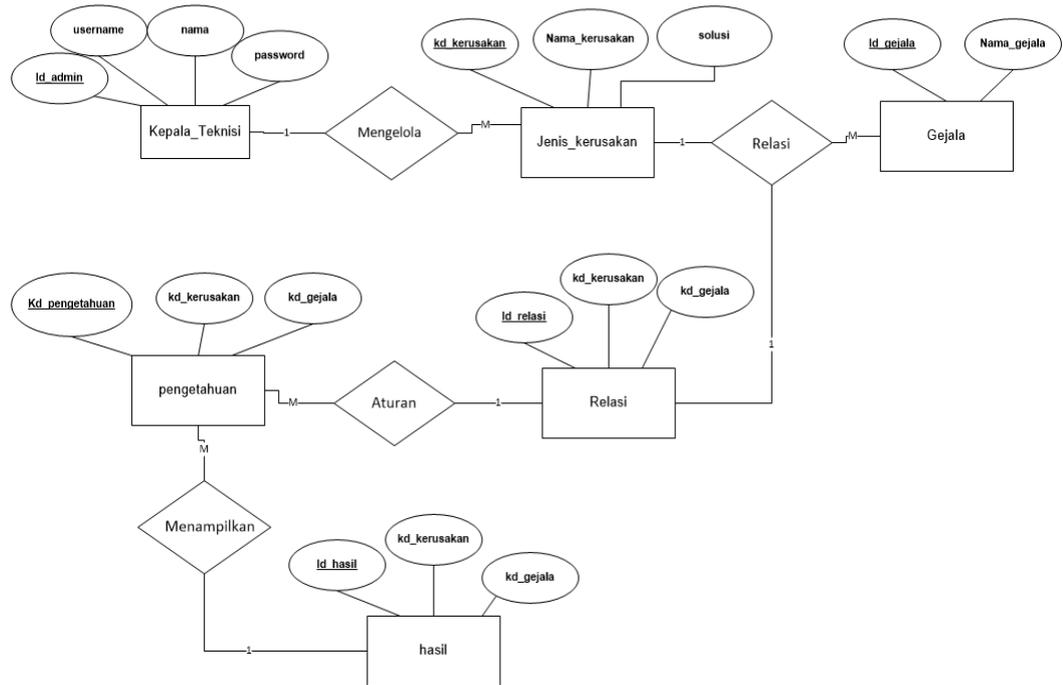
Gambar 3.4 Activity Diagram Kepala Teknisi.

Berdasarkan *Activity Diagram* pada gambar 3.4 dapat dijelaskan secara singkat tentang alur *Activity Diagram* sebagai berikut :

1. Kepala teknisi melakukan login dengan menginput username dan password, lalu system melakukan verifikasi untuk menampilkan halaman utama.
2. Kepala teknisi memilih menu jenis kerusakan untuk mengelola data base jenis kerusakan seperti tambah, ubah dan hapus kemudian *system* akan menampilkan *data base* jenis kerusakan yang telah diatur oleh kepala teknisi.
3. Kepala teknisi kembali ke menu utama untuk mengelola *data base* gejala seperti tambah, ubah dan hapus kemudian *system* akan menampilkan *data base* gejala yang telah diatur oleh kepala teknisi.
4. Kepala teknisi kembali ke menu utama untuk mengelola data base data pengetahuan yang berisi data pengetahuan yaitu rule yang ada pada *system*.
5. Kepala teknisi melakukan *logout*.

d. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) untuk mendeskripsikan data-data atau objek-objek seperti entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi. *Entity Relationship Diagram* pada Sistem Penentu Kerusakan Software Komputer Berbasis Windows Menggunakan Backward Chaining digambarkan dengan rancangan sebagai berikut :



Gambar 3.5 Entity Relationship Diagram.

Berdasarkan *Entity Relationship Diagram* pada gambar 3.5 dapat dijelaskan secara singkat tentang alur *Entity Relationship Diagram* sebagai berikut :

1. Entitas dan Atribut

- Kepala_teknisi (*id_admin* (*primary*), nama, username dan password)
- Jenis_kerusakan (*kd_kerusakan* (*primary*), Nama_kerusakan dan solusi)
- Gejala (*kd_gejala* (*primary*) dan Nama_gejala)
- Relasi (*id_relasi* (*primary*), *kd_kerusakan* (*foreign*), *kd_gejala* (*foreign*))
- Pengetahuan (*kd_pengetahuan* (*primary*), *kd_kerusakan* (*foreign*), *kd_gejala* (*foreign*))
- Hasil (*id_hasil* (*primary*), *kd_kerusakan* (*foreign*), *kd_gejala* (*foreign*))

2. Relasi

- Relasi Mengelola (Relasi 1 : M)
- Relasi Relasi (Relasi jenis_kerusakan dengan gejala 1 : M), (Relasi Jenis_kerusakan dengan relasi 1 : 1), (Relasi gejala dengan relasi M : 1)
- Relasi Aturan (Relasi 1 : M)
- Relasi Menampilkan (Relasi M : 1)

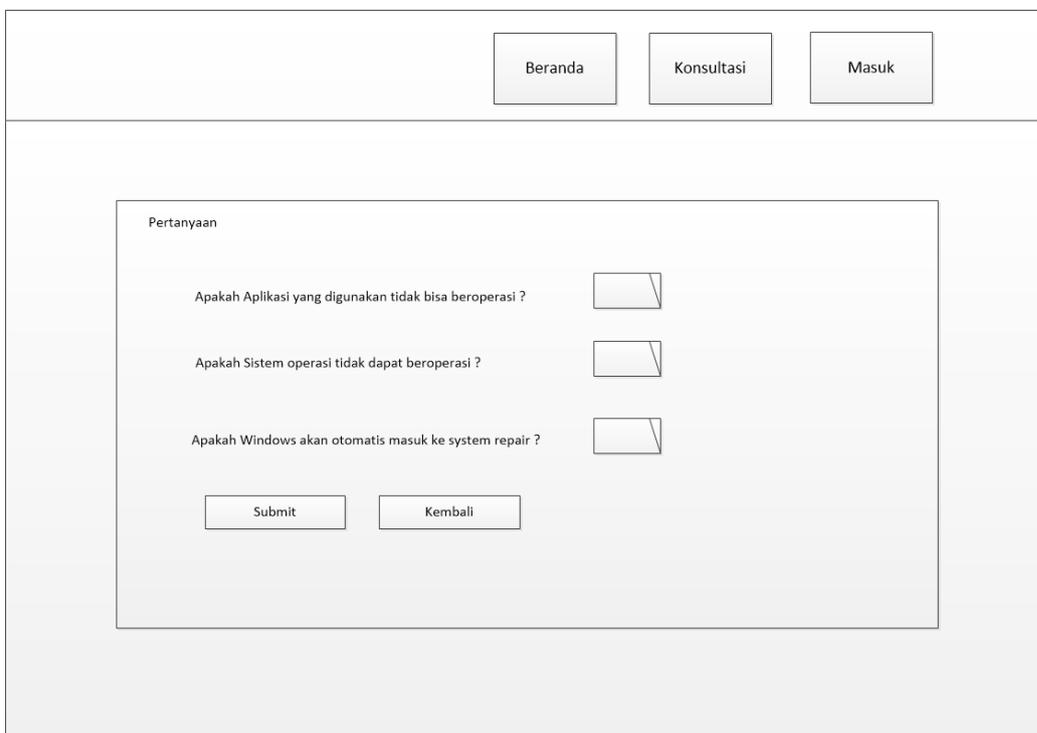
3.5 Perancangan Antarmuka (Desain *Interface*)

Perancangan antarmuka merupakan rancangan dari perangkat lunak yang hendak dibangun dan dalam proses perancangan ini pengembang membagi kebutuhan-kebutuhan perangkat lunak sehingga menghasilkan sebuah arsitektur perangkat lunak dan dapat diterjemahkan kedalam kode-kode program dan *interface*.

Perancangan antarmuka Sistem Penentu Kerusakan Software Komputer Berbasis Windows Menggunakan Backward Chaining digambarkan dengan rancangan sebagai berikut :

3.5.1 Rancangan Tampilan Pertanyaan Gejala

Rancangan tampilan halaman utama dapat dilihat pada gambar 3.6



The image shows a web interface for a software damage diagnosis system. At the top, there is a navigation bar with three buttons: 'Beranda', 'Konsultasi', and 'Masuk'. Below this is a main content area titled 'Pertanyaan'. It contains three diagnostic questions, each with a radio button for selection:

- Apakah Aplikasi yang digunakan tidak bisa beroperasi ?
- Apakah Sistem operasi tidak dapat beroperasi ?
- Apakah Windows akan otomatis masuk ke system repair ?

At the bottom of the question area, there are two buttons: 'Submit' and 'Kembali'.

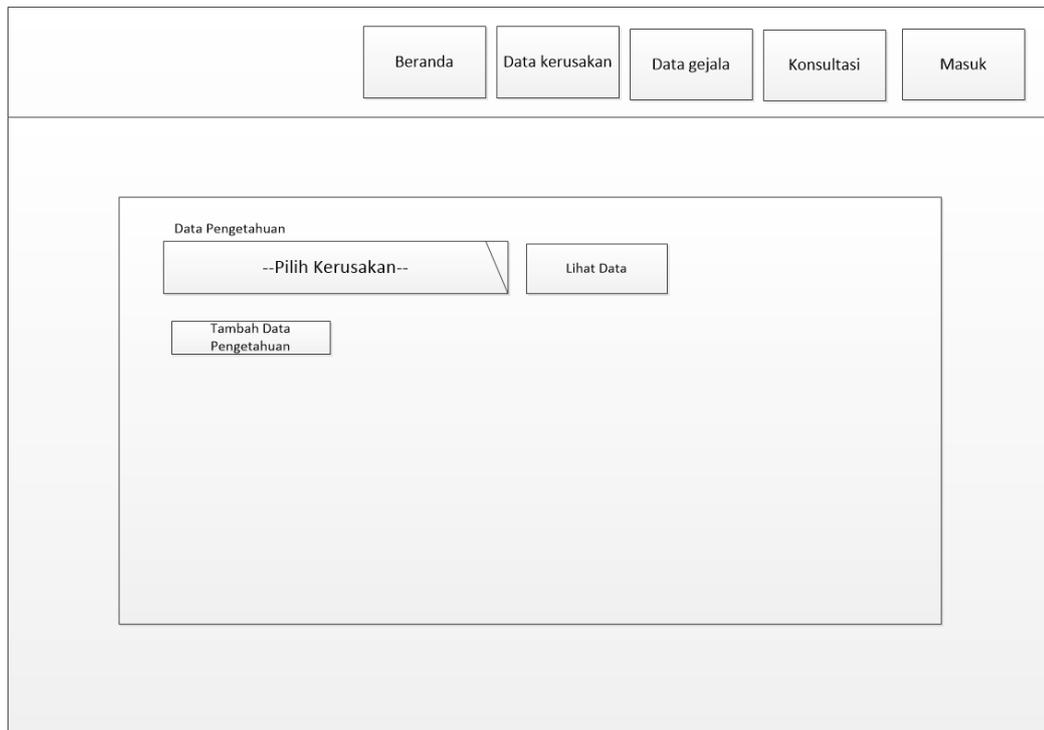
Gambar 3.6 Tampilan Halaman Pertanyaan

Pada gambar 3.6 menampilkan halaman pertanyaan kepada teknisi yaitu pertanyaan tentang gejala yang terjadi saat terjadi kerusakan yang sudah terlebih dahulu dipilih oleh teknisi, jika teknisi menjawab “ya” maka sistem akan menampilkan hasil yaitu solusi dari kerusakan yang di pilih, jika teknisi

menjawab “tidak” pada semua pertanyaan maka sistem akan menampilkan pesan bahwa kerusakan yang anda pilih tidak sesuai.

3.5.2 Rancangan Tampilan Data Pengetahuan (Rule)

Rancangan tampilan halaman utama dapat dilihat pada gambar 3.7.



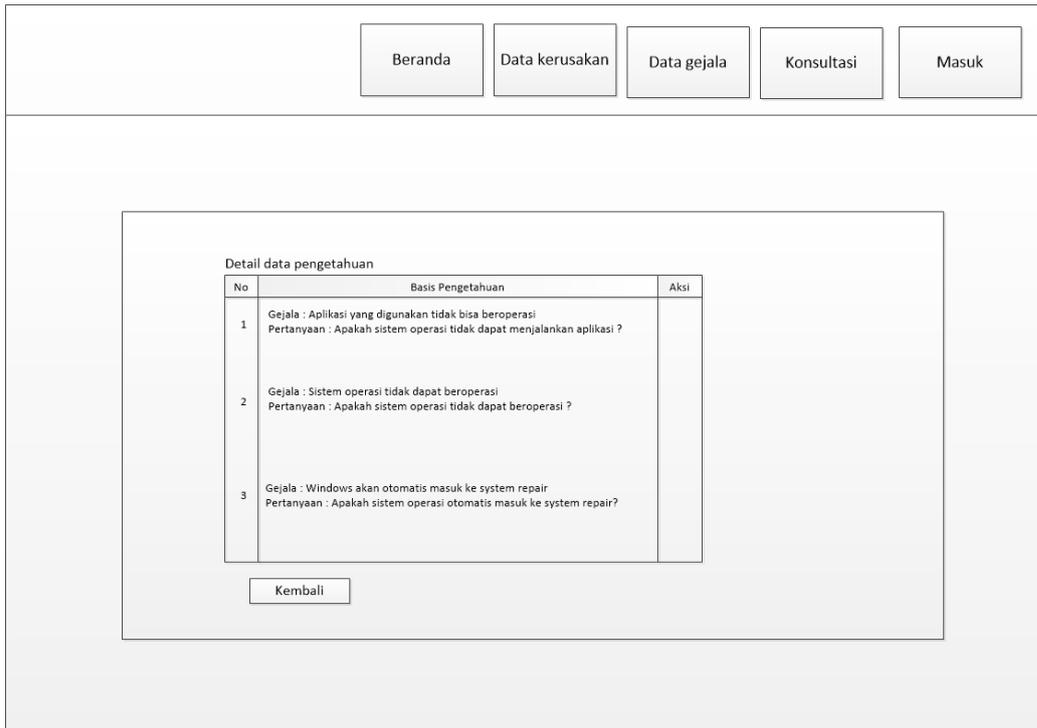
The image shows a web application interface for 'Data Pengetahuan (Rule)'. At the top, there is a navigation bar with five buttons: 'Beranda', 'Data kerusakan', 'Data gejala', 'Konsultasi', and 'Masuk'. Below this, the main content area is titled 'Data Pengetahuan'. It features a dropdown menu with the text '--Pilih Kerusakan--', a 'Lihat Data' button, and a 'Tambah Data Pengetahuan' button.

Gambar 3.7 Tampilan Data Pengetahuan (Rule)

Pada gambar 3.7 menampilkan halaman data pengetahuan yaitu berisi implementasi rule “jika memilih jenis kerusakan maka akan tampil gejala yang sesuai kerusakan”, pilih kerusakan kemudian lihat data akan tampil gejala yang sesuai dengan kerusakan.

3.5.3 Rancangan Tampilan Data Pengetahuan (Rule)

Rancangan tampilan halaman utama dapat dilihat pada gambar 3.8.



Gambar 3.8 Tampilan Data Pengetahuan

Pada gambar 3.8 menampilkan data gejala dan pertanyaan tentang gejala tersebut untuk meyakinkan teknisi akan gejala yang terjadi saat memilih kerusakan.

3.6 Construction of Prototype

Pada tahap ini dimulai dengan pembuatan sistem dan pembuatan script coding, disesuaikan dengan desain sistem yang telah dikerjakan pada tahap *modelling quick design*. Sistem akan dibangun dengan menggunakan Visual code.

3.7 Evaluation of Prototype

Pengujian sistem akan dilakukan dengan menggunakan metode pengujian *black box testing*, yaitu pengujian yang menitikberatkan pada uji fungsionalitas dari program yang dibuat. Hal yang perlu dilakukan dalam pengujian adalah menguji interface program untuk memastikan suatu masukan diproses oleh sistem dengan benar dan menghasilkan keluaran yang sesuai dengan perancangan.

3.8 *Deployment, Delivery and Feedback*

Tahapan ini dilakukan setelah semua tahapan dari *communication*, *quick plan*, *modelling quick design*, dan *construction of prototype* telah selesai dilakukan.

Pada tahapan *deployment*, *system* diserahkan kepada teknisi untuk di evaluasi, tahapan ini dilakukan agar teknisi dapat memberikan masukan tentang kekurangan dari *system* yang di buat.

Pada tahapan *delivery and feedback*, setelah *system* di evaluasi oleh teknisi maka *developer* akan mengetahui kekurangan dan apa yang diinginkan oleh teknisi terhadap *system* yang dibuat, sehingga *developer* mengetahui apa yang harus diperbaiki dengan *system* tersebut.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Hasil penelitian merupakan lanjutan tahap dari perancangan, pada tahap ini sistem pakar yang telah dirancang akan di implementasikan ke dalam bentuk website online.

Sistem pakar tersebut dapat diakses melalui: <http://sigerproject.com/pakar/>

4.1.1 Rancangan Interface

4.1.1.1 Halaman Utama

Halaman ini merupakan halaman *default* yang akan ditampilkan pertama kali ketika teknisi membuka *website*. Tampilan halaman utama dapat dilihat pada gambar 4.1 berikut ini.

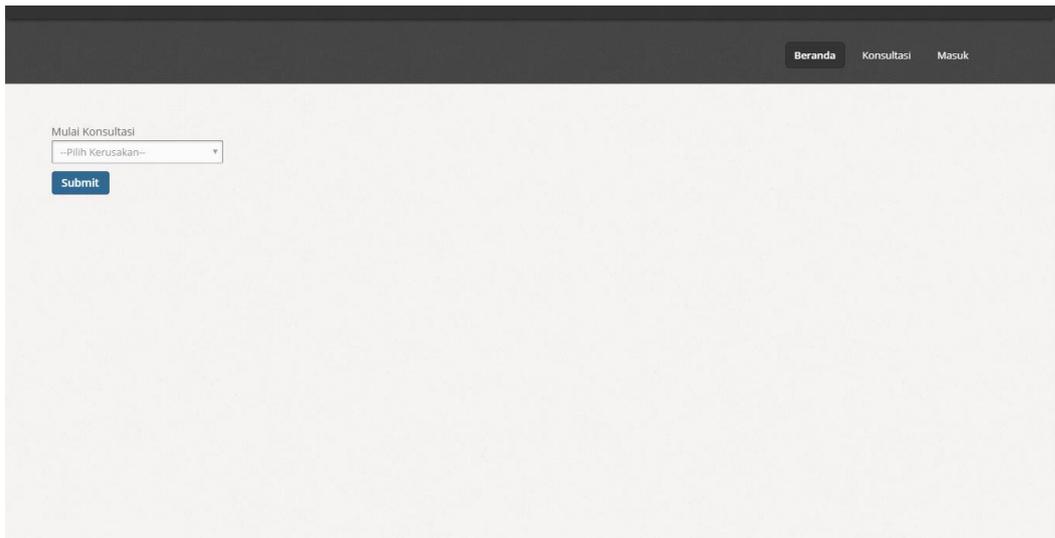


Gambar 4.1 Halaman Utama

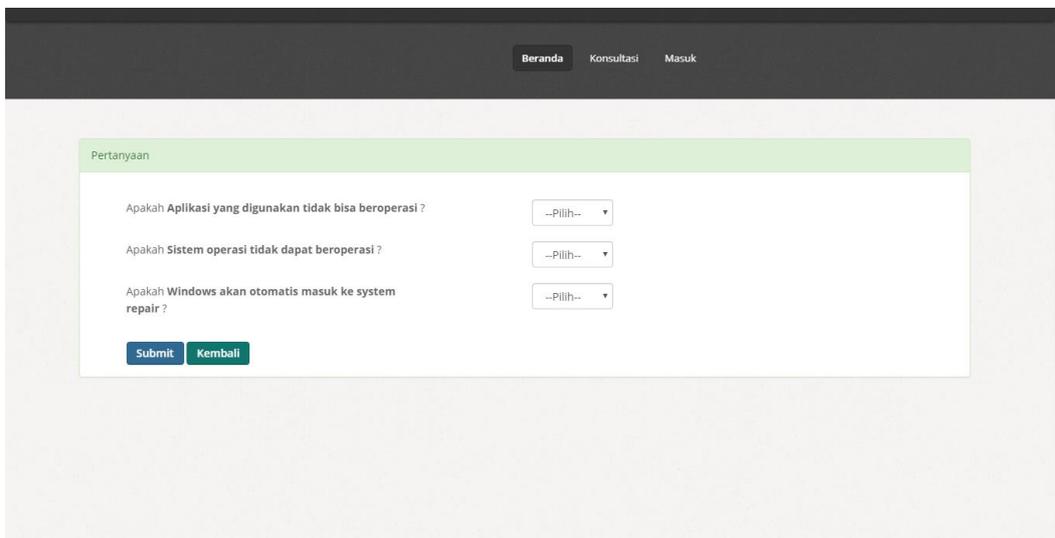
4.1.1.2 Halaman Konsultasi

Halaman konsultasi merupakan halaman untuk teknisi dalam melakukan proses diagnosa kerusakan sistem operasi *windows* 10, teknisi dapat memilih jenis kerusakan yang terjadi, kemudian sistem akan masuk ke dalam menu pertanyaan.

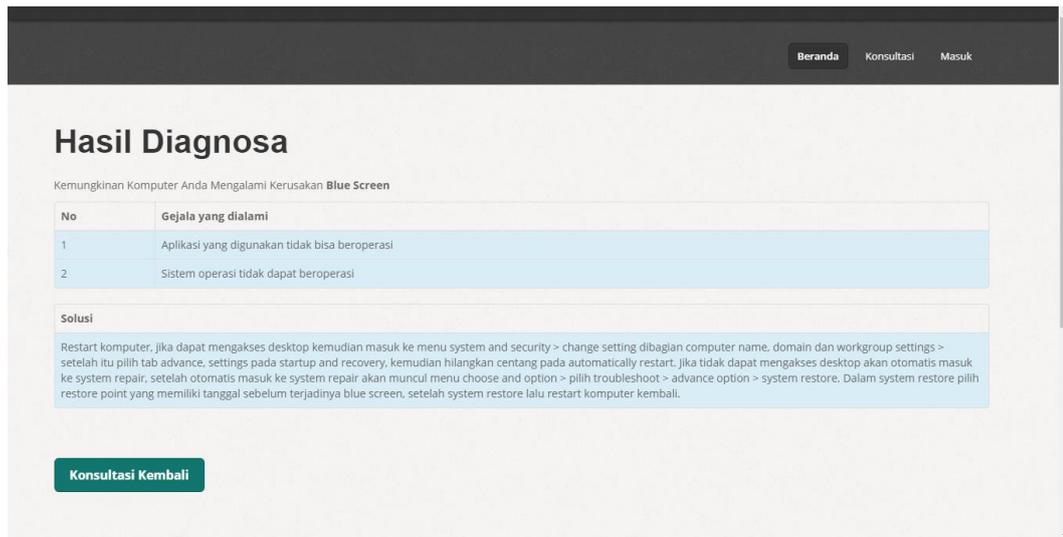
Pada menu pertanyaan, sistem akan menanyakan perihal tentang data gejala yang sesuai dengan kerusakan yang dipilih oleh teknisi, jika teknisi menjawab “ya” maka sistem akan menampilkan hasil diagnosa yaitu solusi dari kerusakan yang di pilih, jika teknisi menjawab “tidak” maka sistem akan menampilkan pesan “kerusakan yang anda pilih tidak sesuai”. Menu konsultasi dapat dilihat pada gambar 4.2, 4.3 dan 4.4 berikut.



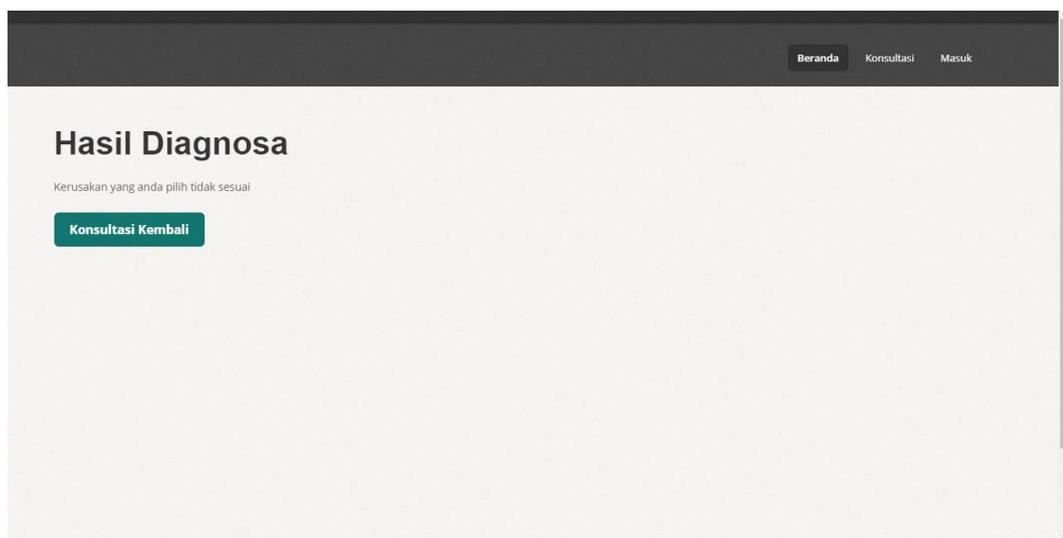
Gambar 4.2 Halaman Menu Konsultasi (Pilih kerusakan)



Gambar 4.3 Halaman Menu Konsultasi (Menjawab Pertanyaan)



Gambar 4.4 Halaman Menu Konsultasi (Hasil Dagnosa Jawaban “ya”)



Gambar 4.5 Halaman Menu Konsultasi (Hasil Diagnosa Jawaban “tidak”)

4.1.1.3 Halaman *Login* Kepala Teknisi

Menu *Login* kepala teknisi merupakan halaman yang menampilkan *username* dan *password* milik kepala teknisi sesuai ketentuan yang ada sehingga dapat mengakses *website* sebagai *administrator*. Menu *login* kepala teknisi dapat dilihat pada gambar 4.6.

Gambar 4.6 Halaman *Login* Kepala Teknisi

4.1.1.4 Halaman Data Kerusakan

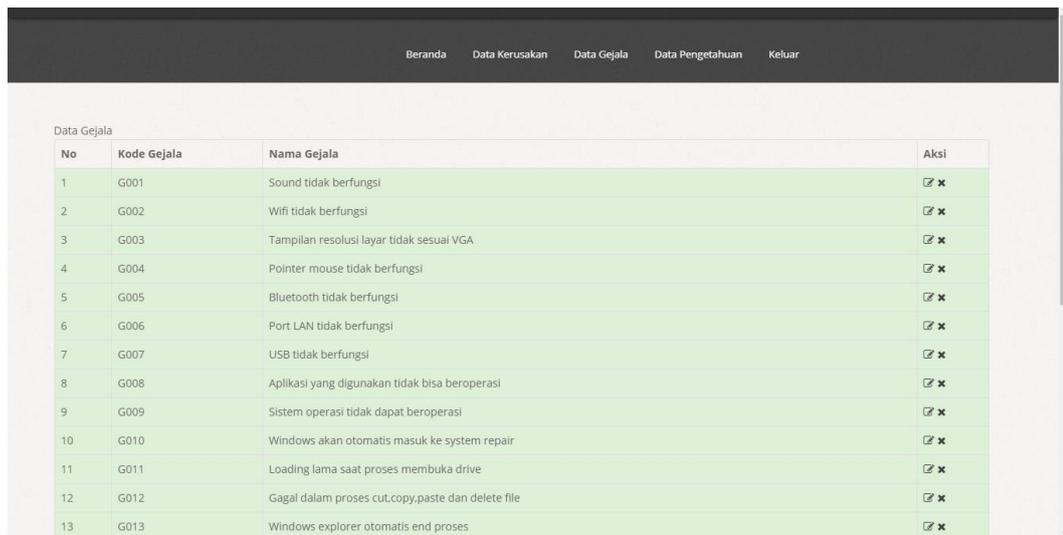
Halaman ini merupakan halaman yang berisi tentang data - data kerusakan seperti No, Kode Kerusakan, Nama Kerusakan, Solusi. Pada halaman ini kepala teknisi mempunyai hak akses untuk menambah, menghapus, dan mengubah data jenis kerusakan.

| No | Kode Kerusakan | Nama Kerusakan | Solusi | Aksi |
|----|----------------|---------------------------------|--|---------------------------------------|
| 1 | K001 | Error Driver | Install ulang driver yang bermasalah, dengan cara masuk ke menu control panel > system and security > system > device manager. Kemudian update driver yang bermasalah lalu terdapat dua pilihan yaitu search driver automatically for driver dan browse my computer for driver. Driver automatically for driver untuk menginstall driver secara otomatis menggunakan koneksi internet, browse my computer for driver untuk menginstall driver yang sudah ada dalam bentuk file yang sudah jadi di dalam drive komputer. | <input checked="" type="checkbox"/> x |
| 2 | K002 | Blue Screen | Restart komputer, jika dapat mengakses desktop kemudian masuk ke menu system and security > change setting dibagian computer name, domain dan workgroup settings > setelah itu pilih tab advance, settings pada startup and recovery, kemudian hilangkan centang pada automatically restart. Jika tidak dapat mengakses desktop akan otomatis masuk ke system repair, setelah otomatis masuk ke system repair akan muncul menu choose and option > pilih troubleshoot > advance option > system restore. Dalam system restore pilih restore point yang memiliki tanggal sebelum terjadinya blue screen, setelah system restore lalu restart komputer kembali. | <input checked="" type="checkbox"/> x |
| 3 | K003 | Windows Explorer Not Responding | Buka task manager dengan cara tekan tombol ctrl+alt+del, kemudian pilih processes > windows explorer > button end task untuk mengakhiri operasi windows explorer. Buka control panel > search > ketik "file explorer option" > general lalu klik button clear dan restore defaults lalu apply dan ok, kemudian restart komputer. | <input checked="" type="checkbox"/> x |
| 4 | K004 | Error Update Windows | Restart komputer, jika dapat mengakses desktop bersihkan berbagai sampah file melalui tool disk clean up. Bersihkan device driver packages, previous windows installations dan temporary windows installations. Setelah itu cabut berbagai kabel peripheral lalu lakukan update windows kembali, jika tidak mau update windows lakukan nonaktif update windows dengan cara melalui task manager > open services > windows update > startup type menjadi disable > service status menjadi stopping > apply lalu oke. Jika ada driver yang bermasalah install kembali driver melalui device manager. Jika tidak dapat mengakses desktop akan otomatis masuk ke system repair, setelah otomatis masuk ke system repair akan muncul menu choose and option, pilih troubleshoot > advance option > system restore. Dalam system restore pilih restore point yang memiliki tanggal sebelum terjadinya blue screen, setelah system restore lalu restart | <input checked="" type="checkbox"/> x |

Gambar 4.7 Halaman Data Kerusakan

4.1.1.5 Halaman Data Gejala

Halaman ini merupakan halaman yang berisi data - data gejala yang terjadi jika kerusakan dipilih oleh teknisi, isi dari data - data gejala yaitu seperti No, Kode Gejala, Nama Gejala. Pada halaman ini kepala teknisi mempunyai hak akses untuk menambah, menghapus, dan mengubah data gejala kerusakan.

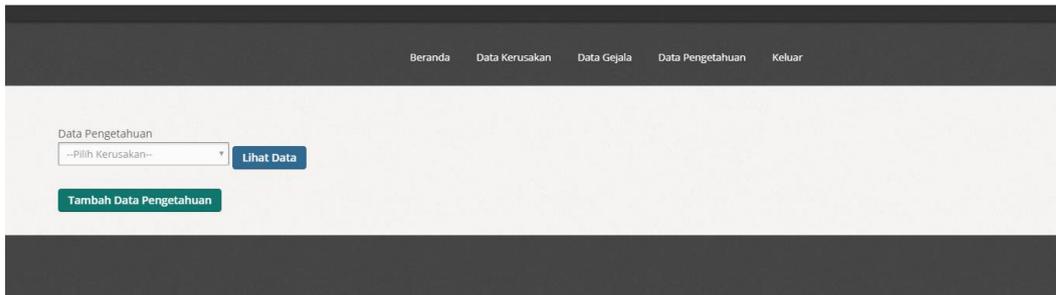


| No | Kode Gejala | Nama Gejala | Aksi |
|----|-------------|---|------|
| 1 | G001 | Sound tidak berfungsi | ✎ ✕ |
| 2 | G002 | Wifi tidak berfungsi | ✎ ✕ |
| 3 | G003 | Tampilan resolusi layar tidak sesuai VGA | ✎ ✕ |
| 4 | G004 | Pointer mouse tidak berfungsi | ✎ ✕ |
| 5 | G005 | Bluetooth tidak berfungsi | ✎ ✕ |
| 6 | G006 | Port LAN tidak berfungsi | ✎ ✕ |
| 7 | G007 | USB tidak berfungsi | ✎ ✕ |
| 8 | G008 | Aplikasi yang digunakan tidak bisa beroperasi | ✎ ✕ |
| 9 | G009 | Sistem operasi tidak dapat beroperasi | ✎ ✕ |
| 10 | G010 | Windows akan otomatis masuk ke system repair | ✎ ✕ |
| 11 | G011 | Loading lama saat proses membuka drive | ✎ ✕ |
| 12 | G012 | Gagal dalam proses cut,copy,paste dan delete file | ✎ ✕ |
| 13 | G013 | Windows explorer otomatis end proses | ✎ ✕ |

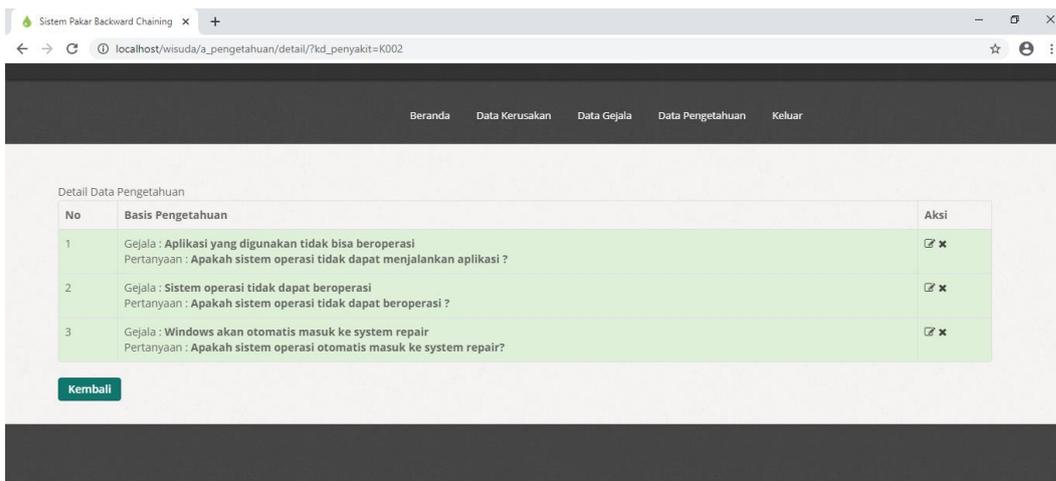
Gambar 4.8 Halaman Data Gejala

4.1.1.6 Halman Data Pengetahuan

Didalam halaman ini berisi implementasi aturan dari Sistem Pakar Penentu Kerusakan Software Komputer Berbasis Windows Menggunakan Backward Chaining, yaitu jika memilih kerusakan maka akan muncul gejala yang disertai oleh pertanyaan sesuai dari kerusakan tersebut.



Gambar 4.9 Halaman Data Pengetahuan (Pilih Kerusakan)



Gambar 4.10 Halaman Data Pengetahuan (Gejala dan Pertanyaan)

4.1.1.7 Source Code

Source Code adalah kumpulan dari beberapa kode bahasa pemrograman tertentu yang membentuk sebuah deklarasi atau perintah yang dapat dibaca oleh komputer.

Pada penelitian ini bahasa pemrograman yang di gunakan yaitu HTML5 dan PHP. Adapun *Source Code* aplikasi (*terlampir*).

4.2 Pembahasan

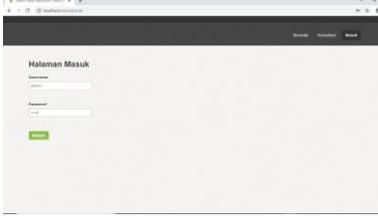
Hasil pembahasan adalah hasil dari pengamatan atau penelitian yang terdapat dalam teks ilmiah. Sebagai laporan observasi tentang penilaian terhadap sesuatu. Hasil pengamatan merupakan isi bagian yang penting dari teks ilmiah. Hasil pembahasan mempunyai sifat objektif atau subjektif. Hasil pembahasan sebagai pertimbangan atau acuan, untuk dijadikan sebagai sebuah teori.

4.2.1 Pengujian Black Box

a. Pengujian Pada Menu Login Kepala Teknisi

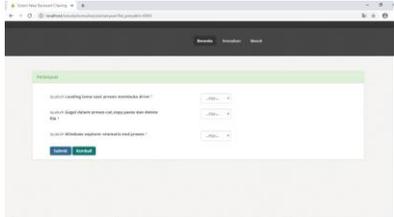
Tabel 4.1 Pengujian *Black Box* Pada Menu *Login* Kepala Teknisi

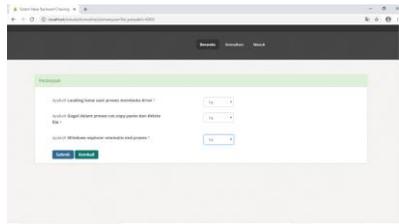
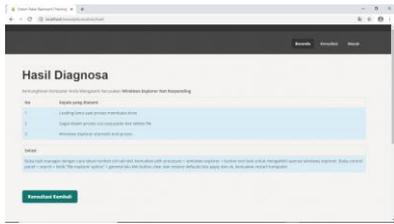
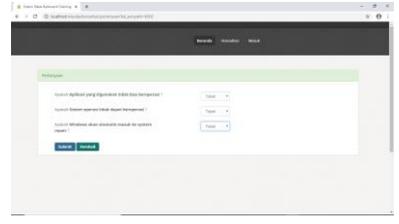
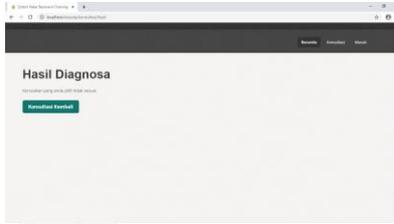
| No | Skenario Pengujian | Hasil | Kesimpulan |
|----|---|--|--------------|
| 1 | Mengosongkan <i>Username</i> dan <i>Password</i> <i>Test Case :</i>  | Sistem akan menolak akses <i>login</i> <i>Hasil :</i>  | <i>Valid</i> |
| 2 | Mengisi Hanya <i>User Name</i> tanpa <i>password</i> | Sistem akan menolak akses <i>login</i> | <i>Valid</i> |

| | | | |
|---|--|--|--------------|
| | <p><i>Test Case :</i></p>  | <p><i>Hasil :</i></p>  | |
| 3 | <p>Mengisi <i>Password</i> dan <i>Username</i> yang telah ditentukan</p> <p><i>Test Case:</i></p>  | <p>Sistem akan masuk ke halaman <i>admin</i></p> <p><i>Hasil :</i></p>  | <i>Valid</i> |

b. Pengujian *Black Box* Pada Halaman Konsultasi

Tabel 4.2 Pengujian *Black Box* Pada Halaman Konsultasi.

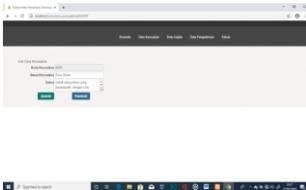
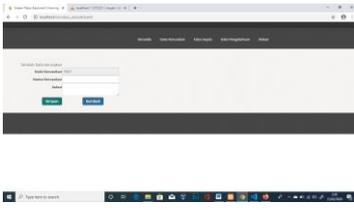
| No | Skenario Pengujian | Hasil | Kesimpulan |
|----|--|---|--------------|
| 1 | <p>Teknisi memilih jenis kerusakan</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan pertanyaan tentang gejala yang sesuai kerusakan</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |
| 2 | <p>Teknisi menjawab salah satu pertanyaan dengan jawaban</p> | <p>Sistem akan menampilkan hasil diagnosa</p> | <i>Valid</i> |

| | | | |
|---|---|---|--------------|
| | <p>“ya”</p> <p><i>Test Case :</i></p>  | <p>Hasil :</p>  | |
| 3 | <p>Teknisi menjawab semua pertanyaan dengan jawaban “tidak”</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan pesan “kerusakan yang anda pilih tidak sesuai”</p> <p>Hasil :</p>  | <i>Valid</i> |

c. Pengujian *Black Box* Pada Data Kerusakan

Tabel 4.3 Pengujian *Black Box* Pada Halaman Data Kerusakan

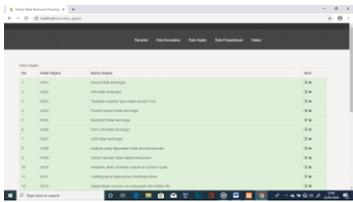
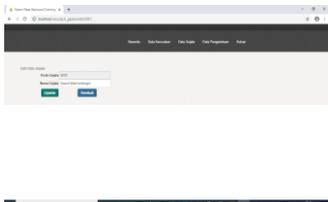
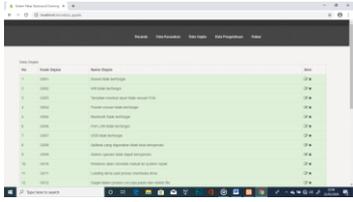
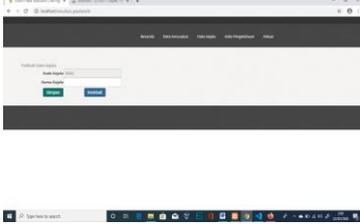
| No | Skenario Pengujian | Hasil | Kesimpulan |
|----|--|--|--------------|
| 1 | Kepala teknisi memilih edit data kerusakan | Sistem akan menampilkan data kerusakan yang akan di edit | <i>Valid</i> |

| | | | |
|---|---|---|--------------|
| | <p><i>Test Case :</i></p>  | <p><i>Hasil :</i></p>  | |
| 2 | <p>Kepala teknisi memilih hapus data kerusakan</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan pesan “anda yakin ? “ jika ok maka terhapus jika <i>cancel</i> maka tidak terhapus</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |
| 3 | <p>Kepala teknisi memilih tambah data kerusakan</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan form tambah data kerusakan</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |

d. Pengujian *Black Box* Pada Halaman Data Gejala

Tabel 4.4 Pengujian *Black Box* Pada Halaman Data Gejala

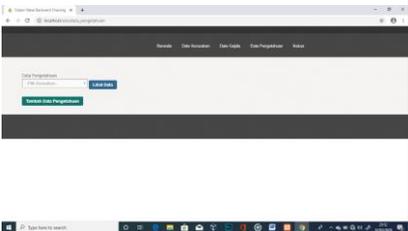
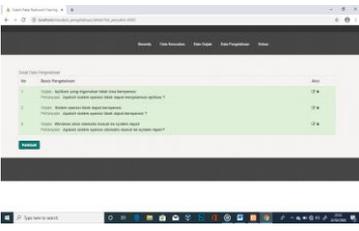
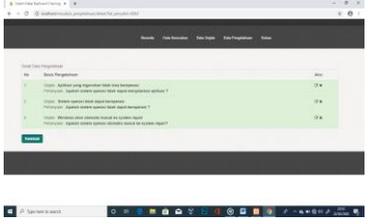
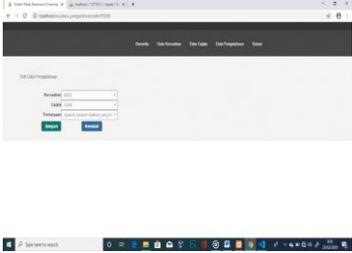
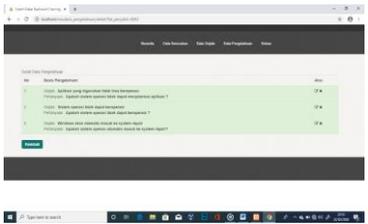
| No | Sekenario Pengujian | Hasil | Kesimpulan |
|----|---|---|--------------|
| 1 | Kepala teknisi memilih edit data gejala | Sistem akan menampilkan data gejala yang akan di edit | <i>Valid</i> |

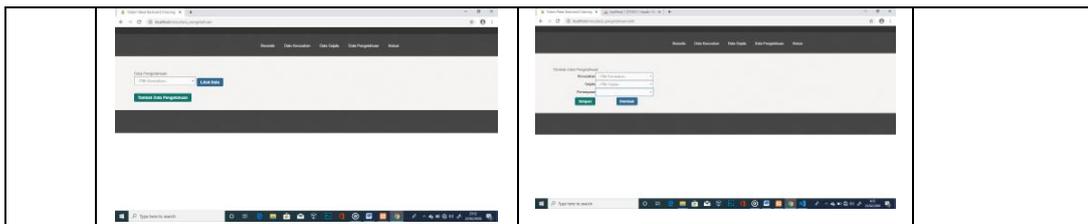
| | | | |
|---|---|---|--------------|
| | <p><i>Test Case :</i></p>  | <p><i>Hasil :</i></p>  | |
| 2 | <p>Kepala teknisi memilih hapus data gejala</p> <p><i>Test Case:</i></p>  | <p>Sistem akan menampilkan pesan “anda yakin ? “ jika ok maka terhapus jika <i>cancel</i> maka tidak terhapus</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |
| 3 | <p>Kepala teknisi memilih tambah data gejala</p> <p><i>Test Case:</i></p>  | <p>Sistem akan menampilkan <i>form</i> tambah data gejala</p> <p><i>Hasil:</i></p>  | <i>Valid</i> |

c. Pengujian *Black Box* Pada Halaman Data Pengetahuan

Tabel 4.5 Pengujian *Black Box* Pada Data Pengetahuan

| No | Skenario Pengujian | Hasil | Kesimpulan |
|----|--------------------------------|---|--------------|
| 1 | Kepala teknisi jenis kerusakan | Sistem akan menampilkan data gejala dan pertanyaan mengenai gejala yang | <i>Valid</i> |

| | | | |
|---|--|--|--------------|
| | <p><i>Test Case :</i></p>  | <p>sesuai dengan kerusakan</p> <p><i>Hasil :</i></p>  | |
| 2 | <p>Kepala teknisi memilih edit data pengetahuan</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan edit data pengetahuan berupa kode kerusakan, kode gejala dan pertanyaan.</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |
| 3 | <p>Kepala teknisi memilih hapus data pengetahuan</p> <p><i>Test Case :</i></p>  | <p>Sistem akan menampilkan pesan “anda yakin ? “ jika ok maka terhapus jika <i>cancel</i> maka tidak terhapus</p> <p><i>Hasil :</i></p>  | <i>Valid</i> |
| 4 | <p>Kepala teknisi memilih tambah data pengetahuan</p> <p><i>Test Case :</i></p> | <p>Sistem akan menampilkan data pengetahuan yang akan di tambah</p> <p><i>Hasil :</i></p> | <i>Valid</i> |



4.3 Kelebihan Dan Kekurangan Sistem Pakar

Setelah dilakukan pengujian sistem identifikasi ini ternyata aplikasi dapat berjalan dengan baik. Banyak manfaat yang dapat diambil dengan adanya sistem ini, khususnya bagi teknisi komputer Berkah Com dalam proses mendiagnosa kerusakan *software* khususnya sistem operasi *windows 10*.

4.3.1 Kelebihan Sistem Pakar

Kelebihan dari sistem ini adalah sebagai berikut :

- a) Sistem yang telah dibuat berbentuk web dapat diakses dengan mudah oleh teknisi komputer.
- b) Sistem diagnosa ini dapat membantu teknisi komputer dalam menyelesaikan pekerjaannya dengan lebih cepat.
- c) Sistem diagnosa ini dapat membantu pengguna komputer untuk lebih cepat menerima hasil diagnosa teknisi komputer

Tabel 4.6 Perbandingan diagnosa menggunakan aplikasi dan tanpa aplikasi

| Kerusakan | Tanpa Aplikasi | Menggunakan Aplikasi |
|--|----------------|----------------------|
| <i>error driver</i> | 30 menit | 20 menit |
| <i>blue screen</i> | 40 menit | 25 menit |
| <i>windows explorer not responding</i> | 30 menit | 20 menit |
| <i>error windows update</i> | 25 menit | 15 menit |
| <i>windows button not responding</i> | 20 menit | 15 menit |
| <i>error application</i> | 25menit | 20 menit |

4.3.2 Kelemahan Sistem Pakar

Kekurangan dari sistem ini adalah sebagai berikut :

- a) Sistem diagnosa ini masih berbasis *website*.
- b) Data jenis kerusakan dan gejala pada sistem ini masih sangat terbatas.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan pembahasan dari bab-bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut :

1. Pemanfaatan Sistem Pakar Penentu Kerusakan Software Komputer Berbasis Windows Menggunakan Backward Chaining berjalan sesuai dengan rancangannya, yaitu dapat menampilkan hasil diagnosa kerusakan sistem operasi *windows 10*.
2. Pemanfaatan sistem ini dapat membantu teknisi komputer dalam menyelesaikan pekerjaannya dengan lebih cepat.
3. Membantu pengguna komputer untuk mendapatkan hasil diagnosa lebih cepat.
4. Sistem yang di rancang dalam penelitian ini menggunakan metode *backward chaining* yang digunakan untuk menentukan jenis kerusakan yang terjadi pada sistem operasi *windows 10*.

5.2 Saran

Saran yang diberikan sesuai dengan adanya penelitian yang telah di lakukan adalah:

1. Membandingkan atau menggabungkan dengan algoritma lain misalkan algoritma *Forward Chaining, Naive Bayes*, dan sebagainya.
2. Menambah data-data jenis kerusakan dan gejala yang terjadi pada sistem operasi *windows 10*.
3. Aplikasi Implementasi backward chaining menentukan kerusakan sistem operasi *windows 10* perlu dilakukan update basis data secara berkala untuk memelihara dan menjaga keakuratan data.
4. Mengembangkan lagi sistem pakar ini dengan menambah jenis kerusakan komputer yang lainnya, seperti kerusakan *hardware* dan aplikasi lainnya.
5. Mengembangkan aplikasi ini pada perangkat *mobile* dan *platform* lain, seperti *ios* dan *windows phone*.