

## Sistem Pakar Mendeteksi Gejala Kerusakan Pada Laptop Berbasis Web

Dewi Apriyanti<sup>1</sup>, Novi Yona Sidratul Munti<sup>2\*</sup>

<sup>1,2</sup>Fakultas Teknik, Teknik Informatika, Universitas Pahlawan  
Tuanku Tambusai, Kab. Kampar, Provinsi Riau, Indonesia

E-mail : <sup>1</sup>dewiapriyanti636@gmail.com, <sup>2\*</sup>sikumbang\_ona@yahoo.com

(\* : corresponding author)

### Abstrak

Sekarang ini sudah tidak dipungkiri bahwa teknologi informasi dan komputer merupakan salah satu kebutuhan manusia yang sangat mendasar dalam membantu pekerjaan dalam bidang komputerisasi. Bagian atau komponen hardware maupun software laptop dalam jangka waktu tertentu akan mengalami perubahan fisik maupun kerusakan, yang menyebabkan laptop harus diperbaiki. Oleh karena itu, sangat dianjurkan bagi pengguna laptop untuk mengetahui cara merawat dan memperbaiki ketika laptop bermasalah, sebelum memutuskan untuk menyerahkannya ke tempat servis. Maka Diperlukan sebuah sistem berbasis web untuk membantu pengguna mengetahui gejala kerusakan pada laptop beserta solusi dan cara penanganannya. Sistem Pakar adalah sistem yang yang mampu mengambil kemampuan dan pengetahuan seorang Pakar untuk menyelesaikan suatu masalah. Dalam penelitian ini metode yang digunakan adalah metode Certainty Factor yang dapat menggambarkan tingkat keyakinan seorang pakar terhadap masalah yang sedang dihadapi. Sistem pakar untuk mendeteksi gejala kerusakan pada laptop yang sudah diimplementasikan dapat berjalan dengan baik dan dapat mendeteksi gejala kerusakan laptop serta memberikan solusinya. Dengan sistem ini, penulis berharap dapat membantu pengguna laptop untuk mendeteksi gejala kerusakan pada perangkatnya.

Kata kunci: sistem pakar, kerusakan laptop, *certainty factor*, web

### Abstract

*Now it is undeniable that information technology and computers are one of the very basic human needs in helping work in the field of computerization. Parts or components of laptop hardware or software within a certain period of time will experience physical changes or damage, which causes the laptop to be repaired. Therefore, it is highly recommended for laptop users to know how to care for and repair when a laptop has a problem, before deciding to hand it over to a service center. So a web-based system is needed to help users find out the symptoms of damage to the laptop along with solutions and how to handle them. Expert system is a system that is able to take the ability and knowledge of an expert to solve a problem. In this study, the method used is the Certainty Factor method which can describe the level of confidence an expert on the problem at hand. Expert systems to detect symptoms of damage to laptops that have been implemented can run well and can detect symptoms of laptop damage and provide solutions. With this system, the author hopes to help laptop users to detect symptoms of damage to their devices.*

*Keywords: expert system, laptop damage, certainty factor, web*

## 1. PENDAHULUAN

Pada saat ini sudah tidak dapat dipungkiri lagi bahwa teknologi informasi dan komputer merupakan salah satu kebutuhan manusia yang paling mendasar terutama dalam membantu setiap pekerjaan dalam bidang komputerisasi. Dalam hal ini laptop merupakan salah satu jenis komputer yang banyak diminati oleh masyarakat, harganya yang semakin terjangkau dan memiliki mobilitas yang tinggi menjadi alasan paling kuat masyarakat untuk memiliki sebuah laptop.

Bagian atau komponen *hardware* laptop dalam jangka waktu tertentu akan mengalami perubahan fisik maupun kerusakan, yang menyebabkan laptop tersebut harus diperbaiki. Oleh karena itu, sangat dianjurkan bagi pengguna/*user* untuk mengetahui cara merawat dan memberikan pertolongan pertama ketika laptop-nya bermasalah, sebelum memutuskan untuk

menyerahkannya ke tempat servis. Informasi yang diharapkan dapat mengatasi kerusakan yang terjadi disaat ini masih tidak lengkap, bahkan buku manual yang disertakan pun tidak dapat mengakomodasi terhadap semua kemungkinan kerusakan, oleh karena itu dirasakan perlu dibuat sebuah sistem yang dapat membantu memecahkan permasalahan kerusakan pada laptop.

Dalam hal ini sistem akan membantu pengguna dalam menemukan informasi penyebab (ciri) kerusakan berdasarkan gejala kerusakan pada setiap jenis komponen kerusakan sampai ditemukannya solusi (hasil diagnosa) berupa informasi mengenai cara perbaikannya. Tentunya sistem yang dibuat harus dapat menyajikan solusi yang tepat, masuk akal dan efisien.

Sistem pakar dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan. Sistem Pakar yang merupakan sistem yang mampu mengambil kemampuan dan pengetahuan seorang Pakar untuk mampu menyelesaikan suatu masalah. Sistem Pakar merupakan salah satu cabang *Artificial Intelligence* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang Pakar. Seorang Pakar adalah orang yang mempunyai keahlian dalam bidang tertentu yaitu Pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya [1].

Sistem Pakar adalah sistem berbasis komputer yang mampu menyamai atau meniru kemampuan seorang pakar dalam menyelesaikan suatu masalah. Sistem ini dirancang untuk meniru keahlian seorang Pakar dalam menjawab pertanyaan dan menyelesaikan suatu permasalahan baik dibidang kesehatan, bisnis, ekonomi dan sebagainya. Peran penting seorang Pakar dapat digantikan oleh program komputer yang prinsip kerjanya untuk memberikan solusi seperti yang dilakukan oleh pakar [2].

Kecerdasan buatan atau *Artificial Intelligence* dimunculkan oleh seorang profesor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada *Dartmouth Conference* yang dihadiri para peneliti kecerdasan buatan. [3]. Kecerdasan buatan berasal dari kata *artificial intelligence* yang mengandung arti tiruan dan kecerdasan. Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Ada tiga tujuan kecerdasan buatan yaitu membuat komputer cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Kecerdasan disini dimaksud dengan kemampuan untuk belajar atau mengerti dari pengalaman, mampu memahami pesan yang kontradiktif ataupun ambigu, mampu menanggapi dengan cepat dan baik atas situasi baru dan menggunakan penalaran yang baik dalam memecahkan masalah serta menyelesaikannya dengan efektif [4].

Perancangan yang digunakan dalam penelitian ini adalah perancangan UML (*Unified Modelling Language*). UML (*Unified Modeling Language*) merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML (*Unified Modeling Language*) merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [7].

UML (*Unified Modeling Language*) adalah sebuah bahasa standard untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*. UML (*Unified Modeling Language*) tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database* [8].

Terdapat beberapa diagram UML (*Unified Modeling Language*) yang sering digunakan dalam pengembangan sebuah sistem, yaitu: *usecase diagram*, *activity diagram*, dan *class diagram*.

Berdasarkan hasil penjelasan yang telah diuraikan pada latar belakang tersebut maka dibuatlah “Sistem Pakar Mendeteksi Gejala Kerusakan Pada Laptop Berbasis Web” untuk dapat

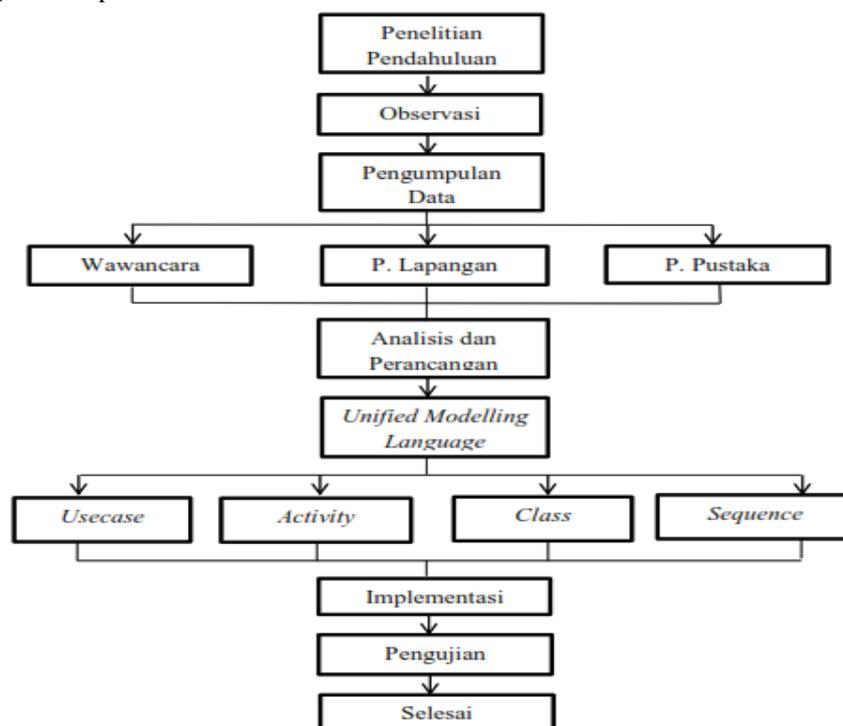
menekan waktu dan memudahkan teknisi dalam melakukan deteksi terhadap gejala kerusakan laptop dengan cepat, akurat, dan efisien, selain itu membantu pengguna laptop dalam mengatasi masalahnya sehingga kinerja penanganan servis/perbaikan laptop dapat berjalan secara optimal.

Pada penelitian terdahulu yang dilakukan oleh [5] dengan judul Penerapan Metode *Forward Chaining* Untuk Deteksi Kerusakan Pada Laptop. Tujuan dari penelitian ini adalah membangun sebuah sistem pakar dengan basis penerapan metode *Forward Chaining* yang mampu mendeteksi kerusakan dan cara perbaikannya pada laptop. Selain itu, juga memberikan kemudahan serta solusi bagi pengguna laptop untuk mengetahui masalah atau kerusakan yang terjadi. Sistem yang dibangun mudah untuk digunakan oleh semua kalangan karena tampilan serta menu-menu yang ada pada sistem mudah untuk dipahami oleh para pengguna yang akan berkonsultasi dengan sistem.

Penelitian kedua dilakukan oleh [6] dengan judul Penerapan Metode *Certainty Factor* Dalam Perancangan Sistem Pakar Mendiagnosa Kerusakan Laptop. Tujuan dari penelitian ini adalah untuk mempermudah pengguna dalam menangani kerusakan pada *hardware* laptop. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Microsoft Visual FoxPro 9.0. Teknik analisis data yang digunakan adalah *Unified Modelling Language* (UML). Penerapan metode *certainty factor* untuk menentukan nilai kepastian. Aplikasi untuk mendiagnosa kerusakan *hardware* laptop yang dibangun diharapkan dapat membantu dan mempermudah teknisi komputer maupun pengguna yang ingin mengetahui penyebab dan solusi kerusakan.

## 2. METODE PENELITIAN

### 2.1 Kerangka Tahapan Penelitian



Gambar 1. Kerangka Tahapan Penelitian

### 2.2 Metode *Certainty Factor*

Metode *certainty factor* merupakan suatu metode untuk membuktikan ketidakpastian pemikiran seorang pakar, dimana untuk mengakomodasi hal tersebut seseorang biasanya menggunakan *certainty factor* untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

*Certainty Factor* merupakan metode untuk menampung pernyataan-pernyataan dari seorang ahli dalam penyampaian. Hasil analisis mendeskripsikan tingkat kepercayaan ahli terhadap masalah yang dihadapi. *Certainty Factor* berfungsi untuk mengakomodasikan ketidakpastian pemikiran seorang pakar. *Certainty Factor* menyatakan kepercayaan dalam sebuah kejadian (fakta atau *hipotesa*) berdasarkan bukti atau penilaian pakar. *Certainty Factor* digunakan untuk menyatakan seberapa akurat, jujur, atau dapat diandalkan dalam menilai suatu predikat.[9].

*Certainty Factor* menunjukkan ukuran kepastian terhadap fakta atau peraturan. Rumus umum metode *Certainty Factor* adalah sebagai berikut:

$$CF[h,e] = MB[h,e] - MD[h,e] \quad (1)$$

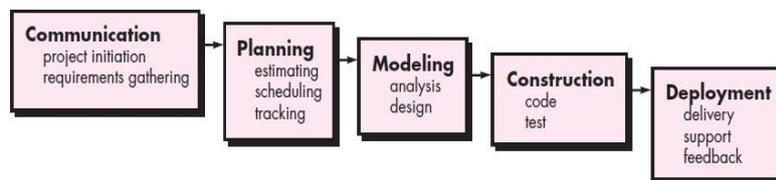
CF[h,e] = Faktor kepastian.

MB[h,e] = Ukuran kepercayaan/tingkat keyakinan Terhadap hipotesis h, jika diberikan/dipengaruhi *evidence* e (antara 0 dan 1).

MD[h,e] = Ukuran ketidakpercayaan/tingkat Ketidakyakinan Terhadap hipotesis h, jika diberikan/dipengaruhi *evidence* e (antara 0 dan 1).

### 2.3 Metode Pengembangan Perangkat Lunak

Dalam penelitian ini menggunakan metode pengembangan sistem *Waterfall*. Nama lain dari Model *Waterfall* adalah Model Air Terjun, kadang dinamakan siklus hidup klasik (*classic life cycle*), Model *waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan. Model *waterfall* ini juga dikenal dengan nama model tradisional atau model klasik. Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*Classic cycle*)". Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*) [10]. Pengembangan perangkat lunak dimulai dari analisa kebutuhan pengguna, desain sistem perangkat lunak, lalu dilanjutkan dengan pembuatan program atau penulisan kode program (*coding*), kemudian dilakukannya pengujian program dan yang terakhir penerapan program serta pemeliharaan. Gambar model pengembangan sistem *Waterfall* dapat dilihat pada dibawah ini.



Gambar 2. Model Pengembangan Waterfall

### 2.4 Proses Cara Kerja Sistem

Berikut proses cara kerja sistem menggunakan metode *certainty factor* pada sebuah kasus dengan mempresentasikan kedalam tabel.

Tabel 1. Data Kerusakan

No	Kode Kerusakan	Nama Kerusakan
1.	K001	Monitor Bermasalah
2.	K002	RAM Bermasalah
3.	K003	HDD Bermasalah
4.	K004	VGA Bermasalah
5.	K005	Sound Card Bermasalah
6.	K006	OS / Windows Bermasalah
7.	K007	Power Supply Bermasalah

8.	K008	<i>Processor</i> Bermasalah
9.	K009	<i>Keyboard</i> Bermasalah
10.	K010	<i>Mouse / Touchpad</i> Bermasalah
11.	K011	<i>Motherboard / IC Regulator</i> Bermasalah
12.	K012	<i>Charger</i> Bermasalah
13.	K013	Kipas Bermasalah / Kotor
14.	K014	Baterai Laptop Bermasalah

Tabel 2. Data Gejala Kerusakan

No	Kode Gejala	Nama Gejala Kerusakan
1.	G001	Sering mati sendiri setiap beberapa Waktu
2.	G002	Tampil <i>bluesreen</i> pada OS Windows (macam-macam pesan error)
3.	G003	Bunyi suara beep-beep
4.	G004	PC atau laptop hidup, tapi tidak tampil di layar atau monitor
5.	G005	Masuk Windows, tapi layar <i>blank</i> hitam
6.	G006	Tombol keyboard sebagian atau seluruh tidak berfungsi
7.	G007	Mati total / tidak bisa dihidupkan
8.	G008	Berhenti beroperasi atau hang
9.	G009	PC atau laptop lambat atau berat
10.	G010	PC atau laptop hidup tapi tidak mau masuk Windows atau gagal booting
11.	G011	Mengalami proses <i>loading</i> yang lama saat dihidupkan
12.	G012	Sering <i>restart</i> sendiri
13.	G013	Tidak ada tampilan awal BIOS
14.	G014	Terdengar suara aneh pada <i>Harddisk</i> atau HDD
15.	G015	Selalu <i>Scandisk</i> ketika <i>booting</i>
16.	G016	Muncul pesan error pada BIOS (macam- macam pesan error)
17.	G017	Tidak ada tanda-tanda dari sebagian/seluruh perangkat bekerja pada CPU
18.	G018	Kinerja grafis terasa sangat berat (biasanya dalam game atau manipulasi gambar)
19.	G019	Hanya sebagian perangkat yang bekerja
20.	G020	<i>Pointer mouse</i> tidak merespon gerakan <i>Mouse</i>
21.	G021	Tampak blok hitam, dan gambar tidak simetris / acak
22.	G022	Keluar bunyi beep panjang pada saat laptop dinyalakan
23.	G023	Dihidupkan agak sulit
24.	G024	Baterai tidak mau di-charge
25.	G025	Tidak ada indikasi masuk <i>power</i>
26.	G026	Laptop di-charge posisi hidup kemudian tiba-tiba mati layar
27.	G027	Terdapat garis horizontal / vertikal di tengah monitor
28.	G028	<i>Device driver</i> informasi tidak terdeteksi dalam <i>device manager</i>
29.	G029	<i>Sound</i> atau suara tidak keluar
30.	G030	Muncul pesan <i>error</i> saat menjalankan aplikasi audio
31.	G031	Belum sampai Windows sudah <i>restart</i> Lagi
32.	G032	Muncul pesan <i>error</i> pada Windows/OS
33.	G033	PC atau laptop sangat panas serta kotor atau banyak debu
34.	G034	Kipas tidak berfungsi

35.	G035	Baterai di-charge tidak bisa penuh 100%
36.	G036	Indikator baterai menampilkan silang (x) merah
37.	G037	Tombol <i>keyboard</i> lengket dan terpicet sendiri
38.	G038	Laptop bunyi beep panjang 1x dan beep pendek 2x
39.	G039	Laptop bunyi beep panjang 1x dan beep pendek 3x

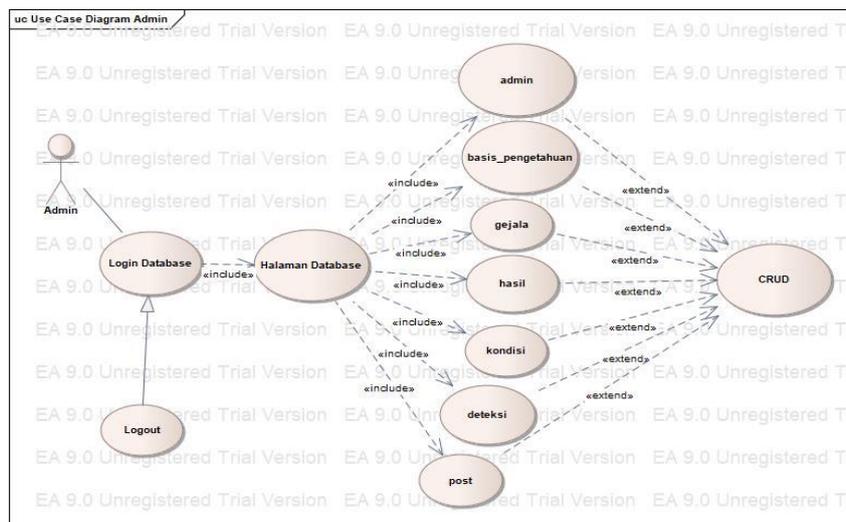
Tabel 3. Tabel Aturan

No	Aturan (Rule)	Kendala
1.	Aturan 1 (R1)	If G004 and G021 and G027 then K001
2.	Aturan 2 (R2)	If G002 and G003 and G004 and G009 and G012 and G013 and G016 and G022 then K002
3.	Aturan 3 (R3)	If G008 and G009 and G010 and G011 and G014 and G015 and G031 then K003
4.	Aturan 4 (R4)	If G002 and G003 and G004 and G013 and G018 and G028 and G038 then K004
5.	Aturan 5 (R5)	If G028 and G029 and G030 then K005
6.	Aturan 6 (R6)	If G005 and G009 and G010 and G012 and G032 then K006
7.	Aturan 7 (R7)	If G001 and G007 and G017 and G019 and G023 and G025 then K007
8.	Aturan 8 (R8)	If G004 and G013 and G016 then K008
9.	Aturan 9 (R9)	If G003 and G006 and G022 and G028 and G037 and G039 then K009
10.	Aturan 10 (R10)	If G020 and G028 then K010
11.	Aturan 11 (R11)	If G007 and G023 then K011
12.	Aturan 12 (R12)	If G024 and G025 and G026 then K012
13.	Aturan 13 (R13)	If G001 and G012 and G033 and G034 then K013
14.	Aturan 14 (R14)	If G024 and G035 and G036 then K014

## 2.5 Perancangan Proses

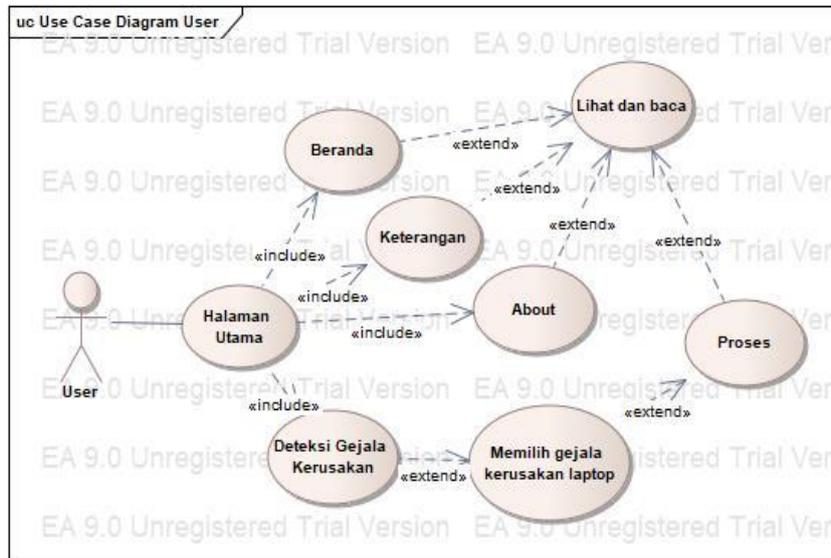
### a. Use Case Diagram

*Use Case* merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. *Use case* digunakan untuk mengetahui fungsi dari apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



Gambar 3. Use Case Diagram Admin

Pada Gambar 3, *admin* melakukan *login* terhadap sistem *database* dengan memasukan *username* dan *password*. Maka sistem akan menampilkan halaman *dashboard* untuk dapat melakukan CRUD (*Create, Read, Update, Delete*) pada *database*.

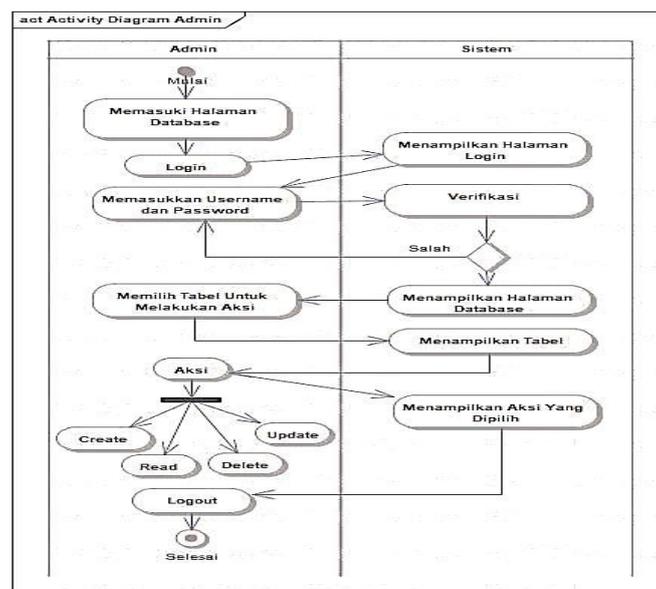


Gambar 4. Use Case Diagram User

Pada Gambar 4, *user* masuk ke halaman utama di *browser* lalu sistem akan menampilkan menu Beranda, Keterangan, About, Deteksi Kerusakan. Pada menu Deteksi Gejala Kerusakan, *user* akan diminta untuk memilih gejala-gejala apa saja yang sedang dihadapi oleh laptopnya. Setelah memilih gejala, sistem akan menampilkan hasil deteksi dari pemilihan gejala tersebut. Sedangkan pada menu yang lainnya, *user* hanya dapat membaca dan melihat.

b. Activity Diagram

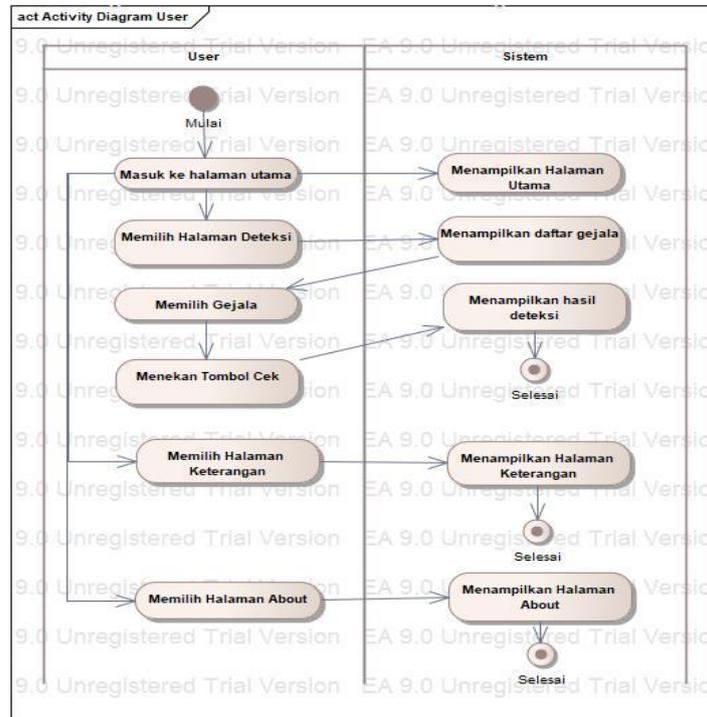
*Activity diagram* merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan. *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.



Gambar 5. Activity Diagram Admin

Pada Gambar 5 proses melakukan *login*, terdapat tahap proses yang dilakukan admin

terhadap sistem yaitu memasukkan *username* dan *password* yang menampilkan halaman *dashboard database*. Pada halaman *dashboard* tersebut, admin dapat melakukan aktivitas seperti menambah, mengedit dan dan menghapus data (CRUD).

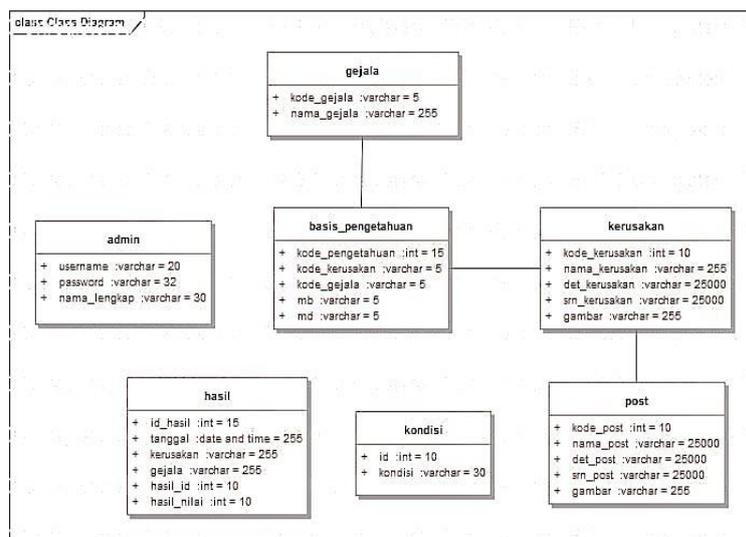


Gambar 6. Activity Diagram User

Pada Gambar 6 merupakan *activity* yang dilakukan *user* adalah ketika *user* memasuki halaman utama, *user* akan memilih gejala kerusakan lalu mengklik cek. Maka sistem akan menampilkan halaman hasil perhitungan dari pemilihan gejala tersebut. Kemudian *user* juga dapat melihat dan membaca menu beranda, keterangan dan halaman *about*.

c. Class Diagram

*Class Diagram* merupakan gambaran struktur dan deskripsi dari *class*, *package* dan objek yang saling berhubungan seperti diantaranya pewarisan, asosisasi dan lainnya.

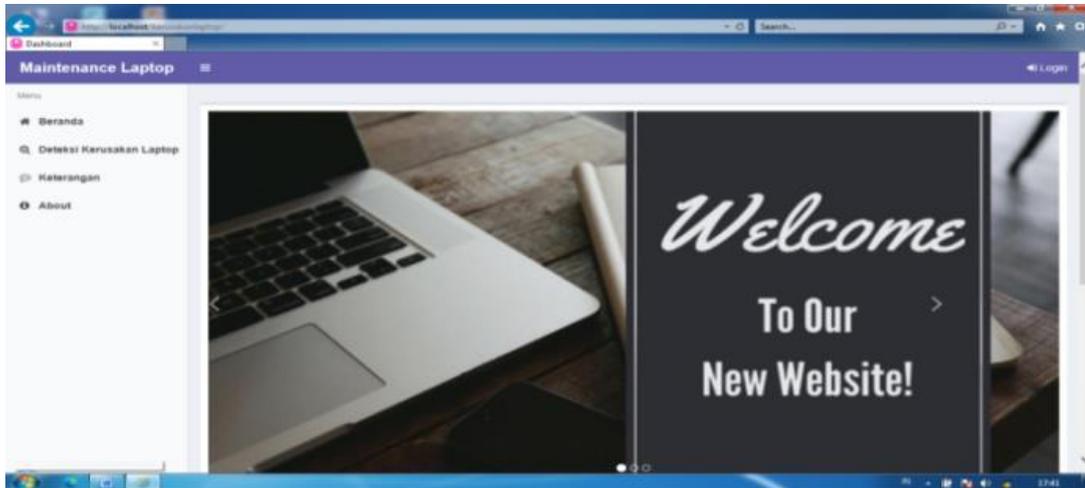


Gambar 7. Class Diagram

Pada gambar 7 *class* diagram terdapat tabel gejala, tabel admin, basis pengetahuan, hasil, kondisi, kerusakan dan post yang saling berhubungan.

### 3. HASIL DAN PEMBAHASAN

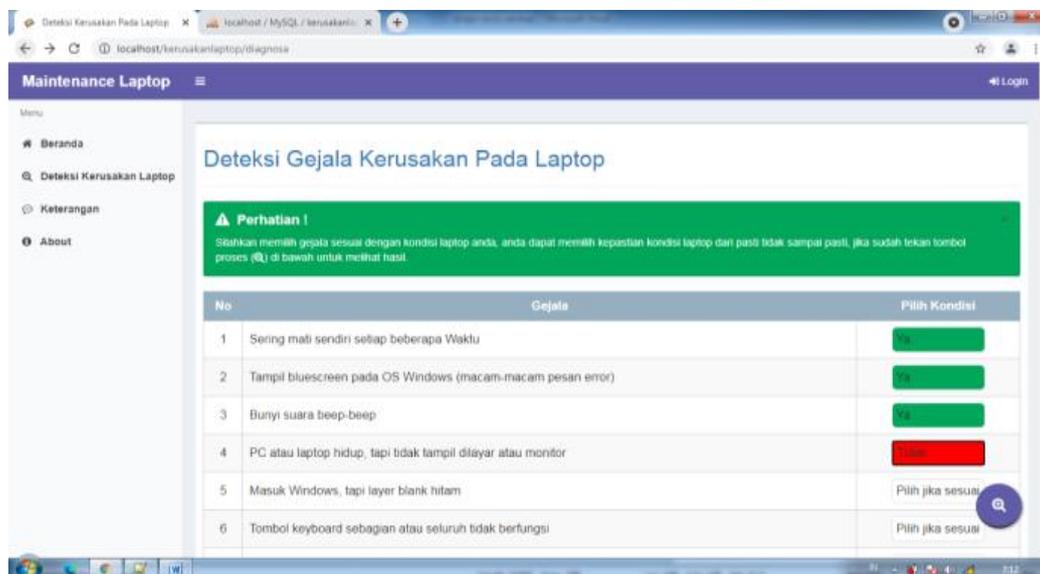
#### 3.1 Halaman Utama/Beranda



Gambar 8. Halaman Utama

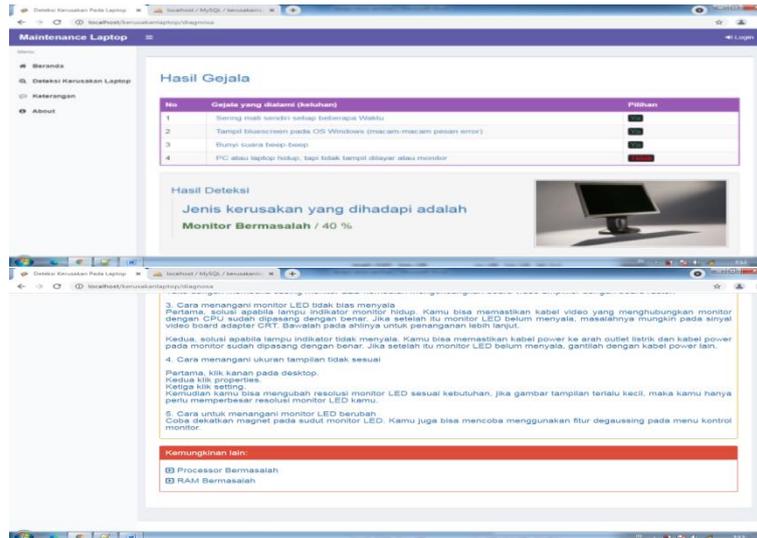
Pada Gambar 8 merupakan halaman yang terdapat menu beranda, deteksi kerusakan laptop, keterangan dan tentang. Pada halaman ini juga terdapat gambar slide untuk memperindah *background* serta *button login* untuk pakar.

#### 3.2 Halaman Deteksi Kerusakan Laptop



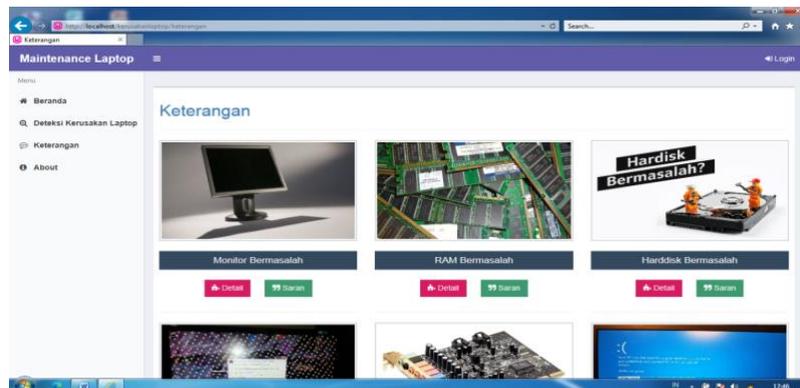
Gambar 9. Halaman Deteksi Kerusakan

Pada Gambar 9 merupakan halaman yang terdapat tabel untuk deteksi kerusakan. Didalam tabel terdapat kolom no, gejala dan pilih kondisi. *User* akan diminta untuk memilih gejala apa saja yang sedang dialami laptop kemudian memilih kondisi antara ya, ragu-ragu, tidak tahu dan tidak. Kemudian *user* diminta menekan tombol pencarian dipojok kanan bawah untuk memproses gejala yang dipilih.



Gambar 10. Halaman Hasil Deteksi

### 3.3 Halaman Keterangan



Gambar 11. Halaman Keterangan

Pada Gambar 11 merupakan halaman yang terdapat berbagai macam kerusakan. Terdapat 2 *button*, yang pertama *button* detail untuk melihat detail dari kerusakan, serta *button* saran untuk melihat saran penanganannya.

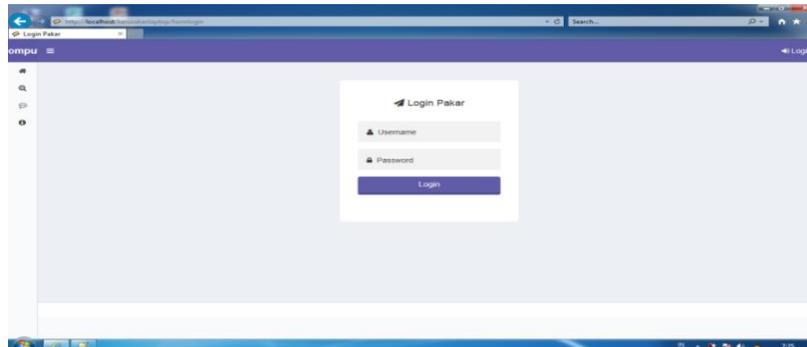
### 3.4 Halaman About



Gambar 12. Halaman About

Pada Gambar 12 merupakan halaman *about* yang terdapat informasi tentang sistem dan para pakar.

### 3.5 Halaman Login



Gambar 13. Halaman Login

Pada Gambar 13 merupakan halaman yang hanya dilakukan oleh admin. Admin akan diminta untuk mengisi kolom yang harus diisi, kemudian menekan tombol *login* dibawahnya. Jika berhasil, maka akan menampilkan *dashboard* admin.

### 3.6 Pengujian

Setelah dilakukan pengujian sistem kepada user. Maka didapatkan hasil pengujian seperti pada tabel dibawah ini

Tabel 4. Tabel Pengujian

No	Nama	Pekerjaan	Hasil	Keterangan
1	Gusrahayu	Mahasiswa	Berhasil	Sistem dapat dipahami dan dijalankan sesuai dengan yang diharapkan.
2	Maharani	Siswa	Berhasil	Sistem dapat dipahami dan dijalankan sesuai dengan yang diharapkan.
3	Elsa putri	Siswa	Berhasil	Sistem dapat dipahami dan dijalankan sesuai dengan yang diharapkan.
4	Erlina	Mahasiswa	Berhasil	Sistem dapat dipahami dan dijalankan sesuai dengan yang diharapkan.
5	Agnes	Mahasiswa	Berhasil	Sistem dapat dipahami dan dijalankan sesuai dengan yang diharapkan.

## 4. KESIMPULAN DAN SARAN

### 4.1. Kesimpulan

Berdasarkan penelitian yang dilakukan, penulis dapat menyimpulkan bahwa sistem pakar mendeteksi gejala kerusakan pada laptop berbasis web dapat diimplementasikan dengan baik dan dapat memberikan solusi cara mengatasi kerusakan.

Sistem pakar mendeteksi kerusakan laptop dapat membantu untuk mengetahui penyebab kerusakan pada laptop dan mampu mendeteksi gejala kerusakan laptop yang sedang terjadi serta memberikan solusi untuk mengatasinya, namun sistem pakar ini tidak dapat menggantikan seorang pakar atau teknisi karena tidak semua kerusakan dapat diperbaiki sendiri dan memerlukan seorang teknisi yang ahli untuk memperbaiki kerusakan berat yang memerlukan keahlian khusus.

### 4.2. Saran

Berdasarkan penelitian yang penulis lakukan, web ini dapat dikembangkan dan bisa diharapkan untuk:

- Adanya Menu *login* untuk *user* agar *user* dapat menyimpan data yang dimasukkan atau saling berinteraksi di dalam sebuah sistem *web*.
- Aplikasi dapat dikembangkan dalam bentuk versi *online*.

## DAFTAR PUSTAKA

- [1] N. A. Hasibuan, H. Sunandar, S. Alas, and S. Sugiman, "Sistem Pakar Mendiagnosa Penyakit Kaki Gajah Menggunakan Metode Certainty Factor," *JURASIK*, vol. 2, no. 1, pp. 29–39, 2017.
- [2] H. T. Sihotang, "Sistem Pakar Mendiagnosa Penyakit Kolesterol Pada Remaja Dengan Metode Certainty Factor (CF) Berbasis Web," *Jurnal Mantik Penusa*, vol. 15, no. 1, pp. 16–23, 2014.
- [3] N. Yona Sidratul Munti, "Perancangan Sistem Pakar Diagnosa Penyakit Lupus Eritmatosus Sistem(Les) Dengan Metode Forward Chaining Menggunakan Pemrograman Php dan Mysql," *Jurnal Fasilkom*, vol. 9, no. 2, pp. 407–428, 2019.
- [4] M. Septiani And S. J. Kuryanti, "Sistem Pakar Untuk Mendiagnosa Penyakit Saluran Pernapasan Pada Anak (Studi Kasus: RSAB Harapan Kita Jakarta)," vol. 2, no. 2, pp. 23–27, 2018.
- [5] Y. Yunita, "Penerapan Metode Forward Chaining Untuk Deteksi Kerusakan Pada Laptop," *Jurnal Techno Nusa Mandiri*, vol. 11, no. 1, pp. 1–10, 2014.
- [6] Yanto, Hendry and A. Ifbro, "Penerapan Metode Certainty Factor Dalam Perancangan Sistem Pakar Pendiagnosa Kerusakan Laptop," *Jurnal InTekSis*, vol. 3, no. 2, pp. 30–38, 2016.
- [7] M. T. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 3, no. 1, pp. 126–129, 2018.
- [8] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek Abdul Mubarak," *JIKO: Jurnal Informatika dan Komputer*, vol. 2, no. 1, pp. 19–25, 2019.
- [9] A. R. Fahindra, I. H. A. Amin, "Sistem Pakar Deteksi Awal Covid-19 Menggunakan Metode Certainty Factor," *Jurnal Teknokompak*, vol. 15, no. 1, pp. 92–103, 2019.
- [10] M. Susilo, R. Kurniati and Kasmawi, "Rancang Bangun Website Toko Online Menggunakan Metode Waterfall," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 2, no. 2, pp. 98-105, 2018.