

# Sistem Pakar Diagnosa Kerusakan Laptop Menggunakan Metode *Naïve Bayes-Certainty Factor* Berbasis Website

Novitasari<sup>1)</sup>, Ria Arafiah<sup>2)</sup>, Mulyono<sup>3)</sup>

<sup>123</sup>*Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Jakarta  
Jalan Raya Rawamangun Muka, Jakarta Timur DKI Jakarta 13220*

Email: <sup>a)</sup>sari.novita9911@gmail.com, <sup>b)</sup>riaarafiah@unj.ac.id, <sup>c)</sup>Mulyono@unj.ac.id

## Abstract

Laptop damage is a condition where the laptop malfunctions caused by damaged components in the laptop. According to the ICT Survey and Its Implications for Socio-Cultural and Economic Aspects of Society in Indonesia, 41.55% of Indonesians are computer/laptop users. Seeing that this number is quite large and not everyone can diagnose damage and handle laptop damage, this research is here to build an expert system for diagnosing laptop damage. This study discusses the development of a laptop-based expert system for diagnosing damage *website* as an effort to provide laptop damage diagnostic services along with suggestions or solutions that can be repair the damage experienced by the *user's* laptop. This expert system was developed using the method *Naïve Bayes* and *Certainty factor*. The input given to the system is in the form of symptoms of damage experienced, along with its intensity. Method *naïve bayes* used to find the highest probability of the diagnosis. The results of this diagnosis are then processed using *certainty factor* to determine the level of confidence in the results of the diagnosis. This expert system comes in the form *website* built using *framework* CodeIgniter and MySQL. Based on the results of system testing using 40 expert data, it shows that the expert system for diagnosing laptop damage uses the method *Naïve Bayes-Certainty factor* shows the suitability of the results of system diagnostics with expert diagnoses that is equal to 85%.

**Keywords:** *Expert System, Laptop Damage, Diagnosis, Naïve Bayes, Certainty factor*

## Abstrak

Kerusakan laptop merupakan kondisi dimana laptop mengalami malfungsi yang diakibatkan oleh rusaknya komponen yang ada di dalam laptop. Menurut Survei TIK Serta Implikasinya Terhadap Aspek Sosial Budaya dan Ekonomi Masyarakat di Indonesia tercatat 41,55% dari masyarakat Indonesia merupakan pengguna komputer/laptop. Melihat angka tersebut yang cukup besar dan tidak semua orang dapat mendiagnosa kerusakan serta penanganan pada kerusakan laptop, maka penelitian ini hadir untuk membangun sistem pakar diagnosa kerusakan laptop. Sistem pakar ini dikembangkan menggunakan metode *Naïve Bayes* dan *Certainty factor*. Masukkan yang diberikan kepada sistem yakni berupa gejala kerusakan yang dialami beserta dengan intensitasnya. Metode *naïve bayes* digunakan untuk mencari probabilitas tertinggi dari hasil diagnosa. Hasil diagnosa ini selanjutnya diproses menggunakan *certainty factor* untuk mengetahui tingkat keyakinan terhadap hasil diagnosa tersebut. Sistem pakar ini hadir dalam bentuk *website* yang dibangun menggunakan *framework* CodeIgniter dan MySQL. Berdasarkan hasil pengujian sistem menggunakan 40 data pakar menunjukkan bahwa sistem pakar diagnosa kerusakan laptop menggunakan metode *Naïve Bayes-Certainty factor* menunjukkan adanya kekesuaian hasil diagnosa sistem dengan diagnosa pakar yakni sebesar 85%.

**Kata-kata kunci:** *Sistem pakar, kerusakan laptop, diagnosis, Naïve Bayes, Certainty factor.*

## PENDAHULUAN

Komputer atau laptop menjadi salah satu alat yang sangat diperlukan dalam era globalisasi saat ini. Hal ini dikarenakan komputer memiliki peran penting dalam membantu memudahkan kehidupan sehari-hari seperti halnya sekolah dan bekerja. Selain itu, memasuki masa pandemi covid yang dimulai dari tahun 2020 hingga 2022 menyebabkan berbagai kegiatan harus dilakukan secara daring dan tentunya membutuhkan komputer. Komputer saat ini merupakan sekumpulan sistem yang dapat membantu manusia menyelesaikan permasalahan dan mampu mengolah data menjadi informasi, serta menjadi tempat penyimpanan besar yang dapat dengan mudah diakses.

Menurut Survei TIK Serta Implikasinya Terhadap Aspek Sosial Budaya dan Ekonomi Masyarakat di Indonesia yang dilakukan oleh Pusat Penelitian dan pengembangan Aplikasi Informatika dan Informasi Komunikasi Publik pada tahun 2019 dengan jumlah responden sebanyak 5.091 orang menunjukkan bahwa ada 41,55% masyarakat Indonesia yang merupakan pengguna komputer/laptop (Kemenkominfo, Implikasi Penggunaan TIK, 2019).

Dikarenakan tidak semua orang dapat mendiagnosa kerusakan laptop dibutuhkan sebuah sistem pakar untuk membantu orang mengidentifikasi kerusakan laptop beserta dengan cara penanganannya.

Pada pembangunan sistem pakar untuk diagnosa kerusakan laptop ini ada beberapa metode yang umum digunakan yakni Forward Chaining, Backward Chaining dan Dempster Shafer. Selain itu, metode lain yang sudah pernah digunakan untuk mendiagnosa kerusakan laptop yaitu Naïve Bayes dan *Certainty factor*.

Metode Naïve Bayes dan *Certainty factor* merupakan metode yang akan penulis gunakan dalam membangun sistem pakar diagnosa kerusakan komputer. Hal ini dikarenakan naïve bayes ialah teknik pengklasifikasian menggunakan metode probabilitas dan statistik yang dapat memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. Algoritma ini tidak memerlukan jumlah data yang banyak untuk menentukan estimasi yang diperlukan pada proses pengklasifikasian. Selain itu, naïve bayes juga memiliki kelebihan yakni algoritmanya sederhana, cepat dan berakurasi tinggi (Hamzah, n.d., 2020). Namun, naïve bayes juga memiliki kelemahan seperti ketidakpastian dalam prediksi dan asumsi independensi fitur (Ashari Muin, 2016). Kelemahan tersebut dapat diperbaiki dengan penggunaan *certainty factor*. *Certainty factor* sendiri ialah metode yang digunakan untuk mengukur kepastian terhadap suatu fakta atau aturan (Setyaputri et al., n.d., 2018). *Certainty factor* ini merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. Para pakar atau ahli biasanya menganalisis informasi yang ada dengan kata mungkin, kemungkinan besar, hampir pasti, dan lain sebagainya. Maka dari itu, *certainty factor* digunakan untuk mengakomodasi hal tersebut dengan menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Kedua metode ini dipilih karena laptop sendiri merupakan alat yang sangat kompleks seperti tubuh manusia di mana dibeberepa jenis kerusakan yang berbeda namun memiliki beberapa kesamaan gejala. Hal ini nantinya akan di atasi oleh klasifikasi naïve bayes. Selain itu, untuk menentukan tingkat keyakinan terhadap hasil klasifikasi naïve bayes tadi ditambah dengan interpretasi dari *user* pada *certainty factor* menjadikan kedua metode ini tepat untuk digunakan pada kasus ini.

Sistem pakar yang dibangun oleh penulis yakni sistem pakar dengan media *website*. Hal ini dikarenakan *website* merupakan salah satu media yang dapat dengan mudah diakses selama perangkat terhubung ke jaringan internet.

## KAJIAN PUSTAKA

### Sistem Pakar

Sistem pakar (expert system) merupakan sebuah aplikasi komputer yang digunakan untuk membantu mengambil keputusan atau memecahkan persoalan dalam bidang yang spesifik yang bekerja seperti seorang pakar atau ahli. Cara kerja sistem ini yakni menggunakan pengetahuan dan metode analisis yang terlebih dahulu didefinisikan sesuai dengan bidang keahliannya. Sehubungan dengan cara kerja sistem ini yang menirukan seorang pakar, maka sistem ini harus memuat

pengetahuan, pengalaman, dan pemecahan suatu permasalahan layaknya seorang pakar (B. Herawan Hayadi, 2018). Sistem pakar memiliki beberapa komponen penting yakni *user interface*, basis pengetahuan (*knowledge base*), basis data atau memori kerja (*working memory*), mesin inferensi, akuisisi pengetahuan, fasilitas penjelasan, *workplace*, perbaikan pengetahuan, dan *knowledge engineer*. Selain itu, pada sistem pakar juga terdapat representasi pengetahuan yang merupakan cara atau metode untuk mewakili atau mengkodekan pengetahuan dalam sebuah sistem pakar berbasis pengetahuan. Adapun tujuan dari representasi ini yakni untuk menangkap sifat-sifat penting dari sebuah masalah dan membuat informasi tersebut dapat di akses oleh prosedur pemecahan masalah (kusrini 2006).

### Naïve Bayes dan *Certainty factor*

*Naïve Bayes* merupakan salah satu metode yang digunakan dalam data mining yang didasarkan pada teorema bayes. Teorema bayes yaitu memprediksi peluang terjadinya sebuah kejadian di masa depan berdasarkan kejadian yang terjadi di masa lampau. Teorema ini dikombinasikan dengan “naive” yang mana kondisi antaratribut diasumsikan saling bebas. Maka dari itu, hanya variasi dari variabel untuk masing-masing kelas harus ditentukan, bukan seluruh matriks kovarians (Zhang, 2004). Rumus untuk naïve bayes *classifier* dapat dilihat pada persamaan 1.

$$P(a_i|v_j) = \frac{n_c + m_p}{n + m} \quad (1)$$

Keterangan:

$n_c$  : jumlah record pada data  $v = v_j$  dan  $a = a_i$

$m_p$  :  $1/x$ ,  $x$  adalah banyaknya jenis kerusakan

$m$  : total gejala/jumlah parameter

$n$  : jumlah record pada data di setiap kelas

Persamaan tersebut dapat diselesaikan melalui beberapa proses tahapan sebagai berikut.

- 1) Menentukan nilai  $N$ ,  $m$ ,  $x$ ,  $n_c$  setiap class dan  $P(v_j)$ .
- 2) Menghitung nilai  $P(a_i|v_j)$ .
- 3) Menghitung  $P(a_i|v_j) \times P(v_j)$  untuk tiap  $v$ .
- 4) Memutuskan klasifikasi berdasarkan hasil perkalian dari  $v$  yang memiliki nilai terbesar.

*Certainty factor* adalah salah satu metode yang digunakan dalam sistem pakar untuk menyatakan derajat keyakinan dengan hasil akhirnya berupa persentase dari kepastian. Pada suatu permasalahan seorang pakar yang biasa menganalisis informasi yang ada dengan kata mungkin, kemungkinan besar, hampir pasti, dan lain sebagainya. Metode ini mengakomodasi hal tersebut dengan menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

*Certainty factor* diperkenalkan pada tahun 1970-an oleh Shortliffe Buchanan yang digunakan untuk pembuatan MYCIN yang merupakan program sistem pakar yang dapat melakukan diagnosis penyakit meningitis dan penyakit bacteremia yang bertugas untuk menentukan kemungkinan identitas patogen pada pasien infeksi dan untuk membantu pemilihan rejimen terapi yang sesuai dalam mengobati organisme yang sedang dipertimbangkan (Shortliffe & Buchanan, 1984). Adapun untuk pengukurannya yakni menggunakan kemungkinan, keyakinan (*belief*) dan ketidakpercayaan (*disbelief*). Rumus untuk *certainty factor* dapat dilihat pada persamaan 2.

$$CF(h,e) = MB(h,e) - MD(h,e) \quad (2)$$

Keterangan:

$CF(h,e)$  : *Certainty factor* dari hipotesis  $h$  yang dipengaruhi oleh bukti  $e$ .

$MB(h,e)$  : ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis  $H$  yang dipengaruhi bukti  $e$ .

$MD(h,e)$  : ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis  $H$  yang dipengaruhi bukti  $e$ .

Pada sebuah sistem berbasis pengetahuan yang didalamnya terdapat beberapa aturan yang saling terkait yang mana masing-masing membuat kesimpulan yang sama namun dengan faktor kepastian yang berbeda-beda, maka dari itu setiap aturan dapat dipandang sebagai bagian dari bukti pendukung kesimpulan bersama. Untuk menghitung nilai atau kesimpulan dari faktor kepastian itu sendiri

diperlukan penggabungan bukti. Dalam probabilitistik, ketika kita menggabungkan dua probabilitas dependen (join probabilitas) yang dapat dituliskan pada persamaan 3 (Turban et al., 2007).

$$CF(R1, R2) = [CF(R1)] \times [CF(R2)] \quad (3)$$

Keterangan:

CF(R1,R2) : CF gabungan 2 probabilitas

CF(R1) : CF probabilitas 1

CF(R2) : CF probabilitas 2

Setelah penggabungan bukti, dilakukan penghitungan kombinasi untuk setiap bukti yang didapatkan. Fungsi kombinasi tersebut dapat dilihat pada persamaan 4.

$$MB[h, s_1 \& s_2] = \begin{cases} 0 & \text{if } MD[h, s_1 \& s_2] = 1 \\ MB[h, s_1] + MB[h, s_2](1 - MB[h, s_1]) & \text{sebaliknya} \end{cases} \quad (4)$$

Keterangan:

MB[h,s1&s2 ] : nilai kepercayaan terhadap hipotesis berdasarkan bukti 1 dan bukti 2

MD[h,s1&s2 ] : nilai ketidakpercayaan terhadap hipotesis berdasarkan bukti

Hasil dari *Certainty factor* dapat disajikan dengan bentuk persentase dengan mengubah nilai CF ke persentase dapat dilakukan dengan mengalikan dengan nilai seratus seperti pada persamaan 4.

$$CFprecentage = CF \times 100\% \quad (4)$$

### Website

*Website* pertama kali dibangun oleh Berners-Lee pada tahun 1991. *Website* merupakan kumpulan halaman-halaman yang menampilkan berbagai macam informasi dalam bentuk data, teks, gambar diam atau bergerak, suara, data animasi, video atau gabungan dari semuanya. Halaman-halaman ini dapat bersifat statis ataupun dinamis yang membentuk suatu rangkaian yang saling berkaitan yang masing-masing halamannya dihubungkan dengan jaringan halaman (*hyperlink*). *Website* ini terangkum dalam sebuah domain atau subdomain atau lebih tepatnya berada di WWW (*World Wide Web*) yang terdapat didalam internet (Nurhadi, 2017).

### UML

Dalam perancangam sebuah *website* tentunya diperlukan sebuah konsep atau model, adapun salah satu model yang banyak digunakan untuk pengembangan sebuah *website* adalah Unified Modelling Language (UML). Bahasa permodelan ini diadopsi oleh Object management Group (OMG) pada tahun 1997 sebagai standar bahasa pemrograman. UML yakni seperti pembuatan cetak biru (*blue print*) yang berguna untuk memvisualisasikan, menspesifikasikan, dan menjadi dokumentasi dari sebuah software. Contoh diagram UML yakni *use case diagram*, *activity diagram*, dan lain sebagainya.

### Pengujian

Pengujian sistem ini dilakukan untuk mengetahui apakah sistem berjalan dengan baik atau tidak. Adapun metode yang digunakan untuk pengujian sistem ini yakni blackbox testing. Blackbox testing adalah pengujian yang dilakukan dengan melihat hasil eksekusi melalui pengujian pada data dan validasi fungsi perangkat lunak. Pengujian ini difokuskan pada apa yang diamati terutama pada fungsi dan output. Fokus pengujian ini sebagian besar terdapat pada perangkat lunak yang diciptakan dan *user experience*.

Selain pengujian sistem, ada juga penguian akurasi yang bertujuan untuk menunjukkan bahwa sistem yang dibuat sudah berjalan sesuai dengan kebutuhan. Pengujian ini dilakukan dengan mencari nilai akurasi pada sistem. Nilai ini digunakan untuk melihat kesesuaian antara hasil dari sistem dengan hasil diagnosa pakar, hal ini bertujuan untuk menunjukkan bahwa sistem yang dikembangkan ini dapat mengenali input yang diberikan oleh *user*. Perhitungan nilai akurasi dapat dilihat pada persamaan 5 (Maulana, et al., 2019).

$$Akurasi = \frac{\text{Jumlah data benar}}{\text{Jumlah seluruh data}} \times 100\% \quad (5)$$

## IMPLEMENTASI

### Data Kerusakan dan Gejala

Data yang digunakan dalam penelitian ini yakni data sekunder yang didapatkan dari jurnal-jurnal terkait dan divalidasi oleh 2 orang pakar yakni bapak Robi Adi Ramdhani, A.md.T dan bapak Budi Subagja, A.md.T. Data yang sudah divalidasi oleh pakar ,selanjutnya dimasukkan kedalam basis pengetahuan atau *knowledge base*. Basis pengetahuan ini merupakan pengetahuan pakar secara spesifik yang diperoleh melalui akuisisi pengetahuan. Dalam hal ini penulis membuat basis pengetahuan yang didalamnya terdiri dari daftar kerusakan serta gejala-gejala dari kerusakan laptop dengan memberikan kode-kode pada keduanya. Kode kerusakan dapat dilihat pada Tabel 1.

**TABEL 1.** Kode dan Nama kerusakan

Kode Kerusakan	Nama Kerusakan
K1	LCD Rusak
K2	RAM ( <i>Random Access Memory</i> ) Rusak
K3	HDD ( <i>Hard Disk Drive</i> ) Rusak
K4	VGA ( <i>Video Graphic Accelerator</i> ) Rusak
K5	Sound Card Rusak
K6	OS Bermasalah
K7	Aplikasi Rusak
K8	Baterai dan CMOS Bermasalah
K9	Processor Bermasalah
K10	Kekurangan Memori
K11	Kekurangan Memori VGA
K12	Clock Processor Kurang Tinggi
K13	Adapter Charger Bermasalah
K14	Kekurangan Daya Baterai
K15	Perangkat USB Rusak
K16	Keyboard Rusak
K17	Touchpad Rusak
K18	Motherboard Rusak
K19	Kabel Power Rusak
K20	Kabel Sata/Ide Rusak
K21	CD/DVD/ROM/RW Rusak
K22	Bios Error

Selain kerusakan, terdapat gejala-gejala kerusakan. Kode dari setiap gejala dapat dilihat pada Tabel 2.

**TABEL 2.** Kode dan Nama Gejala

Kode Gejala	Nama Gejala
G1	Tombol hidup tapi tidak ada gambar yang ditampilkan terpantau
G2	Terdapat garis horizontal / vertikal di monitor
G3	Tidak ada tampilan awal bios
G4	Muncul pesan Error pada bios (pesan selalu berbeda tergantung kondisi tertentu)
G5	Alarm bios berbunyi
G6	Terdengarsuara aneh di HDD
G7	Sering hang/crash saat menjalankan aplikasi
G8	Selalu pindai disk saat boot

G9	Muncul pesan error saat menjalankan aplikasi
G10	Informasi driver perangkat tidak terdeteksi di manajer perangkat, meskipun driver telah diinstal
G11	Tiba-tiba tiba OS restart otomatis
G12	Keluarnya blue screen pada OS (pesan selalu berbeda tergantung kondisi tertentu)
G13	Suara tetap keluar meskipun driver perangkat dan pengaturan telah dilakukan sesuai dengan instruksi
G14	Muncul pesan error saat menjalankan aplikasi audio
G15	Muncul pesan kesalahan saat pertama kali OS dimuat dari HDD
G16	Daya tidak tersimpan
G17	Tidak bisa connect internet di browser
G18	Muncul pesan di windows, bahwa windows virtual memory kekurangan
G19	Aplikasi berjalan dengan lambat, respons yang lambat terhadap input
G20	Performa grafis sangat berat (biasanya di internet terbuka dan manipulasi gambar)
G21	Perangkat tidak terdeteksi di bios
G22	Deteksi informasi salah di bios
G23	Muncul peringatan batre saat proses booting
G24	Sebagain / seluruh karakter input mati
G25	Pointer touchpad tidak merespon
G26	Sebagain / seluruh karakter input mati
G27	Keluar bip panjang saat laptop dihidupkan
G28	Dihidupkan agak susah
G29	Kabel power mengalami goresan
G30	Tidak ada indikasi daya masuk
G31	Mati total
G32	Posisi komputer hidup dan kemudian tiba-tiba layar mati
G33	Bunyi bip keluar berulang kali
G34	Lupakan sampai sistem operasi telah di-restart lagi
G35	Tanggal dan jam selalu ter-reset saat restart dan dimatikan
G36	Indikator power nyala tapi tidak display
G37	CD / DVD / ROM / RW tidak terdeteksi di OS
G38	CD / DVD / ROM / RW drive yang tidak bisa keluar CD
G39	Pesan "Kesalahan checksum BIOS ROM - Sistem dihentikan"
G40	Sistem Operasi Tidak Muncul
G41	Saat di tekan tombol power tidak menyala.

Setelah itu, ada relasi yang terjadi antara tabel 2 dan 3. Relasi tersebut dijadikan sebagai aturan atau *rule* untuk pengelompokan gejala dari setiap kerusakan. Relasi tersebut dapat dilihat pada Tabel 3.

**TABEL 3.** Relasi antara Kerusakan dan Gejala

<b>Kode</b>	<b>Rule Kerusakan</b>
R1	<b>IF G01 OR G02 OR G26 THEN K1</b>

R2	<b>IF G03 OR G04 OR G05 OR G11 OR G12 OR G33 THEN K2</b>
R3	<b>IF G06 OR G07 OR G08 OR G10 OR G21 OR G22 OR G34 THEN K3</b>
R4	<b>IF G03 OR G09 OR G12 THEN K4</b>
R5	<b>IF G10 OR G13 OR G14 THEN K5</b>
R6	<b>IF G11 OR G15 OR G40 THEN K6</b>
R7	<b>IF G07 OR G12 THEN K7</b>
R8	<b>IF G16 OR G17 OR G35 THEN K8</b>
R9	<b>IF G01 OR G03 OR G04 THEN K9</b>
R10	<b>IF G18 OR G19 THEN K10</b>
R11	<b>IF G09 OR G20 THEN K11</b>
R12	<b>IF G19 THEN K12</b>
R13	<b>IF G41 THEN K13</b>
R14	<b>IF G05 OR G23 THEN K14</b>
R15	<b>IF G10 THEN K15</b>
R16	<b>IF G10 OR G24 OR G27 THEN K16</b>
R17	<b>IF G10 OR G25 THEN K17</b>
R18	<b>IF G28 OR G31 OR G36 THEN K18</b>
R19	<b>IF G29 OR G30 OR G32 THEN K19</b>
R20	<b>IF G10 OR G21 THEN K20</b>
R21	<b>IF G37 OR G38 THEN K21</b>
R22	<b>IF G39 OR G03 THEN K22</b>

Para pakar atau ahli biasanya menganalisis informasi yang ada dengan kata mungkin, kemungkinan besar, hampir pasti, dan sebagainya. Untuk mengakomodasi hal tersebut maka dihadirkan interpretasi pakar dengan cara memberikan pembobotan terhadap setiap term tersebut. *Term* interpretasi pakar tersebut dapat dilihat pada Tabel 4.

**TABEL 4.** Bobot Interpretasi Pakar

No	<i>Certainty Term</i>	Bobot Pakar
1	Tidak diketahui	0,2
2	Mungkin	0,4
3	Kemungkinan besar	0,6
4	Hampir pasti	0,8
5	Pasti	1

Pada saat *user* melakukan diagnosa melalui *website* nantinya akan diberikan pilihan intensitas jawaban berdasarkan gejala-gejala kerusakan laptop yang dialami. Intensitas ini mengandung bobot yang diberikan untuk mengakomodasi *user* dalam menggambarkan intensitas gejala kerusakan laptop yang dialami. Adapun interpretasi *user* dapat dilihat pada tabel 5.

**TABEL 5.** Interpretasi *User*

No	<i>Certainty Term</i>	Bobot
1	Tidak Ada	0
2	Jarang	0,25
3	Cukup Sering	0,5
4	Sering	0,75
5	Sangat sering	1

### Proses Diagnosa

Proses ini dimulai dengan *user* memilih gejala beserta intensitas yang dialami oleh laptop. Proses selanjutnya yakni dilakukan perhitungan menggunakan metode *naïve bayes* terlebih dahulu yang nantinya akan menghasilkan klasifikasi kerusakan misal kerusakan RAM (K02). Setelah itu, hasil ini akan diproses menggunakan metode *certainty factor* yang nantinya akan menentukan hasil nilai keyakinan atas kerusakan tersebut. Nilai keyakinan ini berupa persentase seberapa besar laptop itu mengalami kerusakan tersebut. Hasil ini akan ditampilkan di laporan diagnosa pada *website*. Berikut pembahasan dari setiap tahapan pada proses diagnosa ini.

1. *User* memilih gejala dan intensitas

Pada *website* sistem pakar ini nantinya akan disediakan sejumlah gejala beserta intensitas waktu yakni jarang, sering, cukup sering, dan sangat sering.. Contoh gejala yang dipilih dapat dilihat pada Tabel 6.

**TABEL 6.** Contoh Gejala dan Intensitas yang Dipilih

Kode Gejala	Gejala	Intensitas
G17	Tidak bisa connect internet di browser	Sangat sering
G35	Tanggal dan jam selalu ter-reset saat restart dan dimatikan	Sering
G07	Hang/crash saat menjalankan aplikasi	Jarang

2. Menentukan nilai  $N$ ,  $m$ ,  $x$ ,  $n_c$  setiap class dan  $P(v_j)$

Dari gejala yang telah dipilih pada tabel 6 akan dicari nilai  $N$ ,  $m$ ,  $x$ ,  $n_c$  setiap class dan  $P(v_j)$  dari setiap kerusakan laptop, contoh perhitungannya sebagai berikut.

Kerusakan ke-1 : LCD / Monitor Rusak

$$N = 1, m = 41, x = 22, n_c 17 = 0, n_c 35 = 0, n_c 07 = 0, P(v_j) = 1/x = 0,045$$

Kerusakan ke-2 : RAM Rusak

$$N = 1, m = 41, x = 22, n_c 17 = 0, n_c 35 = 0, n_c 07 = 0, P(v_j) = 1/x = 0,045$$

.....

3. Menghitung nilai  $P(ai|vj)$  menggunakan persamaan 1. Contoh perhitungannya sebagai berikut.

Kerusakan ke-1 : LCD / Monitor Rusak

$$P(17|L) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

$$P(35|L) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

$$P(07|L) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

Kerusakan ke-2 : RAM Rusak

$$P(17|R) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

$$P(35|R) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

$$P(07|R) = \frac{0 + 41 \cdot 0,045}{1 + 41} = 0,0439$$

4. Menghitung  $P(ai|vj) \times P(v_j)$  untuk tiap  $v$ .

Kerusakan ke-1: LCD / Monitor Rusak

$$P(L) = 0,045 \times (0,0439 \times 0,0439 \times 0,0439) = 0,00000380720$$

Kerusakan ke-2: RAM Rusak

$$P(R) = 0,045 \times (0,0439 \times 0,0439 \times 0,0439) = 0,00000380720$$

.....

5. Menentukan hasil klasifikasi

Untuk menentukan hasil klasifikasi *naïve bayes* yakni dengan menentukan nilai probabilitas terbesar dari hasil perhitungan  $P(ai|vj) \times P(v_j)$  untuk tiap  $v$ . Berdasarkan perhitungan probabilitas di atas diperoleh nilai tertinggi yakni pada kerusakan Baterai dan CMOS Bermasalah dengan nilai *naïve bayes* sebesar 0,00000905428. Untuk mengetahui seberapa besar keyakinan terhadap hasil tersebut maka akan dilakukan perhitungan menggunakan metode *certainty factor*.

6. Menentukan nilai CF pakar, CF *user*, dan CF gejala

Nilai CF pakar diperoleh dari pakar sesuai dengan *term* pada Tabel 4. CF *user*



didapatkan dari intensitas yang dipilih *user*. Adapun untuk CF gejala didapatkan dengan cara mengalikan CF pakar dan CF *user* sebagai penggabungan bukti. Nilai CF untuk setiap gejala dapat dilihat pada tabel 7.

TABEL 7. Nilai CF pakar, CF *user*, dan CF gejala

Kode Gejala	Gejala	CF Pakar	CF User	CF Gejala
G17	Tidak bisa connect internet di browser	0,6	1	0,6
G35	Tanggal dan jam selalu ter-reset saat restart dan dimatikan	1	0,75	0,75
G07	Hang/crash saat menjalankan aplikasi	1	0,25	0,25

7. Menentukan nilai CF *Combine*

Langkah menghitungnya yakni perhitungan pertama gejala 1 sebagai  $s_1$  dan gejala 2 sebagai  $s_2$ . Perhitungan selanjutnya yakni nilai yang dihasilkan pada perhitungan pertama akan dikombinasikan dengan gejala 3 sebagai  $s_3$  lalu dilakukan perhitungan menggunakan fungsi kombinasi 1. Setelah itu maka dihasilkan nilai keyakinannya. Contoh perhitungannya yakni sebagai berikut.

- $CF[h, s_1 \& s_2] = MB[h, s_1] + MB[h, s_2](1 - MB[h, s_1])$   
 $CF[h, s_1 \& s_2] = 0,6 + 0,75 * (1 - 0,6)$   
 $CF[h, s_1 \& s_2] = 0,9$
- $CF[h, s_1 \& s_2 \& s_3] = MB[h, s_1 \& s_2] + MB[h, s_3](1 - MB[h, s_1 \& s_2])$   
 $CF[h, s_1 \& s_2 \& s_3] = 0,9 + 0,25 * (1 - 0,9)$   
 $CF[h, s_1 \& s_2] = 0,925$

8. Menghitung Persentase Nilai Kepercayaan

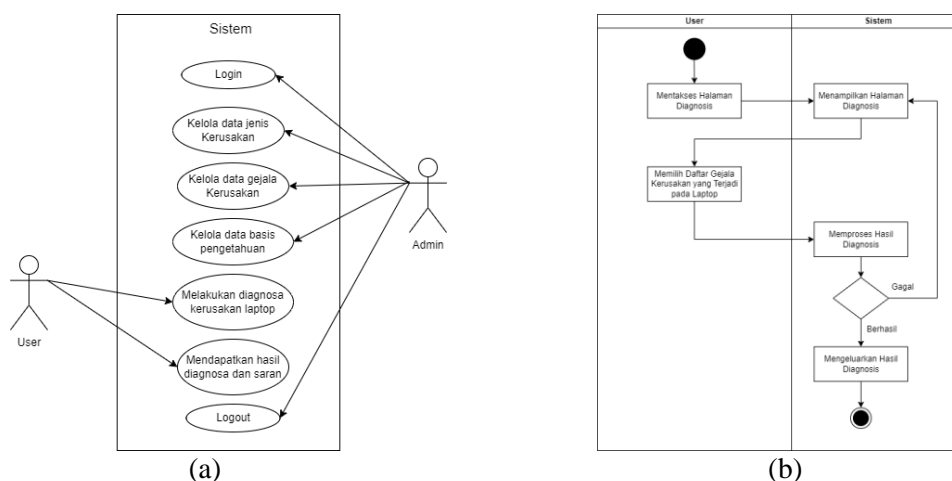
Untuk menghitung presentase nilai kepercayaan dari kerusakan yang diperoleh yakni menggunakan persamaan 4.

$$\begin{aligned}
 \text{Persentase} &= 0,925 * 100\% \\
 &= 92,5\%
 \end{aligned}$$

Maka hasil yang diperoleh kam dari perhitungan certainty faktor ini yakni kerusakan pada Baterai dan CMOS dengan keyakinan sebesar 92,5%

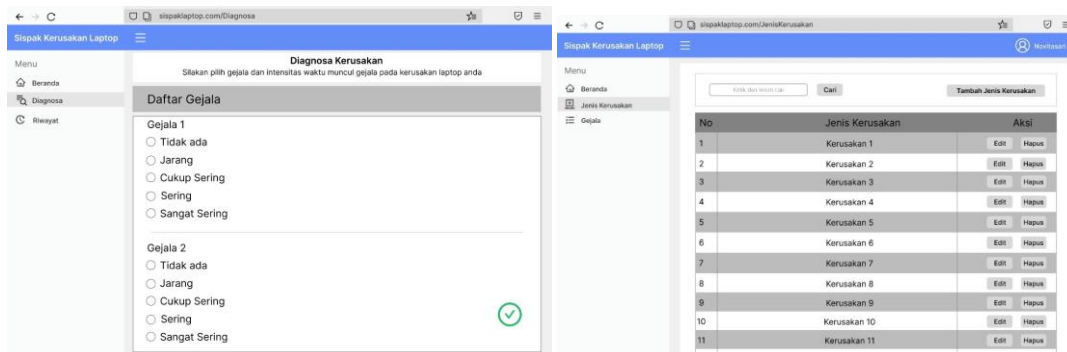
Desain

Pada tahap ini penulis membuat 2 desain yaitu desain sistem seperti pembuatan *use case diagram*, *activity diagram*, dan *entity relationship diagram*, serta pembuatan desain tampilan atau *mockup* sistem yang akan dikembangkan.



GAMBAR 1. (a) use case diagram, (b) activity diagram

Setelah desain sistem, maka dibuat desain tampilan atau yang biasa dikenal dengan sebutan *mockup*.



(a)

(b)

GAMBAR 2. (a) halaman diagnosa kerusakan, (b) halaman Kelola admin

### Pengembangan Sistem

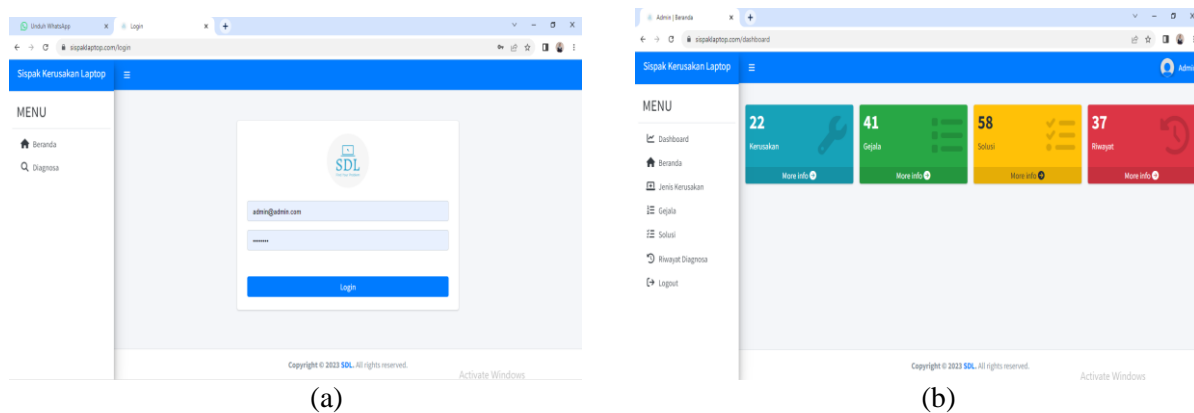
Dalam proses pengembangan sistem diperlukan beberapa alat atau tools seperti bahasa pemrograman yang digunakan yakni PHP dengan framework CodeIgniter dan untuk database menggunakan MySQL serta pengelolaannya yakni menggunakan interface phpMyAdmin. Adapun untuk tampilan web ini menggunakan bahasa HTML, CSS dan JavaScript.

## HASIL DAN PENGUJIAN

### Implementasi Sistem

Implementasi sistem merupakan sebuah tahapan untuk menyelesaikan rancangan sistem yang telah dibuat sebelumnya. Desain sistem yang telah dibuat kemudian diimplementasikan melalui coding menggunakan bahasa pemrograman.

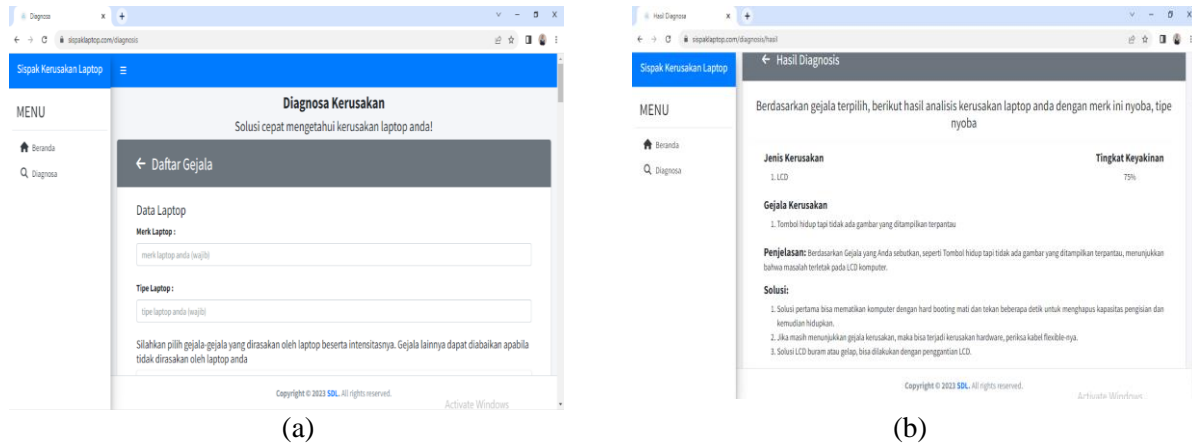
Berikut ini merupakan hasil dari implementasi sistem pakar diagnosa kerusakan komputer berbasis *website* dengan 2 *role* yakni *user* dan *admin*.



(a)

(b)

GAMBAR 3. (a) halaman login admin, (b) halaman dashboard admin



GAMBAR 4. (a) halaman diagnosa *user*, (b) halaman hasil diagnosa

### Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan metode Blackbox Testing terhadap 5 pengguna karena menurut (Nielsen, 2000) pengujian cukup dilakukan terhadap 5 pengguna untuk menguji suatu aplikasi bebas dari masalah sehingga fungsionalitas dari aplikasi berjalan dengan baik. Hasil dari pengujian ini menunjukkan bahwa fungsionalitas sistem baik pada *role user* maupun admin telah berjalan dengan dengan baik.

Pengujian akurasi ini bertujuan untuk mengetahui ketepatan hasil *output* untuk proses diagnosa kerusakan laptop pada sistem dengan cara membandingkan hasil *output* yang diberikan oleh pakar dan sistem. Pengujian ini dilakukan menggunakan 40 data uji dan didapatkan bahwa 34 dari 40 data menunjukkan bahwa hasil diagnosa sistem sama dengan hasil diagnosa pakar. Berikut perhitungan untuk akurasi menggunakan persamaan 5.

$$Akurasi = \frac{34}{40} \times 100\% = 85\%$$

Hasil tersebut menunjukkan kesesuaian sistem pakar dalam mendiagnosa kerusakan laptop selayaknya pakar yakni sebesar 85%.

## KESIMPULAN DAN SARAN

### Kesimpulan

1. Sistem pakar diagnosa kerusakan laptop berbasis *website* dengan metode naïve bayes-*certainty factor* telah berhasil dikembangkan menggunakan bahasa pemrograman PHP dengan framework CodeIgniter dan basis data MySQL.
2. Metode yang digunakan untuk mendignosa kerusakan laptop ini yakni naïve bayes dan *certainty factor*. Naïve bayes digunakan untuk pengklasifikasian kerusakan, sedangkan *certainty factor* digunakan untuk menentukan nilai keyakinan atas hasil diagnosa yang disajikan dalam bentuk persentase.
3. Pengujian sistem ini dilakukan menggunakan metode blackbox dan didapatkan kesimpulan bahwa seluruh fitur yang ada dalam sistem dapat berjalan dengan baik. Untuk pengujian akurasi sistem dilakukan dengan membandingkan kesesuaian hasil diagnosa antara sistem dengan pakar terhadap suatu gejala yang dialami. Adapun hasil yang didapatkan dalam uji akurasi tersebut yakni 85% sistem akurat dengan hasil diagnosa pakar.

### Saran

Terdapat beberapa saran dalam penelitian ini guna memperbaiki kekurangan yang ada, serta dapat mengembangkan penelitian serupa agar menjadi lebih baik kedepannya. Saran-saran tersebut diantaranya sebagai berikut.

1. Diperlukan pemeliharaan secara berkala mengenai basis pengetahuan dalam sistem agar keakuratan sistem pakar tetap terjaga dan selalu terbaru mengikuti perkembangan teknologi pada bidang ini.
2. Penambahan fitur konsultasi atau diskusi dengan pakar maupun sesama pengguna terkait kerusakan ataupun gejala yang dialami oleh laptop.
3. Pengembangan mengenai desain antarmuka maupun fungsionalitas sistem agar pengguna maupun admin dapat dengan nyaman menggunakan sistem pakar ini serta dapat memberikan manfaat bagi masyarakat luas.
4. Menggunakan metode lain guna mendapatkan nilai akurasi sistem yang lebih baik.

### REFERENSI

- Arhami, M. (2005). Konsep Dasar Sistem Pakar. Andi Andriani, A., Anastasya, M., Sardiarinto, Susanto, W. E., & Supriyanta. (2018, August 7-8). *Certainty factors* in Expert System to Diagnose Disease of Chili Plants. International Conference on Cyber and IT Service Management.
- Hayadi, B. H. (2018). Sistem Pakar Penyelesaian Kasus Minat Baca, Kecenderungan, dan Karakter Siswa dengan Metode Forward Chaining. Deepublish.
- Kemenkominfo.(2019). Survei TIK Serta Implikasinya Terhadap Aspek Sosial Budaya dan Ekonomi Masyarakat. Jakarta. Puslitbang Aptika dan IKP.
- Kiray, D., & Sianturi, F. C. (2020). Diagnose Expert System Computer Malfunction *Certainty factor* Method. Journal of Computer Networks, Architecture and High Performance Computing (Vol. 2, No.1).
- Kusrini. (2006). Sistem Pakar Teori dan Aplikasi. C.V Andi Offset.
- Maulana, F., Hidayat, N., & Wijoyo, S. (2019). Diagnosis Penyakit Pada Bawang Merah Dengan Menggunakan Metode Fuzzy Tsukamoto (Studi Kasus: UPTD. Pembibitan Ternak dan Hijauan Makanan Ternak Kec. Singosari Malang). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, pp. 220-226.
- Nielsen, J. (2000). Why You Only Need to Test with 5 Users.
- Nurhadi. (2017). Pondasi Dasar Pemrograman *Website*. CV. Garuda Mas Sejahtera . Surabaya.
- Shortliffe, E. H., & Buchanan, B. G. (1984). Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.
- Turban, E., Aronson, J. E., & Liang, T.-P. (2007). Decision Support Systems and Intelligent Systems 7th Edition (7th ed.). Prentice-Hall.