

Implementasi Metode *Certainty Factor* Untuk Diagnosis Kerusakan Software Pada Smartphone Berbasis Web

Rizky Afwan ¹⁾, Indah Fitri Astuti ²⁾, Dedy Cahyadi ³⁾

Informatika, Fakultas Teknik, Universitas Mulawarman
Jalan Kuaro Gunung Kelua, Samarinda, 75119.

E-Mail : rizkyafwan88@gmail.com ¹⁾; indahfitriastuti@unmul.ac.id ²⁾; dedy.cahyadi@unmul.ac.id ³⁾;

ABSTRAK

Smartphone merupakan alat komunikasi antara dua orang atau lebih, saat ini tidak hanya berkomunikasi, beberapa fungsi lainnya sudah tersedia, serta beberapa *smartphone* memiliki kinerja layaknya komputer. Komponen teknologi umumnya tidak dapat terhindar dari kerusakan. Kerusakan yang terjadi pada *smartphone* sebagian besar masyarakat awam tidak tahu pasti kerusakan apa yang dialami. Tujuan yang ingin dicapai dalam penelitian ini adalah membentuk sebuah sistem yang dapat melakukan diagnosis terhadap suatu kerusakan *smartphone* berdasarkan pola pikir pakar atau ahli. Penelitian ini menggunakan metode *Certainty Factor* untuk mengakomodasi ketidakpastian seorang pakar. Data yang digunakan merupakan hasil wawancara bersama pakar selaku teknisi *Smartphone Care* dan ditemukan 6 jenis kerusakan yang dapat disebabkan oleh *software* yaitu *Light Bootloop*, *Medium Bootloop*, *Hard Bootloop*, *Soft Brick*, *Hard Brick*, dan *Freeze*. Hasil pengujian dari 6 data, diperoleh hasil diagnosis jenis kerusakan yang terjadi berdasarkan gejala yang dialami oleh user. Hasil akurasi dari perhitungan manual dan sistem memiliki tingkat akurasi 83,3%, sedangkan hasil akurasi dari perhitungan sistem dengan hasil diagnosis pakar memiliki tingkat akurasi 100%. Hasil diagnosis kerusakan metode *certainty factor* ini bisa menjadi bahan pertimbangan untuk para pengguna *smartphone* awam menggunakan sistem dalam melakukan diagnosis pada *smartphone* lebih dini tanpa datang ke teknisi untuk melakukan perbaikan.

Kata Kunci – Sistem Pakar, Diagnosis Kerusakan, *Certainty Factor*, *Smartphone*

1. PENDAHULUAN

Teknologi dan Komunikasi saat ini sudah menjadi kebutuhan masyarakat, *smartphone* merupakan salah satu media komunikasi yang populer di anak muda hingga orang tua. *Smartphone* merupakan alat komunikasi antara dua orang atau lebih guna saling bertukar informasi, *smartphone* saat ini tidak hanya untuk berkomunikasi saja, fungsi-fungsi tambahan yang tersedia seperti kamera, video bahkan fasilitas internet sudah tersedia, serta beberapa *smartphone* memiliki kinerja layaknya seperti komputer (Wati dan Tusaadiah, 2017).

Sebagaimana sebuah komponen teknologi pada umumnya, maka *smartphone* tidak dapat terhindar dari kerusakan yang terjadi, terutama pada *software* atau perangkat lunak. *Software* atau perangkat lunak dapat diartikan sebagai istilah dalam dunia komputer yang mencerminkan data yang diolah secara khusus, disimpan secara digital serta dalam pembuatannya diperlukan bahasa pemrograman tertentu yang dipahami oleh komputer (Herliana dan Rasyid, 2016).

Beberapa kerusakan yang terjadi pada *smartphone*, sebagian besar pengguna *smartphone* awam tidak tahu pasti kerusakan apa yang sedang dialami. Jasa perbaikan *smartphone* memerlukan biaya serta kehadiran ahli atau pakar tidak selalu tersedia 24 jam. Salah satu teknik yang memungkinkan untuk mendapatkan hasil akurat berdasarkan pola pikir ahli atau pakar adalah teknik *certainty factor* yang merupakan program komputer

yang mengandung pengetahuan berdasarkan pola pikir pakar. Metode sistem pakar menggambarkan keyakinan seorang pakar dengan memberikan bobot sesuai pengetahuan. Sebagai alat bantu memberikan keputusan diperlukan metode *forward chaining* yang berfungsi sebagai metode penelusuran pengumpulan data untuk mencari kesimpulan serta solusi dari masalah yang dihadapi (Yuwono dkk, 2017).

Berdasarkan penjabaran terkait, diperlukan sistem yang dapat melakukan diagnosis kerusakan *smartphone* berdasarkan pola pikir pakar untuk memudahkan masyarakat awam mengetahui lebih awal kerusakan apa yang terjadi pada *smartphone* serta solusi penanganan tanpa perlu mendatangi tempat perbaikan *smartphone*, maka dibangun sebuah sistem pakar untuk melakukan diagnosis kerusakan *software* pada *smartphone* menggunakan metode *certainty factor*.

2. TINJAUAN PUSTAKA

Penerapan metode *certainty factor* oleh (Adi dan Diwandari, 2020) melakukan rancang bangun sistem diagnosis kerusakan *handphone* berdasarkan data wawancara dari teknisi Firdaus Cell sebagai pakar. Uji kelayakan sistem dilakukan dengan tujuan mengetahui kelayakan sistem untuk digunakan pengguna dan uji ketepatan diagnosis berdasarkan perhitungan manual dan perhitungan menggunakan sistem. Berdasarkan hasil pengujian diperoleh 100% fungsionalitas sistem pakar diagnosis kerusakan *handphone* berjalan sesuai

dengan daftar kebutuhan sistem dan sistem mempunyai tingkat akurasi sebesar 100%.

Berdasarkan pengujian yang telah dilakukan oleh (Nengsih dan Putra, 2020) dengan tujuan mengamati sejauh mana sistem pakar diagnosis kerusakan *smartphone* dengan menggunakan metode *certainty factor* dan *forward chaining* dapat mempermudah dan mempercepat dalam memperbaiki kerusakan *smartphone*. Sistem pakar dalam mendiagnosis mendapatkan hasil yang cukup baik dengan nilai 73,33% jika dibandingkan dengan representasi yang diberikan oleh seorang tenaga ahli *service* langsung atau pakar.

A. Kerusakan *Smartphone*

Kerusakan yang dapat terjadi pada *smartphone* terdapat 2 buah jenis yang menyebabkan tidak berfungsinya *smartphone* secara optimal. Jenis kerusakan tersebut terdiri dari kerusakan *software* dan kerusakan *hardware* (Nengsih dan Putra, 2020).

Menurut pengalaman Muallif Askar Shodiq selaku teknisi yang telah bekerja selama 2 tahun di salah satu lokasi jasa perbaikan *smartphone* yaitu *smartphone care*, kerusakan yang disebabkan oleh *software* terbagi lagi menjadi 6 jenis yaitu :

1. *Light Bootloop*

Light bootloop merupakan suatu kondisi pada saat *smartphone* digunakan terkadang mengalami *slow respond* (lambat saat digunakan), dalam penggunaan aplikasi dari *smartphone* sering terjadi *force close* (aplikasi yang tertutup paksa pada saat aplikasi diakses), *smartphone* kadang mengalami *restart system* (memulai sistem ulang dengan sendirinya).

2. *Medium Bootloop*

Medium bootloop merupakan suatu kondisi *smartphone* mengalami kendala terkadang tidak dapat memasuki menu aplikasi, terjadi kesalahan pada sistem perangkat seperti *wi-fi* tidak muncul, kamera tidak dapat digunakan, serta terjadi not *responding* secara tiba-tiba (tidak ada tanggapan dari *smartphone* saat menggunakan aplikasi).

3. *Hard Bootloop*

Hard bootloop merupakan suatu kondisi *smartphone* hanya menampilkan logo animasi pada layar utama, tidak dapat memasuki menu aplikasi, dan bisa juga terjadi mati total

4. *Freeze*

Freeze merupakan sebuah gejala yang dialami ketika pengguna menggunakan aplikasi *smartphone* berhenti secara tiba – tiba. Gejala yang muncul biasanya seperti aplikasi tidak dapat ditutup pada saat penggunaan, aplikasi berhenti secara tiba-tiba atau not *responding* (tidak ada tanggapan dari *smartphone* saat menggunakan aplikasi), terjadi *restart* secara tiba-tiba.

5. *Softbrick*

Softbrick merupakan suatu kondisi *smartphone* mengalami stuck di logo merek pabrikan (pada saat *smartphone* dinyalakan hanya sampai logo merek pabrikan), *smartphone* hanya menampilkan animasi pada layar utama, mengalami *restart smartphone* yang berulang-ulang.

6. *Hardbrick*

Hardbrick merupakan suatu kondisi *smartphone* mengalami *restart* sistem yang berulang ulang dan *smartphone* mengalami mati total.

B. Sistem Pakar

Sistem pakar merupakan program komputer yang mengambil pemikiran dari para ahli dengan tujuan memecahkan masalah serta membantu proses pengambilan keputusan terhadap suatu masalah. Umumnya sistem pakar terdiri dari basis pengetahuan, akumulasi pengalaman dan perangkat aturan untuk menetapkan kondisi setiap situasi tertentu yang ditentukan dalam suatu program (Putri, 2018). Sistem pakar dianggap berhasil ketika sistem tersebut bekerja layaknya pakar dalam mengambil keputusan, baik dari sisi proses pengambil keputusan maupun memberikan solusi (Septiani dan Kuryanti, 2018).

C. *Forward Chaining*

Forward Chaining bisa disebut juga Teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian menyesuaikan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*, jika ditemukan fakta yang sesuai dengan bagian *IF*, maka *rule* tersebut akan dieksekusi, apabila sebuah *rule* akan dieksekusi, maka sebuah fakta baru (bagian *THEN*) akan ditambahkan ke dalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. *Forward Chaining* bermakna menggunakan himpunan aturan kondisi dan aksi. Metode ini, data digunakan dalam menentukan aturan mana yang nanti digunakan, lalu aturan tersebut diproses, mungkin proses menambahkan data ke memori kerja, proses diulang sampai ditemukan suatu hasil (Afriansyah dkk, 2019).

D. *Certainty Factor*

Metode *Certainty Factor* merupakan teori metode yang diperkenalkan oleh *Shortlife* dan *Buchanan* pada tahun 1975 di mana metode *Certainty Factor* digunakan untuk mengakomodasi ketidakpastian pemikiran seorang pakar. *Certainty Factor* adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti atau tidak pasti yang berbentuk *metric* yang biasanya digunakan dalam sistem pakar (Arifin, 2016).

Buku yang ditulis oleh (G Buchanan & H Shortliffe, 1984), menerapkan beberapa persamaan untuk menentukan hasil nilai CF. Berikut rumus CF yang ditunjukkan pada persamaan 1,2 dan 3

$$CF[H, E] = MB[H, E] - MD[H, E] \quad 1)$$

$$CF[H, E]_1 = CF[H] * CF[E] \quad (2)$$

$$CF_{combine} CF[H, E]_{1,2} = CF[H, E]_1 + CF[H, E]_2 * [1 - CF[H, E]_1]$$

$$CF_{combine} CF[H, E]_{old,3} = CF[H, E]_{old} + CF[HE]_3 * [1 - CF[H, E]_{old}]$$

E. UML

Unified Modelling Language (UML) merupakan sebuah bahasa yang telah menjadi standar dalam industri dalam membuat visualisasi, rancangan dan dokumentasi sistem perangkat lunak. UML memberikan sebuah standar dalam merancang model sebuah sistem. *Unified Modelling Language* merupakan alat untuk merancang sistem pada objek. Dengan menggunakan alat bantu *Unified Modelling Language* kita dapat membuat sesuatu model untuk semua jenis aplikasi perangkat lunak, aplikasi ini dapat berjalan pada perangkat keras, sistem operasi dan jaringan apa saja, serta ditulis dalam bahasa pemrograman (Zulkifli, 2018).

3. Metode Penelitian

A. Tahapan Penelitian

Tahapan ini dilakukan untuk menganalisis sistem yang akan dibangun dengan metode *Certainty Factor*. Pelaksanaan penelitian ini mencakup Langkah-langkah dari awal sampai akhir, adapun langkah dalam pelaksanaan penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan langkah-langkah pelaksanaan penelitian pada Gambar 3.1, maka dapat diuraikan pembahasan dari langkah-langkah tahapan dalam penelitian adalah :

1. Melakukan Wawancara

Tahapan ini wawancara dilakukan sebagai landasan dasar penelitian untuk menentukan kriteria-kriteria dan alternatif yang dibutuhkan.

2. Menentukan Kriteria dan Alternatif

Setelah melakukan wawancara maka disusunlah beberapa kriteria yang dibutuhkan berdasarkan pengetahuan ahli atau pakar.

3. Implementasi Program

Tahapan ini dilakukan dengan membuat program sistem pakar menggunakan *web*

application untuk dapat memberikan kemudahan akses sistem untuk masyarakat awam.

4. Perhitungan *Certainty Factor*

Program sistem pakar pada tahapan sebelumnya, setelah melakukan pengisian data yang dibutuhkan akan melakukan perhitungan algoritma dengan metode *certainty factor*.

5. Menampilkan Hasil Diagnosis

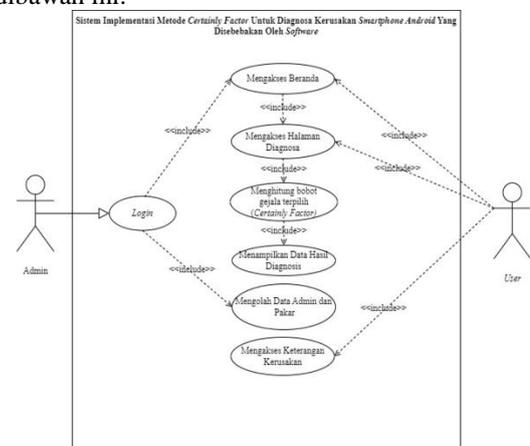
Hasil data perhitungan pada tahapan sebelumnya akan ditampilkan dan diurutkan berdasarkan persentase kemungkinan terbesar.

B. Perancangan Data

Penelitian ini menggunakan wawancara bersama pakar yaitu salah satu teknisi dari *Smartphone Care* sebagai landasan data yang akan digunakan sebagai perhitungan menggunakan metode *certainty factor*. Data yang digunakan berupa informasi dan data-data berupa gejala-gejala apa saja yang disebabkan dari kerusakan *smartphone*, serta solusi penanganan yang ditawarkan.

C. Perancangan Proses

Perancangan Proses berfungsi untuk menggambarkan rancangan penelitian yang telah dikumpulkan untuk membangun sistem pakar diagnosis kerusakan *smartphone*. Perancangan digambarkan dengan diagram *usecase*, beberapa proses sistem penelitian ini terdapat pada gambar dibawah ini:



Gambar 2. Usecase Diagram

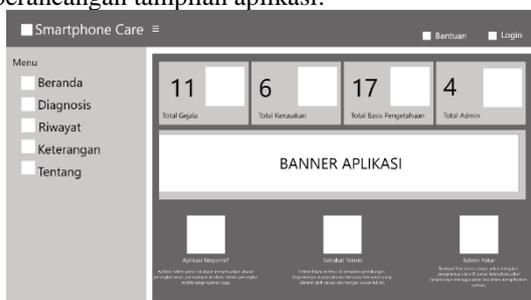
1. *Login*, admin terlebih dahulu melakukan *login* untuk masuk ke dalam *dashboard* sistem pakar, sedangkan *user* tidak perlu melakukan *login* karena sistem bersifat terbuka bagi pengunjung untuk melakukan diagnosis.
2. Mengakses beranda, halaman beranda berisi informasi yang menampilkan total gejala, total kerusakan, total pengetahuan, total admin, *button* diagnosis untuk melakukan diagnosis, *button* keterangan untuk mengakses keterangan, *button* tentang untuk mengakses tentang aplikasi
3. Mengakses halaman diagnosis, sistem akan menampilkan seluruh gejala yang terdapat pada

sistem dan *user* memilih parameter keyakinan *user* terhadap suatu gejala yang dialami seperti pasti ya, hampir pasti, kemungkinan besar, mungkin, hampir mungkin, tidak tahu atau tidak yakin. Setelah memilih seluruh parameter keyakinan *user* terhadap gejala yang dialami lalu tekan *button submit*.

4. Melakukan perhitungan dari setiap nilai bobot gejala yang akan dirasakan *user* oleh sistem menggunakan metode *Certainty factor*.
5. Menampilkan data hasil diagnosis beserta solusi, setelah sistem melakukan perhitungan berdasarkan gejala yang dipilih dan menampilkan hasil bahwa *smartphone* android mengalami salah satu kerusakan, menampilkan detail dari kerusakan, nilai dari kepercayaan pakar, solusi dari penanganan kerusakan, serta gambar dari kerusakan.
6. Mengolah data admin dan pakar, admin dapat menambahkan, mengubah, dan menghapus isi di dalam sistem.
7. Mengakses keterangan kerusakan, *user* dapat melihat seluruh detail dari kerusakan yang dapat terjadi serta solusi penanganan untuk setiap kerusakannya. Use case diagram dapat dilihat pada Gambar 2.

D. Perancangan Tampilan

Perancangan Tampilan pada aplikasi ini terdiri dari tampilan halaman beranda, halaman diagnosis, dan halaman hasil diagnosis. Berikut perancangan tampilan aplikasi:



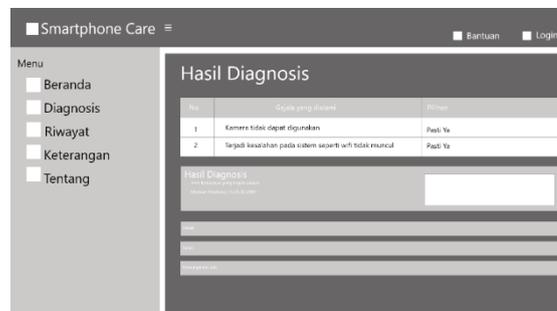
Gambar 3. Rancangan Halaman Beranda

Halaman beranda merupakan tampilan awal pada saat menjalankan aplikasi dimana terdapat beberapa menu yang ditampilkan seperti diagnosis, keterangan, tentang dan beberapa fitur dari aplikasi lainnya.



Gambar 4. Rancangan Halaman diagnosis

Halaman diagnosis merupakan halaman yang muncul ketika pengguna sistem ingin melakukan diagnosis kerusakan *smartphone*, terdapat beberapa gejala yang muncul berdasarkan pengetahuan dan pilihan kondisi untuk dipilih sesuai kondisi yang terjadi.



Gambar 5 Rancangan Halaman Hasil

E. Perancangan Pengujian

Pelaksanaan pengujian pada penelitian ini dilakukan dengan tujuan menentukan perhitungan manual menggunakan metode *certainty factor* berdasarkan basis pengetahuan, menentukan perbandingan hasil perhitungan manual dan perbandingan sistem, dan menentukan perbandingan sistem dan diagnosis dari pakar. Pengujian ini berfokus untuk menentukan seberapa akurat sistem melakukan diagnosis berdasarkan pola pikir pakar.

4. Hasil dan Pembahasan

A. Pengumpulan data

Data-data yang diperoleh selama pengumpulan data terdiri dari jenis kerusakan, gejala tiap kerusakan, solusi kerusakan, tingkat keyakinan, basis pengetahuan, dan representasi pengetahuan. Data-data tersebut diperoleh dari wawancara yang berhubungan dengan kerusakan pada *smartphone*.

1. Data Jenis Kerusakan

Data-data jenis kerusakan yang digunakan dalam sistem pakar ini mendeteksi kerusakan pada *smartphone* ini berjumlah 6 jenis kerusakan. Adapun data-data jenis kerusakan tersebut dapat dilihat pada Tabel 1 dibawah ini:

Tabel 1. Data Jenis Kerusakan
Kode Kerusakan Jenis Kerusakan

JK01	<i>Light Bootloop</i>
JK02	<i>Medium Bootloop</i>
JK03	<i>Hard Bootloop</i>
JK04	<i>Soft Brick</i>
JK05	<i>Hard Brick</i>
JK06	<i>Freeze</i>

2. Data Gejala Tiap Kerusakan

Data gejala merupakan data yang berisi tentang gejala tiap kerusakan. Setiap kerusakan terdiri dari beberapa gejala yang menjadi penyebab kerusakan pada *smartphone*.

Gejala	Kerusakan					
	<i>Light Bootloop</i>	<i>Medium Bootloop</i>	<i>Hard Bootloop</i>	<i>Softbrick</i>	<i>Hardbrick</i>	<i>Freeze</i>
	CF PAKAR					
G01	1	0	0	0	0	0
G02	1	0	0	0	0	0.2
G03	0	0.8	0	0	0	0
G04	0	0.8	0	0	0	0
G05	0	0	1	0	1	0
G06	0	0	0	0	0	1
G07	0	0	0	1	0	0
G08	0	0	0.4	1	0.8	0
G09	0	0	0	0.8	1	0
G10	1	0	0	0	0	1
G11	0	0.6	0.2	0	0	0
G12	0	0.8	0	0	0	1

3. Data Solusi Kerusakan

Data Solusi kerusakan merupakan data yang berisikan tentang solusi dari setiap kerusakan yang terjadi. Data solusi dapat dilihat pada Tabel 3.

Kode Solusi	Kode Kerusakan	Solusi
S01	JK01	1. Melakukan pembaruan aplikasi 2. Melakukan pembaruan sistem 3. Melakukan <i>restart smartphone</i>
S02	JK02	1. Melakukan <i>flashing</i> dengan <i>firmware</i> yang sama 2. Melakukan <i>factory reset</i> / reset pabrik
S03	JK03	1. Melakukan pengecekan <i>hardware</i> dari <i>power</i> hingga <i>data memory processor</i>
S04	JK04	1. Masuk ke dalam <i>recovery mode</i> 2. <i>Install</i> menggunakan SPflashtool
S05	JK05	1. Pengecekan pada <i>hardware</i> 2. Memperbaiki menggunakan JTAG
S06	JK06	1. Menyesuaikan penggunaan dengan spesifikasi <i>smartphone</i> 2. Membersihkan RAM/ penyimpanan memori sementara dengan aplikasi 3. Tidak menggunakan <i>smartphone</i> secara berlebihan

4. Data Tingkat Keyakinan Pakar

Data tingkat keyakinan merupakan data yang diperoleh dari pakar untuk mengukur tingkat keyakinan seorang pakar berdasarkan gejala yang terjadi terhadap suatu kerusakan. Data tingkat keyakinan terdiri dari 6 tingkat keyakinan. Data tingkat keyakinan dapat dilihat pada Tabel 4.

Tabel 3. Data Tingkat Keyakinan
Tingkat Keyakinan Nilai Keyakinan

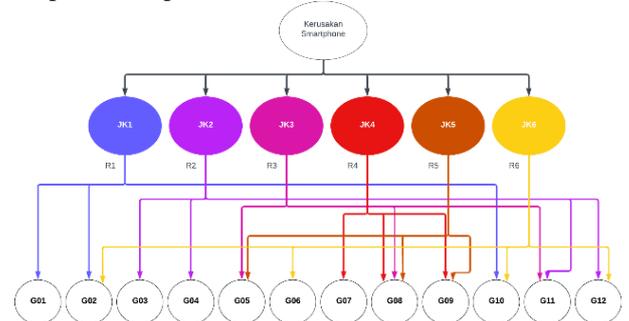
Pasti Ya	1
Hampir Pasti Ya	0.8
Mungkin Ya	0.6
Mungkin Tidak	0.4
Hampir Pasti Tidak	0.2
Pasti Tidak	0

5. Data Basis Pengetahuan

Data basis pengetahuan merupakan data yang diolah dari hasil wawancara bersama pakar dan disusun berdasarkan tingkat keyakinan seorang pakar dari suatu gejala terhadap suatu kerusakan *smartphone*. Data basis pengetahuan inilah yang nantinya akan digunakan untuk perhitungan menggunakan persamaan dari rumus *certainty factor*, serta menjadi landasan basis pengetahuan untuk membentuk *decision tree* dengan rule *IF THEN*.

6. Representasi Pengetahuan

Berdasarkan basis pengetahuan dibentuklah representasi pengetahuan yang digambarkan dengan *decision tree* untuk digunakan sebagai pohon keputusan yang berkaitan dengan masalah diagnosis kerusakan *smartphone*. *Decision tree* dapat dilihat pada Gambar 6.



Gambar 6. *Decision Tree*

Berdasarkan representasi tiap – tiap gejala yang dialami oleh pengguna *smartphone* yang disebabkan oleh *software*, disusunlah aturan – aturan yang akan digunakan sebagai dasar aturan penelusuran *forward chaining*, antara lain sebagai berikut

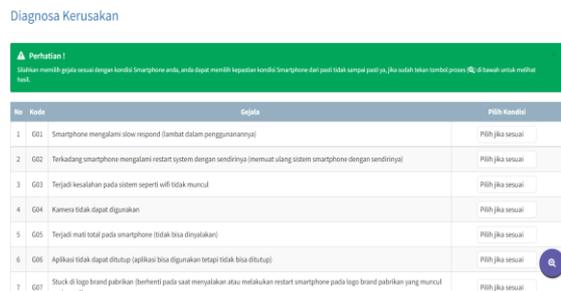
1. IF G01 OR G02 OR G10 OR JK01
2. IF G03 OR G04 OR G11 OR G12 OR JK02
3. IF G05 OR G08 OR G11 OR JK03
4. IF G07 OR G08 OR G09 OR JK04
5. IF G05 OR G08 OR G09 OR JK05
6. IF G02 OR G06 OR G10 OR G12 THEN JK06

B. Penerapan Tampilan

Tampilan yang menjadi fitur utama pada aplikasi ini terdiri dari halaman diagnosis dan halaman hasil diagnosis. Penerapan tampilan merupakan implementasi dari perancangan tampilan. Berikut beberapa tampilan yang menjadi fitur utama sistem pakar.

1. Tampilan Diagnosis

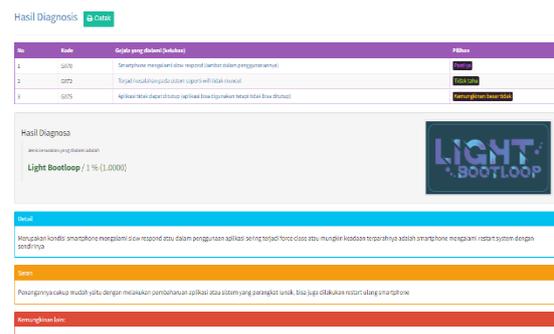
Tampilan diagnosis merupakan tampilan yang muncul ketika pengguna sistem memilih menu diagnosis pada halaman utama. Tampilan diagnosis berisikan petunjuk penggunaan sistem pakar. Tampilan diagnosis dapat dilihat pada Gambar 8.



Gambar 7. Tampilan Diagnosis

2. Tampilan Hasil

Tampilan hasil merupakan tampilan yang ditampilkan ketika pengguna sistem selesai memilih gejala. Tampilan hasil menampilkan gejala yang dipilih, kerusakan yang dialami, dan beberapa kemungkinan lain jika ada. Tampilan hasil dapat dilihat pada Gambar 9.



Gambar 8. Tampilan Hasil Diagnosis

C. Pengujian Perhitungan Manual

Perhitungan manual bertujuan untuk mendapatkan hasil berdasarkan kondisi keadaan *smartphone* tiap *client* menggunakan metode *certainty factor* dan penelusuran dari *forward chaining*. Hasil dari perhitungan nantinya akan dibandingkan dengan hasil yang didapatkan dari perhitungan menggunakan sistem aplikasi yang telah dirancang. Berikut merupakan contoh *client* yang melakukan diagnosis.

Client (masyarakat awam) pada saat melakukan diagnosis, mengisi gejala-gejala yang dialami pada *smartphone* berdasarkan kondisi yang dialami. Terdapat 6 tingkat keyakinan yang harus dipilih pada saat melakukan diagnosis yaitu pasti tidak (0), hampir pasti tidak (0,2), mungkin tidak (0,4), mungkin ya (0,6), hampir pasti ya (0,8), dan pasti ya (1,0).

Contoh kasus untuk melakukan perhitungan ini menggunakan 6 buah data *client* yang memiliki beberapa gejala. Data uji ditampilkan dalam bentuk tabel dan masukan langsung ke dalam rumus persamaan ke-2 *certainty factor*.

Tabel 4. Perkalian Nilai Bobot Keyakinan Pakar CF[H] dengan Nilai Bobot Keyakinan user CF[E]

Client A								
Kemungkinan 1 Light Bootloop (JK1)			Kemungkinan 2 JK3 Hard Bootloop (JK2)					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	CF[H]	CF[E]		
1	G01	1	0,6	0,6	G05	1	0,6	0,6
2	G02	1	0,8	0,8	G08	0,4	0,4	0,16
3	G10	1	0,4	0,4	G011	0,2	0,6	0,12
Client B								
Kemungkinan 1 Medium Bootloop			Kemungkinan 2 Freeze					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	CF[H]	CF[E]		
1	G03	0,8	0,8	0,64	G02	0,2	0,6	0,12
2	G04	0,8	0,8	0,64	G06	1	0,6	0,6
3	G11	0,6	0,6	0,36	G10	1	0,6	0,6
4	G12	0,8	0,6	0,48	G12	1	0,6	0,6
Client C								
Kemungkinan 1 Hard Bootloop			Kemungkinan 2 Hard Brick					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	Kode Gejala	CF[H]	CF[E]	
1	G05	1	0,8	0,8	G05	1	0,8	0,8
2	G08	0,4	0,6	0,24	G08	0,8	0,6	0,48
3	G11	0,2	0,8	0,16	G09	1	0,4	0,4
Client D								
Kemungkinan 1 Soft Brick			Kemungkinan 2 Light Bootloop					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	Kode Gejala	CF[H]	CF[E]	
1	G07	1	0,6	0,6	G01	1	0,6	0,6
2	G08	1	0,8	0,8	G02	1	0,4	0,4
3	G09	0,8	0,6	0,48	G10	1	0,4	0,4
Client E								
Kemungkinan 1 Hard Brick			Kemungkinan 2 Soft Brick					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	Kode Gejala	CF[H]	CF[E]	
1	G05	1	0,8	0,8	G07	1	0,4	0,4
2	G08	0,8	0,8	0,64	G08	1	0,8	0,8
3	G09	1	0,4	0,4	G09	0,8	0,4	0,48
Client F								
Kemungkinan 1 Freeze			Kemungkinan 2 Medium Bootloop					
No	Kode Gejala	CF[H]	CF[E]	CF[H]E ₁ = CF[H]xCF[E]	Kode Gejala	CF[H]	CF[E]	
1	G02	0,2	0,8	0,16	G03	0,8	0,4	0,32
2	G06	1	0,6	0,6	G04	0,8	0,6	0,48
3	G10	1	0,6	0,6	G11	0,6	0,4	0,24
4	G12	1	0,8	0,8	G12	0,8	0,8	0,64

Persamaan kedua merupakan langkah yang digunakan dengan melakukan perkalian terhadap gejala yang dipilih *client* dengan nilai hipotesa pakar berdasarkan basis pengetahuan. Bentuk tabel merupakan bentuk setelah dilakukan klasifikasi gejala yang dipilih dengan kerusakan terkait dan di sesuaikan dengan kemungkinan yang ada.

Setelah dilakukan perhitungan pada persamaan kedua pada Tabel 6, selanjutnya adalah kombinasi dari hasil perkalian sebelumnya menggunakan rumus persamaan ke . Kombinasi hasil rumus persamaan ke-3 dapat dilihat pada tabel 7.

Kombinasi dilakukan dengan persyaratan ditemukan 2 atau lebih nilai CF. Pembobotan nilai maksimum pada penelitian ini adalah 1,0. Setelah menemukan hasil dari perhitungan kombinasi maka, langkah terakhir adalah mengubah nilai CF ke dalam bentuk persentase yaitu melakukan perkalian dengan nilai 100 yang di tunjukan pada 8.

Tabel 5 Persamaan ke-3 Rumus CF

Client A			
Kemungkinan 1 Light Bootloop		Kemungkinan 2 Hard Bootloop	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,6 + 0,8 * [1 - 0,6] = 0,76	1	0,6 + 0,16 * [1 - 0,6] = 0,664
2	0,92 + 0,4 * [1 - 0,92] = 0,952	2	0,664 + 0,12 * [1 - 0,664] = 0,70
Client B			
Kemungkinan 1 Medium Bootloop		Kemungkinan 2 Freeze	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,48 + 0,64 * [1 - 0,48] = 0,81	1	0,08 + 0,2 * [1 - 0,08] = 0,26
2	0,81 + 0,24 * [1 - 0,81] = 0,85	2	0,26 + 0,6 * [1 - 0,26] = 0,704
3	0,85 + 0,16 * [1 - 0,85] = 0,87	3	0,704 + 0,2 * [1 - 0,704] = 0,76
Client C			
Kemungkinan 1 Hard Bootloop		Kemungkinan 2 Hard Brick	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,8 + 0,24 * [1 - 0,8] = 0,848	1	0,8 + 0,48 * [1 - 0,8] = 0,896
2	0,848 + 0,16 * [1 - 0,848] = 0,872	2	0,896 + 0,4 * [1 - 0,896] = 0,9376
Client D			
Kemungkinan 1 Soft Brick		Kemungkinan 2 Light Bootloop	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,6 + 0,8 * [1 - 0,6] = 0,92	1	0,6 + 0,4 * [1 - 0,6] = 0,76
2	0,92 + 0,48 * [1 - 0,92] = 0,9584	2	0,76 + 0,4 * [1 - 0,76] = 0,856
Client E			
Kemungkinan 1 Hard Brick		Kemungkinan 2 Soft Brick	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,8 + 0,64 * [1 - 0,8] = 0,928	1	0,4 + 0,8 * [1 - 0,4] = 0,88
2	0,928 + 0,4 * [1 - 0,928] = 0,9568	2	0,88 + 0,48 * [1 - 0,88] = 0,9376
Client F			
Kemungkinan 1 Medium Bootloop		Kemungkinan 2 Freeze	
Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]	Iterasi	CF[H] ₁ + CF[H] ₂ * [1 - CF[H] ₁]
1	0,16 + 0,6 * [1 - 0,16] = 0,664	1	0,32 + 0,48 * [1 - 0,32] = 0,6464
2	0,664 + 0,6 * [1 - 0,664] = 0,8656	2	0,6464 + 0,24 * [1 - 0,6464] = 0,731264
3	0,8656 + 0,8 * [1 - 0,8656] = 0,97312	3	0,731264 + 0,64 * [1 - 0,731264] = 0,932550

Tabel 6 Mengubah Bentuk Nilai CF ke Bentuk Persentase

<i>Client</i>	Kemungkinan 1 Hasil Kombinasi * 100	Kemungkinan 2 Hasil Kombinasi * 100
A	0.952 * 100 = 95,2 %	0.676 * 100 = 67,6 %
B	0,87 * 100 = 87 %	0.76 * 100 = 76 %
C	0,872 * 100 = 87,2 %	0.9376 * 100 = 93,8 %
D	0.9584 * 100 = 95,8 %	0.856 * 100 = 85,6 %
E	0.9568 * 100 = 95,8 %	0.9376 * 100 = 93,8 %
F	0.97312 * 100 = 97,3 %	0.93255 * 100 = 93,3 %

Pembobotan nilai maksimum pada penelitian ini adalah 1,0 atau = 100%. Berdasarkan hasil akhir setelah mengubah nilai CF ke dalam bentuk nilai persentase maka hasil yang tertinggi merupakan kemungkinan jenis kerusakan yang terjadi. Kerusakan yang terjadi pada *smartphone client* A adalah *Light Bootloop* dengan persentase 95 %. Kerusakan yang dialami *client* B adalah *Medium Bootloop* dengan persentase 87 %. Kerusakan yang dialami *client* C adalah *Hard Brick* dengan persentase 93,8 %. Kerusakan yang dialami *client* D adalah *Soft Brick* dengan persentase 95,7 %. Kerusakan yang dialami *client* E adalah *Hard Brick* dengan persentase 95,7 %. Kerusakan yang dialami *client* F adalah *Freeze* dengan persentase 97,3 %.

D. Pengujian Perbandingan Perhitungan Manual Dan Sistem

Perbandingan hitung manual dengan sistem merupakan bentuk perbandingan yang akan menunjukkan seberapa besar akurasi sistem dalam melakukan komputasi perhitungan menggunakan metode *certainty factor*. Perbandingan sistem dengan hitung manual dapat dilihat pada Tabel 9.

Tabel 7 Perbandingan Data Sistem dan Manual

Data <i>Client</i>	Hitung Manual	Hasil Sistem	Sesuai/Tidak Sesuai
<i>Client A</i>	95,2 %	95,2 %	Sesuai
<i>Client B</i>	87 %	88 %	Tidak Sesuai
<i>Client C</i>	93,8 %	93,8 %	Sesuai
<i>Client D</i>	95,8 %	95,8 %	Sesuai
<i>Client E</i>	95,7 %	95,7 %	Sesuai
<i>Client F</i>	97,3 %	97,3 %	Sesuai

E. Uji Perbandingan Sistem dan Pakar

Perbandingan sistem dilakukan untuk mengetahui sejauh mana sistem dapat melakukan diagnosis berdasarkan pola pikir pakar yang dituangkan ke dalam sebuah sistem yang menggunakan metode *certainty factor*. Hasil diagnosis dari pakar terhadap *client*. Perbandingan hasil penelitian dengan pakar dicantumkan pada Tabel 10.

Tabel 8 Perbandingan Hasil Sistem dengan Pakar

Data <i>Client</i>	Hasil Sistem	Hasil Pakar	Sesuai/Tidak Sesuai
<i>Client A</i>	<i>Light Bootloop</i>	<i>Light Bootloop</i>	Sesuai
<i>Client B</i>	<i>Medium Bootloop</i>	<i>Medium Bootloop</i>	Sesuai
<i>Client C</i>	<i>Hard Brick</i>	<i>Hard Brick</i>	Sesuai
<i>Client D</i>	<i>Soft Brick</i>	<i>Soft Brick</i>	Sesuai
<i>Client E</i>	<i>Hard Brick</i>	<i>Hard Brick</i>	Sesuai
<i>Client F</i>	<i>Freeze</i>	<i>Freeze</i>	Sesuai

Berdasarkan Tabel 10 hasil yang didapatkan dari perbandingan data uji sistem dengan data diagnosis yang didapat dari pakar memiliki kesamaan dalam melakukan diagnosis. Berdasarkan hasil tersebut dari 6 data uji, terdapat 6 data *client* yang memperoleh diagnosis yang sama sehingga tingkat akurasi yang didapatkan sebesar 100% sistem dapat melakukan diagnosis berdasarkan pola pikir pakar.

5. Kesimpulan

Berdasarkan pada hasil perancangan, implementasi, dan pengujian pada penelitian "Implementasi Metode *Certainty Factor* untuk Diagnosis Kerusakan *Software* pada *Smartphone* Berbasis Web", dapat ditarik kesimpulan bahwa sistem yang dibuat dengan menuangkan pola pikir pakar dalam mendiagnosis sebuah kerusakan yang terjadi pada *smartphone* dapat mendiagnosis sebanyak 6 kemungkinan jenis kerusakan yang terjadi yaitu *Light Bootloop*, *Medium Bootloop*, *Hard Bootloop*, *Soft Brick*, *Hard*, dan *Freeze*, hasil perhitungan merupakan persentase keyakinan sebuah sistem dalam melakukan diagnosis. Pengujian dilakukan menggunakan data perhitungan manual dan sistem kemudian dicocokkan hasilnya dengan diagnosis pakar, tingkat akurasi yang diperoleh dari 2 buah pengujian adalah 83,3 %, sedangkan untuk pencocokan hasil yang diperoleh dari sistem dan pakar memiliki tingkat akurasi 100% sistem dapat melakukan diagnosis berdasarkan pola pikir pakar, sehingga masyarakat awam dapat melakukan diagnosis tanpa harus bertemu teknisi *smartphone*.

6. Daftar Pustaka

- Adi, H. P., & Diwandari, S. (2020). Naskah publikasi rancang bangun sistem diagnosa kerusakan *handphone* menggunakan metode *certainty factor* [Teknologi Yogyakarta]. In *Technologi*. <http://eprints.uty.ac.id/id/eprint/6295>

- Afriansyah, A., Annisa, R., & Mair, Z. R. (2019). Sistem Pakar Deteksi Kerusakan (Troubleshooting) Pada Smartphone Berbasis Mobile. *Teknologi Informasi Dan Komputer*, 9(1), 17–25. <https://jurnal.polsky.ac.id/index.php/index/article/view/186>
- Arifin, J. (2016). Sistem Pakar Diagnosa Penyakit Gigi dan Mulut Manusia Menggunakan Knowledge Base System dan Certainty Factor. *Jurnal Ilmiah Teknologi Informasi Asia*, 10(2), 50–64. <https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/86>
- G Buchanan, B., & H Shortliffe, E. (1984). *Rule-based Expert System – The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley. https://www.researchgate.net/publication/44466702_Rule-based_Expert_System_-_The_MYCIN_Experiments_of_the_Stanford_Heuristic_Programming_Project
- Herliana, A., & Rasyid, P. M. (2016). Sistem Informasi Monitoring Pengembangan Software Pada Tahap. *Jurnal Informatika*, 3(1), 41–50. <https://doi.org/10.31294/ji.v3i1.281>
- Nengsih, Y. G., & Putra, N. (2020). Sistem Pakar Menggunakan Forward Chaining Dan Certainty Factor Untuk Diagnosa Kerusakan Smartphone. *Sistem Informasi Dan Manajemen*, 8(2), 61–69. <https://doi.org/doi.org/10.47024/js.v8i2.205>
- Putri, N. A. (2018). Sistem Pakar untuk Mengidentifikasi Kepribadian Siswa Menggunakan Metode Certainty Factor dalam Mendukung Pendekatan Guru. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 78–90. <https://doi.org/10.31539/intecom.v1i1.164>
- Septiani, M., & Kuryanti, S. J. (2018). Sistem Pakar untuk Mendiagnosa Penyakit Saluran Pernapasan pada Anak. *Publikasi Jurnal & Penelitian Teknik Informatika*, 2(2), 23–27. <https://jurnal.polgan.ac.id/index.php/sinkron/article/download/99/62/>
- Wati, V. W., & Tusaadiah, H. (2017). Sistem Pakar Mendeteksi Kerusakan Pada Smartphone Dengan Metode Forward Chaining. *Jurnal Sains Dan Teknologi: Jurnal Keilmuan Dan Aplikasi Teknologi Industri*, 17(2), 98. <https://doi.org/10.36275/stsp.v17i2.31>
- Yuwono, D. T., Fadlil, A., & Sunardi, S. (2017). Penerapan Metode Forward Chaining Dan Certainty Factor Pada Sistem Pakar Diagnosa Hama Anggrek *Coelogyne Pandurata*. *Klik - Kumpulan Jurnal Ilmu Komputer*, 4(2), 136. <https://doi.org/10.20527/klik.v4i2.89>
- Zulkifli. (2018). Rancang Bangun Website E-Learning Dengan Pemodelan Uml (Studi Kasus Di Stkip Muhammadiyah Muara Bungo). *IntecomS: Journal of Information Technology and Computer Science*, 1(2), 159–167. <https://doi.org/10.31539/intecom.v1i2.291>