

METODE FORWARD CHAINING PADA APLIKASI MOBILE PENDETEKSI KERUSAKAN HANDPHONE

Divania Alfiza Sumarno^{1*}

¹Program Studi Teknik Informatika, Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: ^{1*}divaniaalfiza04@gmail.com

(* : corresponding author)

Abstrak-Kemajuan teknologi, terutama dalam bidang *handphone Android*, telah merubah cara hidup kita dengan cepat dan mendalam. Meskipun begitu, kerusakan yang kerap menimpa *handphone* dapat menjadi hambatan yang mengganggu, dengan para pengguna sering kali mengalami kesulitan dalam mengatasi masalah tersebut. Dalam rangka mengatasi permasalahan ini, hadirilah solusi inovatif berupa "Sistem Pengambilan Keputusan untuk Mendiagnosa Kerusakan pada *Handphone*", sebuah platform berbasis *mobile* yang menggunakan metode *forward chaining* yang canggih. Penelitian ini bertujuan memberikan panduan komprehensif bagi pengguna *Android* dalam mendiagnosa serta memperbaiki gangguan yang terjadi pada perangkat *handphone*. Dengan pendekatan *forward chaining*, sistem pakar dibangun untuk mengambil keputusan berdasarkan informasi gejala dan fakta yang diberikan oleh pengguna. Data mengenai beragam gejala dan jenis kerusakan diperoleh melalui wawancara dengan teknisi *handphone* yang berpengalaman. Hasilnya adalah struktur pohon keputusan yang menjadi panduan berharga bagi pengguna dalam memilih gejala yang relevan, sehingga mereka dapat mendiagnosa permasalahan yang mungkin terjadi pada *handphone* mereka. Sistem ini diimplementasikan sebagai aplikasi *Android* yang menggunakan *database Firebase*. Melalui serangkaian pengujian *blackbox*, sistem terbukti mampu memberikan diagnosa yang akurat dan memberikan panduan berharga bagi pengguna dalam mengatasi masalah kerusakan pada perangkat *Android* mereka. Keberhasilan sistem ini diharapkan mampu menghemat waktu, tenaga, dan biaya yang kerap kali terbuang sia-sia saat menghadapi permasalahan *handphone*. Dengan adopsi solusi ini, diharapkan para pengguna *handphone Android* di seluruh dunia dapat merasakan manfaat positif yang signifikan dalam mengatasi masalah kerusakan perangkat mereka.

Kata Kunci: Sistem Pengambilan Keputusan, Diagnosa, *Forward Chaining*

FORWARD CHAINING METHOD IN MOBILE APPLICATIONS FOR PHONE DAMAGE DETECTION

Abstract-The advancement of technology, particularly in the realm of *Android smartphones*, has rapidly and profoundly transformed our way of life. However, amidst this progress, the frequent occurrence of phone damage can pose a disruptive hurdle, often leaving users struggling to resolve such issues. To address this challenge, an innovative solution emerges in the form of a "Decision-Making System for Diagnosing Phone Damage", a mobile-based platform employing advanced forward chaining methodology. This research aims to provide a comprehensive guide for *Android* users in diagnosing and rectifying disruptions experienced by their mobile devices. Through the forward chaining approach, an expert system is constructed to make decisions based on symptom-related information and facts provided by users. Data concerning various symptoms and types of damage are gathered through interviews with experienced phone technicians. The outcome is a decision tree structure that serves as a valuable guide for users in selecting relevant symptoms, enabling them to diagnose potential problems their phones might be encountering. This system is implemented as an *Android* application utilizing the *Firebase* database. Through a series of *blackbox* testing, the system proves its capability in delivering accurate diagnoses and valuable guidance to users in resolving issues related to their *Android* devices. The success of this system is expected to save time, effort, and costs often wasted when dealing with phone-related problems. With the adoption of this solution, it is hoped that *Android* phone users worldwide can experience significant positive benefits in addressing issues related to device damage.

Keywords: Decision Making System, Diagnosis, *Forward Chaining*

1. PENDAHULUAN

Pada saat ini, teknologi termasuk *handphone Android*, berkembang dengan cepat dan pesat. Menurut statcounte.com, pengguna *android* menyumbang 91,85% data di Indonesia, menunjukkan dominasi pengguna *android* di pasar *handphone*. Mengingat banyaknya harga *handphone* yang semakin murah dan canggih, jumlah ini masih dapat meningkat. Kerusakan pada *handphone* sering terjadi karena kesalahan pabrik atau kelalaian pengguna. Ironisnya, banyak pengguna yang tidak tahu cara terbaik untuk memperbaiki kerusakan ini dan cenderung memilih membeli perangkat baru, meskipun kerusakan yang dialami hanya sedikit. Maka dari itu

dibutuhkan suatu sistem yang dapat menghasilkan pengetahuan untuk menangani permasalahan yang timbul dari *handphone* [1]

Dalam hal ini, penelitian tentang “Sistem Pengambilan Keputusan Untuk Diagnosa Kerusakan Pada *Handphone*”. Tujuan dari penelitian ini adalah untuk memberi orang – orang yang menggunakan ponsel Android dan yang bekerja untuk memperbaiki ponsel mereka saran yang tepat. Sistem yang akan dibangun akan berbasis *mobile* dan dapat diakses secara luas melalui metode *forward chaining*. Sistem pakar adalah cabang AI yang berusaha mengadopsi pengetahuan manusia ke computer untuk menyelesaikan masalah seperti yang biasa diselesaikan oleh para pakar [2], Kecerdasan buatan yaitu kecerdasan buatan yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia [3].

Tujuan dari pembuatan sistem pakar ini adalah. Membangun sebuah sistem pakar yang akan membantu pengguna mendiagnosa kerusakan pada *handphone* dengan menggunakan metode *forward chaining*. Sehingga mereka dapat membuat *diagnose* yang lebih valid. Dan juga menguji kekuatan sistem dan ketepatan diagnosa sistem pakar untuk mendiagnosa kerusakan pada *handphone*. Dan nantinya bermanfaat untuk membantu menjawab pengguna untuk menghindari penipuan teknisi yang tidak bertanggung jawab, serta menghemat tenaga, waktu, dan biaya bagi pengguna. Dengan mempertimbangkan informasi sebelumnya, masalah dapat diidentifikasi sebagai berikut:

- a. Merekomendasikan perbaikan *handphone* android yang tepat bagi pengguna.
- b. Belum adanya sistem yang handal untuk mendiagnosa *handphone*.

Seperti yang disebutkan sebelumnya, dapat dirumuskan masalah berikut. Pertama bagaimana merealisasikan aplikasi yang dapat mendiagnosa kerusakan *handphone*? Kedua, bagaimana menguji kelayakan sistem dan ketepatan *diagnose* pada sistem pakar *diagnose* kerusakan *handphone*?

2. METODE

Dalam penelitian kali ini menggunakan metode *forward chaining*, metode ini dimulai dari fakta-fakta yang telah diketahui, untuk mendapatkan suatu fakta baru dengan memakai *rule-rule* yang memiliki ide dasar yang cocok dengan fakta dan terus dilanjutkan sampai mendapatkan tujuan atau sampai tidak ada *rule* yang punya ide dasar yang cocok atau sampai mendapatkan fakta [4]. Metode ini adalah yang paling banyak digunakan dalam sistem pakar. Dengan pendekatan ini, proses pelacakan berawal dari premis menuju kesimpulan akhir, juga dikenal sebagai pencarian yang dikendalikan oleh data yang diberikan. Aktivitas sistem dimulai dengan mencari semua aturan yang kondisinya disimpan dalam *database*, lalu memilih salah satunya dan melakukan tindakan yang sesuai dengan aturan tersebut. Strategi penyelesaian konflik adalah pendekatan permanen untuk memilih aturan yang akan diterapkan [5]. Dikutip dari Amriana A *forward chaining* merupakan salah satu Teknik yang digunakan dalam mekanisme inferensi untuk pengujian aturan [6].

Lalu aplikasi ini juga berjalan di sistem operasi android, yang merupakan sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Dan untuk *databasenya* menggunakan *firebase* dikenal juga sebagai BaaS (*Backend as a Service*), adalah layanan yang ditawarkan oleh google yang membantu para pengembang aplikasi mempercepat proses pengembangan aplikasi mereka. Setiap pengguna yang terhubung menerima data yang disinkronkan secara real – time, yang disimpan dan dieksekusi dalam format JSON.

Untuk metode pengembangan perangkat lunak, dapat menggunakan metode *extreme programming*, yang bertujuan menyederhanakan berbagai tahapan pengembangan untuk menjadi lebih fleksibel dan adaptif [7] selanjutnya, seperti yang dinyatakan oleh Ferdinana dalam Lubis[8], *extreme programming* dikenal sebagai cara suatu tim teknis mengembangkan perangkat lunak dengan memanfaatkan berbagai prinsip dan Teknik praktis pengembang perangkat lunak. Terdapat empat tahapan yang dikerjakan pada metode *extreme programming* yaitu:

- a. *Planning*: Langkah awal dalam pembangunan sistem adalah perencanaan, dimana beberapa tugas perencanaan dilakukan, seperti mengidentifikasi masalah, mengevaluasi kebutuhan, dan menetapkan jadwal pelaksanaan pembangunan sistem.
- b. *Design*: Langkah berikutnya dalam desain adalah perancangan, dimana kegiatan permodelan dilakukan, mulai dari permodelan sistem, arsitektur, dan basis data, permodelan sistem dan arsitektur dilakukan dengan menggunakan *diagram Unified Modeling Language* (UML). Sedangkan permodelan basis data dilakukan dengan *Entity Relationship Diagram* (ERD)
- c. *Coding*: Proses penerapan permodelan yang sudah dibuat ke dalam UI dengan menggunakan bahasa pemrograman adalah PHP dengan metode terstruktur. Piranti lunak MySQL digunakan untuk mengelola sistem basis data.

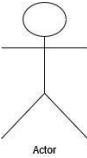
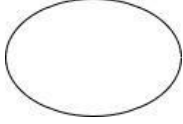
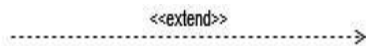
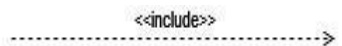
- d. *Testing*: Setelah tahapan selesai, langkah berikutnya adalah pengujian sistem. Ini dilakukan untuk mengidentifikasi kesalahan apa pun yang terjadi saat aplikasi berjalan, serta untuk memastikan apakah sistem yang dibangun sudah memenuhi kebutuhan pengguna. Pengujian dilakukan terhadap berbagai form untuk memastikan bahwa semuanya berjalan sesuai dengan fungsinya.

Unified Modelling Language sendiri adalah Bahasa permodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Permodelan sesungguhnya digunakan untuk penyederhanaan permasalahan – permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Tujuan utama dalam desain UML adalah:

- Menyediakan bagi pengguna (analisis dan desain sistem) suatu Bahasa permodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
- Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
- Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
- Memberikan dasar ormal untuk pemahaman bahasa pemodelan.
- Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek.
- Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola, dan komponen terhadap suatu sistem.
- Memiliki integrasi praktik terbaik.

UML mempunyai berbagai macam model, contohnya adalah *usecase diagram* yang merupakan permodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi[9]. Berdasarkan definisi tersebut maka dapat disimpulkan bahwa *usecase diagram* merupakan *diagram* yang menggambarkan fungsi apa saja dan actor siapa saja yang ada pada sistem yang akan dibangun. Berikut ini Tabel 1 merupakan simbol – simbol pada *usecase diagram* menurut Lubis: [8]

Table 1 Tabel Simbol *Usecase Diagram*

No	Gambar	Nama	Keterangan
1		Actor	Mewakili peran orang, system yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2		Use case	Abstraksi dan interaksi antara system dan actor.
3		Association	Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
4		Extend	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
5		Generalization	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
6		Include	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

Lalu untuk pengetesannya sendiri dapat menggunakan *blackbox testing* untuk pengujian perangkat lunak dengan menggunakan *blackbox testing* dapat menemukan error (kesalahan) dalam beberapa kategori, antara lain fungsi yang salah ataupun hilang, kesalahan dalam struktur data atau dalam mengakses *database* eksternalnya, kesalahan inisialisasi dan terminasi, dan kesalahan dalam *interface*[10]. Menurut Mustaqbal dalam *blackbox testing* digunakan untuk mendeteksi permasalahan berikut:

- Fungsi yang salah atau hilang

- b. Kesalahan pada *interface*
- c. Kesalahan struktur data dan basis data
- d. Kesalahan fungsi
- e. Kesalahan deklarasi dan terminasi

3. PEMBAHASAN DAN HASIL

3.1. Langkah – Langkah *Forward Chaining*

Jika ingin membuat sistem menggunakan penerapan perhitungan *forward chaining*, maka langkah – langkah yang harus disiapkan adalah:

- a. Mencari pakar yang sesuai dengan tema sistem yang akan dibuat.
- b. Mendapatkan data hipotesa / konklusi / kesimpulan sesuai bidang ahli si pakar.
- c. Mendapatkan data premis / gejala / evidence pada tiap hipotesa.
- d. Mendapatkan data *rule* / aturan dari pakar berdasarkan data premis dan konklusi.
- e. Setelah mendapat semua data yang dibutuhkan, langkah selanjutnya adalah menghitung data tersebut ke dalam teori *forward chaining*, sehingga menjadi konklusi paling tepat dengan berdasarkan premis inputan user.

3.2. Analisis Data

Berdasarkan penelitian yang sudah di terangkan sebelumnya, maka data yang dibutuhkan untuk penelitin ini adalah 15 kerusakan yang umum terjadi di *handphone* beserta gejala, dan penyebab. Data ini diperoleh melalui wawancara dengan teknisi *handphone* yang akan bertindak sebagai pakar di dalam penelitian ini. Berikut tabel 2 adalah data – data yang telah diperoleh.

Tabel 2 Jenis Kerusakan

Kode	Kerusakan
K001	Sinyal Hilang (<i>No Signal / No Network</i>)
K002	Sinyal Berubah
K003	Sinyal Tidak Stabil
K004	Sinyal Penuh Tapi Tidak Dapat Melakukan Panggilan
K005	Ponsel Mati Total Karena Mati Sendiri
K006	Ponsel Mati Total Kena Air
K007	Ponsel Boros Baterai
K008	Speaker Rusak
K009	Ponsel Tidak Bisa Bergetar
K010	Ringtone Mati
K011	Lampu Atau Led Mati
K012	Charging Gagal
K013	Kegagalan Sirkuit Sim Card
K014	Tampilan Informasi " <i>Contact Service</i> "
K015	Masalah <i>Keypad</i>

Data Tabel 3 diperoleh setelah mengolah data wawancara 15 kerusakan yang umum terjadi di *handphone*. Tiap jenis kerusakan dilambangkan dengan menggunakan kode K01 sampai K15, seperti yang tertulis di tabel 2. Selanjutnya adalah data gejala – gejala kerusakan yang dirangkum dari 15 kerusakan yang terdapat di tabel 3. Data gejala ini diperoleh melalui wawancara dengan pakar yaitu teknisi *handphone*.

Tabel 3 Gejala

No	Kode	Pertanyaan
1	G-01	Apakah ponsel anda bermasalah di bagian sinyal?
2	G-02	Apakah ponsel anda mati total atau tidak dapat hidup?
3	G-03	apakah ponsel anda bermsalan di baterai?

No	Kode	Pertanyaan
4	G-04	apakah ponsel anda bermasalah di bagian audio?
5	G-05	apakah ponsel anda bermasalah di bagian vibra?
6	G-06	apakah ponsel anda bermasalah di lampu LED?
7	G-07	apakah ponsel anda bermasalah di bagian SOFTWARE?
8	G-08	apakah ponsel anda bermasalah di bagian keypad?
9	G-09	Apakah komponen konektor dan antenna terhubung dengan baik ?
10	G-10	Dapatkah ponsel mencari jaringan secara manual ?
11	G-11	Apakah ponsel dapat mencari jaringan secara manual,namun kadang gagal?
12	G-12	Apakah ponsel hanya dapat menangkap satu jaringan operator/penyedia jasa layanan selular ?
13	G-13	Apakah tampilan pada LCD ponsel level sinyal penuh, tetapi saat melakukan panggilan tiba-tiba semua sinyal hilang ?
14	G-14	Apakah ponsel pernah jatuh ke dalam air ?
15	G-15	Apakah ponsel tidak pernah jatuh ?
16	G-16	Apakah seringkali sinyal ponsel terkadang kuat kemudian melemah ?
17	G-17	Apakah ponsel sudah berada dalam jangkauan sinyal, tetapi sinyal tetap tidak stabil ?
18	G-18	Apakah ponsel tidak dapat mencari jaringan secara manual ?
19	G-19	Apakah indikator sinyal ponsel anda penuh tetapi tidak dapat melakukan panggilan ?
20	G-20	Apakah pengaturan sudah benar, tetapi ponsel tetap tidak dapat melakukan panggilan ?
21	G-21	Apakah ponsel tidak dapat mencari sinyal secara manual ?
22	G-22	Apakah baterai masih dalam keadaan baik ?
23	G-23	Apakah pada konektor terlihat baterai berkarat ?
24	G-24	Apakah pada saat ponsel di charge, indikator proses isi ulang tampil tetapi ponsel tidak dapat dinyalakan ?
25	G-25	Apakah proses charging tidak berjalan ?
26	G-26	Apakah ponsel Anda pernah terjatuh ke dalam air ?
27	G-27	Apakah setelah dinyalakan kembali ponsel mati total ?
28	G-28	Apakah ponsel dapat hidup kembali tetapi layar blank ?
29	G-29	Apakah baterai ponsel anda cepat sekali habis ?
30	G-30	Apakah power supply menunjukan keadaan baterai normal tetapi baterai tetap boros ?
31	G-31	Apakah pada saat melakukan charge lama penuhnya?
32	G-32	Apakah saat melakukan panggilan lawan bicara kita tidak dapat mendengar suara apapun ?
33	G-33	Apakah komponen MIC dalam keadaan baik ?
34	G-34	Apakah komponen Audio dalam keadaan baik ?
35	G-35	Apakah dalam pengaturan ponsel, fungsi getar telah di aktifkan ?
36	G-36	Apakah saat menerima sms/telepon ponsel tidak bergetar ?
37	G-37	Apakah komponen dan jalur VIBRA dalam keadaan baik tetapi ponsel masih tidak bergetar ?
38	G-38	Apakah ponsel tidak mengeluarkan bunyi saat ada sms/panggilan masuk ?
39	G-39	Apakah komponen BUZZER tidak berfungsi dengan baik ?
40	G-40	Apakah jalur komponen BUZZER tidak berfungsi dengan baik ?
41	G-41	Apakah IC UI masih dalam keadaan baik ?
42	G-42	Apakah lampu LED tidak menyala ?
43	G-43	Apakah komponen LED tidak berfungsi dengan baik ?
44	G-44	Apakah jalur komponen LED tidak berfungsi dengan baik ?
45	G-45	Apakah IC UI masih dalam keadaan baik ?

No	Kode	Pertanyaan
46	G-46	Apakah alat charger anda tidak dapat mengisi baterai pada ponsel anda?
47	G-47	Apakah saat anda mencoba melakukan isi ulang baterai pada ponsel lain sejenis, alat charger anda berfungsi dengan baik ?
48	G-48	Apakah pada layar ponsel muncul pesan “Not Charging” ?
49	G-49	Apakah ponsel telah selesai diisi ulang, tetapi indicator pengisian tetap berjalan ?
50	G-50	Apakah saat ponsel tidak dalam keadaan isi ulang (<i>charge</i>), <i>indicator</i> isi ulang jalan terus ?
51	G-51	Apakah ponsel selalu mati setiap kali akan diisi ulang (<i>charge</i>) ?
52	G-52	Apakah lempengan sim <i>card reader</i> bengkok atau kotor ?
53	G-53	Apakah resisten SIM <i>lines</i> ke <i>ground</i> nilainya < 200 ?
54	G-54	Apakah tegangan VSIM naik menjadi 3/5 volt setelah ponsel dinyalakan?
55	G-55	Apakah ada keterangan “SIM card not accepted” pada LCD ponsel ?
56	G-56	Apakah Checksum MCU ROM gagal dijalankan ?
57	G-57	Apakah <i>interface</i> CCONT gagal dijalankan ?
58	G-58	Apakah parallel/serial COBRA gagal ?
59	G-59	Apakah Checksum sec/tune Eeprom gagal ?
60	G-60	Apakah salah satu atau seluruh keypad ponsel tidak berfungsi ?
61	G-61	Apakah salah satu keypad, jika ditekan output ke LCD tidak sesuai ?
62	G-62	Apakah jika salah satu tombol ditekan, menyebabkan ponsel mati ?

Terdapat 62 gejala dari 15 Kerusakan yang ada pada tabel 2 Gejala – gejala yang telah didapatkan diberikan dengan menggunakan kode G-01 sampai G-62 seperti yang telah diuraikan pada tabel 3.

3.3. Penentuan Rule Dengan Menggunakan Forward Chaining

Dari data di tabel 1 dan tabel 2 sudah bisa melakukan penentuan *rule* dengan metode *forward chaining*, berikut Tabel 4 adalah *rule* yang telah ditentukan dari pengolahan data di tabel sebelumnya.

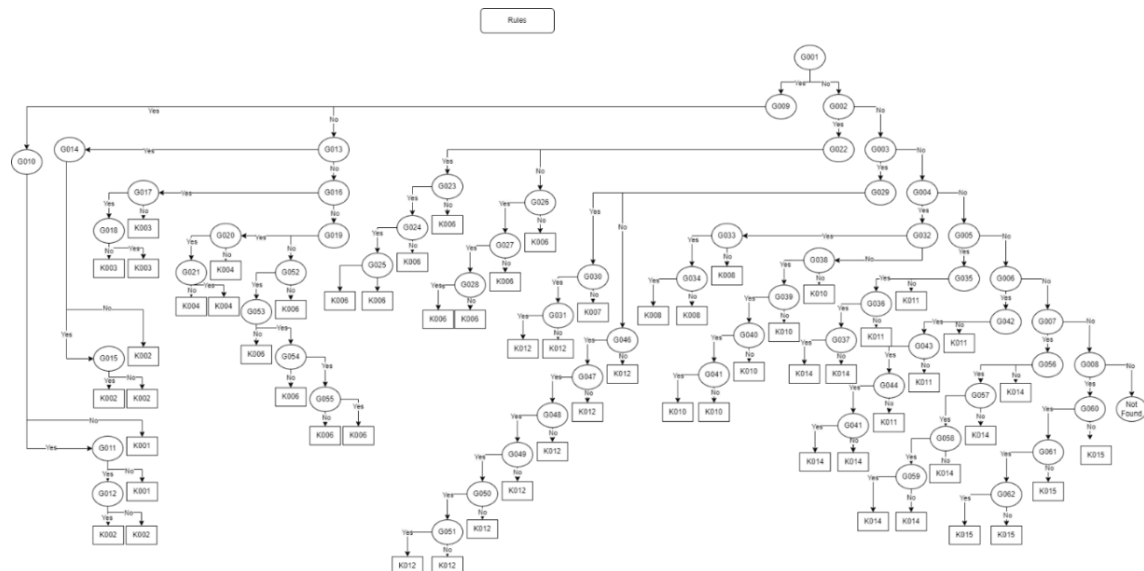
Tabel 4 Rule

No	Rules
1	IF [G001] AND [G010] AND [G011] AND [G012] THEN [K001]
2	IF [G001] AND [G013] AND [G014] AND [G015] THEN [K002]
3	IF [G001] AND [G016] AND [G017] AND [G018] THEN [K003]
4	IF [G001] AND [G019] AND [G020] AND [G021] THEN [K004]
5	IF [G002] AND [G022] AND [G023] AND [G024] AND [G025] THEN [K005]
6	IF [G002] AND [G026] AND [G027] AND [G028] THEN [K006]
7	IF [G003] AND [G029] AND [G030] AND [G031] THEN [K007]
8	IF [G004] AND [G032] AND [G033] AND [G034] THEN [K008]
9	IF [G005] AND [G035] AND [G036] AND [G037] THEN [K009]
10	IF [G004] AND [G038] AND [G039] AND [G040] AND [G041] THEN [K010]
11	IF [G006] AND [G042] AND [G043] AND [G044] AND [G045] THEN [K011]
12	IF [G003] AND [G046] AND [G047] AND [G048] AND [G049] AND [G050] AND [G051] THEN [K012]
13	IF [G001] AND [G052] AND [G053] AND [G054] AND [G055] THEN [K013]
14	IF [G007] AND [G056] AND [G057] AND [G058] AND [G059] THEN [K014]
15	IF [G008] AND [G060] AND [G061] AND [G062] THEN [K015]

Dan dari tabel 4 di atas sudah bisa membuat pohon keputusan *forward chaining* dengan mengolah data – data sebelumnya.

3.4. Perancangan Pohon Keputusan Diagnosa Kerusakan *Handphone*

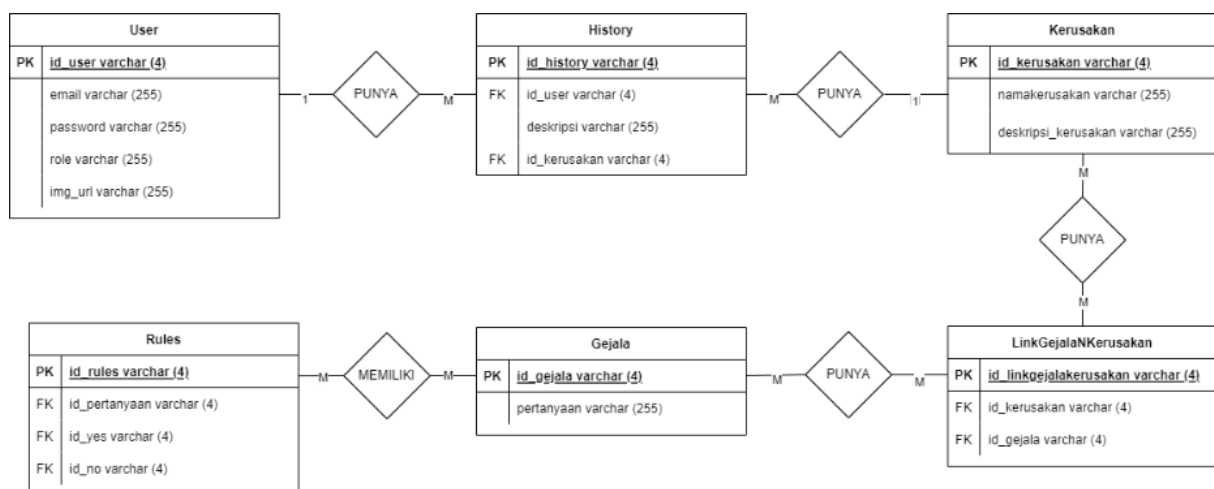
Diagram pohon keputusan adalah suatu rancangan dalam bentuk pohon (*tree*) yang digunakan untuk membangun sistem pakar, *diagram* pohon keputusan akan memberikan kemudahan dalam menyusun kumpulan aturan yang diperlukan untuk menentukan faktor kepastian dari diagnosa kerusakan pada *handphone* berdasarkan gejala-gejala yang ada. Untuk melakukan diagnose kerusakan pada *handphone* maka digunakan *binary tree* yaitu pembentukan pohon keputusan didalam perancangan sistem pakar. Pohon keputusan ini dikombinasikan dengan metode pencarian *best first search*.



Gambar 1 Pohon Keputusan Forward Chaining

3.5. ERD (Entity Relationship Diagram)

Entity-Relationship Diagram (ERD) adalah salah satu jenis *diagram* yang sangat bermanfaat dalam desain *database* atau perencanaan bisnis karena merupakan *diagram* yang lebih struktural dan digunakan untuk menunjukkan hubungan antara entitas (objek) dalam suatu sistem. ERD juga menunjukkan hubungan dan ketergantungan antara entitas, serta atribut yang dimiliki oleh setiap entitas. ERD membantu dalam desain *database* dengan memastikan integritas data dan merancang skema *database* yang efektif.

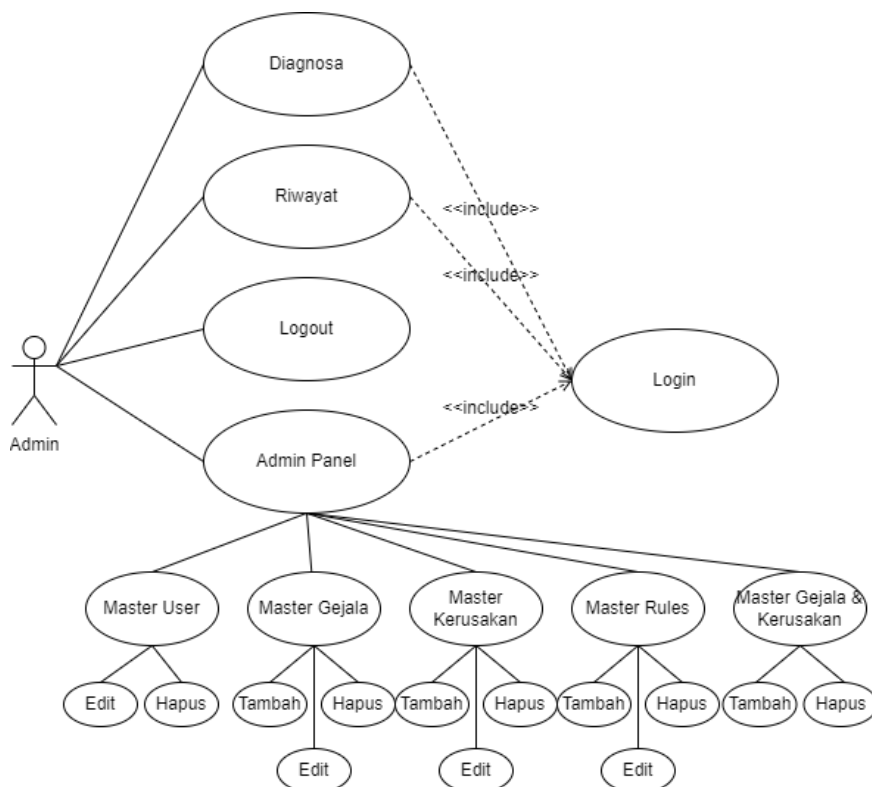


Gambar 2 Entity Relationship Diagram

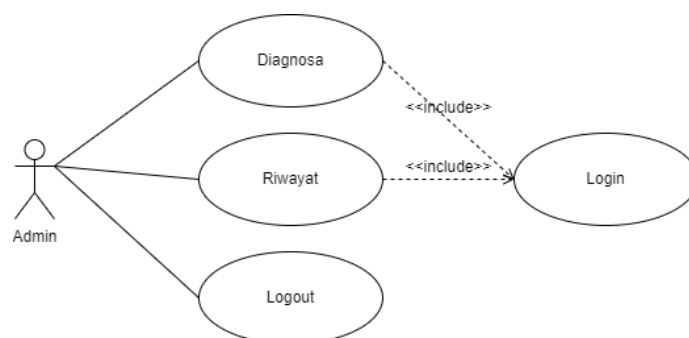
3.6. Use case Diagram

Salah satu jenis *diagram* UML yang sangat berguna untuk menggambarkan hubungan interaksi antara sistem yang sedang dirancang dengan aktor-aktor yang terlibat adalah *use case diagram*. *Diagram* ini menyajikan

skenario-skenario yang menggambarkan interaksi antara aktor dan sistem, dengan aktor dianggap sebagai entitas luar yang berinteraksi dengan sistem. Ini juga membantu dalam memvisualisasikan fungsionalitas sistem. Aktor dapat berupa pengguna, perangkat keras, sistem eksternal, atau bahkan bagian sistem lainnya. *Diagram* ini membantu pengembang perangkat lunak atau analis sistem memahami apa yang dibutuhkan pengguna. Ini membantu mereka merancang dan menerapkan sistem yang sesuai dengan kebutuhan pengguna. Gambar 3 menjelaskan *Usecase Diagram Admin* dan Gambar 4 menjelaskan *Usecase Diagram User*.



Gambar 3 *Usecase Diagram Admin*



Gambar 4 *Usecase Diagram User*

4. KESIMPULAN

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

- Dengan menggunakan aplikasi Sistem Pengambilan Keputusan diagnosa kerusakan *Handphone*, dapat memudahkan pengguna, serta cepat dan akurat untuk mendiagnosa kerusakan *handphone* yang dialami oleh pengguna tanpa harus bertemu dengan mekanik.
- Aplikasi ini mengimplementasikan metode *forward chaining* yang bertujuan untuk memudahkan para pengguna dalam mendiagnosa kerusakan *handphone* yang dalam penyajiannya terdiri dari gejala-gejala yang harus dipilih sesuai dengan kerusakan yang dialami *handphone* pengguna. Lalu proses untuk mendapatkan hasil diagnosa kerusakan dari *handphone* serta penyebabnya.

DAFTAR PUSTAKA

- [1] A. Santika, F. M. Hamundu, and H. Budiman, "SISTEM PAKAR DIAGNOSA KERUSAKAN *HANDPHONE* DENGAN METODE *FORWARD CHAINING*," 2023.
- [2] S. Kusumadewi, *Artificial intelligence: teknik dan aplikasinya*, 1st ed. Yogyakarta: Graha Ilmu, 2003.
- [3] Mustofa and R. A. N. Pamudji, "Sistem Pakar Diagnosa Kerusakan Smartphone Android Menggunakan Metode *Forward Chaining*," *IKRAM: Jurnal Ilmu Komputer Al Muslim*, vol. 1, 2022.
- [4] A. NurJumala, N. A. Prasetyo, and H. W. Utomo, "Sistem Pakar Diagnosis Penyakit Rhinitis Menggunakan Metode *Forward Chaining* Berbasis Web," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 1, 2022, doi: 10.30865/jurikomv9i1.3815.
- [5] A. Amriana, A. Y. E. Dodu, and P. R. Mas, "Pendeteksian Kerusakan Printer menggunakan Metode *Forward Chaining*," *ILKOM Jurnal Ilmiah*, vol. 12, no. 1, 2020, doi: 10.33096/ilkom.v12i1.523.47-57.
- [6] N. Nurhidayati and A. M. Nur, "Pemanfaatan Aplikasi Android Dalam Rancang Bangun Sistem Informasi Persebaran Indekos di Wilayah Pancor Kabupaten Lombok Timur," *Infotek: Jurnal Informatika dan Teknologi*, vol. 4, no. 1, pp. 51–62, Jan. 2021, doi: 10.29408/jitv4i1.2989.
- [7] I. Carolina and A. Rusman, "Penerapan Extreme Programming Pada Sistem Informasi Penjualan Pakaian Berbasis Web (Studi Kasus Toko ST Jaya)," *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, vol. 4, no. 2, 2019.
- [8] I. Carolina, A. Supriyatna, J. Kamal Raya No, and R. Barat Cengkareng Jakarta Barat, "PENERAPAN METODE EXTREME PROGRAMMING DALAM PERANCANGAN APLIKASI PERHITUNGAN KUOTA SKS MENGAJAR DOSEN," 2019.
- [9] Vikasari, "Pengujian Sistem Informasi Magang Industri dengan Metode Blackbox Testing Boundary Value Analysis," 2018.
- [10] M. Nurudin, W. Jayanti, R. D. Saputro, M. P. Saputra, and D. Yulianti, "Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis," *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 4, pp. 143-148, 2019, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika/article/view/3841>.