

Sistem Pakar Pendeteksi Kerusakan Handphone Berbasis Android

Mercyidian Pangkey⁽¹⁾, Vecky Poekoel⁽²⁾, Oktavian Lantang⁽³⁾
Teknik Informatika, Universitas Sam Ratulangi, Manado, Indonesia.

Email: maimunaobit@gmail.com, vecky_poekoel@unsrat.ac.id, oktavian_lantang@unsrat.ac.id

Abstrak --- *Handphone* tidak hanya memiliki fungsi untuk menelepon dan mengirim pesan. Pada beberapa *handphone* kelas atas (*smartphone*) bahkan hampir memiliki fungsi seperti komputer. Dengan adanya *handphone* sangat membantu kelancaran kegiatan manusia. Akan tetapi, hanya sedikit dari orang yang memakai *handphone* yang peka akan gejala – gejala kerusakan pada *handphone*, sehingga kebanyakan orang tidak sadar dan cuek akan gejala kerusakan tersebut hingga *handphone* tersebut benar-benar mati. Karena hal itulah dibutuhkan suatu perangkat lunak yang berupa aplikasi program atau sistem pakar yang sesuai dengan permasalahan yang terjadi pada *handphone*.

Sistem pakar merupakan kecerdasan buatan yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan dari penelitian ini adalah membuat aplikasi sistem pakar berbasis android dalam mendeteksi kerusakan *handphone*. Dengan menggunakan teknik penalaran *forward chaining*, deteksi dilakukan dengan memulai dari sekumpulan gejala-gejala, nantinya dapat melihat kesimpulan jenis kerusakan pada *handphone*. Metode yang digunakan sebagai tahapan penelitian ini adalah metode *Extreme Programming* (XP) yang merupakan metode rancang bangun perangkat lunak yang menekankan pada 4 tahapan dalam pengembangan perangkat lunak.

Kata Kunci: *Handphone*, Sistem Pakar, *Forward Chaining*, *Extreme Programming*

I. PENDAHULUAN

Handphone atau biasa disebut dengan telepon genggam yang dahulu termasuk barang mewah, namun dengan seiring berkembangnya zaman dan semakin bertambahnya perusahaan pembuat *handphone*, membuat harga *handphone* semakin murah sehingga hampir semua orang bisa memilikinya. Bahkan tidak jarang dari mereka memiliki *handphone* lebih dari satu.

Seiring berkembangnya teknologi saat ini *handphone* tidak hanya memiliki fungsi untuk menelepon dan mengirim pesan. Pada beberapa *handphone* kelas atas (*smartphone*) bahkan hampir memiliki fungsi seperti komputer. Dengan adanya *handphone* sangat membantu kelancaran kegiatan manusia. Akan tetapi, hanya sedikit dari orang yang memakai *handphone* yang peka akan gejala – gejala kerusakan pada *handphone*, seperti kerusakan IC Power, LCD, memori, keypad, speaker, Bluetooth dan lain-lain sehingga kebanyakan orang tidak sadar dan cuek akan gejala kerusakan tersebut hingga *handphone* tersebut benar-benar mati.

Kerusakan – kerusakan pada sebuah *handphone* sering kali mengganggu pengguna *handphone*, sehingga

penggunanya hanya membawa *handphone* tersebut ke service *handphone* untuk mengetahui kerusakan apa yang terjadi pada perangkat tersebut. Waktu perbaikan yang habis terpakai selama *handphone* pengguna di tempat service juga dapat menyita waktu pengguna. Belum juga biaya yang akan dikeluarkan untuk memperbaiki perangkat tersebut serta penipuan yang banyak terjadi dikala kita membawa *handphone* ke tempat service.

Berdasarkan permasalahan diatas, dibutuhkan suatu aplikasi yang dapat menghasilkan pengetahuan untuk menangani masalah yang timbul dari kerusakan *handphone*. Aplikasi yang dibuat harus mampu menangani masalah jarak, waktu, tenaga dan biaya yang dikeluarkan serta mudah digunakan bagi seluruh kalangan pengguna *handphone*.

Salah satu sistem operasi *mobile* yang digunakan oleh *smartphone* adalah Android. Android adalah sebuah sistem operasi untuk *smartphone* yang berbasis Linux. Kelebihan Android dibanding sistem operasi *mobilephone* atau *smartphone* lainnya adalah Android bersifat *open source code* sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur – fitur yang belum ada di sistem operasi Android sesuai dengan keinginan mereka sendiri.

Maka dari itu sebagai upaya, diterapkan pembuatan Sistem Pakar Pendeteksi Kerusakan *Handphone* Berbasis Android. Dengan banyaknya pengguna *handphone* canggih (*Smartphone*) saat ini, dapat membantu para pengguna untuk lebih memahami lagi akan setiap gejala kerusakan pada *handphone* agar terhindar dari berbagai macam hal yang tidak diinginkan.

II. LANDASAN TEORI

1. Sistem Pakar

Menurut Kusumadewi (2003), sistem pakar adalah suatu cabang dari *Artificial Intelligence* yang berusaha mengadopsi pengetahuan manusia ke komputer agar dapat menyelesaikan masalah seperti yang biasa diselesaikan oleh para pakar. Dengan adanya sistem ini, orang awam pun dapat menyelesaikan masalah yang rumit yang sebenarnya hanya dapat diselesaikan oleh para ahli. Sedangkan bagi para ahli, sistem pakar dapat membantu aktivitasnya sebagai asisten yang berpengetahuan. Ada beberapa dasar mengapa sistem pakar menggantikan seorang pakar diantaranya adalah : [1]

1. Dapat menyediakan kepakaran setiap waktu diberbagai lokasi
2. Secara rutin mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar
3. Seorang pakar akan pensiun atau pergi
4. Seorang pakar adalah mahal

5. Kepakaran juga dibutuhkan pada lingkungan yang tidak bersahabat (hostile environment)

Dengan demikian seorang awam sekalipun bisa menggunakan sistem pakar itu untuk memecahkan berbagai persoalan yang ia hadapi dan bagi seorang ahli, sistem pakar dapat dijadikan alat untuk menunjang aktivitasnya yaitu sebagai asisten yang berpengalaman.

2. Mesin Inferensi

Menurut Turban menyebutkan mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk merumuskan kesimpulan. Terdapat dua pendekatan dalam mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu :[4]

- Pelacakan kedepan (forward chaining)

Pelacakan kedepan adalah pendekatan yang terkendali-data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian *if* dari aturan *if-then*.

- Pelacakan kebelakang (backward chaining)

Pelacakan kebelakang adalah pendekatan terkendali-tujuan (*goal driven*). Dalam pendekatan ini, pelacakan dimulai dari tujuan selanjutnya dicari aturan yang memiliki tujuan tersebut untuk tujuannya.[4]

3. Android

Android merupakan sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi linux, namun telah dimodifikasi. Android diambil alih oleh google pada tahun 2005 dari android. Ini sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja android termasuk tim yang mengembangkan android. Secara garis besar sistem operasi android menjadi 5 tingkatan:[3]

- a) Linux Kernel

Linux kernel adalah kernel dasar dari android. Tingkat ini berisi semua driver perangkat tingkat rendah untuk komponen-komponen perangkat android.

- b) Libraries

Libraries berisi semua kode program yang menyediakan layanan-layanan utama sistem operasi android.

- c) Android Runtime

Android runtime kedudukannya sama dengan libraries, android runtime menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi dengan bahasa pemrograman java.

- d) Application Framework

Tingkat dimana proses pengerjaan pembuatan aplikasi dilakukan..[3]

4. Extreme Programming

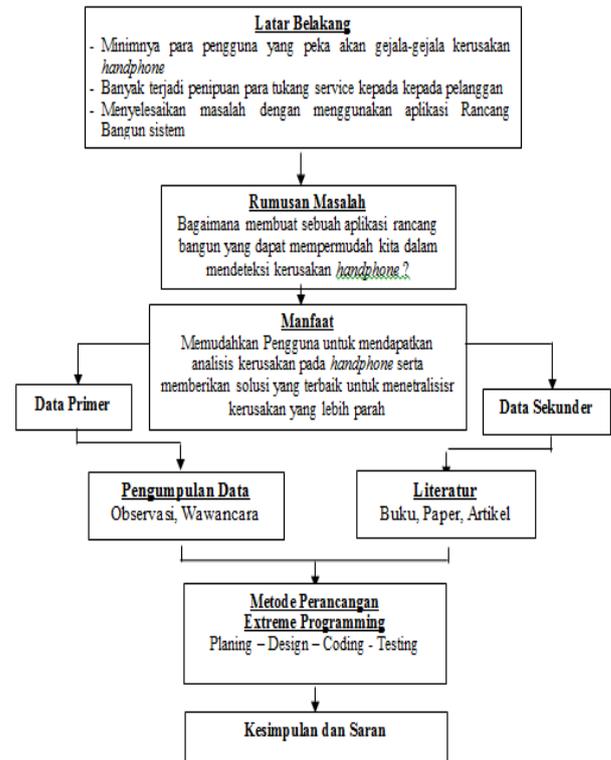
Model proses ini diciptakan dan dikembangkan oleh Kent Beck. *Extreme Programming (XP)* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih

adaptif dan fleksibel. Walaupun menggunakan kata *programming*, *XP* tidak hanya terfokus pada *coding* tetapi meliputi seluruh area pengembangan perangkat lunak.

Menurut Kent Beck, *XP* ringan, efisien, resiko rendah, mudah disesuaikan, dapat diprediksi, ilmiah dan mudah dikembangkan. Suatu model yang menekankan pada keterlibatan *user* secara langsung. [2]

III.METODOLOGI PENELITIAN

1. Kerangka Pemikiran



Gambar 1. Kerangka Pikir

2. Metode Perancangan Sistem

Dalam metode perancangan sistem ini, penulis menggunakan metode Extreme Programming (XP). Model proses *extreme programming* terbentuk dari sebuah kerangka kerja yang memiliki 4 konteks aktivitas utama, yaitu :

1. Planning.

Aktivitas planning pada model proses XP berfokus pada mendapatkan gambaran fitur serta fungsi dari perangkat lunak yang akan dibangun. Pada aktivitas ini dimulai dengan membuat kumpulan cerita atau gambaran yang diberikan klien yang kemudian akan menjadi gambaran dasar dari perangkat lunak

2. Design

Proses desain pada XP mengikuti prinsip KIS (*Keep It Simple*). Desain akan berisikan semua implementasi dari stories tanpa ada pengurangan maupun penambahan. Desain yang memiliki fungsi tambahan tidak disarankan. *XP* menggunakan *CRC (Class-Responsibility-Collaborator) Cards* untuk mengidentifikasi dan mengorganisasikan kelas berorientasi objek yang berkaitan dengan proses pengembangan *software*. Jika terdapat kesulitan untuk melakukan desain terhadap *stories*, *XP* menyarankan untuk membuat *prototype* dari desain tersebut. Hal ini disebut sebagai *spike solution*,

prototype nantinya akan diimplementasikan dan dievaluasi. Perubahan desain dapat terjadi walaupun sudah memasuki tahap *coding/* implementasi. Hal tersebut dilakukan untuk mendapat desain yang baik dan kode yang bersih.

3. Coding

Setelah menyelesaikan pengumpulan cerita dan menyelesaikan *design* untuk aplikasi secara keseluruhan, XP lebih merekomendasikan tim untuk terlebih dahulu membuat modul unit tes yang bertujuan untuk melakukan uji coba setiap cerita yang didapat dari klien. Setelah berbagai unit tes selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan *coding* aplikasi. XP menerapkan konsep *pair programming* dimana setiap tugas sebuah modul dikembangkan oleh dua orang *programmer*. XP beranggapan, 2 orang akan lebih cepat dan baik dalam menyelesaikan sebuah masalah. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.

4. Testing

Setelah semua modul telah dikumpulkan dalam sebuah sistem yang sempurna, barulah pengujian penerimaan (*acceptance test*) dilakukan. Pada tahapan pengujian ini aplikasi langsung diuji coba oleh pengguna atau klien dan mendapat tanggapan langsung mengenai penerapan cerita yang telah digambarkan sebelumnya.

3. Definisi dan Aturan Gejala Kerusakan

Dalam pembahasan ini akan menjelaskan data mentah mengenai jenis-jenis kerusakan dan gejalanya yang ada pada *handphone*. Data inilah yang nantinya akan dibuat program dan didahului dengan adanya relasi data antara gejala dan kerusakan yang terjadi.

1. Data Gejala Kerusakan

Dibawah ini adalah daftar gejala-gejala semua jenis kerusakan yang terjadi :

Table 1. Daftar Kerusakan

Kode	Jenis Kerusakan
Q1	Tidak Bisa Cars
Q2	LCD Bergaris
Q3	Insert SIM
Q4	Pada saat melakukan panggilan, hp langsung mati
Q5	Mati total
Q6	Bercak hitam pada LCD
Q7	Lampu LCD berkedip-kedip
Q8	Penekanan pada huruf menjadi acak
Q9	Data tiba-tiba hilang
Q10	Tulisan bergaris/ berantakan
Q11	Sebagian tombol tidak berfungsi
Q12	Tidak bisa diakses pada computer atau laptop
Q13	LCD blank/ mati
Q14	Layar tidak jelas
Q15	Sebagian tombol sulit ditekan
Q16	Tidak dapat membuka gallery pada HP
Q17	Koneksi Bluetooth sering gagal dan terputus
Q18	Kinerja HP lambat dan tidak optimal
Q19	HP tidak dapat booting ke menu

Q20	HP hidup mati sendiri
Q21	Hanya bisa mengirim file tertentu saja
G22	Sinyal hilang tiba-tiba
G23	Saat mengaktifkan bluetooth HP menjadi hang atau macet total
G24	Hanya bisa menerima file tertentu saja
G25	HP tiba-tiba mati padahal isi baterai masih ada
G26	Saat Bluetooth diaktifkan, HP terestart sendiri
G27	Tidak ada reaksi ketika ditekan

2. Data Kerusakan Handphone

Pada bagian ini di sajikan daftar kerusakan yang sering terjadi pada *handphone* secara umum. Daftar nama kerusakan ini di beri nomor urut otomatis, saya menggunakan 'A' untuk urutan pertama, 'B' untuk urutan kedua dan seterusnya

Tabel 2. Daftar Kersakan Handphone

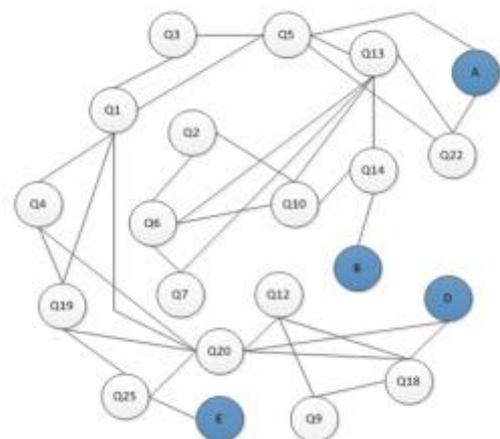
Kode	Jenis Kerusakan
A	IC Power
B	LCD
C	Keypad
D	Memory
E	Baterai
F	Bluetooth

3. Representasi Pengetahuan

Representasi pengetahuan yang akan digunakan dalam sistem pakar ini adalah dengan menggunakan jaringan semantik, pohon keputusan, dan kaidah produksi. Representasi pengetahuan dapat menemukan solusi atau memberikan penalaran yang tepeat dalam pengambilan kesimpulan.

a. Jaringan Semantik

Suatu proses terhadap basis pengetahuan atau informasi yang didapat dari pakar, terlebih dahulu diubah kedalam bentuk jaringan semantik. Jaringan semantik menunjukkan hubungan antara gejala dan kerusakan.

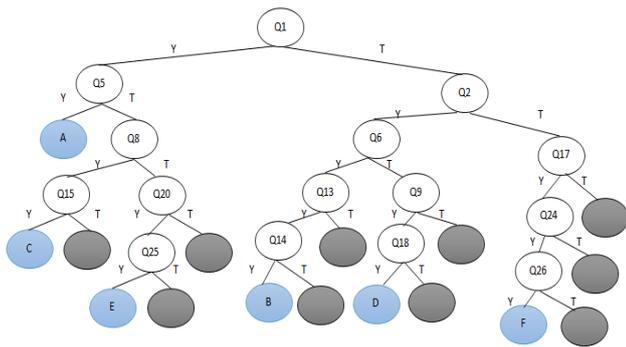


Gambar 2. Jaringan Semantik

b. Pohon Keputusan

Setelah proses terhadap basis pengetahuan telah diubah kedalam bentuk jaringan semantik, selanjutnya diubah kedalam bentuk pohon keputusan, sehingga didalam penyelesaian masalah

lebih mudah dilakukan penelusuran untuk mendapatkan solusi atau kesimpulan akhir yang terbaik.



Gambar 3. Pohon Keputusan

IV. HASIL DAN PEMBAHASAN

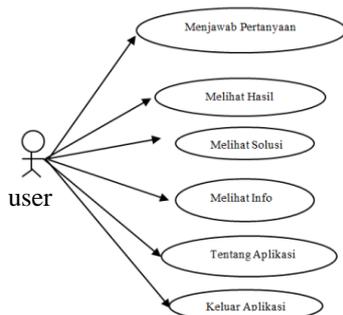
Pada bab ini, akan dibahas secara detail dan terperinci mengenai sistem yang akan penulis bahas dengan menerapkan metode pengembangan sistem *Extreme Programming (XP)* sebagai proses modelnya. Metode ini menekankan suatu pendekatan interaktif dari proses pengembangan perangkat lunak yang dimaksud untuk mengurangi resiko pada aplikasi dan memaksimalkan hasil akhir dari sebuah aplikasi. Optimalisasi juga dilakukan pada bagian teknologi perangkat lunak pendukung dengan cara menggunakan *API (Application Programming Interface)* dan *Framework open source* sehingga penulis bisa lebih focus pada ide dan konsep bisnis proses. Focus pada ide dan konsep bisnis proses memberikan ruang lebih banyak pada penulis untuk mencurahkan gagasan-gagasan baru dan kemudian dapat menghasilkan *API, module* dan teknik-teknik baru sehingga bisa terbebas dari paradigma *reinvent the wheel*. *API* dan *Framework* diperoleh dari www.codeplex.com yang merupakan *repository* hasil komunitas pengguna teknologi berbasis *Microsoft*. Adapun tahapan yang dilakukan dan dijelaskan antara lain

1. Tahap Planning

Tahapan *planning* pada *XP* merupakan tahapan awal dari semua proses pengembangan sistem. Tahapan ini merupakan gabungan dari tahap analisis dan *planning* pada metode pengembangan klasik. Tahap *planning* dapat diketahui gambaran besar dari aplikasi yang hendak dikembangkan dan teknologi-teknologi pendukung yang akan.

a. Use Case

Use case ini menggambarkan tentang interaksi antara actor (user) dengan system. Disini digambarkan actor (user) melakukan beberapa kegiatan seperti melihat gejala, melihat hasil, melihat solusi, melihat tentang aplikasi dan keluar



Gambar 4. Use Case

b. User Stories

Dari penjelasan analisis *use case* di atas, penulis dapat menyimpulkan melalui *user stories* sebagai berikut

1. *User* dapat menjawab pertanyaan dari gejala kerusakan *handphone*
2. *User* dapat melihat hasil dari gejala kerusakan yang dipilih
3. *User* dapat melihat Solusi dari hasil kerusakan yang terjadi
4. *User* dapat melihat Info dari hasil kerusakan yang terjadi
5. *User* dapat melihat ‘Tentang Aplikasi’
6. *User* dapat Keluar dari aplikasi

2. Tahap Desain

Dalam mendesain aplikasi ini penulis menggunakan pendekatan yaitu membuat *class design* dan *GUI design*. Maksudnya adalah untuk mempermudah dan mempercepat penyelesaian proses desain. Penulis juga berusaha membuatnya sangat detail supaya pada saat proses pengkodean lebih mudah karena gambaran fungsi-fungsinya sudah jelas.

1. Design CRC Card

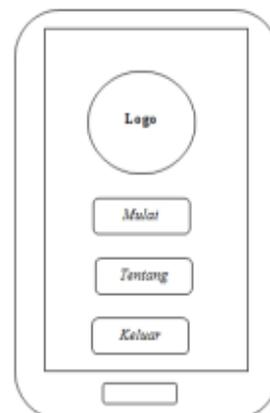
Proses desain pada tahap ini yaitu menggunakan *CRC Responsibilities, and Collaboration (CRC) card*. Penggunaan *CRC card* hanya dipakai jika pengembangan *software* berbasis objek. Setiap *CRC card* akan menjelaskan tiap objek yang dibutuhkan.

Table 3. CRC Card

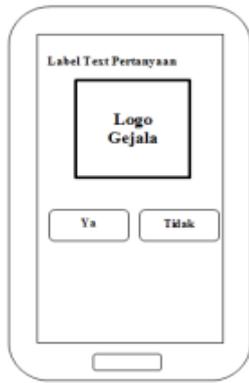
Class : Pertanyaan	
Resposibilities	Collaborators
Menampilkan Pertanyaan	Mesin Inferensi
Menjawab Pertanyaan	

2. Desain GUI

Pada tahap ini, keseluruhan perancangan secara grafis aplikasi dilakukan. Dalam merancangnya digunakan prinsip-prinsip dari bidang ilmu Interaksi Manusia dan Komputer. Contohnya dalam hal pemilihan warna, besar huruf, kebiasaan pengguna dan lain sebagainya. Tahap ini dimaksudkan agar waktu pengerjaan lebih singkat dan tidak ada dokumentasi yang sia-sia.



Gambar 5. Rancangan Tampilan Awal



Gambar 6. Rancangan Pertanyaan



Gambar 7. Rancangan Kesimpulan

3. Coding

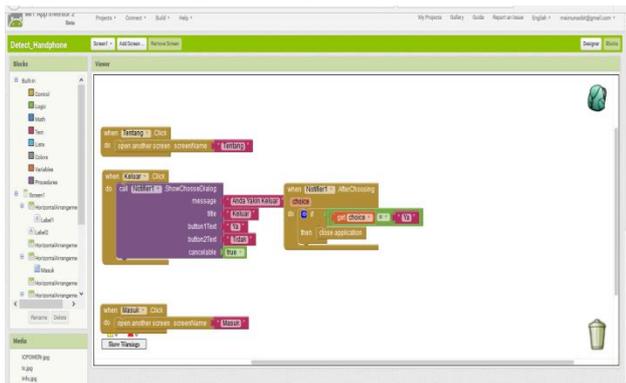
Implementasi Aplikasi Sistem Pakar Pendeteksi Kerusakan Handphone menggunakan web *open-source App Inventor* yang dijalankan pada platform Android.

a. Perancangan Antar Muka di Lingkungan APPInventor

Pada bagian ini, pengembang melakukan perancangan antar muka program melalui komponen App Inventor bernama *Designer*, dimana di bagian kiri layar terdapat serangkaian palet yang didalamnya terdapat bagian-bagian yang diantaranya adalah label, tombol, textbox, pilihan pengaturan layout, media penyimpanan dan sebagainya. Masing-masing komponen yang diperlukan di *drag and drop* ke area *viewer*

b. Penyusunan Kode Blok

Kode blok digunakan untuk menentukan bagaimana *behavior* atau fungsi dari masing-masing komponen yang telah dirancang pada komponen *designer*. Penyusunan kode blok juga dengan metode *drag and drop* dari palet yang berada di bagian kiri layar



Gambar 8. Penyusunan Kode Blok

4. Testing

Pada tahap ini dilakukan pengujian masing-masing modul (*unit*) program apakah sesuai dengan tugasnya. Pengetesan ini dilakukan dengan menggunakan metode *black box* testing yaitu pengetesan menggunakan metode pengujian logika program yaitu perkasus atau masalah yang diajukan. Pada tahap akhir akan diujikan pada kondisi sebenarnya yaitu dengan *implementation testing*.

1. Black Box Testing

Karena penulis menggunakan *extreme programming* sebagai model prosesnya maka sebenarnya pengujian hanya dilakukan dengan metode *unit testing*. Table dibawah merupakan hasil pengujian dengan *unit testing*.

Tabel 4. Hasil Pengujian dengan Unit Testing

No.	Nama Pengujian	Hasil Yang Diharapkan	Hasil
1.	Fungsi menu pada halaman awal	Tiap menu dapat dijalankan tanpa pesan error	Valid
2.	Input Data Pengguna	Sistem mampu menerima input data pengguna berupa nama dan ciri/karakteristik untuk proses analisa	Valid
3.	Proses Analisa	Sistem mampu menampilkan hasil analisa berdasarkan input yang dimasukkan pengguna	Valid
4.	Data "Tentang" pengguna	Sistem menampilkan jenis-jenis kerusakan yang dibahas dalam kerusakan handphone	Valid

2. Implementation Testing

Berikut ini ilustrasi implementasi aplikasi yang berjalan oleh seorang pengguna yakni aplikasi dimulai dengan tampilan awal seperti dibawah ini :

3.



Gambar 9. Tampilan Awal Aplikasi

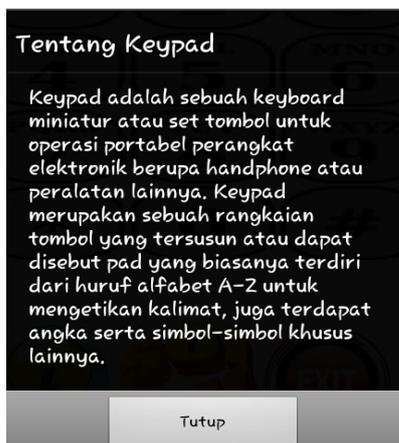


Gambar 10. Tampilan Pertanyaan

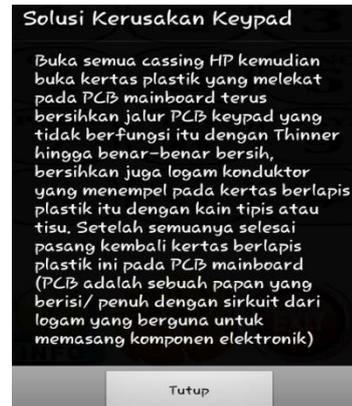
Mesin inferensi kemudian mengolah hasil input dari pengguna, kemudian menarik kesimpulan hasil analisa yaitu jenis kerusakan handphone.



Gambar 11. Tampilan Kesimpulan



Gambar 12. Tampilan Info/tentang jenis kerusakan



Gambar 13. Tampilan Solusi Kerusakan

BAB V KESIMPULAN DAN SARAN

1. Kesimpulan

Berdasarkan hasil pengujian dan implementasi sistem pakar pendeteksi kerusakan handphone ini dapat disimpulkan sebagai berikut :

1. Aplikasi sistem pakar Pendeteksi Kerusakan Handphone berbasis Android berjalan dengan baik sesuai rancangan.
2. Metode inferensi *Forward Chaining* yang digunakan pada penelitian ini menentukan gejala-gejala kerusakan terlebih dahulu, setelah itu dicocokkan dengan fakta-fakta dan aturan yang ada dalam basis pengetahuan kemudian menarik kesimpulan berupa jenis kerusakan yang terjadi.

Dengan adanya aplikasi sistem pakar ini maka dapat membantu efisiensi waktu dalam memperoleh hasil dari kerusakan-kerusakan yang terjadi pada handphone serta meminimalisir biaya untuk konsultasi ke teknisi. Namun demikian tidak sepenuhnya menggantikan peran pakar (Teknisi) untuk hasil analisa yang lebih

2. Saran

Berdasarkan kesimpulan-kesimpulan yang telah dikemukakan, dapat diajukan beberapa saran untuk pengembangan lebih lanjut antara lain :

1. Untuk penelitian selanjutnya agar melakukan perbaikan pengetahuan, untuk meningkatkan keakuratan sistem pakar
2. Jika memungkinkan bagi yang selanjutnya akan mengembangkan sistem pakar menggunakan App Inventor agar menggunakan fasilitas TinyWebDB supaya data dapat disimpan secara terus menerus di database web.
3. Ke depan aplikasi sistem pakar pendeteksi kerusakan handphone dapat didesain dengan menggunakan metode backward chaining untuk data kerusakan lebih kompleks.

DAFTAR PUSTAKA

- [1] Kusumadewi. 2003. Artificial Intelligent (Teknik dan aplikasinya). Edisi Pertama. Yogyakarta: Graha Ilmu.
- [2] Pressman, Roger S. 2010. Software Engineering : A Practitioner's Approach, Seventh Edition. Mc Graw Hill Higher Education

- [3] Safaat, Nazzarudin. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- [4] Turban, E. 2005. Decision Support System and Intlegence Systems. Yogyakarta: Penerbit Andi Yogyakarta.



Sekilas dari penulis dengan nama lengkap Mercydian Pangkey, lahir pada tanggal 26 Juni 1991 di Manado. Dengan pendidikan pertama di Taman Kanak-Kanak Nafiri, kemudian melanjutkan ke SD Negeri GMIM 23 Manado, melanjutkan ke SMP Negeri 8 Manado, dan pada tahun 2007 melanjutkan ke SMK Negeri 2 Manado.

Setelah lulus sekolah tingkat kejuruan pada tahun 2010 penulis melanjutkan ke Perguruan Tinggi tepatnya di Universitas Sam Ratulangi Manado, Fakultas Teknik, Jurusan Elektro, Program Studi Teknik Informatika. Penulis membuat skripsi untuk memenuhi syarat sarjana (S1) dengan judul Sistem Pakar Pendeteksi Kerusakan Android Berbasis Android yang di bimbing oleh dua dosen Teknik Informatika yaitu Dr.Eng. Vecky C. Poekoel, ST,MT dan Oktavian A Lantang, ST, MTI sehingga pada tanggal 29 Juni 2016 penulis resmi lulus di Teknik Informatika Universitas Sam Ratulangi Manado.