

Sistem Pakar Mendeteksi Kerusakan Pada iPhone Dengan Menggunakan Metode *Certainty Factor*

Feby Trianisa ^{#1}, Ade Supriatna ^{#2}
Program Studi Teknik Informatika
Fakultas Teknik Universitas Suryakencana Cianjur
¹trianisafeby@gmail.com, ²adespr@yahoo.com

Abstrak

Sistem pakar merupakan bidang yang dicirikan oleh sistem berbasis pengetahuan (Knowledge Base System), memungkinkan komputer dapat berpikir dan mengambil kesimpulan dari sekumpulan kaidah. Sistem pakar ini mampu mendeteksi kerusakan pada iPhone yang dapat memberikan solusi terbaik terhadap kerusakan iPhone dan dapat membantu memberikan informasi kepada para teknisi dan sebagai sarana untuk menyimpan pengetahuan tentang kerusakan iPhone yang berkenaan dengan jenis penyakit mata dari pakar atau ahlinya.

Pada saat ini proses pendeteksian kerusakan pada iPhone di Halife Store iDevice Solution masih dilakukan secara manual. Terdapat beberapa kekurangan pada sistem manual ini yaitu pengguna iPhone datang ke Halife Store iDevice Solution lalu berkonsultasi dengan teknisi atau pegawainya. Setelah itu teknisi akan melakukan pemeriksaan, dan menyimpan hasil pemeriksaan.

*Dalam perancangan aplikasi ini, representasi pengetahuan menggunakan metode *Certainty Factor*, pohon keputusan, dan kaidah produksi. Untuk proses penalaran menggunakan metode *Certainty Factor* yaitu sebagai salah satu teknik sistem pakar ini mengambil nilai bobot dari setiap gejala kerusakan iPhone. Aplikasi sistem pakar mendeteksi kerusakan pada iPhone menggunakan metode *Certainty Factor* ini dibangun dengan bahasa pemrograman PHP menggunakan framework CodeIgniter dan Bootstrap sebagai framework CSS untuk mengatur tampilan yang lebih menarik. Local server menggunakan tools XAMPP.*

Kata kunci: Sistem Pakar, Certainty Factor, iPhone, PHP.

1. Pendahuluan

Tidak ada angka pasti jumlah pengguna iPhone di dunia, menurut data pasar Amerika Serikat yang dirilis BI Intelligence mencatat bahwa pada kuartal pertama tahun 2015 pengguna iPhone meningkat sebanyak 1,3% (sumber: tekno.liputan6.com). Angka tersebut diyakini adalah pengguna iPhone yang beralih menggunakan iPhone sejak diluncurkannya iPhone 6 dan 6 plus. Semakin banyaknya pengguna iPhone tidak dibarengi dengan *support center* yang Apple miliki. Di kota Cianjur sendiri tidak adanya *service counter handphone* yang khusus untuk produk Apple dalam hal ini iPhone membuat para pengguna iPhone harus membawanya ke *service counter* biasa. Padahal tidak semua *service counter handphone* pada umumnya dapat mendiagnosa bahkan memperbaiki kerusakan yang terjadi pada iPhone. Sebenarnya iPhone merupakan suatu perangkat komunikasi yang paling rentan akan kerusakan, hal ini disebabkan karena kurangnya pengetahuan pengguna iPhone mengenai gejala-gejala kerusakan yang timbul, seperti kerusakan pada layar, IC, antenna, dan lain-lain. Sehingga kebanyakan orang tidak sadar dan acuh akan gejala kerusakan tersebut, dan memakai iPhone hingga benar-benar mati total.

Berdasarkan permasalahan yang diuraikan di atas maka sistem pakar perlu dibuat menjadi suatu acuan untuk menyelesaikan permasalahan-permasalahan tersebut dapat dijadikan solusi sebagai alat bantu para teknisi, dan para pengguna iPhone dalam mendeteksi kerusakan

iPhone. Dengan adanya sistem pakar secara terkomputerisasi maka kita akan dapat mendeteksi kerusakan pada iPhone.

2. Kajian Pustaka

2.1. Sistem Pakar

Sistem merupakan bagian-bagian yang saling berkaitan yang saling beroperasi bersama untuk mencapai sasaran dan tujuan yang sama (Gordon B. Davis).

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah (Turban, 1995).

Sistem yang menggunakan pengetahuan manusia yang dimasukkan kedalam komputer untuk memecahkan masalah-masalah yang biasanya diselesaikan oleh pakar (Turban dan Aronson (2001)).

2.2. Faktor Kepastian (*Certainty Factor*)

Faktor kepastian (*Certainty Factor*) menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesa) berdasarkan bukti atau penilaian pakar (Turban, 2005).

Pemberian nilai *Certainty Factor* kaidah bisa sangat bervariasi dalam arti nilai yang diberikan oleh pakar bisa berbeda-beda untuk suatu kaidah yang sama, karena bersifat subyektif. Nilai *Certainty Factor* ada 2, yaitu :

1. Nilai *Certainty Factor* yang nilainya melekat pada suatu kaidah atau *rule* tertentu dan besarnya nilai yang diberikan oleh pakar.

2. Nilai *Certainty Factor* yang diberikan oleh pengguna untuk mewakili derajat kepastian atau keyakinan atas premis (misalnya : gejala, kondisi, ciri) yang dialami pengguna.

Pada tahun 1977 definisi asli dari *Certainty Factor* diubah dalam MYCIN menjadi :

$$CF = \frac{MB - MD}{1 - \min (MB - MD)}$$

Jika ada kaidah lain termasuk kedalam hipotesis yang sama tetapi berbeda dalam faktor kepastian maka perhitungan kepastian dari kaidah yang sama dihitung dari penggabungan fungsi untuk faktor kepastian yang didefinisikan sebagai berikut:

$$CF_{combine}(CF_1, CF_2) = \begin{cases} \frac{CF_1 + CF_2(1 - CF_1)}{CF_1 + CF_2} & \text{2 nilai} \geq 0 \\ \frac{1 - \min(|CF_1|, |CF_2|)}{CF_1 + CF_2(1 + CF_1)} & \text{Salah satu} < 0 \end{cases}$$

Dimana rumusan $CF_{combine}$ digunakan bergantung pada apakah faktor kepastian negatif atau positif.

2.3. iPhone

iPhone adalah nama ponsel pintar (*smartphone*) buatan Apple. iPhone memiliki fitur kamera dan kemampuan berkomunikasi sama seperti *handphone*, dan segala fitur yang dimiliki oleh *handphone* dan ipod juga ada didalam iPhone.

2.4. Voice Recognition

Voice Recognition dalam Bahasa Inggrisnya, *automatic speech recognition* (ASR) adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah komando untuk melakukan suatu pekerjaan, misalnya penekanan tombol pada telepon genggam yang dilakukan secara otomatis dengan komando suara.

Alat *voice recognition*, yang sering disebut dengan *speech recognizer*, membutuhkan sampel kata sebenarnya yang diucapkan dari pengguna. Sampel kata akan digitalisasi, disimpan dalam komputer, dan kemudian digunakan sebagai basis data dalam mencocokkan kata yang diucapkan selanjutnya. Sebagian besar alat *voice recognition* sifatnya masih tergantung kepada pembicara. Alat ini hanya dapat mengenal kata yang diucapkan dari

satu atau dua orang saja dan hanya bias mengenal kata-kata terpisah, yaitu kata-kata yang dalam penyampaian terdapat jeda antar kata. Hanya sebagian kecil dari peralatan yang menggunakan teknologi ini yang sifatnya tidak tergantung pada pembicara. Alat ini sudah dapat mengenal kata yang diucapkan oleh banyak orang dan juga dapat mengenal kata-kata kontinu, atau kata-kata yang dalam penyampaian tidak terdapat jeda antar kata.

Pengenalan ucapan dalam perkembangan teknologinya merupakan bagian dari pengenalan suara (proses identifikasi seorang berdasarkan suaranya). Pengenalan suara sendiri terbagi menjadi dua, yaitu pengenalan pembicara (identifikasi suara berdasarkan orang berbicara) dan pengenalan ucapan (identifikasi suara berdasarkan kata yang diucapkan)

2.4.1. Speech-to-text

Salah satu tipe dari *speech recognition* adalah *speech-to-text* atau *voice-to-text*, yaitu suatu program yang dapat mengkonversi suara menjadi suatu tulisan. *Speech-to-text* awalnya dikembangkan sebagai teknologi bantu untuk tuna rungu. Aplikasi ini yang terdahulu harus dilatih untuk mengenali suara orang tertentu sebelum mencapai akurasi pada tingkat yang dapat diterima pengenalan suaranya. Aplikasi yang baru-baru sekarang dapat mengenali perkataan seseorang tanpa harus melakukan latihan terlebih dahulu, teknologi ini membuka kemungkinan untuk aplikasi terbaru, termasuk pada fungsi telepon, seperti *speech-to-text* pengiriman pesan dan pencarian berbasis suara.

2.4.2. Text-to-speech

Sistem TTS (Text to Speech) adalah suatu sistem yang dapat melakukan konversi dari teks menjadi ucapan. Saat ini terdapat berbagai sistem yang dapat melakukan fungsi seperti itu, misalnya sistem IVR (*interactive voice response*) yang banyak digunakan untuk layanan inhalamanasi otomatis melalui telepon (*call center*). Sistem IVR biasanya menggunakan rekaman kata atau kalimat yang direkam secara utuh. Untuk mempertegas definisi TTS dalam disertasi ini, berikut ini akan ditinjau beberapa definisi TTS dari beberapa literatur. Konversi text to speech memiliki dua sub sistem, yaitu:

1. Bagian Konverter Teks ke Fonem (*Text to Phoneme*)
2. Bagian Konverter Fonem to Ucapan (*Phoneme to Speech*)

Bagian Konverter Teks ke Fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. Bagian ini bersifat sangat language dependant. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut.

Kemudian, bagian Konverter Fonem ke Ucapan akan menerima masukan berupa kode-kode fonem serta *pitch* dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan.

Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah *formant synthesizer*, serta *diPhone concatenation*.

2.5. Prototyping

Seringkali mengambil halamanat atau mencakup pendekatan pembuatan prototipe. Pembuatan prototipe adalah pengembangan yang cepat dan pengujian terhadap model kerja atau prototipe dari aplikasi baru dalam proses interaktif dan berulang-ulang yang bisa digunakan oleh praktisi bisnis. Pembuatan prototipe membuat proses pengembangan lebih cepat dan lebih mudah, khususnya untuk proses dimana persyaratan pemakai akhir sulit dirumuskan. Pembuatan prototipe menggabungkan berbagai langkah dalam siklus pengembangan sistem dan mengubah berbagai peran pakar serta pemakai akhir yang lama. Suatu protipe adalah bentuk dasar atau model awal dari suatu sistem atau bagian dari suatu sistem setelah dioperasikan, prototipe ditingkatkan terus sesuai dengan kebutuhan pemakai sistem yang juga meningkat.

Prototyping adalah proses pengembangan suatu prototipe secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus sampai didapatkan sistem yang utuh. Proses membangun sistem ini yaitu dengan membuat prototip atau model awal, mencobanya, meningkatkannya dan seterusnya sampai didapatkan sistem yang lengkap disebut dengan proses iteratif (*iterative process*) dari pengembangan sistem.

3. Representasi Pengetahuan

Representasi pengetahuan bertujuan membuat struktur yang akan digunakan dalam sistem untuk membantu pengkodean pengetahuan kedalam program. Pengetahuan di representasikan ke dalam halamanat tertentu dan akan disimpan dalam suatu basis pengetahuan. Langkah-langkah atau *based rules* yang perlu dilakukan untuk membuat representasi pengetahuan sistem pakar adalah :

1. Membuat daftar kerusakan beserta gejala dan menghitung nilai kepastiannya yang muncul pada kerusakan iPhone.
2. Pembuatan tabel keputusan (*Decision Table*) yang berguna untuk mendokumentasikan dan mendeskripsikan pengetahuan.
3. Pembuatan pohon keputusan (*Decision Tree*) yang berguna untuk menghilangkan kaidah-kaidah dengan tujuan untuk mencegah terjadinya pertanyaan yang berulang.
4. Konversi pohon keputusan menjadi kaidah produksi.

Tabel 3. 1 Jenis Kerusakan

Kode Kerusakan	Nama Kerusakan
KI01	IC Power
KI02	Driver LCD
KI03	Baterai
KI04	Kamera
KI05	Bluetooth
KI06	Bootloop
KI07	Proxymith
KI08	IC VCO
KI09	CPU

KI10	IC RF
KI11	IC Flash
KI12	IC RAM
KI13	IC Audio
KI14	Filter RX/TX
KI15	IC UI (User interface)

Tabel 3. 2 Daftar Gejala Kerusakan

Kode Gejala	Nama Gejala
GI01	Mati Total
GI02	Tidak Ada Sinyal
GI03	Masukan Sim Card
GI04	Tidak dapat mengisi daya
GI05	Nada Dering Mati
GI06	Speaker Mati
GI07	Mic Mati
GI08	Tidak Ada Getar
GI09	Lampu Mati
GI10	Contac service
GI11	Imei
GI12	Blue Screen
GI13	Terdapat Noda Dilayar
GI14	Layar Buram
GI15	Sistem mati
GI16	Boros Baterai
GI17	Baterai Panas
GI18	Gambar Tidak Sempurna
GI19	Cahaya Tidak Terang
GI20	Gambar Pecah
GI21	Tidak Dapat Membuka Galeri
GI22	Tidak Bisa Mengirim File
GI23	Modul Bluetooth Panas
GI24	Bluetooth Tidak Dapat Diaktifkan
GI25	Hanya Menampilkan Logo Apple
GI26	Tidak Bisa Masuk Menu
GI27	Sinyal Lemah
GI28	Restart
GI29	Call ended
GI30	Blinking

Tabel 3. 3 Nilai CF Gejala Pada Kerusakan IC Audio (KI13)

No	Gejala	Kerusakan	CF
1	Mati total	IC Audio (COBBA)	0,2
2	Tidak ada sinyal	IC Audio (COBBA)	0,6
3	Contac service	IC Audio (COBBA)	0,8

Contoh penentuan nilai bobot *user* (*Evidence*) dilakukan ketika *user* menggunakan aplikasi:

- Aplikasi : Apakah iPhone anda mati total?
User : Sangat Yakin ($CF_{user} = 1$)
- Aplikasi : Apakah iPhone anda tidak ada sinyal?
User : Cukup Yakin ($CF_{user} = 0,6$)
- Aplikasi : Apakah iPhone anda menampilkan contac service?
User : menampilkan contac service?

Tidak Tahu ($CF_{user} = 0,2$)
 Maka nilai CF_{Pakar} dengan CF_{User} :
 $CF[H,E]_1 = CF[H]_1 * CF[E]_1 = 0,2 * 1 = 0,2$
 $CF[H,E]_2 = CF[H]_2 * CF[E]_2 = 0,6 * 0,6 = 0,36$
 $CF[H,E]_3 = CF[H]_3 * CF[E]_3 = 0,8 * 0,2 = 0,16$

$CF(B) = 0,49 + [0 * (1 - 0,49)] = 0,49$
 $CF(A) + [(CF(3) * (1 - CF(A)))] = 0,49$
 $CF[H,E]_0 * 100\% = 0,49 * 100\% = 49\%$

❖ Dengan kata lain dapat dikatakan bahwa iPhone tersebut mengalami kerusakan IC Audio dengan persentase tingkat keyakinan 49%

Maka nilai CF Kerusakan IC Audio :

$CF(A) = 0,2 + [0,36 * (1 - 0,49)] = 0,2$
 $CF(1) + [(CF(2) * (1 - CF(1)))] = 0,2$

Tabel 3.4 Tabel Keputusan

Kode Gejala	Kode Kerusakan														
	KI01	KI02	KI03	KI04	KI05	KI06	KI07	KI08	KI09	KI10	KI11	KI12	KI13	KI14	KI15
GI01	X		X			X		X	X	X	X	X	X		X
GI02	X				X	X	X	X	X	X			X	X	
GI03	X						X								
GI04	X		X												
GI05	X														X
GI06	X														
GI07	X														
GI08	X												X		
GI09	X														X
GI10	X										X	X	X		
GI11	X										X	X			
GI12		X				X		X			X				X
GI13		X					X								
GI14		X					X								
GI15		X	X							X				X	
GI16			X					X							
GI17			X							X					
GI18				X											
GI19				X											
GI20				X											
GI21					X										
GI22					X										
GI23					X										
GI24					X										
GI25						X									
GI26						X									
GI27							X							X	
GI28								X			X	X			
GI29								X						X	
GI30									X						



3.1. Kaidah Produksi

AND Tidak ada sinyal
AND Modul *bluetooth* panas
AND Bluetooth tidak dapat diaktifkan
THEN Bluetooth

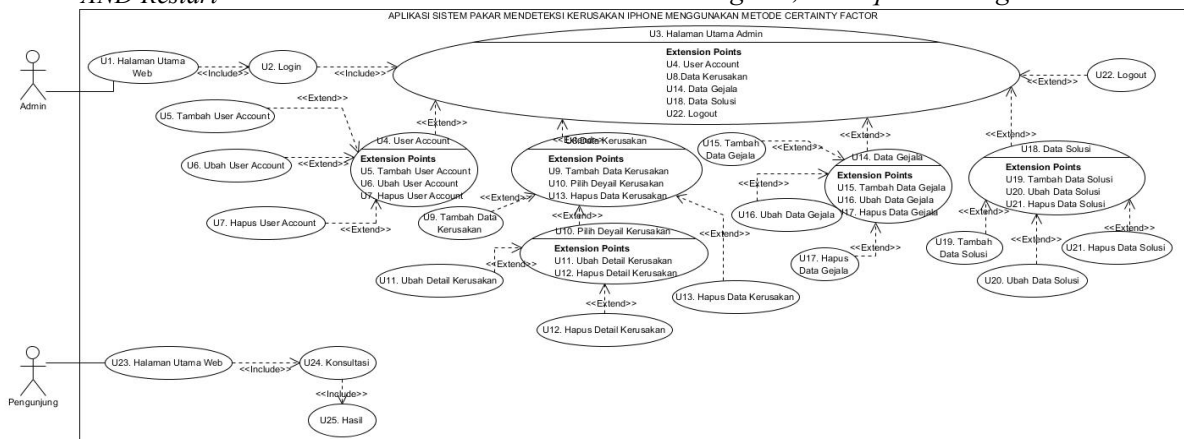
6. Kaidah 6 atau rule 6 (*Bootloop*)
IF Mati Total
AND Bluescreen
AND Hanya menampilkan logo Apple
AND Tidak bisa masuk menu
AND Tidak ada sinyal
THEN *Bootloop*
7. Kaidah 7 atau rule 7 (*Proxymith*)
IF Sinyal lemah
AND Terdapat noda dilayar
AND Tidak ada sinyal
AND Masukan sim card
AND Sistem mati
THEN *Proxymith*
8. Kaidah 8 atau rule 8 (IC VCO)
IF Mati Total
AND *Restart*
AND *Call ended*
AND Tidak ada sinyal
THEN IC VCO
9. Kaidah 9 atau rule 9 (CPU)
IF Tidak ada sinyal
AND Bluescreen
AND Mati total
AND Boros Baterai (Ngedrop)
THEN CPU
10. Kaidah 10 atau rule 10 (IC RF)
IF Mati total
AND Tidak ada sinyal
AND Sistem mati
AND Baterai panas
THEN IC RF
11. Kaidah 11 atau rule 11 (IC Flash)
IF Mati total
AND Blinking
AND Bluescreen
AND Contac service
AND *Restart*

- AND Imei
THEN IC Flash
12. Kaidah 12 atau rule 12 (IC RAM)
IF Mati total
AND Imei
AND Contac service
AND Restart
THEN IC RAM
13. Kaidah 13 atau rule 13 (IC Audio)
IF Mati total
AND Tidak ada sinyal
AND Contac service
THEN IC Audio
14. Kaidah 14 atau rule 14 (Filter RX/TX)
IF Mati total
AND Imei
AND Contac service
AND Restart

- THEN IC RAM
15. Kaidah 15 atau rule 15 (IC UI)
IF Tidak ada getar
AND Nada dering mati
AND Lampu tidak menyala
AND Bluescreen
AND Mati total
THEN IC UI

3.2. Pemodelan Sistem

Permodelan sistem yang dilakukan menggunakan UML (Unified Modeling Language) sebagai penggambaran, perencanaan, dan pembuatan sistem yang akan dibangun. Adapun tahapan analisis sistem menggunakan UML adalah use case diagram, use case scenario, swimlane diagram, class diagram, state diagram, dan sequence diagram.

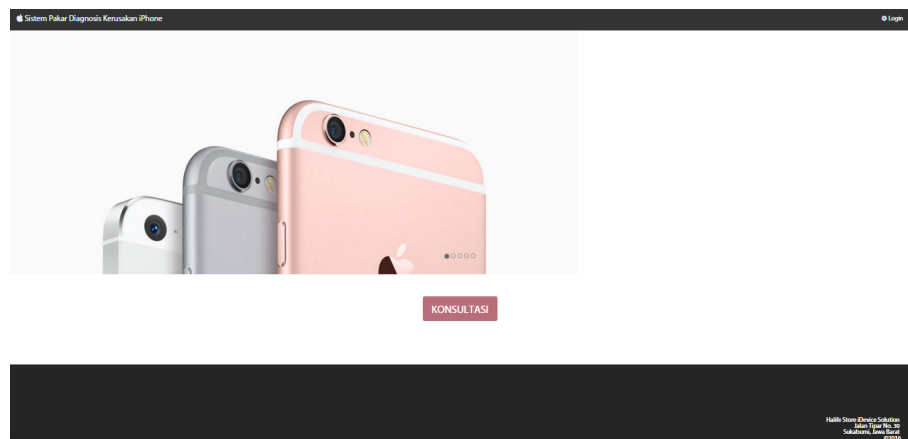


Gambar 3. 2 Use Case Diagram Sistem Pakar Mendeteksi Kerusakan Pada iPhone Dengan Menggunakan Metode Certainty Factor

4. Tampilan Antarmuka

1. Antarmuka Halaman Utama Web

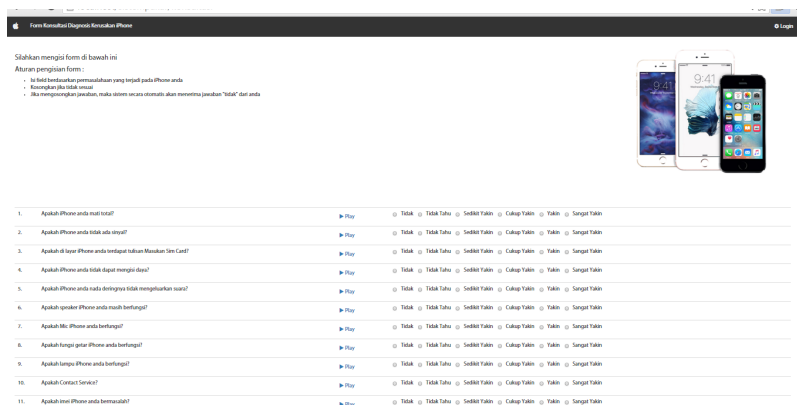
Antarmuka halaman utama web adalah tampilan pertama kali yang digunakan oleh admin ataupun pengunjung sebelum memasuki halaman berikutnya. Berikut adalah antarmuka halaman utama web :



Gambar 4. 1 Antarmuka Halaman Utama Web

2. Antarmuka Halaman Konsultasi

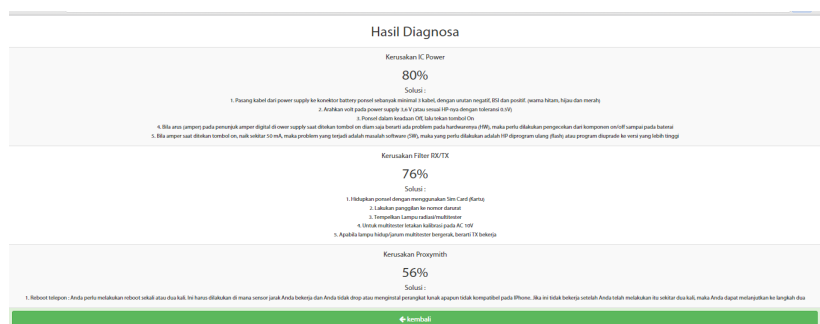
Antarmuka halaman konsultasi merupakan tampilan yang digunakan oleh pengunjung ataupun *admin* untuk melakukan konsultasi mengenai kerusakan iPhone. Berikut adalah antarmuka halaman konsultasi:



Gambar 4. 2 Antarmuka Halaman Konsultasi

3. Antarmuka Halaman Hasil

Antarmuka halaman hasil merupakan tampilan yang digunakan oleh pengunjung ataupun *admin* untuk melihat hasil dari konsultasi mengenai kerusakan iPhone. Berikut adalah antarmuka halaman hasil :



Gambar 4. 3 Antarmuka Halaman Hasil

4.1. Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan menggunakan metode pengujian *black box*.

5. Kesimpulan

Dari hasil penelitian, perancangan dan implementasi yang telah dilakukan, penulis dapat menyimpulkan beberapa simpulan sebagai berikut :

1. Telah berhasil dibangun sebuah aplikasi sistem pakar mendeteksi kerusakan iPhone menggunakan metode *Certainty Factor* sebagai alat bantu pendeteksian atau proses mendiagnosis awal di Halife Store iDevis Solution sesuai dengan kebutuhan pengguna yang dirancang secara efektif dan efisien.
2. Aplikasi yang telah dibuat ini dapat memudahkan para pengunjung untuk melakukan konsultasi mengenai kerusakan iPhone tanpa harus berhubungan langsung dengan teknisi atau pakar.
3. Aplikasi ini menjadi alat bantu atau alternatif kemudahan dalam mendeteksi kerusakan pada iPhone secara cepat dan tepat berguna untuk

pemecahan masalah dan memberikan solusi yang terbaik.

6. Daftar Pustaka

- [1] Arhamni, Muhammad. (2005). Konsep Dasar Sistem Pakar. Yogyakarta : Andi
- [2] Dawn, Mark. (2014). iPhone 6 Beginners Tutorial Guide. New York. Conceptual Kings.
- [3] Hartati, Sri.(2008). Sistem Pakar dan Pengembangannya. Yogyakarta : Graha Ilmu
- [4] <http://www.apple.com/pr/library/2015/09/28Apple-Announces-Record-iPhone-6s-iPhone-6s-Plus-Sales.html> waktu akses 27 Oktober 2015
- [5] <http://tekno.liputan6.com/read/2232821/pengguna-iPhone-berbondong-bondong-hijrah-ke-iPhone> waktu akses 27 Oktober 2015
- [6] Jurick, David. (2010). iPhone Hacks: Pushing the iPhone and iPod Touch Beyond Their Limits. New York. O'Reilly.

- [9] Kendall E, Kendall J. 2007. *Analisis dan Perancangan Sistem*. Klaten: PT Indeks
- [10] Prasetyo, Eko.(2008). *Pemrograman Web PHP & Mysql*. Yogyakarta: Graha Ilmu.
- [11] Pressman, R. S.(2010). *Software Engineering: a practioner's approach 7ed*. Newyork: McGraw Hill
- [12] Sidik, Betha. 2012. "*Pemrograman Web PHP edisi revisi*" Bandung : Informatika