

IMPLEMENTASI BASELINE BACKWARD CHAINING PADA SISTEM PAKAR DIAGNOSIS KERUSAKAN HANDPHONE

Sebastianus Adi Santoso Mola
Jurusan Ilmu Komputer, FST, UNDANA

ABSTRACT

The aim of this research is to implement baseline backward chaining to detect the hand phone failures. The types of hand phone are limited at Nokia hand phone such as 3310, 3315, 3330, 3330, and 8310. The knowledge is presented in the form if-then rule which generated from and-or decision tree and decision table. The result from this research is an expert system containing interactive dialogue between user and system. User guided by the system is to conduct troubleshooting hand phone checking steps and its solution. The level of user's confidence in replying question from the system is expressed by the measurement result.

Keywords: *expert system, hand phone failures, baseline backward chaining.*

Kerusakan pada handphone dapat terjadi karena dua penyebab utama yakni kerusakan *software* dan kerusakan *hardware*. Kerusakan *software* adalah kerusakan yang ditimbulkan akibat tidak dapat berjalannya sebagian atau seluruh program dalam handphone. Kerusakan *hardware* disebabkan karena kerusakan fisik atau tidak bekerjanya sebuah komponen elektroniknya. Kerusakan *hardware* dapat menyebabkan kerusakan *software*.

Diagnosa kerusakan handphone yang sering dilakukan oleh para teknisi berdasarkan dokumen tertulis dari perusahaan pembuat handphone. Langkah-langkah diagnosis kerusakan telah didokumentasikan secara baik dalam bentuk referensi manual yang terdiri dari ratusan halaman dan biasanya terdapat referensi cepat yang berisi ekstrak dari informasi pada referensi manual. Dengan mengacu pada referensi ini, solusi atas kerusakan handphone lebih akurat namun membutuhkan waktu yang lama untuk mencapai solusinya. Cara kedua yang sering dilakukan oleh teknisi adalah dengan mengandalkan pengetahuan dan pengalaman mereka. Langkah-langkah diagnosa seperti yang tertera dalam referensi manual kadang tidak diikuti sepenuhnya. Cara ini menghasilkan solusi dalam waktu cepat namun keakuratan solusinya tidak terjamin.

Penelitian mengenai diagnosis kerusakan sebelumnya yang pernah dilakukan adalah diagnosis kerusakan server HP (Hewlett Packard) berbasis aturan [Dev, 1997], diagnosis kerusakan perangkat keras komputer menggunakan pohon keputusan bercabang jamak [Manik, 2004], dan diagnosis kerusakan mesin mobil Panther berbasis mobile [Yudatama, 2008].

Pada penelitian ini, pendeteksian kerusakan handphone menggunakan metode baseline backward chaining. Pengetahuan mengenai kerusakan handphone disajikan ke dalam pohon keputusan dan tabel keputusan sebelum direpresentasikan ke dalam basis pengetahuan.

MATERI DAN METODE

Jenis Kerusakan

Referensi manual diagnosis kerusakan handphone biasanya disajikan dalam bentuk diagram alir (*flow chart*), namun kadang kala ditemukan juga dalam bentuk lain seperti narasi dan gambar. Bagaimanapun juga, referensi manual ini menjadi sumber pengetahuan dalam diagnosis kerusakan handphone. Jenis-jenis kerusakan pada handphone diantaranya adalah mati total, *flash not possible*, *low operational and standby time*, *not charging*, *contact service*, *SIM card fault*, *audio fault*, *user inteface (display, keypad, backlight, vibrator, buzzer) fault*, *clock fault*, dan *no service*.

Basis Pengetahuan

Agar pengetahuan yang terdapat dalam referensi manual dapat disimpan ke dalam komputer menjadi sebuah basis pengetahuan, referensi manual harus diubah menjadi aturan-aturan yang berisi fakta dan konklusi. Proses mengubah referensi manual menjadi aturan mencakup beberapa langkah yakni:

1. Susun kembali pengetahuan dalam referensi manual ke dalam bentuk diagram alir jika pengetahuan yang diperoleh berbentuk narasi atau gambar
2. Ubah diagram alir menjadi pohon keputusan dengan terlebih dahulu menentukan fakta dan konklusi dengan cara:
 - a. Menentukan fakta dari hasil pengecekan oleh teknisi dimana perintah pengecekannya terdapat dalam simbol *decision* dari diagram alir
 - b. Menentukan konklusi yang berupa pernyataan yang terdapat dalam simbol *process* dari diagram alir
 - c. Membentuk pohon keputusan AND-OR dengan menempatkan konklusi sebagai *root* dan fakta sebagai *leaf*.
3. Ubah pohon keputusan menjadi tabel keputusan. Tabel berikut adalah contoh tabel keputusan untuk kerusakan 'Not Charging' dari Nokia 33XX.

Tabel 1. Tabel Keputusan untuk kerusakan 'Not Charging' Nokia 33XX

GJL	Keruskan / Solusi Panduan / Gejala	SLS005				SLS020	SLS021	SLS022		
		Y	Y	Y	Y	N	N	Y	Y	Y
024	Lakukan proses charging dengan memasang charger pada HP!Ada pesan "Not charging" di layar?	Y	Y	Y	Y	N	N	Y	Y	Y
025	Lakukan proses charging dengan memasang charger pada HP!Ada pesan "Reconnect Charger" di layar					Y	N			N
027	Jalankan Energi Management Calibration!Ada pesan "Battery Temperature failed"	N	N	N	Y			N	N	
028	Jalankan Energi Management Calibration!Ada pesan "Battery Size Failed"	N	N	Y				N	N	
029	Jalankan Energi Management Calibration!Ada pesan "Battery Voltage failed"	N	Y					N	N	
030	Jalankan Energi Management Calibration!Ada pesan "Charge Current Failed"	N						Y	N	
031	Lakukan pengukuran pada R209/210!Tegangan pada R209/210 = 0.8 V	Y					N		N	Y

Keterangan Tabel:

GJL : Gejala/Fakta

SLS : Solusi/Konklusi

Y : Yes/Ya

N : No/Tidak

SLS005 : Ganti N201 (IC CCONT 2iwfd163kg64t/8 lfbga 8x8)

SLS020 : Ganti N200 (Chaps v2.0 u432v230g236t lbga6x6), N201 (IC CCONT 2iwfd163kg64t/8 lfbga 8x8)

SLS021 : Ganti X200 (Konektor charger), F200 (SM fuse f1.5A 32v), V201 (Chips varistor vwm14v vc30v)!

SLS022 : Ganti N200 (Chaps v2.0 u432v230g236t lbga6x6)

4. Ubah tabel keputusan menjadi aturan. Contoh aturan yang didapat dari tabel keputusan di atas dengan konklusi SLS005 adalah:

IF (GJL024 AND GJL027)

OR (GJL024 AND NOT GJL027 AND GJL028)

OR (GJL024 AND NOT GJL027 AND NOT GJL028 AND GJL029)

OR (GJL024 AND NOT GJL027 AND NOT GJL028 AND NOT GJL029 AND NOT GJL030 AND GJL031)

THEN SLS005

5. Menyederhanakan aturan yang memiliki klausa disjungtif (OR) pada bagian premis [Ignizio,1991]. Pada contoh tersebut, aturan dapat dipecah menjadi 4 yaitu:

aturan 1 : **IF** (GJL024 AND GJL027) **THEN** SLS005

aturan 2 : **IF** (GJL024 AND NOT GJL027 AND GJL028) **THEN** SLS005

- aturan 3 : **IF** (GJL024 AND NOT GJL027 AND NOT GJL028 AND GJL029) **THEN** SLS005
- aturan 4 : **IF** (GJL024 AND NOT GJL027 AND NOT GJL028 AND NOT GJL029 AND NOT GJL030 AND GJL031) **THEN** SLS005.

Baseline Backward Chaining

Dalam kecerdasan buatan, terdapat 2 metode penalaran yakni penalaran maju (*forward chaining*) dan penalaran mundur/balik (*backward chaining*). Kedua metode ini merupakan metode penalaran dengan aturan-aturan inferensi dan implikasi logis dan keduanya didasarkan pada aturan inferensi modus ponens. Secara sederhana, modus ponens mengatakan bahwa jika premis dari sebuah aturan benar maka konklusinya benar.

Algoritma penalaran *baseline backward* version sebagai berikut [Ignizio,1991]:

1. Inisialisasi

Buatlah tiga tabel kosong, tabel Memori Kerja, tabel Tujuan, dan tabel Status Aturan/Premis. Tabel Memori Kerja digunakan untuk merekam semua masukan dari pengguna. Tabel Tujuan secara berurutan menyimpan konklusi dari Aturan. Tabel Status Aturan/Premis menyimpan status aturan (ACTIVE, DISCHARDED, TRIGGERED, FIRED), dan status Premis (FREE, TRUE, FALSE). Semua klausa premis dibuat FREE dan semua aturan dibuat ACTIVE.

2. Mulai Inferensi

Tentukan sebuah tujuan akhir (konklusi). Tempatkan pada bagian awal dari tabel Tujuan.

3. Memindai Aturan dan Pengecekan Konvergensi

Pindai klausa-klausa konklusi dari aturan yang ACTIVE (aturan yang belum berstatus FIRED atau DISCHARDED) untuk menemukan kecocokan dari atribut tujuan yang terdapat pada bagian awal dari tabel Tujuan

a. Jika tabel Tujuan kosong, BERHENTI

b. Jika hanya satu aturan yang ditemukan, lompat ke langkah nomor 6.

c. Jika beberapa aturan ditemukan, dan sebagian diantaranya berstatus TRIGGERED, pilih satu dari aturan-aturan yang berstatus TRIGGERED dan lanjutkan ke langkah 6. Jika tidak, secara acak pilih satu diantara aturan-aturan yang ditemukan yang atribut tujuan pada klausa konklusinya dan lanjutkan ke langkah 6.

- d. Jika tidak ada aturan berstatus ACTIVE yang mengandung atribut tujuan pada klausa konklusinya ditemukan, lanjutkan ke langkah 4.
4. Query
Temukan query dari atribut tujuan pada awal tabel Tujuan. Jika tidak ada query ditemukan, BERHENTI. Jika tidak, tampilkan query ke pengguna dan catat masukan pengguna. Pindahkan atribut tujuan dari tabel Tujuan ke dalam tabel Memori Kerja dan catat nilai yang dimasukkan pengguna ke tabel Memori Kerja. Lanjutkan ke langkah 5.
5. Pembaharuan Aturan/Premis
Lakukan pembaharuan tabel Status Aturan/Premis berdasarkan isi dari tabel Memori Kerja. Jika premis dari aturan bernilai FALSE, status aturan dibuat DISCHARGED. Jika premis bernilai TRUE, status aturannya menjadi TRIGGERED. Ulangi langkah 3.
6. Evaluasi Aturan
Untuk semua aturan yang ditemukan pada langkah 3:
 - a. Jika aturan berstatus TRIGGERED, pindahkan atribut tujuan yang terdapat pada awal tabel Tujuan bersama nilainya ke dalam tabel Memori Kerja. Ubah status dari aturan ini dari TRIGGERED menjadi FIRED. Kelmbali ke langkah 5. Jika tidak (aturan tidak berstatus TRIGGERED), lanjutkan ke langkah 6b.
 - b. Jika aturan tidak berstatus TRIGGERED, pilih premis pretama dari aturan beserta nomor aturannya dan tempatkan pada awal tabel Tujuan. Kembali ke langkah 3.

Faktor Kepastian

Untuk menyatakan tingkat kepastian dari keseluruhan proses konsultasi digunakan Faktor Kepastian (*Certainty Factor* (CF)). Faktor kepastian menunjukkan derajat kepercayaan Faktor kepastian mengekspresikan kepercayaan dalam suatu kejadian (fakta atau hipotesis) didasarkan pada bukti. Salah satu formula yang dapat digunakan untuk menentukan besarnya CF adalah [Turban, 2000]:

$$CF(H,E) = MB(H,E) - MD(H,E) \dots\dots\dots(1)$$

dimana:

- CF : faktor kepastian dalam hipotesis H yang dipengaruhi fakta E
- MB : *Measure of Belief* merupakan ukuran kenaikan kepercayaan hipotesis H dipengaruhi fakta E

- MD : *Measure of Disbelief* merupakan ukuran kenaikan dari kepercayaan terhadap ingkaran hipotesis H dipengaruhi oleh fakta E
- E : fakta (*evidence*)
- H : hipotesis (*hypothesis*)

Atau seperti yang digunakan pada MYCIN:

$$CF = \frac{MB - MD}{1 - \min(MB, MD)} \dots\dots\dots(2)$$

Dalam mengkombinasikan fakta-fakta pada bagian premis dari sebuah aturan menggunakan metode seperti yang terdapat dalam tabel 2.

Tabel 2. Perhitungan CF untuk kombinasi fakta

<i>Evidence, E</i>	<i>Premise Certainty</i>
E ₁ AND E ₂	Min[CF(H,E ₁),CF(H,E ₂)]
E ₁ OR E ₂	Max[CF(H,E ₁),CF(H,E ₂)]
NOT E	-CF(H,E)

Rumus-rumus yang digunakan untuk perhitungan CF tersebut hanya berlaku jika setiap fakta dalam premis diketahui dengan pasti (CF(E,e) = 1). Namun jika faktor kepastian dari fakta E yang menyusun premis dari aturan berdasarkan ketidakpastian fakta e (CF(E,e) < 1) maka perhitungan CF-nya adalah [Giarrantano, 1993]:

$$CF(H,e) = CF(E,e) CF(H,E) \dots\dots\dots(3)$$

dimana:

- CF(E,e): faktor kepastian dari fakta E yang menyusun premis basis aturan pada ketidakpastian fakta e
- CF(H,E): faktor kepastian dari hipotesis dengan asumsi fakta diketahui dengan pasti, bila CF(E,e) = 1
- CF(H,e) : faktor kepastian hipotesis yang didasarkan pada ketidakpastian fakta e.

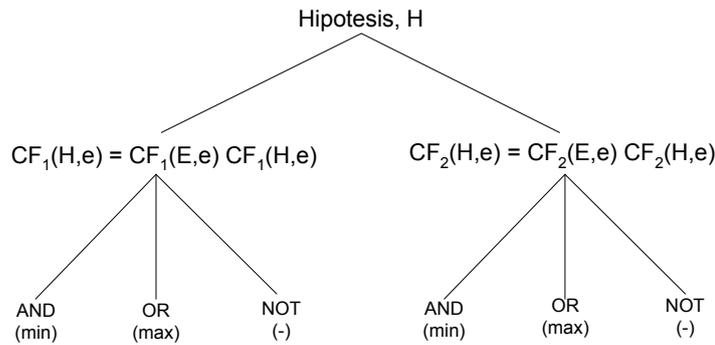
Jika dari fakta-fakta yang diberikan, terdapat beberapa aturan yang diaktifkan namun mempunyai hipotesis yang sama maka digunakan fungsi kombinasi untuk perhitungan faktor kepastiannya [Giarrantano, 1993]:

$$CF_{COMBINE}(CF_1, CF_2) = \begin{cases} \frac{CF_1 + CF_2 (1 - CF_1)}{CF_1 + CF_2} & CF_1, CF_2 > 0 \\ \frac{1 - \min(|CF_1|, |CF_2|)}{CF_1 + CF_2 (1 + CF_1)} & CF_1 \text{ atau } CF_2 < 0 \end{cases} \dots\dots(4)$$

dimana fungsi $CF_{COMBINE}$ tergantung pada masing-masing nilai faktor kepastian apakah positif atau negatif. Namun bila terdapat aturan ketiga dengan hipotesis yang sama maka perhitungan $CF_{COMBINE}$ -nya adalah [Turban, 2001]:

$$CF_{COMBINE}(CF_1, CF_2, CF_3) = f(CF_{COMBINE}(CF_1, CF_2), CF_3) \dots \dots \dots (5)$$

dengan fungsi f sama dengan fungsi kombinasi perhitungan kombinasi untuk CF_1 dan CF_2 . Perhitungan CF untuk dua aturan dengan hipotesis sama ditunjukkan pada gambar 2.



Gambar 1. CF dua aturan dengan hipotesis sama

Dalam penelitian ini digunakan 3 macam CF yakni:

1. CF_Aturan yang terdapat pada setiap aturan,
2. CF_Gejala, tingkat keyakinan terhadap sebuah gejala,
3. CF_User, tingkat keyakinan jawaban pengguna atas gejala yang ditanyakan.

Pengguna menyatakan tingkat kepercayaannya terhadap suatu gejala dengan memberikan nilai hasil pengukuran. Hasil pengukuran pengguna dapat saja berada di luar atau di dalam interval nilai yang diijinkan yang ditentukan oleh nilai_min dan nilai_max ([nilai_min, nilai_max]). Nilai_min dan nilai_max untuk setiap gejala dapat dilihat pada tabel gejala. Apabila hasil pengukuran pengguna berada dalam interval nilai yang diijinkan maka perhitungan CF gejala dapat dilakukan dengan menggunakan metode interpolasi linier yang mempunyai bentuk umum [Borowski, 1991]:

$$y(x) = f(x_0) + (x - x_0) \cdot \frac{f(x_1) - f(x_0)}{x_1 - x_0} \dots \dots \dots (6)$$

atau ditulis dalam bentuk persamaan menggunakan CF dan hasil pengukuran dalam interval nilai yang diijinkan menjadi:

$$CF_Ukur = CF(\min) + (ukur - \min) \cdot \frac{CF(\max) - CF(\min)}{\max - \min} \dots \dots \dots (7)$$

dan karena $CF(\min) = 0.6$ dan $CF(\max) = CF_Gejala$ maka persamaannya dapat ditulis sebagai:

$$CF_Ukur = 0.6 + (CF_Gejala - 0.6) \cdot \frac{(ukur - min)}{max - min} \dots\dots\dots (8)$$

dimana:

CF_Gejala : Tingkat kepercayaan suatu gejala mendukung sebuah hipotesa

CF_Ukur : CF_Gejala dipengaruhi hasil pengukuran

ukur : nilai hasil pengukuran

max : nilai tertinggi yang diijinkan / batas atas interval

min : nilai terendah yang diijinkan / batas bawah interval.

CF konklusi diperoleh dengan perhitngan [Giarrantano, 1993]:

$$CF_Konklusi = (CF_Gejala) \times (CF_Aturan) \dots\dots\dots (9)$$

dan karena pada sistem ini CF_Gejala telah dipengaruhi hasil pengukuran maka perhitungan CF_Konklusi-nya:

$$CF_Konklusi = (CF_Ukur) \times (CF_Aturan) \dots\dots\dots (10).$$

Namun apabila lebih dari aturan yang menghasilkan konklusi yang sama terpicu maka harus ada mekanisme perhitungan CF aturan yang mempertimbangkan semua aturan yang terpicu tersebut. Salah satu cara adalah dengan memakai pola seperti pada rumus (4) jika ada 2 aturan dengan konklusi sama terpicu. Jika terdapat 3 aturan maka dicari kombinasi nilai dari aturan pertama dan kedua terlebih dahulu kemudian dikombinasikan dengan aturan ketiga. Jika nilai CF_1 dan $CF_2 \geq 0$ maka [Turban, 2001]:

$$CF_{COMBINE} = CF_{1,2} + CF_3 - (CF_{1,2} \times CF_3) \dots\dots\dots (11)$$

dimana $CF_{1,2}$ adalah $CF_{COMBINE}$ untuk CF_1 dan CF_2 .

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 2 tipe handphone merk Nokia yakni tipe 33XX (3310, 3315, 3330, dan 3350) dan 8310. Untuk kedua tipe handphone ini, semua pengetahuan mengenai kerusakan handphone dimasukkan ke dalam basis pengetahuan sistem dengan 211 aturan dan 104 solusi.

Dalam sesi konsultasi, setelah pengguna memilih tipe handphone dan macam kerusakannya, maka pengguna harus menjawab pertanyaan yang diberikan oleh sistem. Jawaban yang diberikan dapat berupa jawaban "YA" atau "TIDAK". Jika pertanyaan yang diberikan oleh sistem menuntut adanya tindakan pengukuran tegangan, arus, atau frekuensi maka pengguna harus memasukkan hasil pengukurannya.

Sebagai contoh, dalam diagnosis kerusakan 'not charging' untuk tipe 33XX, proses konsultasi yang terjadi sebagai berikut:

1. Pertanyaan sistem: **Ukur tegangan keluaran dari charger! Apakah Tegangan keluaran charger = 5 V ?**

Jawaban Pengguna: YA

2. Pertanyaan Sistem: **Lakukan pengecekan kondisi kabel dari charger! Apakah Kondisi kabel charger koyak atau putus ?**

Jawaban Pengguna: TIDAK

3. Pertanyaan Sistem: **Lakukan pengecekan koneksi antara charger dan HP! Apakah Koneksi antara charger dan HP longgar?**

Jawaban Pengguna: TIDAK

4. Pertanyaan Sistem: **Lakukan proses charging dengan memasang charger pada HP! Apakah Ada pesan "Not Charging" di layar ?**

Jawaban Pengguna: YA

5. Pertanyaan Sistem: **Jalankan Energi Management Calibration! Apakah Ada pesan "Battery Temperature failed" ?**

Jawaban Pengguna: YA

6. Pertanyaan Sistem: **Jalankan Energi Management Calibration! Apakah Ada pesan "Charge Current Failed" ?**

Jawaban Pengguna: YA

7. Pertanyaan Sistem: **Jalankan Energi Management Calibration! Apakah Ada pesan "Battery Voltage failed" ?**

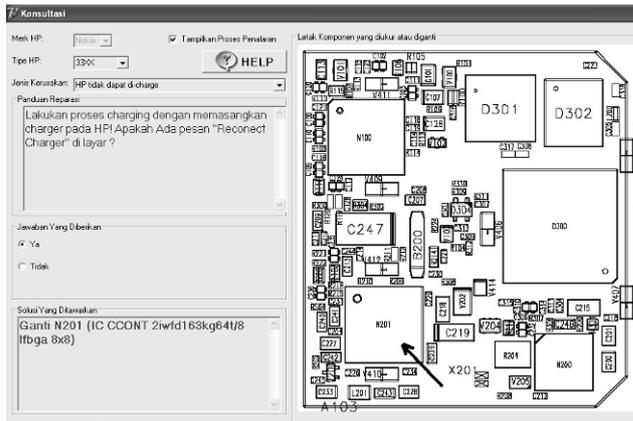
Jawaban Pengguna: YA

8. Pertanyaan Sistem: **Lakukan proses charging dengan memasang charger pada HP! Apakah Ada pesan "Reconnect Charger" di layar ?**

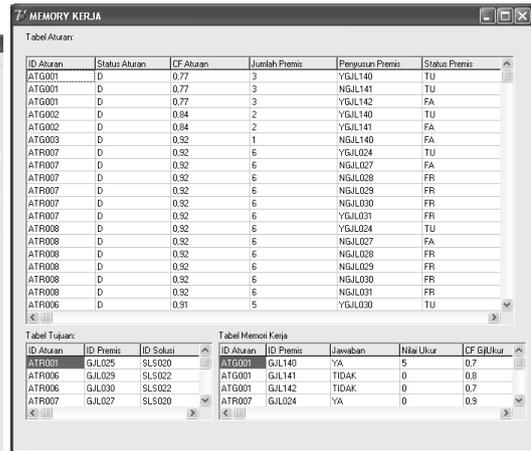
Jawaban Pengguna: YA

9. Solusi dari sistem: **Ganti N201 (IC CCONT 2iwfd163kg64t/8 lfbga 8x8)**

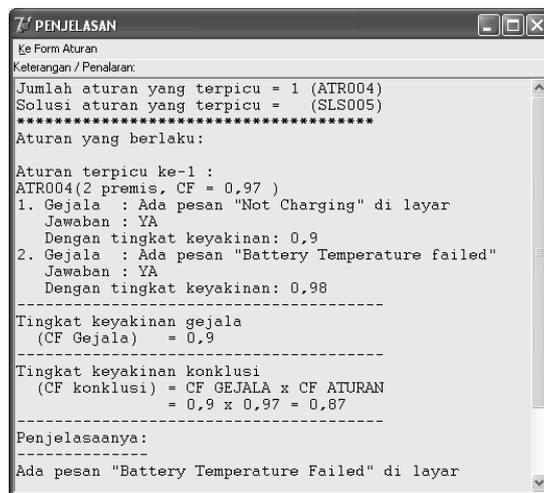
Gambar 3 memperlihatkan solusi yang diberikan oleh sistem, gambar 4 menampilkan proses penalaran baseline backward chaining sedangkan gambar 5 menampilkan fasilitas penjelasan dari sistem pakar.



Gambar 2. Contoh Solusi



Gambar 3. Contoh Penalaran



Gambar 4. Contoh Fasilitas Penjelasan

SIMPULAN

Sistem pakar deteksi kerusakan handphone ini telah dicobakan untuk handphone Nokia tipe 33XX dan tipe 8310 walaupun tidak semua jenis kerusakan pernah dicoba. Hal ini disebabkan karena sulitnya mendapatkan handphone dengan tipe 33XX dan 8310 dengan kerusakan yang bervariasi. Namun keberhasilan dalam reparasi handphone tidak hanya tergantung pada deteksi kerusakan oleh sistem pakar saja namun juga tergantung pada kerja dan ketrampilan dari teknisi reparasi.

Hasil penelitian ini menunjukkan bahwa baseline *backward chaining* dapat digunakan dalam pengembangan sistem pakar deteksi kerusakan handphone. Bila dibandingkan dengan referensi manual yang menjadi dasar pengetahuan deteksi kerusakan handphone, maka penalaran baseline *backward chaining* memberikan kesesuaian 100% untuk solusi yang ditawarkan.

Perlu dikembangkan sistem yang bersifat modular agar pengetahuan sistem dapat ditambahkan menggunakan modul basis pengetahuan berdasarkan tipe handphone. Hal ini memudahkan pengguna dalam memasukkan pengetahuan sistem untuk tipe handphone tertentu.

DAFTAR PUSTAKA

- Arman, M., Djanis Budi. 2004. *Cara Praktis Mamperbaiki Ponsel*. Gava Media. Yogyakarta.
- Borowski, E. J., and J. M, Barnein. 1991. *The Harper Collins Dictionary of Mathematics*. Herper Collins Publishers. New York.
- Dev, Narendra., and Bart Anderson. 1997. Pimtool, an Expert System to Troubleshoot Computer Hardware Failures. Dalam *Proceedings of the IAAI Conference*. 853-860. AAAI Press
- Donel, Hendri. 2005. *Kupas Tuntas Hardware handphone*. Penerbit Vyctoria. Semarang.
- Giarratano, Joseph. 1993. *Expert System, Principle and Programming*. Second Edition. PWS-Kent Publishing Company, Boston.
- Ignizio, James P. 1991. *Introduction to Expert System, The Development and Implementation of Rule-Based Expert System*. McGraw-Hill, Inc. New York.
- Manik, Janangur. 2004. *Sistem Pakar Troubleshooting hardware PC Menggunakan Decision Tree dengan Multiple Branches*. Tesis Magister Ilmu Komputer, UGM.
- Mulyanta, Edi S. 2005. *Kupas Tuntas Telepon Seluler Anda*. Penerbit Andi. Yogyakarta.
- Turban, Efraim., and Jay E. Aronson. 2000. *Decision Support Systems and Intelligent Systems*. Prentice Hall Inc. New Jersey.
- Yudatama, U. 2008. Sistem Pakar untuk Diagnosis Kerusakan Mesin Mobil Panther Berbasis Mobile. *Journal Teknologi* 1,(2),212-218.