

# PENGEMBANGAN SISTEM PAKAR BERBASIS ANDROID UNTUK MENDIAGNOSA KERUSAKKAN PADA PERANGKAT KOMPUTER

## *THE DEVELOPMENT OF ANDROID BASED EXPERT SYSTEM TO DIAGNOSE COMPUTER PROBLEM*

Oleh: Nafis Akhsan, Universitas Negeri Yogyakarta, nafisakhsan@gmail.com

### **Abstrak**

Penelitian ini bertujuan untuk mengembangkan sebuah sistem pakar berbasis android guna mendiagnosa kerusakan pada perangkat komputer. Hal ini dilatarbelakangi karena proses identifikasi sumber masalah pada perangkat komputer bukanlah merupakan hal yang mudah. Sehingga dengan adanya aplikasi sistem pakar ini diharapkan mampu mempermudah proses identifikasi yang dilakukan. Metode pengembangan perangkat lunak yang digunakan adalah metode *waterfall*. Tahap pengembangan meliputi proses analisis kebutuhan, desain, pengkodean, dan pengujian. Guna menghasilkan aplikasi yang baik, standar ISO 9126 dipakai sebagai acuan pada tahap pengujian. Aspek kelayakan pada standar ISO 9126 yang dipakai meliputi aspek *functionality*, *reliability*, *maintainability*, *portability*, dan *usability*. Hasil penelitian yang telah dilakukan menunjukkan hasil bahwa : 1) hasil pengembangan aplikasi sistem pakar telah berhasil melalui serangkaian tahap pada proses pengembangan perangkat lunak yang dimulai dari tahap analisis kebutuhan hingga proses pengujian, 2) hasil pengujian yang dilakukan menunjukkan bahwa aplikasi sistem pakar yang dibuat telah berhasil memenuhi aspek kelayakan standar perangkat lunak yang diacu yaitu aspek *functionality*, *reliability*, *maintainability*, *portability*, dan *usability*.

Kata kunci: sistem pakar, aplikasi android, ISO 9216

### **Abstract**

*This research aims to develop an Android-based expert system to diagnose problem in computer device. This is because the identifying process of the computer device problem are not an easy thing, so with the application of expert system is expected to help solve the problem. Software development method is the waterfall method. The development phases include the process of requirements analysis, design, coding, and testing. In order to produce a standard application, ISO 9126 is used as a reference at the testing stage. Feasibility aspects on the ISO 9126 standard are functionality, reliability, maintainability, portability, and usability. The results of the research that has been conducted shows that: 1) the result of the development of expert system application has been successful through a series of stages in the software development process that starts from the needs analysis phases to the process of testing, 2) the results of testing showed that the application of expert system created has successfully meet the eligibility aspect of software standards referenced which are aspects of functionality, reliability, maintainability, portability, and usability.*

Keywords: expert system, android application, ISO 9126

## PENDAHULUAN

Perbaikan kerusakan pada perangkat komputer dapat menjadi sesuatu yang sulit bila tidak diketahui penyebab masalah secara pasti. Oleh sebab itu perlu dilakukan diagnosa penyebab kerusakan terlebih dahulu sebelum melakukan perbaikan. Meski demikian, proses dalam melakukan diagnosa bukan hal yang mudah. Leng dan Teng (1991:121) menyatakan bahwa melakukan diagnosa sebuah kerusakan merupakan pekerjaan membutuhkan pengetahuan yang sangat intensif, tidak terkecuali diagnosa kerusakan pada perangkat komputer. Oleh sebab itu diperlukan cara atau metode khusus yang dapat mempermudah proses diagnosa.

Program seperti sistem pakar dapat menjadi solusi akan masalah terkait proses diagnosa tersebut. Sebab, “sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia” (Turban dkk, 2005:708). Sehingga pengguna sistem pakar dalam melakukan proses diagnosa tidak hanya mengandalkan pengetahuan dan pengalaman pribadi, namun juga menggunakan pengetahuan dan pengalaman orang lain (ahli).

Selain itu perkembangan *smartphone* berbasis android di Indonesia saat ini sangat pesat. Data dari statista (2014) menunjukkan bahwa android setidaknya menguasai setidaknya 60% pangsa pasar perangkat mobile di Indonesia. Terkait dengan fitur dan kemudahan yang ditawarkan android semisal kemampuan komputasi dan portabilitasnya, memunculkan pertanyaan “mungkinkah sistem pakar dibuat pada perangkat berbasis android?”.

Penelitian yang dilakukan oleh Abdelhamid dan El-Helly (2013) menunjukkan bahwa sistem pakar dapat dikembangkan dalam sebuah perangkat mobile dengan model desain *expert system as standalone application*. Selain itu penelitian lain oleh Singh (2014) yang berhasil mengembangkan aplikasi sistem pakar pada android untuk bidang kesehatan bernama ICare, menunjukkan potensi dapat dikembangkannya

aplikasi sistem pakar di android.

Berangkat dari latar belakang tersebut, melihat masalah yang ada dan potensi solusinya. Maka penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pakar terkait diagnosa kerusakan pada komputer yang akan berjalan pada perangkat berbasis android. Selain itu penelitian ini juga meninjau kelayakan aplikasi yang dibuat berdasarkan aspek kelayakan ISO 9126 yang meliputi aspek *functionality*, *reliability*, *maintainability*, *portability*, dan *usability*.

## METODE PENELITIAN

Metode penelitian berisi tentang jenis penelitian, waktu dan tempat penelitian, subjek penelitian, prosedur, instrumen penelitian, teknik pengumpulan data, dan teknik analisis data untuk pengembangan sistem pakar berbasis android untuk mendiagnosa kerusakan pada perangkat komputer.

### Jenis Penelitian

Penelitian ini dilakukan dengan menggunakan metode *Research and Development*. Metode *Research and Development* adalah metode penelitian yang digunakan untuk mengembangkan dan melakukan uji validitas terhadap suatu produk yang dikembangkan.

### Waktu dan Tempat Penelitian

Penelitian dilakukan pada bulan Februari sampai dengan November 2015. Proses pengembangan dan validasi aplikasi dilakukan di Jurusan Pendidikan Elektronika Fakultas Teknik Universitas Negeri Yogyakarta (UNY). Sedangkan untuk penelitian terkait pengujian aspek *usability* dilakukan di SMK N 2 Depok Sleman pada bulan November 2015.

### Subjek Penelitian

Subjek penelitian untuk aspek *functionality*, *reliability*, *maintainability*, dan *portability* adalah aplikasi sistem pakar yang dikembangkan.

Sedangkan untuk subjek penelitian untuk aspek *usability* adalah siswa kelas x jurusan Teknik Komputer dan Jaringan SMK N 2 Depok Sleman. Sebesar jumlah sampel untuk penelitian aspek *usability* adalah sebesar 32 orang siswa.

### Prosedur

Penelitian dan pengembangan aplikasi sistem pakar ini menggunakan model pengembangan perangkat lunak model *waterfall*. Pressman (2002:37) menyatakan bahwa model *waterfall* memiliki 4 tahapan didalamnya yaitu: 1) analisis kebutuhan, 2) desain, 3) pengkodean, 4) pengujian.

### Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini antara lain:

1. *Checklist* tes pengujian fungsi-fungsi untuk pengujian aspek *functionality*.
2. Program findbugs dan eclipse metrics plug-in untuk menentukan nilai *defect density* pada pengujian aspek *reliability*.
3. Ukuran-ukuran (*metric*) yang dikembangkan oleh Heitlager, dkk (2007) untuk menganalisis aspek *maintainability* aplikasi sistem pakar.
4. Kuisisioner untuk pengujian aspek *usability* yang dibuat berdasarkan *computer usability satisfaction questionnaires* yang dikembangkan oleh Lewis (1993).
5. Perangkat *smartphone* android dan layanan *cloud test* Testdroid untuk pengujian aspek *portability*.

### Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah :

1. Studi literatur, dilakukan untuk mendapatkan pemahaman mengenai sistem pakar yang dibuat. Studi literatur juga digunakan untuk memperoleh sumber pengetahuan yang akan dipakai dalam aplikasi sistem pakar. Literatur yang digunakan untuk sumber pengetahuan adalah buku karangan Rosenthal, Morris (2013) yang berjudul “*Computer Repair With*

2. Observasi, dilakukan untuk mengumpulkan data terkait dengan pengujian aspek kualitas *reliability, maintainability, dan portability*.
3. Kuisisioner, digunakan untuk mengumpulkan data terkait aspek kualitas *functionality* dan *usability*.

### Teknik Analisis Data

Analisis data terkait pengujian aspek kualitas *functionality, portability, dan usability* dihitung dengan melakukan presentase hasil tes. Rumus persentase yang digunakan adalah sebagai berikut :

$$\text{Presentase tes} = \frac{\text{Skor tes yang dicapai}}{\text{Skor tes ideal}} \times 100 \% \quad (1)$$

Hasil penghitungan presentase kemudian dikonversi menjadi pernyataan predikat. Hasil penyusunan konversi tersebut dapat dilihat pada tabel 1 (Guritno, dkk, 2011:122).

Tabel 1. Hasil Penyesuaian Konversi Interpretasi Presentase

No	Persentase	Interpretasi
1	0% - 20%	Sangat Buruk
2	21% - 40%	Buruk
3	41% - 60%	Cukup
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

Analisis data terkait dengan aspek *reliability* adalah dengan menggunakan menghitung nilai *defect density* dari aplikasi sistem pakar. Rumus untuk menghitung nilai *defect density* adalah:

$$\text{Defect density} = \frac{\text{Number of defect}}{\text{KLOC}} \quad (2)$$

Nilai *defect density* kemudian dibandingkan dengan jumlah perkiraan *defect density* untuk setiap ukuran proyek pada tabel 2 McConnel (2004:652).

Tabel 2. Perkiraan jumlah *defect density*

Ukuran Project (KLOC)	Density
< 2K	0-25 Defect per KLOC
2K - 16K	0-40 Defect per KLOC

Ukuran Project (KLOC)	Density
16K - 64K	0.5–50 Defect per KLOC
64K - 512K	2-70 Defect per KLOC
> 512K	4-100 Defect per KLOC

Setelah itu nilai defect density juga dibandingkan dengan standar berikut (McConnel, 2004:558) :

1. *Industry Average* : 1-25 error tiap 1 KLOC
2. *Microsoft Application* : 10-20 error tiap 1 KLOC pada tiap tahap pengujian *in house* dan 0.5 error tiap KLOC pada tahap peluncuran

Analisis aspek *maintainability* menggunakan *metric* yang disusun oleh Heitlager, dkk (2007:33) seperti dapat dilihat pada gambar 1. Cara pengukuran adalah dengan mengukur *source code properties* yang terdiri dari *volume, complexity per unit, duplication, unit size, dan unit testing*.

	source code properties				
	volume	complexity per unit	duplication	unit size	unit testing
ISO 9126 maintainability					
analysability	x		x	x	x
changeability		x	x		
stability					x
testability		x		x	x

Gambar 1. *Metric Maintainability*

*Volume* diukur dengan menghitung jumlah total *Line of Code* (LOC) yang ada pada aplikasi sistem pakar. Hasil pengukuran kemudian dibandingkan dengan standar pada tabel 3 (Heitlager, dkk 2007:34) .

Tabel 3. Kriteria Jumlah Besaran KLOC

Rank	KLOC
	Java
++	0-66
+	66-246
O	246-665
-	665-1310
--	>1310

*Complexity per unit* diukur dengan menghitung rata-rata *cyclomatic complexity* (CC). Hasil penghitungan kemudian dibandingkan dengan tabel 4 (Heitlager, dkk 2007:35).

Tabel 4. Kategori standar *cyclomatic complexity*

CC	Risk Evaluation
1-10	Simple, without much risk
11-20	More complex, moderate risk
21-50	Complex, high risk
>50	Untestable, very high risk

*Duplication* dihitung banyaknya kode yang sama pada *source code* aplikasi. Hasil penghitungan tersebut kemudian dibandingkan dengan tabel 5 (Heitlager, dkk 2007:36).

Tabel 5. Kategori penilaian duplikasi kode

Rank	Persentase Duplikasi
++	0-3%
+	3-5%
O	5-10%
-	10-20%
--	20-100%

*Unit size* dihitung dengan mengukur banyaknya LOC pada *method-method* yang ada di *source code* aplikasi. Hasil perhitungan dibandingkan dengan kriteria pada tabel 3.

*Unit testing* diukur dengan banyaknya *unit test* yang berhasil dijalankan. Hasil pengukuran kemudian dibandingkan dengan tabel 6 (Heitlager, dkk 2007:36).

Tabel 6. Penilaian ketercakupan *unit testing*

Rank	Ketercakupan Unit Testing
++	95%-100%
+	80%-95%
O	60%-80%
-	20%-60%
--	0%-20%

## HASIL PENELITIAN DAN PEMBAHASAN

Hasil penelitian dan pembahasan berisi tentang proses pengembangan perangkat lunak dan pembahasannya. Pengembangan perangkat lunak menggunakan metode pengembangan

*waterfall*. Tahap-tahap pengembangan dalam metode tersebut meliputi tahap analisis kebutuhan, desain, pengkodean, dan pengetesan.

### Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan pengumpulan data baik secara studi literatur maupun observasi. Hal tersebut bertujuan untuk menentukan bagaimana aplikasi sistem pakar akan dibuat, serta fungsi-fungsi yang dibutuhkan. Selain itu pada tahap ini juga ditentukan sumber pengetahuan yang ada digunakan pada aplikasi sistem pakar.

### Desain

Tahap desain adalah tahap dimana desain aplikasi sistem pakar dibuat. Terdapat 3 desain yang dilakukan yaitu terkait desain *Unified Modeling Language* (UML), desain *user interface*, dan desain database.

### Pengkodean

Tahap pengkodean merupakan tahap implementasi dari desain yang telah dibuat. Desain UML diterjemahkan menjadi rancangan yang digunakan untuk membuat seluruh proyek aplikasi sistem pakar. Desain *user interface* diterjemahkan menjadi tampilan program yang dibuat. Serta membuat gambar dan icon yang nanti digunakan pada tampilan. Desain database diterjemahkan menjadi database dengan jenis database yang digunakan yaitu SQLite. Terdapat 9 buah database yang dibuat untuk menyimpan aturan diagnosa yang ada.

### Pengetesan

Pengetesan dilakukan dengan menggunakan ISO 9126 sebagai acuan standar kelayakan. Pengetesan dilakukan pada aspek *functionality*, *reliability*, *maintainability*, *portability*, dan *usability*.

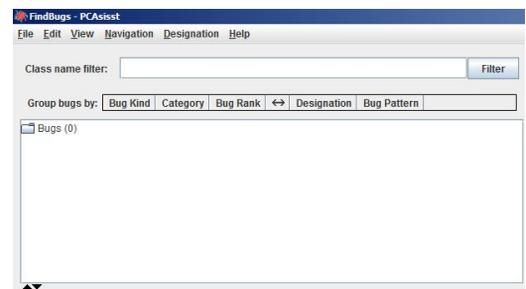
#### 1. Pengetesan Aspek *Functionality*

*Functionality* dites dengan menggunakan *test case* pengujian fungsi-fungsi. Terdapat 20 prosedur pengetesan yang dilakukan. Tes dilakukan oleh 2 orang responden yang

berkecimpung dibidang rekayasa perangkat lunak dan pemrograman. Hasil tes menunjukkan, bahwa semua prosedur tes berhasil dijalankan. Jika dihitung presentase keberhasilannya dengan rumus 1, maka menghasilkan presentase sebesar 100%. Hasil presentase dibandingkan dengan tabel 1, menunjukkan bahwa aspek *functionality* aplikasi sistem pakar “sangat baik”.

#### 2. Pengetesan Aspek *Reliability*

Aspek *reliability* diukur dengan besarnya nilai *defect density* (Malaiya,2005). Proses pengukuran dimulai dengan menghitung terlebih dahulu banyaknya LOC yang ada pada proyek aplikasi dengan bantuan metrics plug-in pada program eclipse. Hasil LOC yang terdapat adalah sebesar 3449 LOC(3,449 KLOC). Jumlah *defect* yang ada dicari dengan bantuan program findbugs. Hasil pencarian ditemukan sebesar 0 *defect*.



Gambar 2. Hasil pencarian *defect* dengan program findbugs.

Jumlah KLOC dan *defect* kemudian dihitung dengan rumus 2. Hasil penghitungan menghasilkan nilai sebesar 0 *defect density*. Nilai tersebut lalu dibandingkan dengan tabel 2. Hasil perbandingan itu menyatakan bahwa nilai *defect density* yang ada masih pada rentang yang sesuai untuk ukuran proyek sebesar 2-16 KLOC. Selain itu jika dibandingkan dengan standar *Industry Average* dan *Microsoft Application*, hasil dapat dilihat pada tabel 7.

Tabel 7. Perbandingan Standar *Defect Density*

Nama Standar	Nilai Standar	<i>Defect Density</i>	Keterangan
<i>Industry Average</i>	1 -25 <i>defect density</i>	0 <i>defect density</i>	<b>Lolos</b>
<i>Microsoft Application</i>	0,5 <i>defect density</i>		<b>Lolos</b>

### 3. Pengetesan Aspek Maintainability

Aspek *maintainability* dites dengan panduan *maintainability metric* pada gambar 1. Lebih lanjut bahwa pengetesan dilakukan dengan mengetes *source code properties* aplikasi. *Source code properties* yang dites meliputi *volume, complexity per unit, duplication, unit size, dan unit testing*. Setiap *source code properties* yang dites mempengaruhi tiap-tiap sub-karakteristik *maintainability*. Secara garis besar semakin baik hasil tes pada *source code properties*, semakin baik pula pengaruhnya pada tiap-tiap sub-karakteristik *maintainability*. Selain itu terdapat sistem ranking yang dipakai yaitu ++ / + / o / - / -- , dengan ranking -- adalah sangat buruk dan ++ adalah sangat baik.

*Volume* dites dengan menentukan banyaknya LOC yang ada. Dengan bantuan metrics plug-in pada program eclipse menunjukkan bahwa banyaknya LOC adalah sebesar 3449 LOC. Hasil tersebut jika dibandingkan dengan tabel 3 memperoleh ranking “++”.

*Complexity per unit* diukur dengan rata-rata besar *cyclomatic complexity* pada proyek aplikasi yang dibuat. Hasil pengukuran dengan menggunakan metrics plug-in pada program eclipse menunjukkan rata-rata sebesar 1,872 CC. Jika hasil tersebut dibandingkan dengan tabel 4, maka *complexity per unit* termasuk kategori “**simple, without much risk**”.

*Duplication* diukur dengan bantuan program *PMD copy paste detector*. Hasil pengukuran dihasilkan bahwa terdapat duplikasi sebesar 131 LOC. Dengan banyaknya jumlah LOC pada proyek aplikasi sebesar 3449LOC, jika dilakukan persentase dihasilkan persentase sebesar 3,79%. Persentase tersebut jika dibandingkan dengan tabel 5, maka diperoleh ranking “+”.

*Unit size* diukur dengan menentukan besarnya LOC untuk method-method yang ada pada proyek aplikasi. Hasil pengukuran dengan metrics plug-in pada program eclipse menunjukkan bahwa besar method-method yang ada adalah 1872 LOC. Jika dibandingkan dengan tabel 3, maka diperoleh ranking “++”.

*Unit testing* dites dengan melakukan *unit test* dengan bantuan framework Junit. Hasil tes menunjukkan bahwa semua *unit* yang dites dapat berjalan dengan baik. Hasil persentase yang dilakukan diperoleh hasil 100%. Jika dibandingkan dengan tabel 6 diperoleh ranking “++”.

### 4. Pengetesan Aspek Portability

Aspek *portability* dites dengan menjalankan aplikasi sistem pakar yang telah dibuat pada perangkat *smartphone* berbasis android yang berbeda-beda. Perbedaan yang mendasar sebagai bahan patokan adalah versi android dan ukuran layar. Tes dijalankan pada 4 perangkat fisik dan 8 perangkat secara *cloud test* dengan bantuan layanan testdroid. Hasil tes dapat dilihat pada tabel 8.

Tabel 8. Hasil tes *portability*

Jumlah Perangkat	Proses Instalasi	Proses Aplikasi Berjalan
12 Perangkat	Aplikasi dapat dipasang disemua perangkat yang ada	Aplikasi dapat dijalankan disemua perangkat yang ada

### 5. Pengetesan Aspek Usability

Aspek *usability* dites dengan menggunakan kuisisioner. Kuisisioner dibagikan kepada responden berjumlah 32 orang siswa. Hasil perhitungan persentase kuisisioner *usability* diperoleh hasil sebesar 76,64%. Hasil persentase tersebut jika dibandingkan dengan tabel 1, maka diperoleh kriteria “**baik**”.

#### Keterbatasan Penelitian

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi sistem pakar yang baik dan dapat diandalkan. Meskipun demikian pada praktiknya terdapat beberapa keterbatasan yang terjadi selama proses penelitian berlangsung. Keterbatasan yang dihadapi tersebut antara lain:

1. Pada proses akuisisi pengetahuan untuk menjadi sebuah aturan yang digunakan dalam proses konsultasi, peneliti hanya

menggunakan sumber pengetahuan terdokumentasi.

2. Pada pengetesan untuk faktor *portability*, perangkat yang digunakan jumlahnya masih terbatas. Oleh karena itu peneliti berfokus bahwa perangkat yang digunakan setidaknya dapat mewakili versi android yang berbeda. Aplikasi sistem pakar dirancang untuk beroperasi pada android versi 4.0 dan di atasnya.
3. Pada pengetesan aplikasi dilapangan atau ketika diujicobakan pada pengguna, pengetesan yang dilakukan hanya dapat sebatas untuk mengetahui tingkat *usability* aplikasi saja. Pengetesan tidak dilakukan dengan menghadapi permasalahan langsung pada perangkat PC. Hal tersebut dikarenakan tidak adanya perangkat PC yang mengalami kerusakan pada saat penelitian dilakukan.

## SIMPULAN DAN SARAN

### Simpulan

Berdasarkan penelitian yang telah dilakukan dalam pengembangan aplikasi sistem pakar untuk diagnosa kerusakan pada komputer, maka peneliti dapat mengambil beberapa kesimpulan yaitu:

1. Aplikasi sistem pakar dikembangkan dengan menggunakan model pengembangan perangkat lunak sekuensial linier atau dikenal juga dengan model *waterfall*. Tahapan pengembangan dalam model ini meliputi proses analisis pengetahuan, desain, pengkodean atau implementasi, dan pengetesan.
2. Kelayakan aplikasi diukur dengan menggunakan kriteria kelayakan perangkat lunak ISO 9126. Pengetesan kelayakan aplikasi oleh pengembang meliputi aspek *functionality*, *reliability*, *maintainability*, serta *portability*. Aspek *functionality* berdasarkan pengetesan yang telah dilakukan, menunjukan semua fungsi yang ada dapat berjalan dengan baik. Aspek *reliability* berdasarkan hasil pengetesan yang dilakukan didapatkan hasil besarnya nilai *defect density* yang dihasilkan

*Pengembangan Sistem Pakar ..... (Nafis Akhsan) 7* adalah sebesar 0 *defect density*. Hasil pengetesan aspek *portability* didapatkan bahwa aplikasi dapat dipasang dan dijalankan pada perangkat-perangkat berbeda yang telah disediakan. Sedangkan untuk aspek *maintability*, aplikasi sistem pakar ini mendapat hasil baik. Aplikasi sistem pakar yang dibuat memiliki tingkat *usability* yang baik. Pernyataan tersebut berdasarkan pada hasil pengetesan terhadap 32 responden. Hasil pengetesan tersebut dihasilkan presentase *usability* sebesar 76,64 %.

### Saran

1. Aplikasi masih dapat dikembangkan dengan model *client-server* agar dapat lebih mudah dan fleksibel dalam melakukan manajemen data.
2. Proses diagnosa masih dapat ditambah lagi dengan proses diagnosa periperal lain seperti input-output device, networking dan sebagainya.
3. Desain aplikasi yang ada saat ini masih sederhana. Oleh karena itu perbaikan desain masih diperlukan agar lebih menarik dan memudahkan pengguna.
4. Penelitian masih dapat dikembangkan untuk meneliti bagaimana pengaruh aplikasi sistem pakar ini terhadap peningkatan kinerja teknis dalam melakukan perbaikan perangkat komputer.

### DAFTAR PUSTAKA

- Abdelhamid, Y & El-Helly, M. (2013) A New Approach for Developinh Diagnostic Expert Systems on Mobile Phones. *Jurnal Communications in Information Science and Management Enginerring*. (Volume 3, Issue 8). Hlm. 374-384.
- Guritno, S., Sudaryono, & Raharja, U. (2011). *Theory and application of IT Research: Metodologi Penelitian Teknologi Informasi*. Yogyakarta: Penerbit ANDI.
- Heitlager, Ilja., Kuipers, Tobias., & Visser,Joost . (2007). A Practical Model for Measuring Maintainability. *QUATIC '07 Proceedings of the 6th International Conference on Quality of Information and Communications Technology*. Hlm. 30-39.

- Leng, G.W & Teen, L.K. (1992). ESPCRM an Expert System for Personal Computer Repair and Maintenance. *Jurnal Engineering Applications of Artificial Intelligence*. (Volume 5, Issue 2). Hlm. 121-133.
- Lewis, J. R. (1993). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human Computer Interaction*. Hlm. 1-39
- Malaiya, Yashwant K. (2005). Software Reliability and Security. *Jurnal Encyclopedia of Library and Information Science*. Hlm 1-12.
- Pressman, Roger S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku I)*. Penerjemah: LN Harnaningrum. Yogyakarta: Penerbit ANDI
- Rosenthal, Morris. (2013). *Computer Repair With Diagnostic Flowcharts Third Edition: Troubleshootig PC Hardware Problems from Boot failure to Poor Performance*. Amerika: Foner Books.
- Singh, Suraj et al. (2014). Healthcare Services using Android Devices. *The International Journal of Engineering and Science*. (Volume 3, issue 4). Hlm 41-45.
- Statista. (2014). *Mobile OS: Market Share in Indonesia 2011-2014*. Diakses pada 31 Oktober 2014, dari <http://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/>
- Turban, E. Aronson, J.E & Liang Peng Ting. (2005). *Decision Support Systems and Intelligent Systems – 7th Ed. Jilid 2 (Sistem pendukung Keputusan dan Sistem Cerdas)*. Penerjemah: Siska Primaningrum. Yogyakarta: Penerbit ANDI.

Penguji Utama



Dessy Irmawati, M.T  
NIP. 19791214 201012 2 002

Pembimbing



Prof. Herman Dwi Surjono, Ph. D  
NIP. 19640205 198703 1 001