PENGEMBANGAN DAN ANALISIS KUALITAS

APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER

Oleh :

Kifni Taufik Darmawan

08520241008

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan bahasa pemrograman Java dan database SQLite dan melakukan analisis kualitas pada aplikasi yang dikembangkan, khususnya pada faktor kualitas *correctness*, *functionality*, *portability*, dan *usability*.

Pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan kaidah rekayasa perangkat lunak (*software engineering*) yaitu dimulai dari proses perencanaan (*planning*), modeling (*perancangan*), konstruksi (*construction*), dan penyebaran (*deployment*). Analisis faktor kualitas correctness dilakukan dengan perhitungan jumlah error / KLOC. Analisis faktor kualitas *functionality* dilakukan dengan pengujian setiap fungsi aplikasi. Analisis faktor kualitas *portability* dilakukan dengan percobaan penjalanan aplikasi pada beberapa sistem operasi yang berbeda. Analisis faktor kualitas *usability* dilakukan dengan metode kuesioner dengan responden Siswa Kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta.

Hasil pengembangan aplikasi yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dalam bentuk file *runnable*. Hasil analisis kualitas menunjukan bahwa aplikasi yang dikembangkan memenuhi semua standar faktor kualitas yang diujikan yaitu *correctness*, *functionality*, *portability*, dan *usability*.

Kata kunci : sistem pakar, java, sqlite, *software quality*, *correctness*, *functionality*, *portability*, *usability*.

THE DEVELOPMENT AND QUALITY ANALYSIS OF

COMPUTER FAILURE DIAGNOSTIC EXPERT SYSTEM APPLICATION

By :

Kifni Taufik Darmawan

08520241008

ABSTRACT

This research aims to develop Computer Failure Diagnostic Expert System Application using Java programming language and SQLite database; and perform quality analysis of developed application, especially on four quality factors : correctness, functionality, portability, and usability.

The development of Computer Failure Diagnostic Expert System performed using software engineering principle that starts from planning, modeling, construction, and deployment processes. The analysis of correctness quality factor performed by calculate the number of error per KLOC (Kilo Lines of Code). The analysis of functionality quality factor performed by testing each function of application. The analysis of portability quality factor performed by testing the application to run on multiple different operating systems. The analysis of usability quality factor conducted by questionnaire method with student of Class XI-TKJ of SMK Muhammadiyah 2 Yogykakarta as the respondents.

The results of the application development processes is a Computer Failure Diagnostic Expert System as runnable file. The result of quality analysis show that application developed meet each standard of quality factor tested : correctness, functionality, portability, and usability.

Keywords : expert system, java, sqlite, software quality, correctness, functionality, portability, usability

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER

SKRIPSI

Diajukan Kepada Fakultas Teknik
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Pendidikan

Oleh :
Kifni Taufik Darmawan
NIM 08520241008

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2012

LEMBAR PERSETUJUAN

Skripsi yang berjudul **"PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER"** yang disusun oleh Kifni Taufik Darmawan, NIM 08520241008 ini telah disetujui oleh pembimbing untuk diujikan.

Yogyakarta, 5 Juli 2012

Menyetujui,

Pembimbing

Handaru Jati, Ph.D.

NIP 19740511 199903 1 002

LEMBAR PENGESAHAN

Skripsi yang berjudul **"PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER"** yang disusun oleh Kifni Taufik Darmawan, NIM 08520241008 ini telah dipertahankan di depan Dewan Penguji pada tanggal 31 Juli 2012 dan dinyatakan lulus.

DEWAN PENGUJI

Nama	Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D	Ketua Penguji	.....	15/8/2012
Herman Dwi Surjono, Ph.D	Sekretaris	.....	15/8 2012
Masduki Zakarijah, MT	Penguji Utama	.....	15/8 2012

Yogyakarta, 15 Agustus 2012

Fakultas Teknik UNY

Dekan

Dr. Moch. Bruri Triyono

NIP 19560216 198603 1 003

HALAMAN PERNYATAAN

Yang bertandatangan di bawah ini saya:

Nama : Kifni Taufik Darmawan

NIM : 08520241008

Program Studi : Pendidikan Teknik Informatika

Fakutas : Fakultas Teknik UNY

Judul Penelitian : **PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER**

Menyatakan bahwa penelitian ini adalah hasil pekerjaan saya sendiri, dan sepanjang pengetahuan saya tidak berisi materi yang telah dipublikasikan atau ditulis oleh orang lain atau telah dipergunakan dan diterima sebagai persyaratan penyelesaian studi pada universitas atau institusi lain, kecuali pada bagian-bagian tertentu yang saya ambil sebagai acuan.

Apabila ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 7 Juli 2012

Yang Menyatakan

Kifni Taufik Darmawan

NIM 08520241008

MOTTO

Ning ndunyo piro suwene, njur bali ning panggonane

Ning akherat yo sejatine, mung amal becik sangune

Ojo ngucap "bodo yo ben", nggolek ilmu kudu telaten

Hidup bagaikan perjalanan bersepeda,

Kadang dihadapkan dengan jalan datar yang mungkin akan membosankan.

Kadang pula dihadapkan dengan tanjakan curam yang seolah menggoda kita untuk menyerah.

Tapi percayalah, setelah tanjakan pasti akan ada turunan sebagai "hadiah" dari kerja keras setelah melewati tanjakan.

Tapi "bonus" turunan tak akan menjadi sangat nikmat jika tak melewati tanjakan sebelumnya.

PERSEMBAHAN

Segala puji bagi Gusti Allah, Tuhan Semesta Alam yang tanpa henti – hentinya memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini.

Karya ini penulis persembahkan pada :

Bapak dan Ibu tercinta yang selalu memberikan do'a, kasih sayang dan dukungannya, semoga selalu diberkahi rahmat-Nya.

Hindun Fatmawati, Adikku tersayang yang selalu memberikan dukunganya selama ini, semoga selalu diberi kelancaran dalam menempuh pilihan hidup.

Istri dan anak (-anakku) kelak nanti.

Inggrit Raberta yang selalu memberikan pengertianya selama ini. Semoga ini menjadi langkah awal yang baik untuk tercapainya masa depan impian.

Teman – teman kelas E PTI 2008 "KOMBRE" yang selalu bersama – sama dalam suka duka selama masa kuliah.

Teman – teman kos Kepuh 839B yang selalu bersama – sama dari awal bulan hingga akhir bulan ☺

Teman – teman komunitas maupun non komunitas sepeda yang telah menghadirkan kebahagiaan lain dalam petualangan bersepeda

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI

SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER

Oleh :

Kifni Taufik Darmawan

08520241008

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan bahasa pemrograman Java dan database SQLite dan melakukan analisis kualitas pada aplikasi yang dikembangkan, khususnya pada faktor kualitas *correctness*, *functionality*, *portability*, dan *usability*.

Pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan kaidah rekayasa perangkat lunak (*software engineering*) yaitu dimulai dari proses perencanaan (*planning*), modeling (*perancangan*), konstruksi (*construction*), dan penyebaran (*deployment*). Analisis faktor kualitas correctness dilakukan dengan perhitungan jumlah error / KLOC. Analisis faktor kualitas *functionality* dilakukan dengan pengujian setiap fungsi aplikasi. Analisis faktor kualitas *portability* dilakukan dengan percobaan penjalanan aplikasi pada beberapa sistem operasi yang berbeda. Analisis faktor kualitas *usability* dilakukan dengan metode kuesioner dengan responden Siswa Kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta.

Hasil pengembangan aplikasi yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dalam bentuk file *runnable*. Hasil analisis kualitas menunjukan bahwa aplikasi yang dikembangkan memenuhi semua standar faktor kualitas yang diujikan yaitu *correctness*, *functionality*, *portability*, dan *usability*.

Kata kunci : sistem pakar, java, sqlite, *software quality*, *correctness*, *functionality*, *portability*, *usability*.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan Kehadirat Tuhan Yang Maha Esa atas kuasa dan limpahan rahmat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan di Fakultas Teknik Universitas Negeri Yogyakarta.

Penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dan peran dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak berikut:

1. Bapak Dr. Moch. Bruri Triyono, selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
2. Bapak Muhammad Munir, M.Pd., selaku Ketua Jurusan Pendidikan Teknik Elektronika FT UNY.
3. Ibu Dr. Ratna Wardani, selaku Ketua Program Studi Pendidikan Teknik Informatika UNY.
4. Ibu Umi Rochayati, M.T., selaku dosen penasehat akademik.
5. Bapak Handaru Jati, Ph.D., selaku dosen pembimbing skripsi.
6. Semua pihak yang telah membantu penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini, namun penulis tetap berharap skripsi ini bermanfaat bagi pengembangan ilmu pengetahuan terutama dalam kaitannya dengan pengembangan sistem pakar.

Yogyakarta, Mei 2012

Penulis

Kifni Taufik Darmawan

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi komputer diikuti pula dengan meningkatnya jumlah pengguna komputer di dunia. Menurut laporan Forester Research, perusahaan yang bergerak dalam riset dan konsultan teknologi informasi, pada tahun 2008 jumlah komputer di dunia telah mencapai 1 Milliar unit dan diperkirakan dengan perkembangan 12% per tahun jumlah komputer di dunia akan lebih dari 2 Milliar unit pada tahun 2015 (Worldometers, 2011). Data tersebut menunjukan bahwa jumlah pengguna komputer semakin meningkat jumlahnya dengan prosentase peningkatan yang cukup besar. Seiring dengan meningkatnya jumlah pengguna komputer, permasalahan kerusakan komputer menjadi masalah yang cukup pelik. Hal ini dapat dimaklumi mengingat banyaknya pengguna komputer yang kurang memiliki pengetahuan tentang komputer, khususnya dalam menangani kerusakan komputer. Permasalahan ini secara umum dialami baik oleh individu, rumah tangga, dan juga institusi baik itu institusi swasta maupun pemerintah. Banyak sekali dana yang dikeluarkan untuk memperbaiki kerusakan komputer, padahal kerusakan komputer yang terjadi belum tentu rumit dan dapat diperbaiki secara mandiri.

Sistem Pakar (*Expert System*) sebagai salah satu hasil dari perkembangan ilmu komputer, khususnya di bidang kecerdasan buatan (*Artificial Intelegence*), dapat memberikan solusi untuk mengatasi masalah tersebut. Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti layaknya para pakar (*expert*). Sistem pakar yang baik dirancang agar dapat

menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para pakar/ahli. Dengan pengembangan sistem pakar, diharapkan bahwa orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli.

Penulis menganggap bahwa diperlukan sebuah aplikasi sistem pakar yang dapat membantu pengguna komputer dalam mendiagnosis kerusakan komputer dan membantu pengguna dalam memperbaikinya. Dengan ini, diharapkan pengguna komputer dapat mengatasi beberapa masalah komputer secara mandiri. Penggunaan sistem pakar ini juga ini tidak lantas menghilangkan peran ahli (expert) dalam hal ini tekniksi komputer, karena tidak semua permasalahan kerusakan komputer dapat ditangani oleh pengguna komputer secara mandiri. Untuk mengatasi kerusakan komputer yang rumit tetap dibutuhkan ahli untuk memperbaikinya.

Seperti halnya perangkat lunak lain, Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini dibuat berdasarkan kaidah rekayasa perangkat lunak (*software engineering*) mulai dari proses awal hingga akhir. Pada akhirnya, aplikasi sistem pakar yang dibuat diharapkan memenuhi standar kualitas perangkat lunak dalam kaidah rakayasa perangkat lunak (*software engineering*). Kualitas perangkat lunak ditentukan oleh berbagai faktor. Beberapa ahli maupun organisasi telah merumuskan kriteria – keriteria pengujian kualias perangkat lunak. McCall, Richards, dan Walter merumuskan kriteria – kriteria untuk melakukan pengujian kualitas perangkat lunak yang terdiri dari beberapa faktor – faktor kualitas yaitu : *maintainability, flexibility, testability, portability, reusability, interoperability, correctness, reliability, usability, integrity, dan efficiency*. Selain itu, *International Standard Organization* (ISO) juga mengeluarkan standar ISO-

9126 yang terdiri dari enam faktor kualitas yaitu *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability*.

Berdasarkan latar belakang tersebut, melalui Tugas Akhir ini Penulis bermaksud untuk mengembangkan Aplikasi Sistem Pakar untuk Diagnosis Kerusakan Komputer dan melakukan analisis kualitas aplikasi tersebut.

B. Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, dapat diidentifikasi beberapa permasalahan antara lain :

1. Permasalahan kerusakan komputer semakin banyak seiring dengan bertambahnya pengguna komputer.
2. Banyaknya pengguna komputer, atau lembaga yang mengeluarkan banyak dana untuk memperbaiki kerusakan komputer yang sebenarnya dapat diatasi sendiri.
3. Belum adanya aplikasi sistem pakar yang dapat membantu pengguna komputer dalam mendiagnosis dan memperbaiki komputer secara mandiri.
4. Belum adanya analisis kualitas pada aplikasi sistem pakar untuk mendiagnosis kerusakan komputer .

C. Pembatasan Masalah

Dalam penyusunan tugas akhir ini diberikan batasan masalah agar dalam penjelasannya nanti akan lebih mudah, terarah dan sesuai dengan yang diharapkan. Aplikasi sistem pakar yang akan dikembangakan akan difokuskan untuk diagnosis kerusakan komputer desktop dimana komponen – komponenya dapat dilepas, diganti, dan dipasang kembali dengan mudah. Aplikasi tersebut

akan dikembangkan menggunakan Bahasa Pemrograman Java dan Database SQlite. Dalam kaitanya dengan pengujian kualitas perangkat lunak, aplikasi tersebut akan diuji pada beberapa faktor kualitas penrangkat lunak yaitu : *correctness*, *functionality*, *portability*,dan *usability*.

D. Perumusan Masalah

Berdasarkan latar belakang masalah yang telah disampaikan sebelumnya, maka dapat diambil beberapa rumusan masalah yaitu :

1. Bagaimana mengembangkan Aplikasi Sistem Pakar untuk Mendiagnosis Kerusakan Komputer menggunakan bahasa pemrograman Java dengan dan database SQLite?.
2. Bagaimana analisis faktor kualitas *correctness* pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
3. Bagaimana analisis faktor kualitas *functionality* pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
4. Bagaimana analisis faktor kualitas *portability* pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
5. Bagaimana analisis faktor kualitas *usability* pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.

E. Tujuan Penelitian

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas akhir ini adalah sebagai berikut :

1. Mengembangkan aplikasi sistem pakar untuk diagnosis kerusakan komputer menggunakan bahasa pemrograman Java dengan dan database SQLite.

2. Mengetahui analisis faktor kualitas correctness pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
3. Mengetahui analisis faktor kualitas functionality pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
4. Mengetahui analisis faktor kualitas portability pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
5. Mengetahui analisis faktor kualitas usabilty pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.

F. Manfaat Penelitian

Berikut merupakan beberapa manfaat dari penulisan tugas akhir ini :

1. Membantu pengguna komputer dalam mendiagnosis dan menyelesaikan permasalahan kerusakan komputer secara mandiri.
2. Memberikan pembelajaran kepada pengguna komputer dalam hal kerusakan komputer dan cara menyelesaikanya.
3. Mendapatkan hasil analisis kualitas aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.
4. Aplikasi Sistem Pakar yang telah dibuat dapat menjadi referensi untuk pengembangan aplikasi sistem pakar (*expert system*) sejenis di kemudian hari.

BAB II

KAJIAN TEORI

A. Deskripsi Teori

1. Kecerdasan Buatan (*Artificial Intelligence*)

Menurut Hartati dan Iswanti (2008), Kecerdasan Buatan (Artificial Intelegence) adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak (software) dan perangkat keras (hardware) untuk menirukan tindakan manusia. Aktivitas menusia yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya.

Kecerdasan Buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Tujuan dari pengembangan kecerdasan buatan adalah :

1. Membuat mesin (komputer) menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan enterpreneurial)

(Winston dan Prendergast dalam (Sutojo, Mulyanto, & Suhartono, 2011))

Berdasarkan definisi tersebut, maka dapat dibedakan antara program atau aplikasi konvensional dengan kecerdasan buatan. Program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik. Jika ada informasi baru, sebuah program konvensional harus diubah untuk menyesuaikan diri dengan informasi baru tersebut. Selain diperlukan waktu yang relatif lama, kemungkinan terjadinya error juga cukup besar. Berbeda dengan kecerdasan

buatan yang memungkinkan program komputer dapat menyimpan informasi baru dalam sebuah basis pengetahuan (*knowledge base*) yang dapat digunakan pada masa yang akan datang.

Saat ini kecerdasan buatan merupakan cabang ilmu komputer yang selalu dikembangkan, dalam implementasinya kecerdasan buatan yang banyak ditemui saat ini dibagi dalam beberapa bidang sebagai beberapa bidang antara lain : Sistem Pakar (*Expert System*); Pengenalan Ucapan (*Speech Recognition*); *Game Playing*, Pengolahan Bahasa Alami (*Natural Language Processing*),dan Logika Fuzzy (*Fuzzy Logic*);

2. Sistem Pakar (*Expert System*)

a. Pengertian sistem pakar

Menurut Sutojo, Mulyanto dan Suhartono (2011), Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu masalah.

Sedangkan menurut Giarratano dan Riley dalam Hartati dan Iswanti (2008), Sistem Pakar merupakan salah satu cabang dari Kecerdasan Buatan (*Artificial Intelligence*) yang menggunakan pengetahuan – pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu.

b. Ciri – ciri sistem pakar

Menurut Sutojo, Mulyanto dan Suhartono (2011) , ciri – ciri sistem pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu.
2. Dapat menjelaskan alasan – alasan dengan cara yang dapat dipahami.

3. Bekerja berdasarkan kaidah/*rule* tertentu.
4. Mudah dimodifikasi.
5. Basis pengetahuan (*knowledge base*) dan mekasisme inferensi terpisah.

c. Komponen sistem pakar

Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar (expert) harus bisa melakukan hal – hal yang dapat dikerjalan oleh seorang pakar. Menurut Giarranato dan Riley dalam Hartati dan Iswati (2008), untuk membangun sistem yang seperti itu maka sebuah sistem pakar harus memiliki komponen – komponen sebagai berikut :

1) *antar muka pengguna (user interface).*

User Interface merupakan mekanisme yang digunakan oleh penggu*n*a untuk dan sistem pakar untuk berkomunikasi. *User interface* menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

2) *basis pengetahuan (knowledge base).*

Menurut Hartati dan Iswanti (2008), basis pengetahuan (*knowledge base*) merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan tersebut diperoleh dari akumulasi pengetahuan pakar dan sumber – sumber pengetahuan lainya.

Basis pengetahuan bersifat dinamis, bisa berkembang dari waktu ke waktu. Perkembangan ini disebabkan karena pengetahuan selalu bertambah. Dalam Sistem Pakar, basis pengetahuan terpisah dari mesin inferensi (*inference*

engine). Hal ini dilakukan agar pengembangan sistem pakar dapat dilakukan dengan leluasa tanpa mengganggu atau mengubah mesin inferensi.

3) *mesin inferensi (inference engine).*

Menurut Sutojo, Mulyanto, dan Suhartono (2011), Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan penalaran. Ada tiga teknik pengendalian yang digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

a) forward chaining.

Sutojo, Mulyanto dan Suhartono (2011) menjelaskan bahwa *forward chaining* adalah teknik inferensi yang dimulai dengan pengumpulan fakta – fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan *rules* yang ada sampai akhirnya didapat konklusi akhir.

b) backward chaining.

Sutojo, Mulyanto dan Suhartono (2011) menjelaskan bahwa *backward chaining* adalah teknik inferensi yang bekerja mundur. Proses dimulai dari goal (konklusi), kemudian pencarian dijalankan untuk mencocokkan apakah fakta – fakta cocok dengan *rules*.

4) *memori kerja (working memory).*

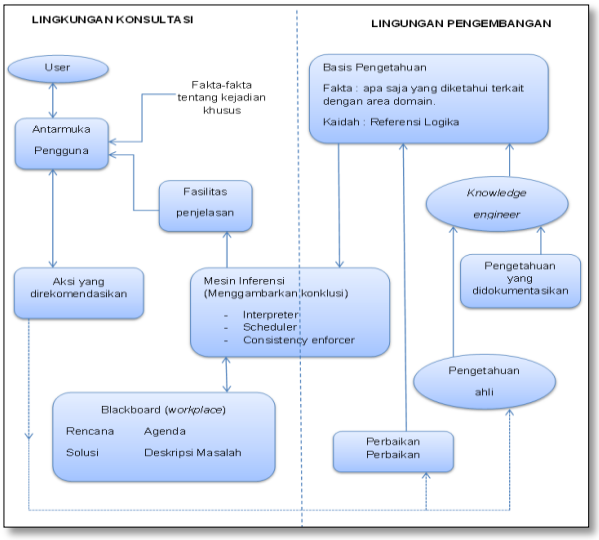
Dalam bukunya Hartati dan Iswanti (2008) menjelaskan bahwa memori kerja merupakan bagian dari sistem pakar yang menyimpan fakta – fakta yang

diperoleh saat dilakukan proses konsultasi. Fakta – fakta inilah yang nantinya akan diolah oleh mesin inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan untuk menentukan suatu keputusan pemecahan masalah. Konklusinya bisa berupa hasil diagnosis, tindakan, ataupun akibat.

d. Struktur sistem pakar

Sutojo, Mulyanto, dan Suhartono (2011) menjelaskan bahwa ada dua bagian penting dari Sistem Pakar, yaitu lingkungan pengembangan dan lingkingan konsultasi. Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen – komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base*. Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguuna mendapatkan pengetahuan dan nasihat dari Sistem pakar layaknya berkonsoltasi dengan seorang pakar.

Gambar 1 berikut menunjukan struktur Sistem Pakar yang menujukan hubungan komponen – komponen dalam sistem pakar.



Gambar 1. Struktur Sistem Pakar

e. Manfaat sistem pakar

Menurut Sutojo, Mulyanto dan Suhartono (2011), Sistem pakar mempunyai beberapa manfaat antara lain :

1. Meningkatkan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam menjadi bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mempu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi dilingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal, karena system pakar tidak pernah menjadi bosan, lelah atau sakit.
8. Meningkatkan kapabilitas sistem komputer.
9. Bisa digunakan sebagai media pelengkap atau pelatihan.
10. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari pakar.

f. Area permasalahan aplikasi sistem pakar

Menurut Sri Hartati dan Sari Hartanti (2008), Secara garis besar aplikasi system pakar dapat dikelompokan ke dalam beberapa kategori, seperti terlihat dalam Tabel berikut :

Tabel 1 : Area Permasalahan Sistem Pakar

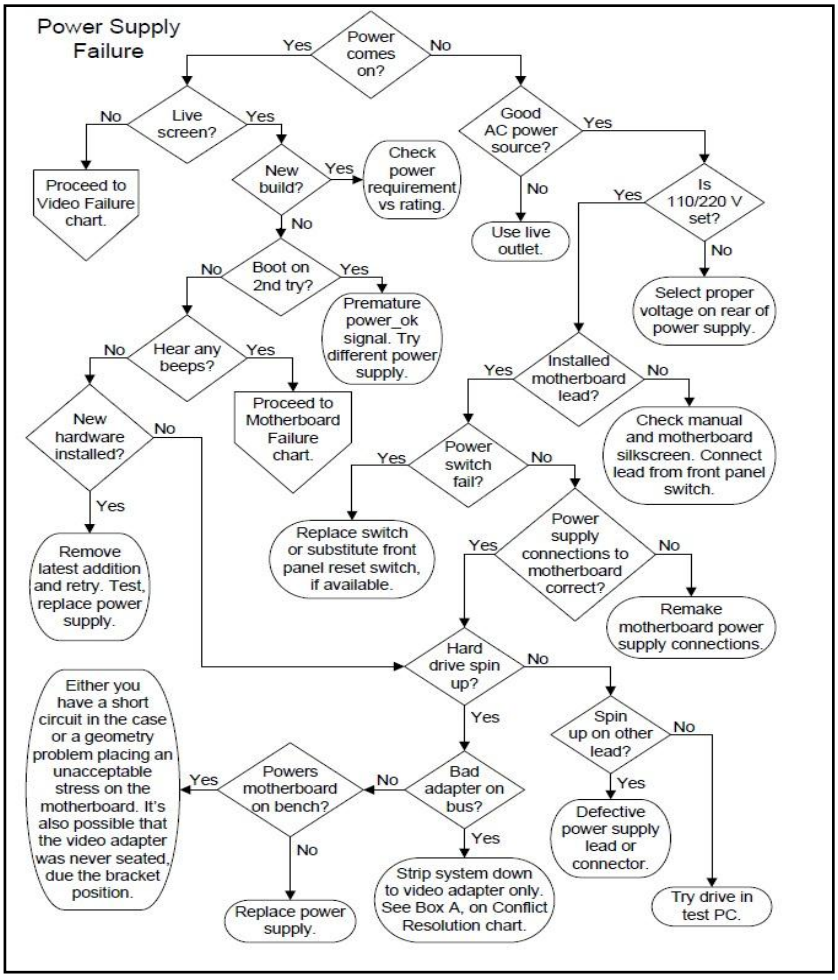
No	Kategori	Keterangan
1	Diagnosis	Menentukan dugaan/hipotesa berdasarkan gejala – gejala yang didapat dari pengamatan.
2	Desain	Menentukan konfigurasi komponen-komponen sistem berdasarkan kendala –kendala yang ada.
3	Debugging	Menentukan cara penyelesaian untuk mengatasi suatu kesalahan.
4	Interpretasi	Membuat deskripsi atau kesimpulan berdasarkan data yang didapat dari hasil pengamatan.
5	Instruksi	Pengajaran yang cerdas; menjawab pertanyaan mengapa, bagaimana, dan what-if sebagaimana yang dilakukan seorang guru.
6	Kontrol	Mengatur pengendalian suatu sistem (lingkungan).
7	Monitoring	Membandingkan hasil pengamatan dengan kondisi yang direncanakan.
8	Perencanaan	Pembuatan rencana untuk mencapai tujuan/sasaran yang telah ditetapkan.
9	Prediksi	Memperkirakan/memproyeksikan akibatyang terjadi dari suatu situasi tertentu.
10	Reparasi	Melakukan perbaikan atas kesalahan yang terjadi pada fungsi atau sistem.

3. Diagnosis Kerusakan Komputer

Menurut Kamus Besar Bahasa Indonesia, Diagnosis berarti penentuan jenis penyakit dengan cara meneliti (memeriksa) gejala-gejalanya. Dengan penyesuaian konteks, Diagnosis kerusakan komputer dapat diartikan sebagai penentuan jenis kerusakan komputer dengan memeriksa gejala – gejala kerusakanya. Morris Rosenthal dalam bukunya "*Computer Repair with Diagnostic Flowchart*" (2010) menjelaskan cara untuk melakukan diagnosis kerusakan komputer. Buku tersebut fokus pada diagnosis keruakan komputer jenis *ATX* dengan model *motherboard plug-n-play* dan *BIOS*. Komputer jenis

ATX tersrbut mulai dijual sejak pertengahan tahun 1990an dan masih menjadi PC (*Personal Computer*) standar yang dijual hingga 2008.

Morris Roshenthal (2010) menggambarkan proses diagnosis kerusakan komputer dalam bentuk *flowchart* (diagram alir). Berikut contoh potongan flowchart dalam buku "*Computer Repair with Diagnostic Flowchart*":



Gambar 2. Potongan Flowchart Diagnosis Kerusakan Komputer

Simbol belah ketupat menunjukan *decision point* yang menggambarkan suatu pertanyaan yang harus dijawab dengan "*ya*" atau "*tidak*". Simbol oval menunjukan jawaban atau solusi yang dapat berisi kesimpulan atau instruksi untuk memperbaikinya. Simbol berbentuk segi lima menujukan instruksi untuk

menjalakan diagnosis lainya karena hasil diagnosis menyimpulkan bahwa tidak terdapat kerusakan pada komponen / bagian yang bersangkutan tetapi ada kemungkinan kerusakan ada pada komponen / bagian lain.

Dalam buku "*Computer Repair with Diagnostic Flowchart*" dijelaskan 17 jenis diagnosis kerusakan komputer antara lain : (1)*power supply failure*; (2)*video failure*; (3)*video performance*; (4)*motherboard, ram, cpu failure*; (5)*motherboard, cpu, ram performance*; (6)*ata drive failure*; (7)*hard drive boot and performance*; (8)*cd or dvd playback*; (9)*recording problem dvd, cd, blu ray*; (10)*modem failure*; (11)*modem performance*; (12)*sound failure*; (13)*sound and game controller performance*; (14)*network hardware diagnostics*; (15)*peripheral failure*; (16)*scsi failure*; (17)*conflict resolution*.

4. Java

a. Pengertian Java

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Avestro, 2007).

b. Keunggulan Java

Bahasa pemrograman Java memiliki berbagai keunggulan. Berikut adalah beberapa keunggulan Java menurut white paper resmi dari SUN :

1. Sederhana

   Sintaks yang digunakan bahasa pomrograman Java mirip dengan C++ tetapi telah banyak perbaikan dan penghilangan penggunaan *pointer* dan *multiple inherintance* yang rumit. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

2. Berorientasi objek (*object oriented*)

   Java merupakan bahasa pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.

3. Robust

   Compiler pada Java mempunya kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. *Architecture Neutral*

   Java merupakan *platform independent* sehingga dapat dijalankan pada platform yang berbeda dengan *Java Virtual Machine*.

4. *Portable*

   Source code maupun aplikasi java dapat dengan mudah dipindahkan ke platform yang berbeda – beda tanpa harus di-*recompile* (kompilasi ulang).

5. *Multithreaded*

   Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

6. Dinamis

   Java didesain untuk dapat dijalankan pada lingkingan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* atau *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

5. SQLite

Menurut Allen dan Owens (2010), SQLite merupakan sebuah RDBMS(Relational Database Management System) yang bersifat *embedded* dan *opensource*. Sampai saat ini SQlite dikenal sebagai RDBMS dengan tingkat portabilitas yang tinggi, mudah digunakan, ringkas, dan reliabel.

Hal inilah yang mendorong penulis untuk memilih SQlite dalam membangun Aplikasi Sistem Pakar untuk Mendiagnosis Kerusakan Komputer.

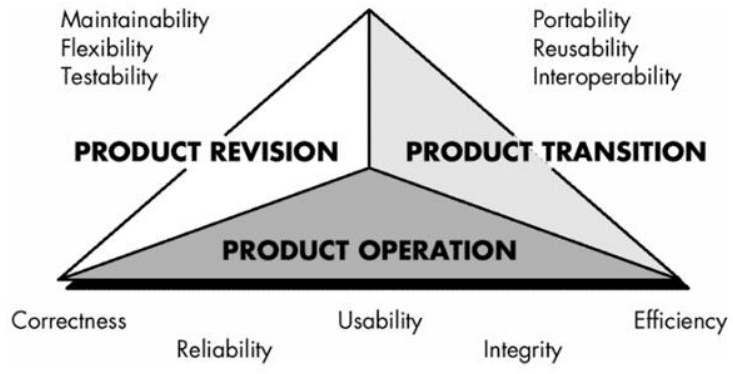
6. Kualitas Perangkat Lunak (Software Quality)

Agarwal, Tayal dan Gupta dalam bukunya (2010) menjelaskan bahwa kualitas perangkat lunak merupakan kesesuaian terhadap persyaratan fungsional dan kinerja secara eksplisit, standar pengembangan yang terdokumentasi secara eksplisit, dan karakteristik implisit yang diharapkan dari perangkat lunak yang dikembangkan secara profesional.

Teori – teori tentang kualitas perangkat lunak telah di kemukakan oleh beberapa ahli. Teori –teori tersebut antara lain :

a. McCall quality factors

McCall, Richards, dan Walter merumuskan serangkaian faktor – faktor yang menunjukan kualitas perangkat lunak. Faktor – faktor kualitas tersebut terkategorisasi menjadi tiga aspek penting dari sebuah perangkat lunak yaitu : karakteristik operasional, kemampuan untuk dalam menangani perubahan, dan kemampuan beradaptasi dengan lingkungan baru. (Pressman, 2010). Kategorisasi tersebut digambarkan dalam Gambar 2.



Gambar 3. Diagram Kategorisasi Faktor Kualitas Perangkat Lunak

Faktor – faktor kualitas yang menunjukan kualitas perangkat lunak tersebut antara lain :

1. *Correctness:* berkaitan dengan bagaimana program mampu memenuhi spesifikasi dan tujuan yang ingin dicapai oleh pengguna.
2. *Reliability* : berkaitan dengan bagaimana sebuah program mampu beroperasi dalam sebuah kondisi yang menuntut presisi tertentu.
3. *Usability* : berkaitan dengan usaha yang diperlukan pengguna untuk mengoperasikan, menyiapkan input, dan menginterpretasikan output dari program.
4. *Integerity* : berkaitan dengan tingkat kontrol terhadap program oleh pengguna, baik yang mendapatkan otorisasi atau tidak.
5. *Efficiency* : berkaitan dengan jumlah sumber daya komputer yang digunakan serta kode yang diperlukan di dalam program untuk menjalankan setiap fungsinya.
6. *Maintainability :* berkaitan dengan usaha yang diperlukan untuk menemukan dan mengatasi kesalahan di dalam program.
7. *Flexibility :* berkaitan dengan usaha yang diperlukan untuk mengubah program yang beroperasi.
8. *Testability :* berkaitan dengan usaha yang diperlukan untuk menguji sebuah program untuk memastikan bahwa program tersebut berfungsi sebagaimana mestinya.
9. *Portability :* berkaitan dengan usaha yang diperlukan untuk dapat mentransfer sebuah program dari sebuah lingkungan perangkat keras atau lunak tertentu ke lingkungan yang lain.

10. *Reusability :* berkaitan dengan bagaimana sebuah bagian program dapat digunakan kembali di dalam program lain.
11. *Interoperability :* berkaitan dengan usaha yang diperlukan untuk menghubungkan sebuah sistem dengan sistem yang lain.

b. ISO 9126 quality factors

International Standard Organization (ISO) mengembangkan Standar ISO 9126 yang mengidentifikasikan enam faktor kualitas yang menentukan kualitas suatu perangkat lunak (Pressman, 2010). Faktor – faktor kualitas tersebut antara lain :

1. *Functionality :* Kemampuan menutupi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan user. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *suitability, accuracy, compliance, security*.
2. *Reliability :* Kemampuan perangkat lunak untuk perawatan dengan level performansi. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *maturity, fault tolerance, recoverability.*
3. *Usability :* Kemampuan yang berhubungan dengan penggunaan perangkat lunak. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *understandability, learnability, operability*.
4. *Efficiency :* Kemampuan software memanfaatkan secara optimal resource yang digunakan, Faktor ini dijunjukan oleh beberapa sub faktor yaitu : *time behavior, resource behavior*.
5. *Maintainability :* Kemudahan suatu perangkat lunak untuk diperbaiki dikemudian hari. Faktor ini dijunjukan oleh beberapa sub faktor antara lain : *analyzability*, *changeability*, *stability*, dan *testability*.

6. *Portability :* Kemampuan yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *adaptability, installability, conformance, replaceability*.

7. Faktor Kualitas *Correctness*

Pressman (2010) menjelaskan bahawa correctness merupakan faktor kualitas yang menunjukan tingkat bagaimana perangkat lunak menjalankan fungsi yang dibutuhkanya. Fakor kualitas *correctness* dapat diukur dengan analisisis *defect per KLOC* (cacat / *error* pada setiap *KLOC/Kilo Line of Code*).

Dalam pengembangan perangkat lunak, terjadinya cacat / *error* adalah hal yang wajar, dan hampir semua developer perangkat lunak pasti menemuinya dalam pengembangan perangkat lunak baik itu yang sederhana sampai yang sangat komplesk. McConnell (2004, hal. 698) dalam bukunya menjelaskan bahwa jumlah *error* yang terjadi dalam pengembangan perangkat lunak, terutama yang kaitanya dengan penulisan code, dapat diperkirakan berdasarkan besar kecilnya *project* perangkat lunak yang sedang dikembangkan. Rentang kemungkinan error yang terjadi dalam suatu *project* digambarkan dalam tabel berikut :

Tabel 2. Perkiraan jumlah Error McConnell

Ukuran Project (*Line of Code / LOC*)	Perkiraan Jumlah *Error*
Lebih kecil dari 2K	0 – 25 *error* / *KLOC*
2K – 16K	0 – 40 *error* / *KLOC*
16K – 64K	0.5 – 50 *error* / *KLOC*
64K – 512K	2 – 70 *error* / *KLOC*
Lebih dari 512K	4 – 100 *error* / *KLOC*

Selain perkiraan jumlah error yang dikemukakan McConnell tersebut, Pressman dalam bukunya (2010, hal. 709) menjelaskan beberapa metode lain yang dapat digunakan untuk menentukan perkiraan jumlah error dalam sebuah project perangkat lunak.

Tabel 3. Metode Perkiraan error tiap KLOC

No	Metode	Rumus
1	Watson – Felix Model	$E = 5.2 \times (KLOC)^{0.91}$
2	Bailey-Basili Model	$E = 5.5 + 0.73 \times (KLOC)^{1.16}$
3	Boehm Simple Model	$E = 3.2 \times (KLOC)^{1.05}$
4	Doty Model(untuk KLOC>9)	$E = 5.28 \times (KLOC)^{1.047}$

McConnel (2004, hal. 564) juga menjelaskan bahwa kemungkinan error yang dapat ditemukan dalam sebuah project tergantung pada kualitas pengembangan perangkat lunak yang dilakukan. Semakin baik kualitas pengembangan perangkat lunak, semakin kecil ditemukan error dalam project tersebut. Berikut adalah beberapa rentang kemungkinan *error* tersebut :

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

Dalam kaitanya dengan pengembangan aplikasi sistem pakar Diagnosis Kerusakan Komputer yang dikembangkan dalam penelitian ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java*. Saat ini telah banyak aplikasi *IDE* (*Integrated Development Environment*) yang mendukung bahasa pemrograman *Java*. Netbeans merupakan salah satu IDE yang banyak

digunakan untuk pengembangan aplikasi *Java*. Fitur – fitur seperti *code assistance, code completion*, dan *error detection* sangat berguna dalam pengembangan aplikasi. Dengan fitur – fitur tersebut deteksi *error* secara langsung daat dilakukan saat penulisan *source code*, karena setiap ada *error* yang terjadi dalam penulisan *source code*, Netbeans akan menunjukan letak *error* yang terjadi dan memberikan penjelasanya. Dengan ini *developer* dapat langsung memperbaiki *error* dalam penulisan *source code* tersebut. Karena java merupakan bahasa pemrograman yang berbasis *compiler*, yaitu bahasa yang harus dicompile dulu sebelum dapat dijalankan, dapat dikatakan bahwa *source code* yang sudah jadi dan dapat dijalankan tidak mempunya *error* yang menyebabkan program tidak dapat dijalankan. Walaupun demikian, masih dimungkinkan kesahalan yang tidak dapat terdeteksi oleh IDE, seperti adanya penulisan kode yang tidak berguna (*unusable*) ataupun kurang efektif.

Basil Vandegriend (2009) dalam tulisanya menyebut *FindBugs* sebagai tools yang cocok digunakan developer Java untuk keperluan analisis kualitas *source code* Java. *FindBugs* merupakan *freeware tools* yang dikembangkan oleh *The University of Maryland*. Dalam website resmi *FindBugs* (2011) dijelaskan bahwa *FindBugs* mengkategorikan jenis bugs menjadi beberapa kategori yaitu : *bad practice*, *correctness, multithreaded correctness*, *experimental*, *internationalization*, *malicious code vulnerability*, *performance*, *security* , *dodgy code.* Dalam hal ini, terdapat dua buah kategori *bugs* yang berkaitan dengan analisis faktor kualitas *correctness* yaitu : ***correctness*** dan ***multithreaded correctness***.

Dalam *FindBugs* versi *2.0.0* yang dirilis pada Desember 2011 terdapat 140 bug pattern dalam kategori correctness, dan 65 *bug pattern* dalam kategori *multithreaded correctness*.

8. Faktor Kualitas *Functionality*

Functionality merupakan faktor kualitas yang menunjukan tingkat kemampuan menyediakan fungsi – fungsi yang diharapkan sehingga dapat memberikan kepuasan kepada pengguna (Pressman, 2010).

Faktor kualitas *functionality* dapat diuji dengan analisis fungsionalitas dari setiap komponen dari suatu perangkat lunak. Metode *black-box testing* merupakan metode yang cocok untuk melakukan pengujian fungsionalitas perangkat lunak. Dalam bukunya, Pressman (2010) menjelaskan bahwa *black-box testing*, atau juga disebut *behavioral testing*, fokus pada kebutuhan fungsional dari suatu perangkat lunak. Pengujian ini memungkinkan analis system memperoleh kumpulan kondisi input yg akan mengerjakan seluruh keperluan fungsional program.

James Bach dalam tulisanya "*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo Desktop Applications Edition*" (2005, hal. 4) membagi fungsi dalam sebuah perangkat lunak menjadi dua yaitu : *primary function* (fungsi primer) dan *contributing function* (fungsi pendukung). Fungsi primer merupakan fungsi yang utama dalam perangkat lunak, kesalahan dalam fungsi ini akan membuat perangkat lunak tidak layak. Sedangkan fungsi pendukung merupakan fungsi yang memberikan kontribusi pada perangkat lunak, tetapi bukan merupakan fungsi utama.

Dalam kaitanya dengan standar yang digunakan untuk menentukan apakah sebuah perangkat lunak lolos dalam pengujian faktor kualitas functionality. James Bach (2005) dalam tulisanya yang berjudul "*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo*" memberikan gambaran bagaimana suatu perangkat lunak dapat dikatakan memenuhi faktor kualitas *functionality* dalam program *Windows Logo Certification*. Berikut tabel kriteria

Tabel 4. Keirteria Lolos / Gagal pada program Windows Logo Certification

Kriteria Lolos	**Kriteria Gagal**
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

9. Faktor Kualitas *Portability*

Pressman (2010) menjelaskan bahwa faktor kualitas *portability* menggambarkan kemampuan perangkat lunak untuk dapat dipindah dan dijalankan di lingkungan yang berbeda, dalam kaitanya dengan penelitian ini adalah pada komputer dengan spesifikasi *hardware* maupun *operating system* yang bebeda – beda.

Bahasa pemrograman java merupakan bahasa dengan keunggulan pada aspek *portability* dan *architecture neutral.* Avestro (2007) menjelaskan bahwa aplikasi yang dibuat dengan *b*ahasa pemrograman java dapat berjalan di berbagai *platform* berbeda, bahkan tanpa perlu adanya proses *recompile*, karena Java bersifat independen dan tidak terikat dengan salah satu *platform*.

Untuk menjalankan aplikasi java pada komputer diperlukan *Java Runtime Environment* (JRE). *Java Runtime Environment* tersedia secara gratis berbagai sistem operasi : *Windows*, *Linux*, *Macintosh*, dan *Solaris.* Dalam kaitanya dengan *hardware requirement* (kebutuhan perangkat lunak), *Java Runtime Environment* tidak membutuhkan hardware dengan spesifikasi yang tinggi. Berdasarkan dokumen dalam website resminya, JRE hanya membutuhkan Processor dengan kecepatan 233 Mhz dan RAM sebesar 126 MB (pada Sistem Operasi Windows dan Macintosh) / 64 MB (Pada Sistem Operasi Linux dan Solaris). Secara umum dapat dikatakan bahwa semua komputer dengan sistem operasi yang didukung *Java Runtime Environment* dapat menjalankan aplikasi Java.

Pengujian faktor kualitas portability dapat dilakukan dengan mencoba aplikasi java pada pada *environtment* yang berbeda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda.

10. Faktor Kualitas *Usability*

Agarwal, Tayal, dan Gupta (2010) menjelaskan bahwa *usability* merupakan faktor kualitas perangkat lunak yang menunjukan kapabilitas untuk dapat dimengerti, dipahami dan digunakan oleh pengguna. Sementara itu, Anne Mette Jonassen Hass (2008) dalam bukunya yang berjudul "*Guide to Advanced Software Testing*" menjelaskan bahwa *usability* merupakan faktor kualitas yang menunjukan kecocokan perangkat lunak dengan penggunanya, dalam hal efektivitas, efisiensi, dan kepuasan pengguna.

Standar ISO 9126 mengkategorikan usability sebagai faktor kualitas nonfungsional. Usability berkaitan langsung dengan bagaimana sebuah perangkat lunak digunakan oleh pengguna. Standar ISO 9126 membagi faktor

kualitas usability menjadi beberapa subfaktor yaitu *understandability*, *learnability*, *operability* dan *attractiveness*. (Hass, 2008).

Understandability berkaitan dengan tingkat kesulitan pengguna dalam mengerti bagaimana menggunakan perangkat lunak dalam konsep logis. Learnability berkaitan dengan bagaimana pengguna dapat belejar dalam menggunakan suatu perangkat lunak. Operability berkaitan dengan bagaimana pengguna dapat menggunakan fungsi – fungsi dalam perangkay lunak. Sementara attractiveness berhubungan dengan bagaimana kemenarikan perangkat lunak sehingga pengguna mau menggunakannya (Hass, 2008).

Anne Mette Jonassen Hass menjelaskan (2008) menjelaskan bahwa faktor kualitas usability dapat diuji dengan metode survey atau kuesioner. Metode survey atau kuisioner digunakan untuk menganalisa faktor kualitas usability dari sisi subjektif pengguna. Pertanyaan – pertanyaan yang digunakan dalam kuisioner harus mencerminkan presepsi pengguna terhadap perangkat lunak yang dikembangkan. Pertanyaan – pertanyaan tersebut juga seharusnya mencakup pada sub faktor kualitas usability yaitu *understandability*, *learnability*, *operability* dan *attractiveness* (Hass, 2008, hal. 254).

B. Penelitian yang Relevan

1. Skripsi, Erni (2003), Pembuatan Sistem Pakar untuk Mendeteksi Kerusakan Komputer dengan Metode Backward Chaining.

Dalam penelitian tersebut, Sistem Pakar dibuat dengan bahasa pemrograman *Object Pascal* dengan *Borland Delphy* sebagai *IDE*. Knowledge base yang digunakan disimpan dalam file teks dan dalam mekanisme inferensinya, digunakan TP Lex dan Yacc.

2. Skripsi, Sendy Radiana (2010), Rancang Bangun Sistem Pakar Troubleshooting Kerusakan Hardware Komputer Berbasis Web.

Dalam penelitian tersebut, Sendy Radiana membuat Sistem Pakar untuk *Troubleshooting* Kerusakan komputer berbasis web menggunakan bahasa pemrograman *PHP* dan *MySql* sebagai database yang digunakan untuk menyimpan *knowledge base*.

C. Kerangka Berpikir

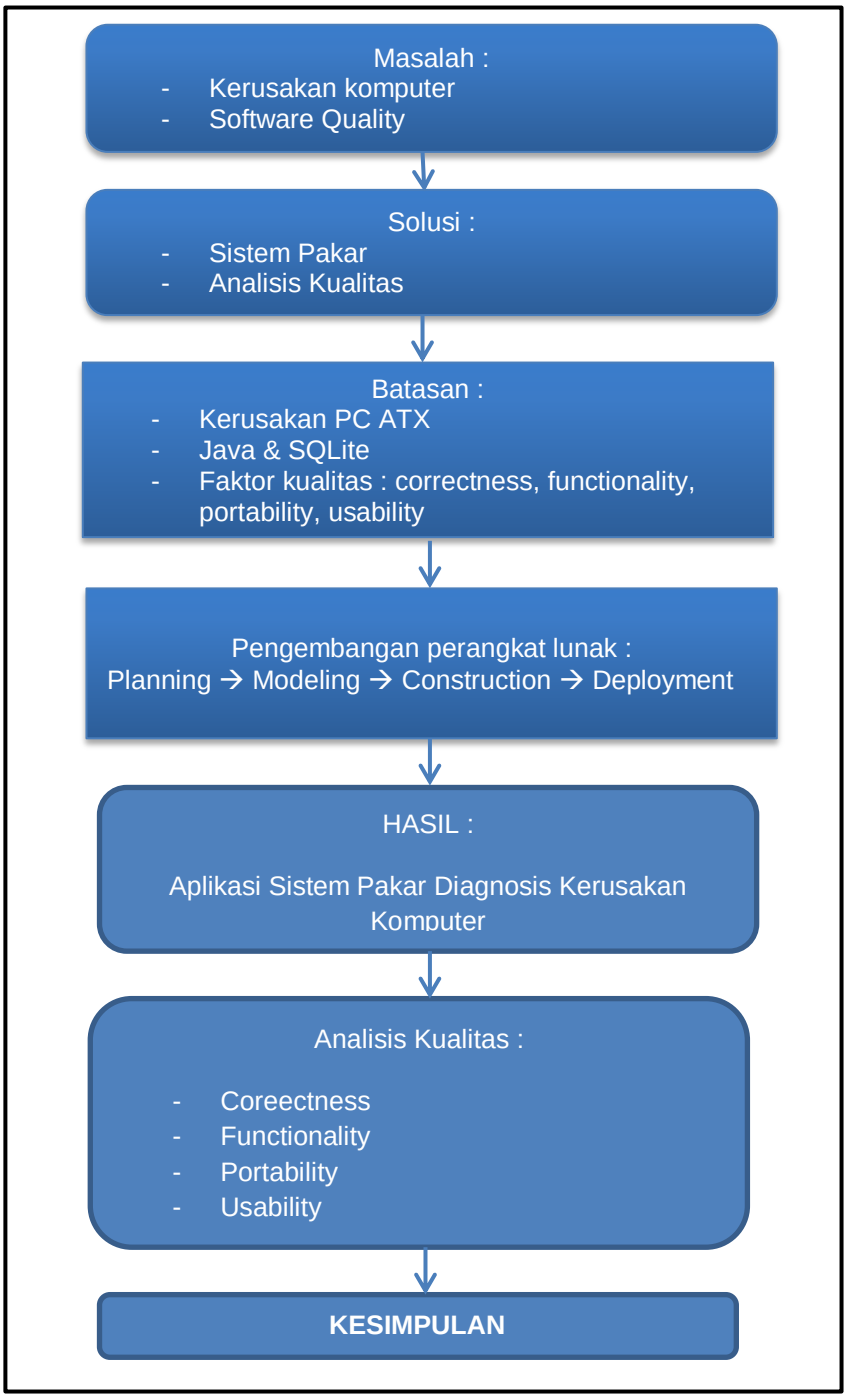
Kerusakan komputer merupakan masalah yang cukup merepotkan bagi sebagian besar pengguna komputer. Sistem Pakar dapat menjadi solusi dari permasalahan tersebut. Penelitian ini bermaksud untuk mengembangkan sebuah Aplikasi Sistem Pakar untuk Diagnosis Kerusakan Komputer. Dengan adanya Sistem Pakar, diharapkan pengguna komputer dapat memperbaiki sendiri kerusakan komputer atau setidaknya dapat mengetahui letak kerusakan komputer yang dialaminya.

Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer akan dibuat untuk dapat dijalankan di Komputer. Pada Pengembangan aplikasi tersebut, digunakan bahasa pemrograman Java dan database SQLite.

Sebagai sebuah produk dari sebuah pengembangan perangkat lunak, Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer diharapkan dapat memenuhi standar kualitas perangkat lunak (*Software Quality)*. Kualitas sebuah perangkat lunak dapat dinalisa dari beberapa faktor disebutkan dalam *McCall Quality Factors* ataupun *ISO-9126 Quality Factors*.

Pada penelitian ini digunakan empat faktor kualitas untuk analisis kualitas aplikasi sistem pakar yang dibuat, yaitu faktor kualitas *correctness, functionality, portability,*dan *usability*. Pengujian dilakukan dari sisi internal dan eksternal

perangkat lunak. Pengujian dari sisi internal perangkat lunak meliputi pengujian faktor kualitas *functionality*, *correctness*, dan *portability*. Sementara pengujian dari sisi ekternal meliputi pengujian faktor kualitas *usabilty* yang berkaitan langsung dengan pengguna aplikasi.



Gambar 4. Kerangka Berpikir

D. Pertanyaan Penelitian

Berdasarkan kerangka berpikir yang telah dijelasakan sebelumnya, Penulis merumuskan beberapa pertanyaan penelitian antara lain :

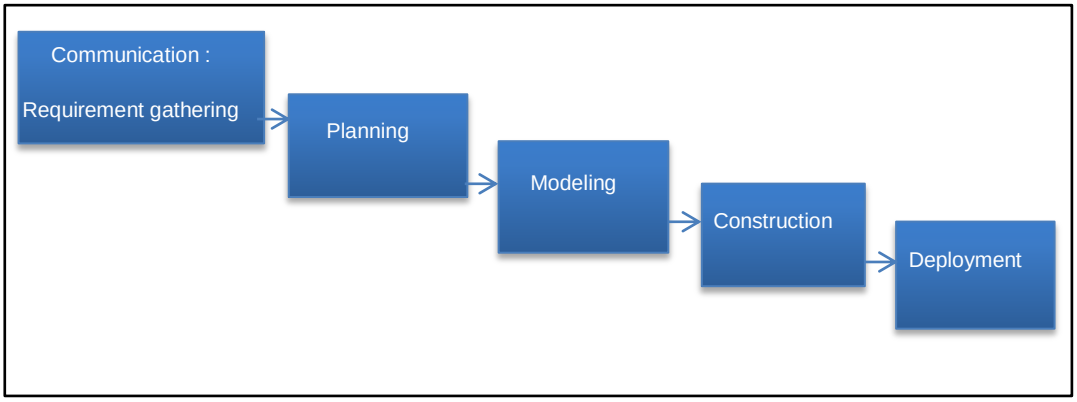
1. Apakah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *correctness*?
2. Apakah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *functionality*?
3. Apakah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *portability*?
4. Apakah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *usability*.

BAB III

METODE PENELITIAN

A. Model Penelitian

Jenis *penelitian* yang digunakan di dalam pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini adalah jenis penelitian riset dan pengembangan *(research and development)*. Penelitian ini bertujuan untuk mengembangkan suatu produk yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Pengembangan perangkat lunak dilakukan menggunakan kaidah *software engineering* (rekayasa perangkat lunak). Dalam teori software engineering terdapat beberapa macam model proses pengembangan perangkat lunak. Penelitan ini menggunakan model *waterfall*. Model waterfall atau yang sering disebut model *classic life cycle* menunjukan pengembangan perangkat lunak secara beruturan dan sistematis dimulai dari tahap analisis kebutuhan , perancanaan, perancangan, konstruksi dan penyebaran (Pressman, 2010).



Gambar 5. Model *Waterfall*

Setelah proses pengembangan aplikasi selesai, kemudian dilakukan analisis kualitas produk tersebut. Analisis kualitas pada penelitian ini difokuskan pada empat faktor kualitas perangkat lunak yaitu *functionality*, *correctness*, *portability*, dan *usability*. Pemilihan beberapa faktor kualitas tersebut didasari pada beberapa pertimbangan yaitu :

1. Faktor kualitas *correctness* dipilih dengan pertimbangan karena pengujianya menunjukan bagaimana kualitas source code aplikasi yang dikembangkan karena dalam pengujianya dilakukan analisis jumlah error pada tiap *KLOC* (*Kilo Lines of Code*).
2. Fakor kualitas functionality dipilih karena faktor kualitas ini menunjukan bagaimana aplikasi memenuhi fungsi – fungsi yang diharapkan dan memastikan bahwa fungsi – fungsi tersebut berjalan dengan baik.
3. Faktor kualitas *portability* dipilih karena pertimbangan bahwa aplikasi sistem pakar diharapkan dapat berjalan di komputer manapun. Pengujian faktor kualitas portability akan memastikan bahwa aplikasi dapat berjalan pada komputer dengan sistem yang berbeda – beda.
4. Faktor kualitas *usability* dipilih dengan pertimbangan bahwa aplikasi harus dapat digunakan oleh pengguna dengan mudah. Pengujian fakor kualitas usability akan menunjukan bagaimana tingkat aplikasi untuk dapat dimengerti, dipahami, dan digunakan oleh pengguna.

B. Variabel Penelitian

Berikut ini ada;ah variabel – variabel yang digunakan dalam penelitian ini :

1. Correctness

Correctness merupakan faktor kualitas yang menunjukan tingkat bagaimana perangkat lunak menjalankan fungsi yang dibutuhkanya.

2. Functionality

Functionality merupakan faktor kualitas yang menunjukan tingkat kemampuan menyediakan fungsi – fungsi yang diharapkan sehingga dapat memberikan kepuasan kepada pengguna.

3. Portability

Faktor kualitas *portability* menggambarkan kemampuan perangkat lunak untuk dapat dipindah dan dijalankan di lingkungan yang berbeda.

4. Usability

Usability merupakan faktor kualitas perangkat lunak yang menunjukan kapabilitas untuk dapat dimengerti, dipahami dan digunakan oleh pengguna

C. Desain Penelitian

1. Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak akan dilakukan dengan kaidah rekayasa perangkat lunak. Berikut tahap – tahap pengembangan perangkat lunak dalam penelitian ini :

a. Komunikasi (*communication*)

Tahap *communication* berisi proses *requrement gathering* (analisis kebutuhan). Pada tahap ini dilakukan analisis masalah yang akan diselesaikan

dengan pengembangan perangkat lunak dan juga fungsi apa saja yang harus dimiliki perangkat lunak yang dikembangkan.

b. Perencanaan (*planning*)

Pada tahap ini dilakukan perencanaan untuk memberikan gambaran umum tentang produk yang akan dikembangkan dan proses – proses yang akan dilakukan dalam proses pengembangan perangkat lunak.

c. Perancangan (*modeling*)

Tahap perancangan dibagi menjadi tiga bagian yaitu :

1. Perancangan Knowledge Base dan Database
2. Perancangan UML (Unified Modeling Language)
3. Perancangan Antar Muka (Graphical User Interface)

d. Konstruksi (*construction*)

Tahap ini merupakan impelentasi dari rancangan yang telah dibuat. Secara umum pada tahap ini dilakukan entry data knowledge base pada database *SQLite* dan pengkodean (*coding*) menggunakan bahasa pemrograman *Java*.

e. Penyebaran (*deployment*)

Produk perangkat lunak yang telah dibuat kemudian dapat disebarkan kepada pengguna.

2. Analisis Kualitas Perangkat Lunak

1. Faktor kualitas *correctness* diuji dengan analisis *error per kilo line of codes* (*KLOC*), yang akan dibandingkan dengan standard *error per kilo line of codes* pada *industry average* dan *Microsoft Application*.

2. Faktor kualitas *functionality* diuji dengan pengujian pada setiap fungsi pada aplikasi yang dibuat. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi pada aplikasi berkerja sebagai mana mestinya.
3. Faktor kualitas *portability* dianalisis dengan melakukan pengujian aplikasi pada beberapa *environtment* yang berbeda, dalam hal ini komputer dengan *sistem operasi* yang berbeda – beda.
4. Faktor kualitas *usability* dikaji dari penilaian pengguna akhir (*end user*) yang didapat melalui kuisioner. Kuisioner yang digunakan mengacu pada *Computer System Usability Questionnaire* yang dipublikasi oleh J.R. Lewis.

D. Teknik Pengumpulan Data

1. Studi Pustaka

Dalam penelitian ini studi pustaka dilakukan penulis pada tahap pengembangan perangkat lunak yaitu pada proses perancangan knowledge-base dengan mempelajari pustaka yang berkaitan yaitu buku "*Computer Repair with Diagnostic Flowchart*" karangan Morris Rosenthal.

2. Observasi

Teknik observasi dalam penelitian ini dalam pengambilan data yang berkaitan dengan analisis faktor kualitas *functionality*, *correctness*, dan *portability*.

3. Kuesioner

Dalam penelitian ini teknik pengumpulan data menggunakan kuisioner dilakukan untuk analisa faktor kualitas *usabilty* pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.

E. Subjek Penelitian

Subjek dalam penelitian adalah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer.

F. Instrumen Penelitian

Dalam penelitian ini dibutuhkan beberapa instrument yang digunakan mulai dari proses pengembangan perangkat lunak dan proses analisis kualitas perangkat lunak. Instrumen yang digunakan pada penelitian ini antara lain :

1. Test case

Dalam pengujian faktor kualitas functionality dengan metode black-box testing, dibutuhkan test case. Argawal, Tayal dan Gupta (2010) menjelaskan bahwa test case merupakan seperangkat instuksi yang disesain untuk mengetahui kesalahan yang dalam perangkat lunak.

Untuk memudahkan dalam melakukan pengujian, seperangkat *test case* perlu didokumentasikan dengan baik, dan sebaiknya dalam format yang sama. Dalam penelitian ini, test case digunakan penulis dalam pengujian faktor kualitas *functionality* dan *portability*. Berikut adalah format test case digunakan dalam penelitian ini :

Tabel 5. Format *test case* yang digunakan dalam pengujian

Test case id	Nama yang unik untuk idektifikasi test case
Purpose	Tujuan dari test case
Assumptions	Syarat kondisi awal yang harus terpenuhi sebelum test dapat dijalankan.
Test data	Variabel atau kondisi yang akan di test.
Steps	Langkah – langkah yang dijalankan.
Expected result:	Hasil yang seharusnya didapatkan (yang menunjukkan bahwa tidak ada kesalahan dalam perangkat lunak)
Actual result:	Hasil yang didapat dalam pengujian.
Pass/Fail:	Keterangan : Lolos atau Gagal.

2. Lines of Code Counter

Tools ini digunakan untuk menghitung *lines of code* dari *source code* aplikasi yang dikembangkan. Tools ini dikembangkan oleh John Roshi yang terdafaftar di repository *Java.Net*.

3. FindBugs

FindBugs merupakan *freeware tools* yang dikembangkan oleh *The University of Maryland*. Tools ini digunakan untuk menganalisa error yang ditemukan dalam source code aplikasi yang dikembangkan. *FindBugs* mengkategorikan jenis bugs menjadi beberapa kategori yaitu : *bad practice*, *correctness, multithreaded correctness*, *experimental*, *internationalization*, *malicious code vulnerability*, *performance*, *security* , *dodgy code.* Dalam penelitian ini akan digunakan dua kategori yaitu kategori *bugs* yang berkaitan dengan analisis faktor kualitas *correctness* : *correctness* dan *multithreaded correctness*.

4. Kuesioner

Kuesioner atau angket merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya (Sugiyono, Metode Penelitian Pendidikan, 2009).

Dalam penelitian ini digunakan kuesioner untuk pengujian faktor kualitas *usability* pada aplikasi yang dikembangkan. Kuesioner yang digunakan mengacu pada *Computer System Usability Questionnaire* yang dipublikasi oleh J.R. Lewis. Kuesioner tersebut kemudian dibagikan kepada responden.

Tabel 6. Computer System Usability Questionnaire

No	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Secara keseluruhan, saya merasa puas dengan kemudahan penggunaan sistem ini.	…	…	…	…	…
2.	Cara penggunaan sistem ini sangat simpel.	…	…	…	…	…
3.	Saya dapat menyelesaikan tugas saya dengan efektif ketika menggunakan sistem ini.	…	…	…	…	…
4.	Saya dapat dengan cepat menyelesaikan pekerjaan saya menggunakan sistem ini.	…	…	…	…	…
5.	Saya dapat menyelesaikan tugas saya dengan efisien ketika menggunakan sistem ini.	…	…	…	…	…
6.	Saya merasa nyaman menggunakan sistem ini.	…	…	…	…	…
7.	Sistem ini sangat mudah dipelajari.	…	…	…	…	…
8.	Saya yakin saya akan lebih produktif ketika menggunakan sistem ini.	…	…	…	…	…
9.	Jika terjadi error, sistem ini memberikan pesan pemberitahuan tentang langkah yang saya lakukan untuk mengatasi masalah.	…	…	…	…	…
10.	Kapanpun saya melakukan kesalahan, saya bisa kembali dan pulih dengan cepat.	…	…	…	…	…
11.	Informasi yang disediakan sistem ini sangat jelas.	…	…	…	…	…
12.	Mudah untuk menemukan informasi yang saya butuhkan.	…	…	…	…	…
13.	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	…	…	…	…	…
14.	Informasi yang diberikan sangat efektif dalam membantu menyelesaikan pekerjaan saya.	…	…	…	…	…
15.	Tata letak informasi yang terdapat di layar monitor sangat jelas.	…	…	…	…	…
16.	Tampilan sistem ini sangat memudahkan.	…	…	…	…	…
17.	Saya suka menggunakan tampilan sistem semacam ini.	…	…	…	…	…
18.	Sistem ini memberikan semua fungsi dan kapabilitas yang saya perlukan.	…	…	…	…	…
19.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini.	…	…	…	…	…

Keterangan :

SS : Sangat Setuju ST : Setuju RG : Ragu – ragu

TS : Tidak setuju STS : Sangat tidak setuju

G. Teknik Analisis Data

1. Analisis Faktor Kualitas *Functionality*

Pengujian faktor kaulitas functionality dilakukan dengan melakukan tes pada setiap fungsi perangkat lunak. Tes yang dilakukan didokumentasikan dalam *test case*. Setiap test case menggambarkan apakah suatu fungsi berjalan sebagaimana mestinya atau tidak.

Berkaitan dengan standar yang digunakan dalam menentukan apakah perangkat lunak telah memenuhi syarat faktor kualitas functionality, penulis menggunakan standar functioality yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo*.

Tabel 7. Standar kriteria faktor kualitas *functionality* dalam *Microsoft Certification Logo* (Bach, 2005)

Kriteria Lolos	**Kriteria Gagal**
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

2. Analisis Faktor Kualitas *Correctness*

Faktor kualitas correctness dianalisa dengan menghitung jumlah *error* tiap *kilo lines of code (KLOC)*. Jumlah *lines of code* dapat dihitung menggunakan *Lines of Code Counter, tools* yang dikembangkan oleh John Roshi yang terdafaftar di repository *Java.Net.* Sedangkan jumlah error dalam suatu

perangkat lunak, dalam hal ini aplikasi java, dapat dihitung dengan FindBugs, aplikasi yang dikembangkan oleh *The University of Maryland*. Jumlah *error / KLOC* yang didapatkan dalam pengujian kemudian dibandingkan dengan standar *error / KLOC* pada *industry average* dan standar *Microsoft Application*.

3. Analisis Faktor Kualitas *Portability*

Pengujian faktor kualitas *portability* pada penelitian difokuskan untuk menjawab pertanyaan apakah perangkat lunak yang dikembangkan dapat berjalan sebagaimana mestinya pada sistem yang berbeda – beda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda.

Berdasarkan acuan bahwa bahasa pemrograman java merupakan bahasa pemrograman dengan tingkat portability yang baik. Penulis menyusun standar pada perangkat lunak yang dikembangkan untuk mementukan apakah perangkat lunak yang dikembangkan lolos atau gagal dalam pengujian faktor kualitas portability.

Tabel 8. Kriteria lolos / gagal pengujian faktor kualitas *portability*

Kriteria Lolos	**Kriteria Gagal**
Perangkat lunak dapat berjalan sebagaimana mestinya pada setiap sistem yang diujikan.	Paling tidak ada satu sistem dimana perangkat lunak tidak dapat berjalan sebagaimana mestinya.

4. Analisis Faktor Kualitas *Usability*

Pengujian faktor kualitas usability dilakukan dengan menggunakan metode kuesioner. Kuesioner akan dibagikan kepada siswa SMK Muhammadiyah 2 Yogyakarta sebagai lokasi penelitian faktor kualitas *usability*. Pengambilan sampel dilakukan dengan teknik *purposive sampling*. Teknik porposive sampling

yaitu teknik penentuan sampel dengan pertimbangan tententu (Sugiyono, 2010). Dalam penelitian ini diambil siswa Kelas XI jurusan Teknik Komputer Jaringan (TKJ). Hal tersebut didasari pertimbangan bahwa aplikasi yang diuji hubunganya dengan diagnosis kerusakan komputer. Pemilihan Kelas XI didasari alasan bahwa siswa kelas XI dianggap sudah paham dasar – dasar komponen komputer sehingga sesuai dengan sasaran pengguna aplikasi yang dikembangkan, siswa kelas X dianggap belum memiliki dasar komponen – komponen komputer, dan siswa kelas XII tidak dipilih karena saat penelitian sudah kelulusan. Jumlah siswa Kelas XI TKJ yang aktif mengikuti kegiatan belajar mengajar sebanyak 55 siswa yang terdiri dari 27 siswa Kelas XI TKJ-1 dan 28 siswa Kelas XI TKJ-28 siswa Kelas XI TKJ-2.

Data yang dihasilkan dari kuesioner tersebut merupakan gambaran pendapat atau presepsi pengguna perangkat lunak, dalam hal ini yang berkaitan dengan faktor kualitas *usability* perangkat lunak yang dikembangkan. Data yang dihasilkan dari kuisioner meruapakan data yang bersifat kuantitatif. Data tersebut dapat dikonversi ke dalam data kualitatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Menurut Sugiyono (2009), Skala Likert digunakan untuk mengukur sikap, pendapat atau presepsi seseorang atau kelompok terhadap sesuatu, dalam hal pendapat pengguna terhadap perangkat lunak yang dikembangkan. Data hasil kuesioner yang berupa jawaban – jawaban pengguna dari setiap item dalam kuesioner mempunyai gradasi nilai dari sangat positif sampai sangat negatif. Dalam kaitanya dengan kuesioner yang digunakan yaitu, *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis, terdapat 5

macam jawaban dalam setiap item keusioner. Data tersebut diberi skor sebagai berikut :

Tabel 9. Konversi jawaban item kuesioner ke dalam nilai kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Skor yang didapatkan pada tiap hasil kuesioner tersebut kemudian diambil nilai rata - rata. Nilai rata – rata tersebut kemudian dijumlahkan. Dengan jumlah responden sebanyak 55 orang maka dapat dihitung nilai tertinggi dan nilai terendah sebagai berikut :

1. Nilai tertinggi = 55 x 19 x 5 = 5225 , dengan asumsi semua responden memberi jawaban sangat setuju pada setiap item kuesioner.
2. Nilai terendah = 55 x 19 x 1 = 1045, dengan asumsi semua responden memberi jawaban “sangat tidak setuju” pada setiap item kuesioner.

Dari data tersbut data tersebut, kemudian dapat disusun kategori penilaian kuesioner berdasarkan perhitungan interval kelas.

1) Menghitung Jumlah Kelas

K = 1 + 3,3 log n

= 1 + (3,3 x log 55)

= 1 + 4,88 = 5,88 ≈ 5

(dibulatkan menjadi 5 agar jumlah kelas sama dengan jumlah pilihan jawaban pada kuesioner)

2) Menghitung Rentang Data

Rentang Data = (Data terbesar – Data terkecil) + 1

= 5225 – 1045

= 4180

3) Menhitung Panjang Kelas

Panjang Kelas = Rentang Data / Jumlah Kelas

= 4180 / 5

= 836

Dengan data tersebut, kemudian disusun kategorisasi penilain faktor kualitas *usability* bedasarkan interval nilai kuesioner.

Tabel 10 Kategori Penilaian Faktor Kualitas *Usability*

Interval Nilai	Ketegori
1045 - 1877	Sangat Tidak Baik
1878- 2714	Tidak Baik
2715- 3551	Cukup
3552- 4388	Baik
4389- 5225	Sangat Baik

Jumlah nilai yang didapat dari hasil konversi jawaban kuisioner ke dalam nilai kuantitatif kemudian dibandingkan dengan kategorisasi penilaian tersebut.

BAB IV

HASIL DAN PEMBAHASAN

A. Pengembangan Perangkat Lunak

Salah satu tujuan dari penelitian ini adalah untuk mengembangkan produk perangkat lunak yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Proses pengembangan perangkat lunak dalam penelitian ini didasarkan pada kaidah rekayasa perangkat lunak (*software engineering*). Proses pengembangan perangkat lunak dalam penelitian ini dibagi menjadi beberapa bagian yaitu : perencanaan (*planning*), perancangan (*modeling*), konstruksi (*construction*), dan penyebaran (*deployment*).

1. Komunikasi (*Communication*)

Tahap *communication* berisi proses *requrement gathering* (analisis kebutuhan). Pada tahap ini dilakukan analisis masalah yang akan diselesaikan dengan pengembangan perangkat lunak dan juga fungsi apa saja yang harus dimiliki perangkat lunak yang dikembangkan. Beriku ini adalah fungsi – fungsi yang diharapkan pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan :

1. Setiap pengguna dapat langsung menggunakan aplikasi untuk menjalankan proses diagnosis.
2. Pengguna dapat memilih jenis – jenis diagnosis kerusakan komputer yang tersedia.
3. Terdapat bagian untuk melakukan pengeditan *knowledge-base* yang hanya dapat diakses oleh pengguna tertentu (admin).

2. Perencanaan (*Planning*)

Produk yang dikembangkan dalam penelitian ini adalah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Aplikasi ini akan dibuat dengan bahasa pemrograman Java dan dirancang untuk dijalankan pada komputer dengan sistem operasi yang dapat mendukung *Java Runtime Environment*.

Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan menpunya dua jenis mode berdasarkan user yang menggunakan, antara lain :

a. Normal Mode

Normal mode adalah mode awal ketika aplikasi dijalankan, dan tidak memerlukan login untuk menggunakanya.

b. Admin Mode

Mode ini adalah mode administrator yang membutuhkan login untuk dapat menggunakanya. Mode admin dibagi menjadi dua macam :

1) *Basic Admin Mode*

Pada mode ini terdapat *live mode* (simulasi normal mode) ditambah dengan beberapa menu administrative seperti untuk mengedit data, menambah data, menghapus data, menambah deteksi dan menghapus deteksi. Semua menu yang ada pada mode ini dalam bentuk *wizard* yang menuntun administrator untuk melakukan pengiditan pada database dengan tidak merusak rule – rule knowledge base yang tersimpan dalam database.

2) *Advance Admin Mode*

Pada mode ini, administrator diberikan menu – menu administratif dasar tanpa dilengkapi dengan wizard aaupun guide. Sehingga hanya administrator yang benar – benar memahami struktur knowledge base dalam database ini yang dapat menggunakan mode ini.

Secara umum spesifikasi produk yang dikembangkan digambarkan dalam Tabel berikut :

Tabel 11. Spesisikasi Perangkat Lunak

Nama	Diagnosis Kerusakan Komputer
Jenis	Sistem Pakar (*Expert System*)
Fungsi	Membantu pengguna komputer mengetahui dan memperbaiki kerusakan komputer secara mandiri.
Karakteristik pengguna	Pengguna komputer yang mengetahui secara umum tentang *hardware* komputer.
Bahasa Pemrograman	*Java*
Database	*SQLite*
Sistem Operasi	Windows, Mac OS, Linux, Solaris yang terinstall *Java Runtime Environment* (JRE)
Pengembang	Kifni Taufik Darmawan
URL	http://kifni.com/portofolio/sistem-pakar-diagnosis-kerusakan-komputer

3. Perancangan (*Modeling*)

Proses perancangan (modeling) aplikasi Sistem Pakar Diagnosis Komputer meliputi perancangan *knowledge base*, *database*, *UML* (*Unified Modeling Language*), dan *user inteface* (antar muka).

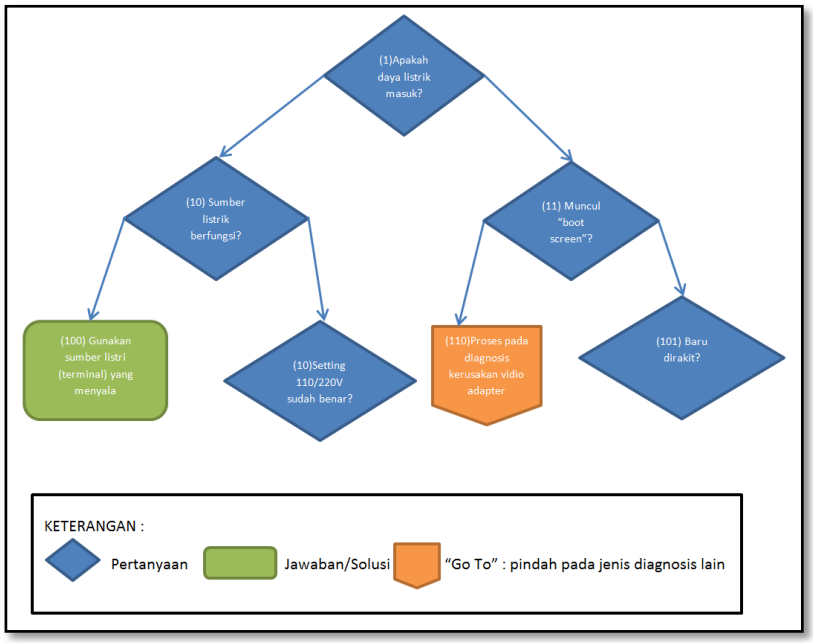
a. Perancangan *knowledge base* dan database

Knowlegde-base (Basis Pengatahuan) yang digunakan dalam Aplikasi Sistem Pakar Pendiagnosis Kerusakan Komputer berdasar pada Buku karangan Mossis Rosenthal yang berjudul *Computer Failure with Diagnostic Flowchart*. Dalam bukunya tersebut Morris Rosenthal (2010) menjelaskan 17 macam diagnosis kerusakan komputer. Secara umum isi dalam buku tersebut lebih difokuskan untuk diagnosis pada PC (*Personal Computer*) dengan *form factor* ATX (jenis umum pada mayoritas PC pertengahan 1990 sampai sekarang) (Rosenthal, 2010). Karena pertimbangan keterbatasan waktu, Penulis hanya

menggunakan 7 jenis diagnosis, tetapi di kemudian hari knowledge-base masih dapat dikembangkan dengan mudah karena struktur knowledge base terpisah dari source code. 7 jenis diagnosis tersebut antara lain :

1) Kerusakan Power Supply
2) Kerusakan Vidio Adapter (VGA)
3) Masalah Kinerja Vidio Adapter (VGA)
4) Kerusakan Motherboard, Processor, dan RAM
5) Masalah Kinerja Motherboard, Processor, dan RAM
6) Kerusakan ATA / SATA Hard Drive
7) Masalah Booting & Kinerja Hard Disk

Bentuk *knowledge base* yang digunakan pada aplikasi Sistem Pakar Diagnosis Kerusakan Komputer didasarkan pada *flowchart* yang menggambarkan tiap – tiap proses diagnosis. *Flowchart* tersebut berisi pertanyaan yang diberikan pada user, dan jawaban (solusi) yang akan diberikan kepada user sebagai hasil dari proses diagnosis. Berikut adalah gambar struktur *knowledge-base* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



Gambar 6. Struktur Knowledge Base

Mekanisme inferensi yang digunakan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer untuk menyajikan knowledge-base pada pengguna adalah teknik *forward chaining*. Proses inferensi dimulai dengan menampilkan pertanyaan – pertanyaan yang menggambarkan gejala – gejala (fakta – fakta) mulai dari yang yang bersifat umum ke khusus sampai ditemukan jawaban (*solusi*).

Untuk menghubungkan teknik inferensi dengan *knowledge base,* setiap data pada *knowledge-base* dikodekan untuk memudahkan dalam menampilkanya pada aplikasi. Data pertama pada suatu tabel diagnosis dimulai dengan kode "**1**", kemudian kode pada data sesudahnya ditambah karkter "**0**" atau "**1**", karakter "**0**" menujukan bahwa pertanyan sebelumnya dijawab "*TIDAK*" dan karakter "**1**" menunjukan bahwa pertanyaan sebelumnya dijawab "***YA***". Selain itu terdapat juga data berupa detail dari pertanyaan atau jawaban yang

berfungsi untuk memberi penjelasan lebih lanjut tentang pertanyaan ataupun jawaban yang ditampilkan oleh aplikasi.

Dalam aplikasi sistem pakar ini, terdapat lebih dari satu macam jenis diagnosis. Tiap diagnosis dan data – datanya akan disimpan dalam tabel. Untuk itu dalam desain database yang akan digunakan terdapat dua jenis tabel yaitu :

1) *table index (table_index).*

Dalam tabel ini disimpan data tentang jenis diagnosis yang terdapat pada sistem pakar. Field *table_code* menunjukan nama tabel dari suatu diagnosis. Sedangkan Field table_title menunjukan nama (judul) suatu diagnosis.

Tabel 12. Struktur Tabel Index

Field	Type	Null	Key	Default
table_code	CHAR	No	UNIQUE	-
table_title	VARCHAR	No	-	-

2) *tabel diagnosis.*

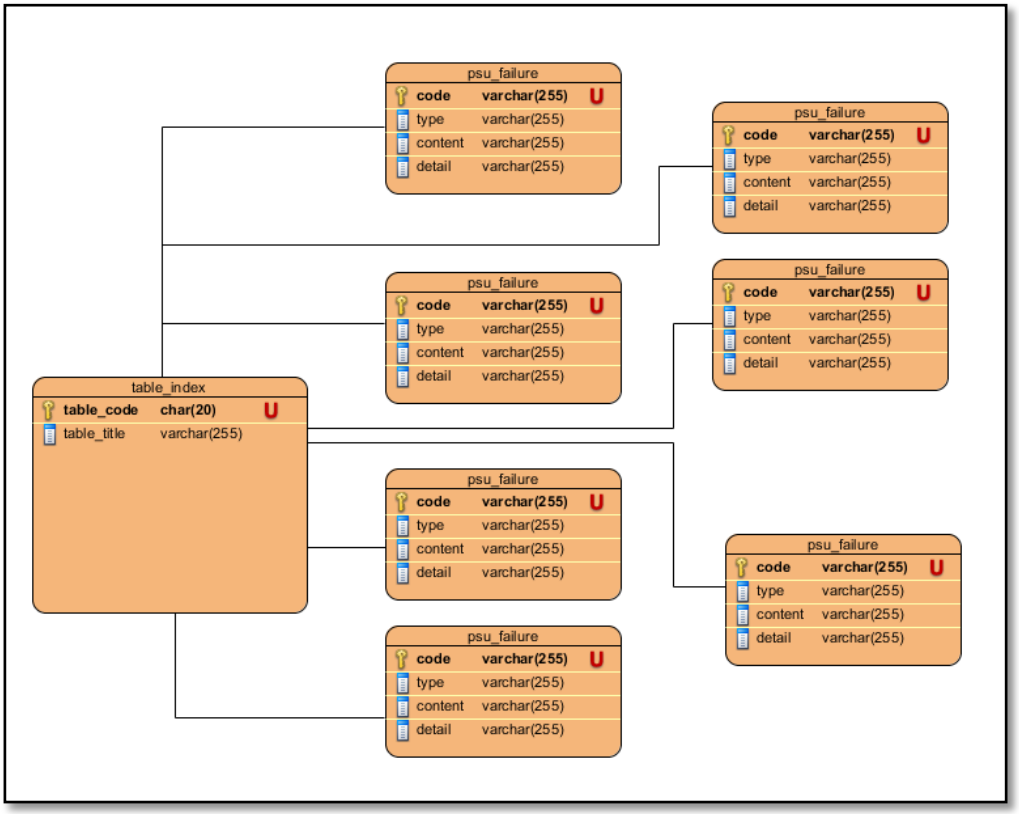
Tabel jenis ini, adalah tabel dimana data – data (pertanyaan, jawaban, dan go_to) dalam suatu diagnosis disimpan. Data berupa pertanyaan disimbolkan dengan karakter "**Q**", data berupa jawaban disimbolkan dengan karakter "**A**", dan data berupa go_to disimbolkan dengan "**G.[table_code]**" (contoh : *G.psu_failure*).

Satu diagnosis menggunakan satu buah tabel. Jadi jumlah tabel jenis ini adalah sama dengan jumlah diagnosis yang tersedia dalam sistem.

Tabel 13. Struktur Tabel Diagnosis

Field	Type	Null	Key	Default
code	VARCHAR	No	UNIQUE	-
type	VARCHAR	No	-	-
content	TEXT	No	-	-
detail	TEXT	No	-	-

Berikut adalah gambar hubungan tabel_index dan tabel diagnosis dalam database aplikasi yang dikembangkan :



Gambar 7. Struktur database

b. Perancangan UML (unified modeling language)

Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar dalam industri untuk visualisasi dan dokumentasi sistem perangkat lunak (software). UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti, 2006). Rancangan sebuah perangkat lunak didefinisikan dalam diagram – diagram dalam UML yaitu : *use case diagram* , *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram*.

1) *use case diagram.*

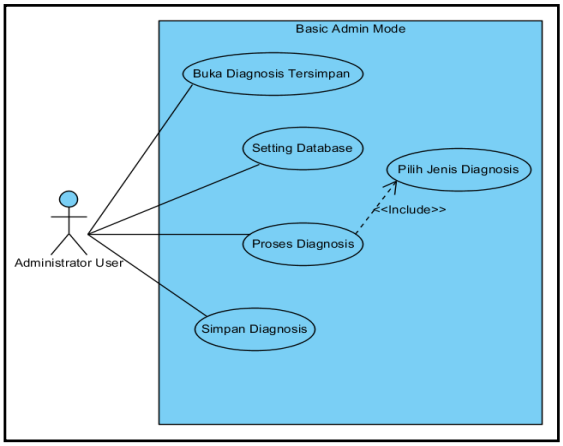
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem (Dharwiyanti, 2006).

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya.. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dibagi menjadi tiga macam yaitu pada *Normal Mode*, *Basic Admin Mode*, dan *Advanced Admin Mode*. Berikut ini merupakan use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :

a) use case : normal mode.

Berikut ini adalah gambar use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Normal Mode :



Gambar 8. Use Case : Normal Mode

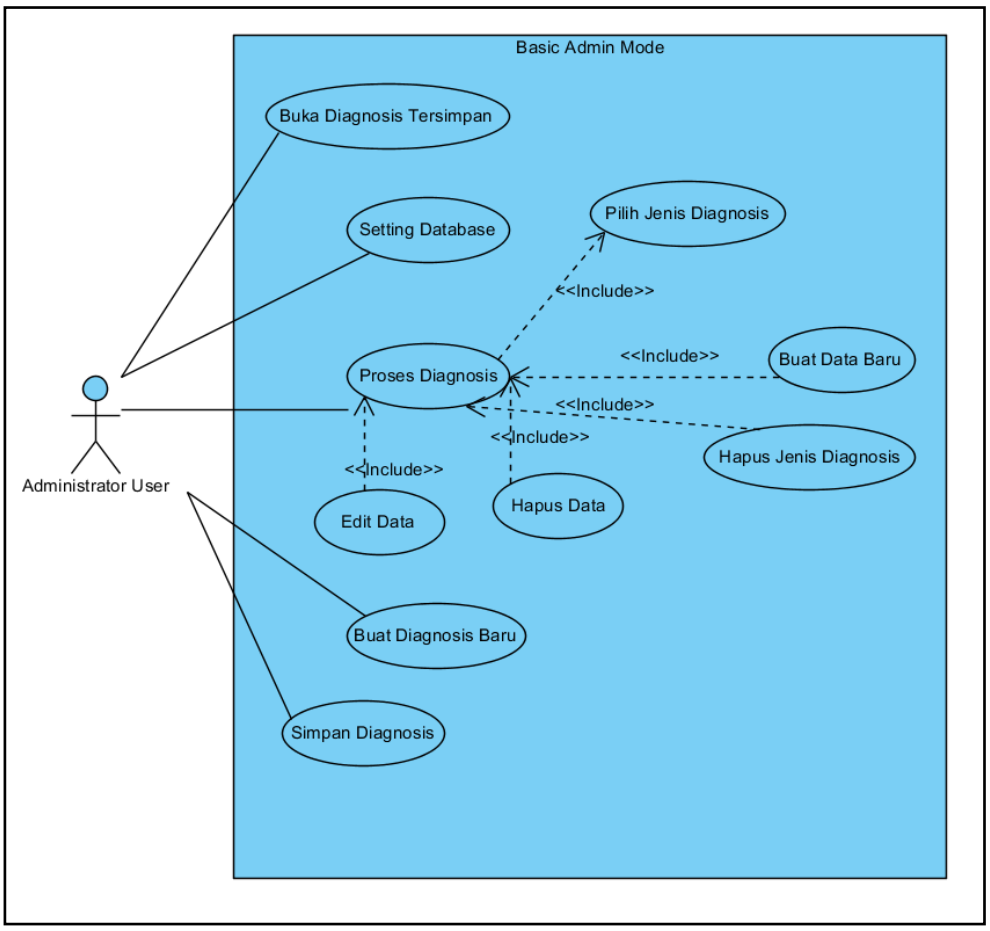
Berikut adalah definisi dari tiap proses pada user case Normal Mode yang dijelaskan dalam bentuk tabel :

Tabel 14. Definisi *Use Case* : *Normal Mode*

No	Use Case	Deskripsi
1	Buka Diagnosis Tersimpan	Proses untuk membuka proses diagnosis yang telah disimpan sebelumnya.
2	Setting Database	Proses untuk melakukan poengaturan database (mengubah file database yang digunakan).
3	Pilih Jenis Diagnosis	Proses untuk memilih salah satu jenis diagnosis yang akan dijalankan.
4	Proses Diagnosis	Proses dimana user (pengguna) mengikuti jalanya proses diagnosis dengan menjawab pertanyaan yang diajukan oleh sistem sampai ditemukan solusi.
5	Simpan Diagnosis	Proses untuk menyimpan proses diagnosis yang sedang berlangsung.

b) use case : basic admin mode.

Berikut ini adalah gambar use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Basic Admin Mode :



Gambar 9. Use Case : Basic Admin Mode

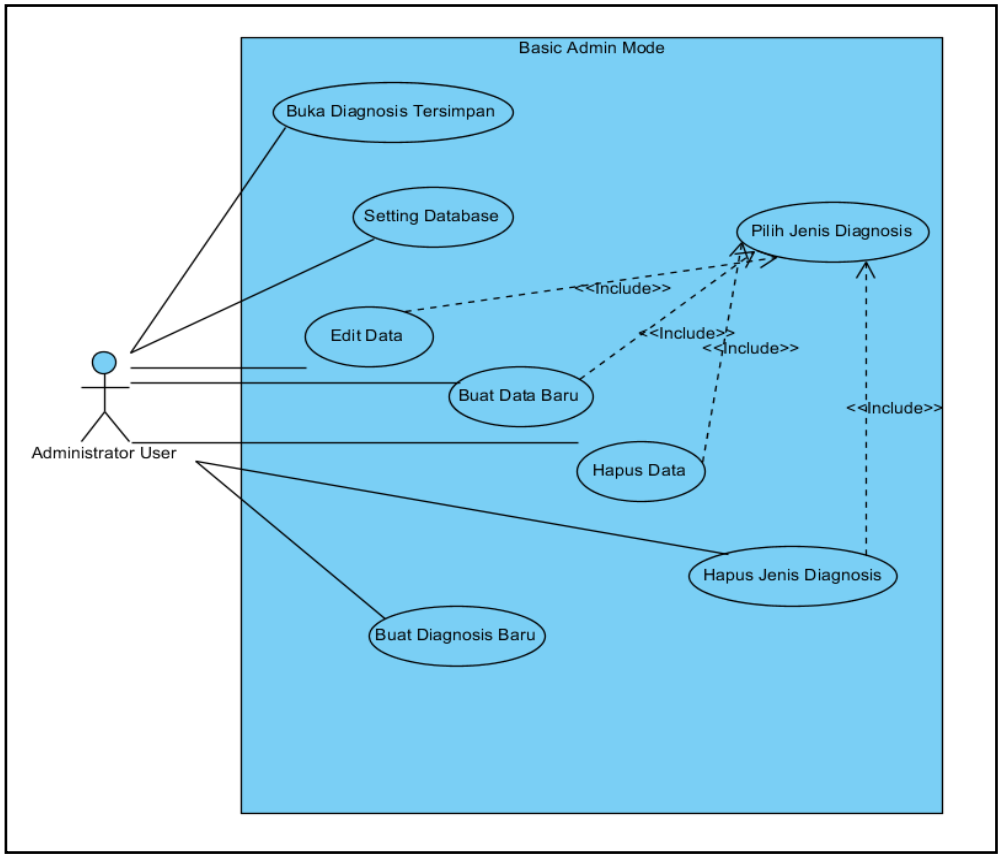
Berikut adalah definisi dari tiap proses pada user case Basic Admin Mode yang dijelaskan dalam bentuk tabel :

Tabel 15. Definisi *use case* : *Basic Admin Mode*

No	Use Case	Deskripsi
1	Buka Diagnosis Tersimpan	Proses untuk membuka proses diagnosis yang telah disimpan sebelumnya.
2	Setting Database	Proses untuk melakukan poengaturan database (mengubah file database yang digunakan).
3	Pilih Jenis Diagnosis	Proses untuk memilih salah satu jenis diagnosis yang akan dijalankan.
4	Proses Diagnosis	Proses dimana user (pengguna) mengikuti jalanya proses diagnosis dengan menjawab pertanyaan yang diajukan oleh sistem sampai ditemukan solusi.
5	Simpan Diagnosis	Proses untuk menyimpan proses diagnosis yang sedang berlangsung.
6	Edit Data	Proses untuk mengedit suatu data baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
7	Hapus Data	Proses untuk menghapus suatu data baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
8	Buat Data Baru	Proses untuk membuat data baru baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
9	Buat Jenis Diagnosis Baru	Proses untuk membuat jenis diagnosis baru pada *knowledge base*.
10	Hapus Jenis DIagnosis	Proses untuk menghapus suatu jenis diagnosis pada *knowledge base*.

c) use case : advanced admin mode.

Berikut ini adalah gambar use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Advanced Admin Mode :



Gambar 10. Use Case : Advanced Admin Mode

Berikut adalah definisi dari tiap proses pada user case Advanced Admin Mode yang dijelaskan dalam bentuk tabel :

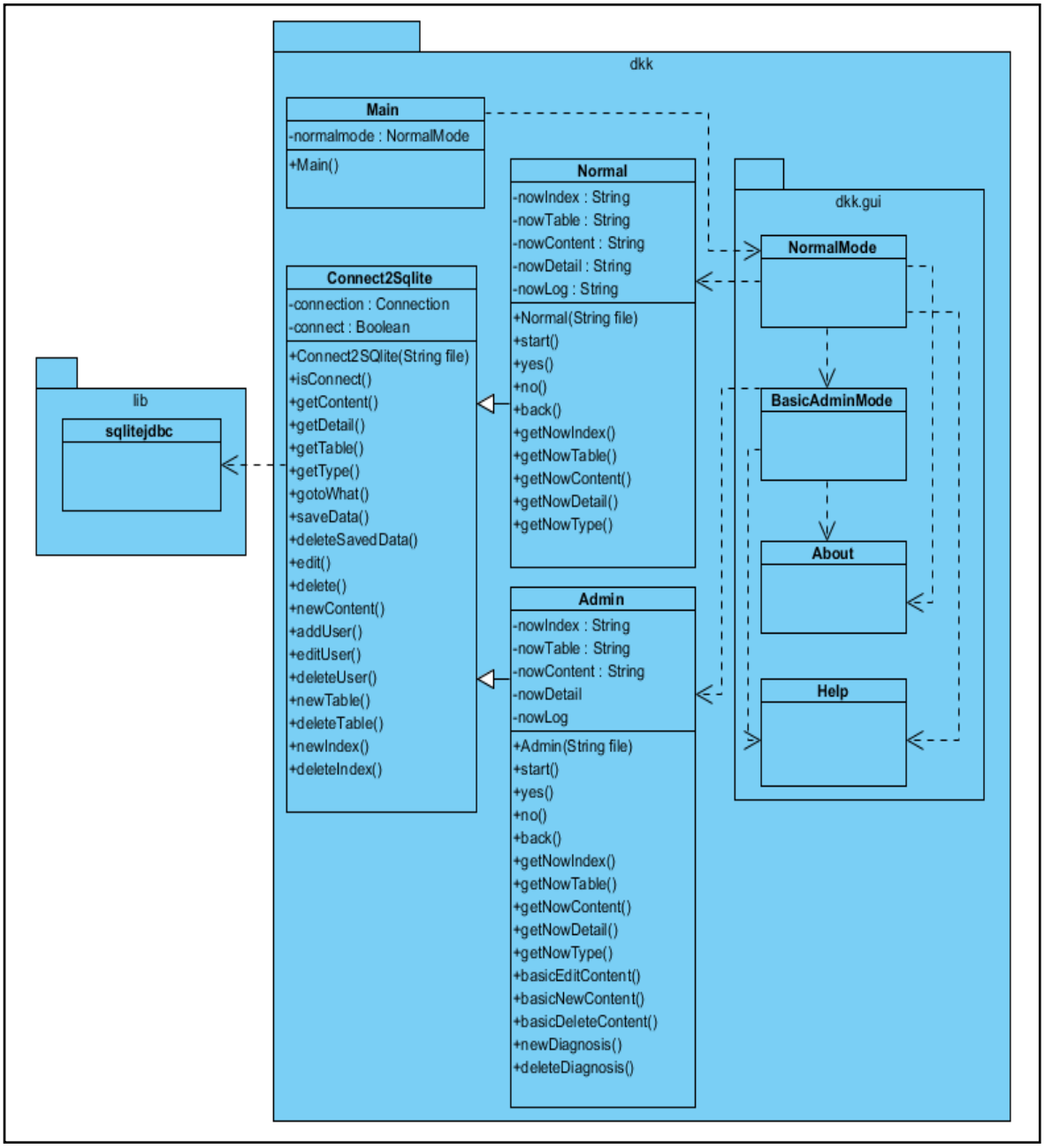
Tabel 16. Definisi *Use Case* : *Advanced Admin Mode*

No	Use Case	Deskripsi
1	Buka Diagnosis Tersimpan	Proses untuk membuka proses diagnosis yang telah disimpan sebelumnya.
2	Setting Database	Proses untuk melakukan poengaturan database (mengubah file database yang digunakan).
3	Pilih Jenis Diagnosis	Proses untuk memilih salah satu jenis diagnosis yang akan dijalankan.
4	Proses Diagnosis	Proses dimana user (pengguna) mengikuti jalanya proses diagnosis dengan menjawab pertanyaan yang diajukan oleh sistem sampai ditemukan solusi.
5	Edit Data	Proses untuk mengedit suatu data baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
6	Hapus Data	Proses untuk menghapus suatu data baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
7	Buat Data Baru	Proses untuk membuat data baru baik itu pertanyaan mapupun jawaban (solusi) pada *knowledge base*.
8	Buat Jenis Diagnosis Baru	Proses untuk membuat jenis diagnosis baru pada *knowledge base*.
9	Hapus Jenis DIagnosis	Proses untuk menghapus suatu jenis diagnosis pada *knowledge base*.

2) *class diagram.*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Dharwiyanti, 2006).

Berikut adalah gambar *class diagram* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



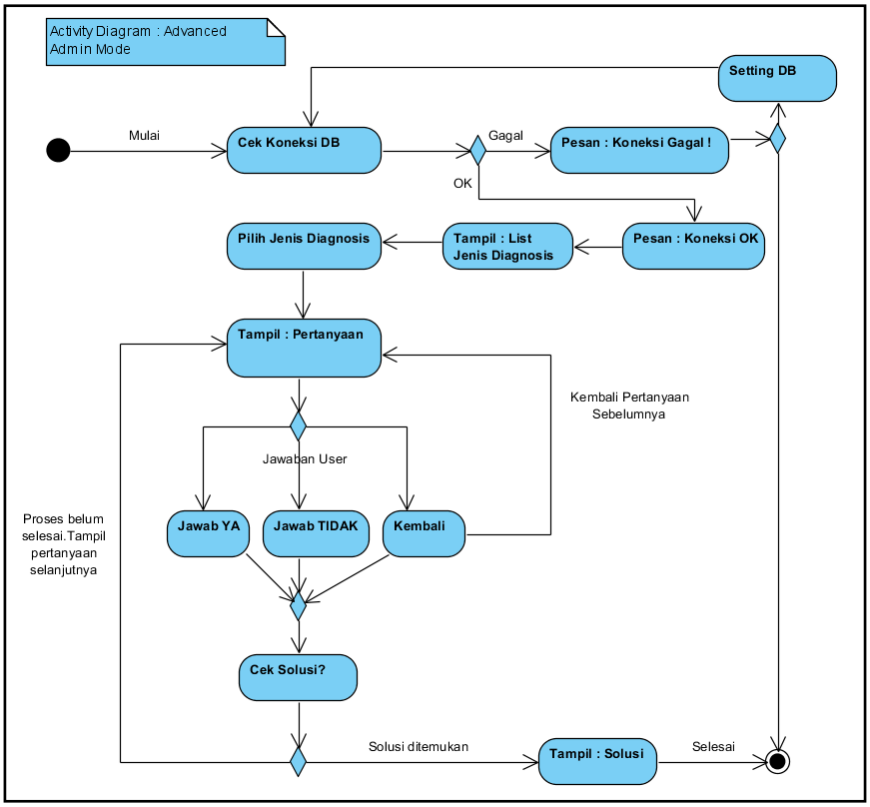
Gambar 11. Class Diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer

3) *avtivity diagram.*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana merekaberakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum (Dharwiyanti, 2006).

a) Activity Diagram : Normal Mode

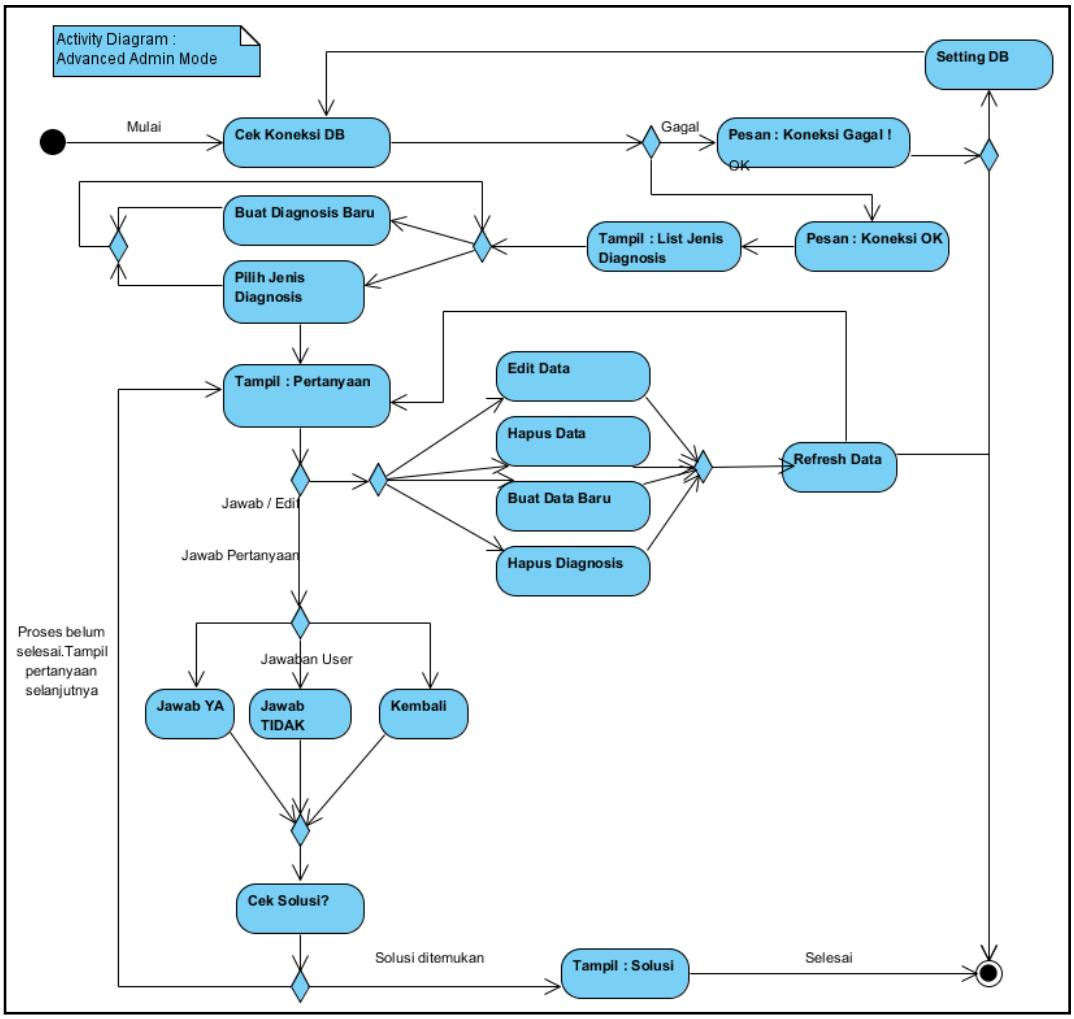
Berikut adalah gambar *activity diagram* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer pada bagian *Normal Mode* :



Gambar 12. Activity Diagram : Normal Mode

b) Activity Diagram : Basic Admin Mode

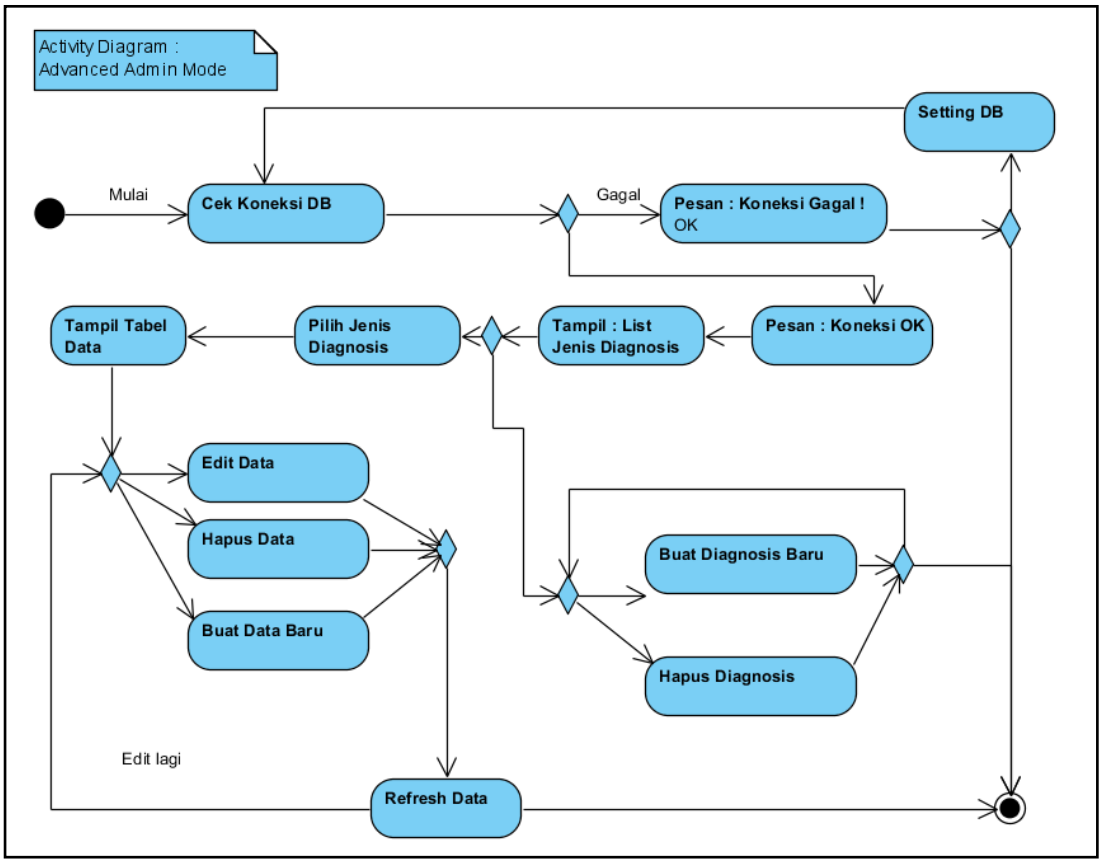
Berikut adalah gambar *activity diagram* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer pada bagian Basic Admin Mode :



Gambar 13. Activity Diagram : Basic Admin Mode

c) Activity Diagram : Advanced Admin Mode

Berikut adalah gambar activity diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer pada bagian Advanced Admin Mode :



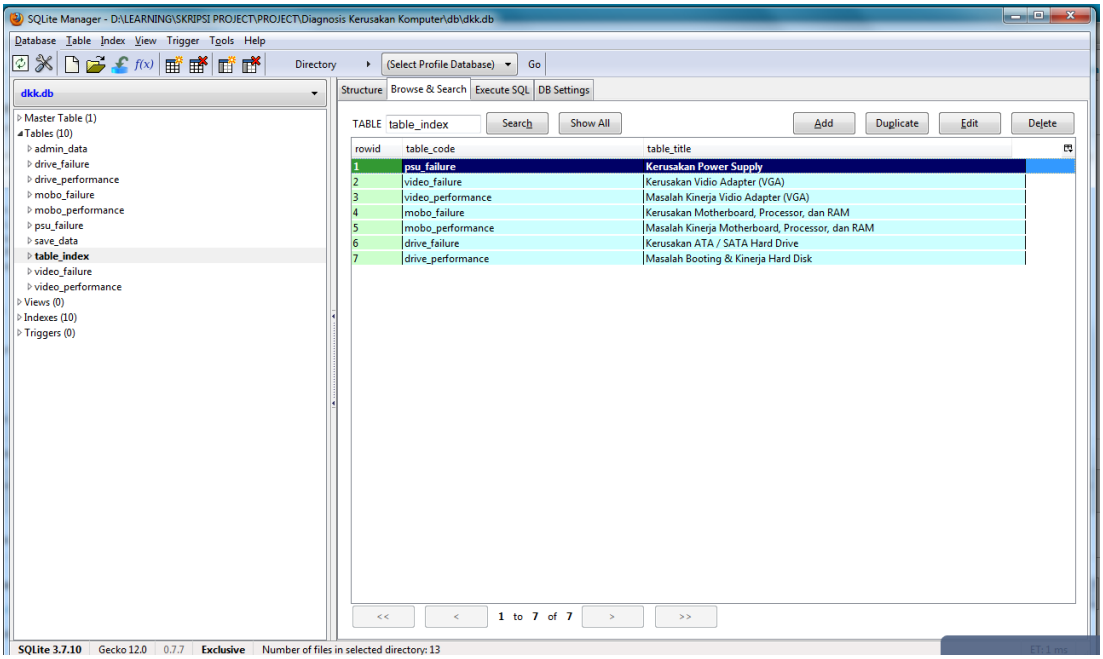
Gambar 14. Activity Diagram : Advanced Admin Mode

4. Konstruksi (*Construction*)

Tahap ini merupakan impelentasi dari rancangan yang telah dibuat. Secara umum pada tahap ini dilakukan entry data knowledge base pada database *SQLite* dan kemudian pembuatan aplikasi dengan bahasa pemrograman Java menggunakan *Netbeans IDE*. Pembuatan aplikasi secara umum terbagi menjadi dua proses utama yaitu desain antar muka (*graphical user interface*) dan pengkodean (*coding*).

a. Entry data knowledge-base pada database sqlite

Sebelum melakukan penulisan data pada database, untuk memudahkan penulisan knowledge-base dilakukan pada program pengolah angka (Microsoft Office Excell) yang kemudian disimpan dalam bentuk *.csv. File tersebut kemudian di-import ke dalam database SQLite menggunakan SQLite Manager.



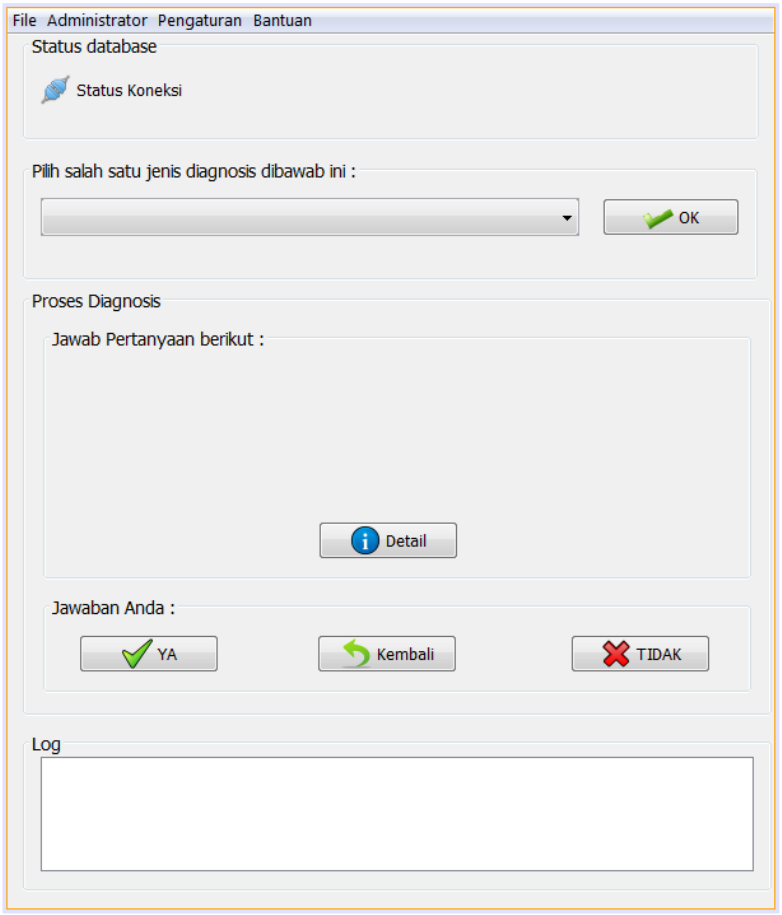
Gambar 15. SQLite Manager

b. Desain Antar Muka (Graphical User Interface)

Proses desain antar muka (*graphical user interface*) Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan bantuan *Netbeans Swing GUI Builder*. Hal itu dilakukan untuk memudahkan proses desain karena sebagian besar komponen yang digunakan dalam antar muka Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah komponen *Swing* (*javax.swing.*)*.

1) *desain antar muka halaman normal mode.*

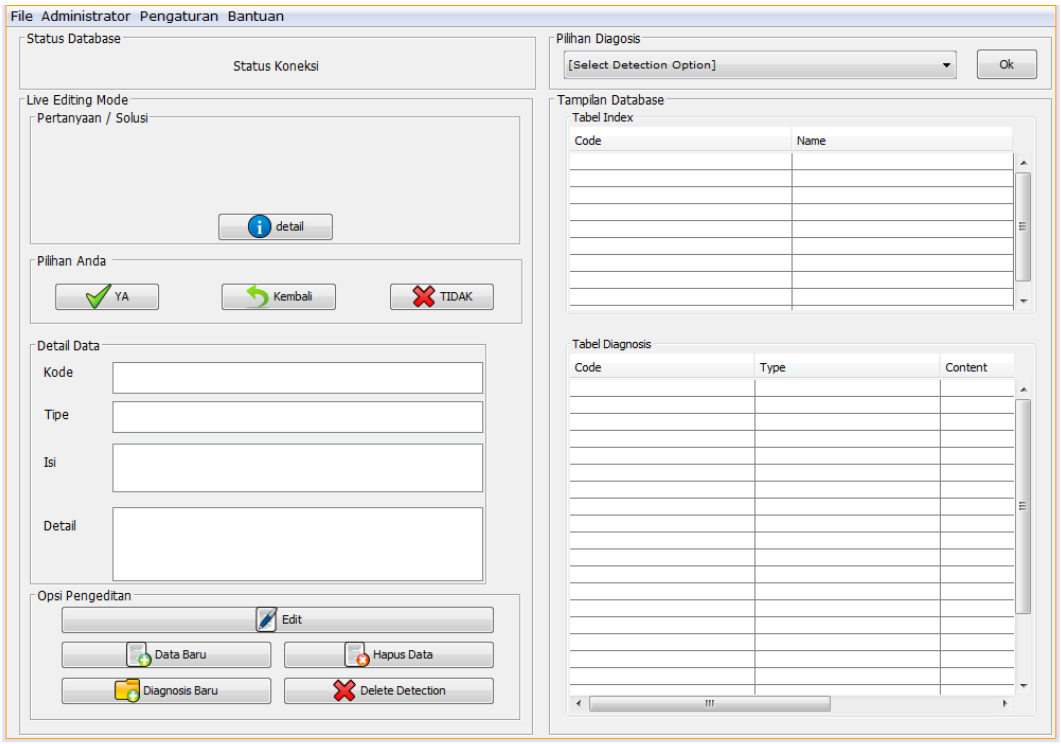
Berikut ini merupakan hasil desain antar muka (*graphical user interface*) Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Normal Mode :



Gambar 16. Desain Antar Muka : Normal Mode

2) ***desain antar muka halaman basic admin mode.***

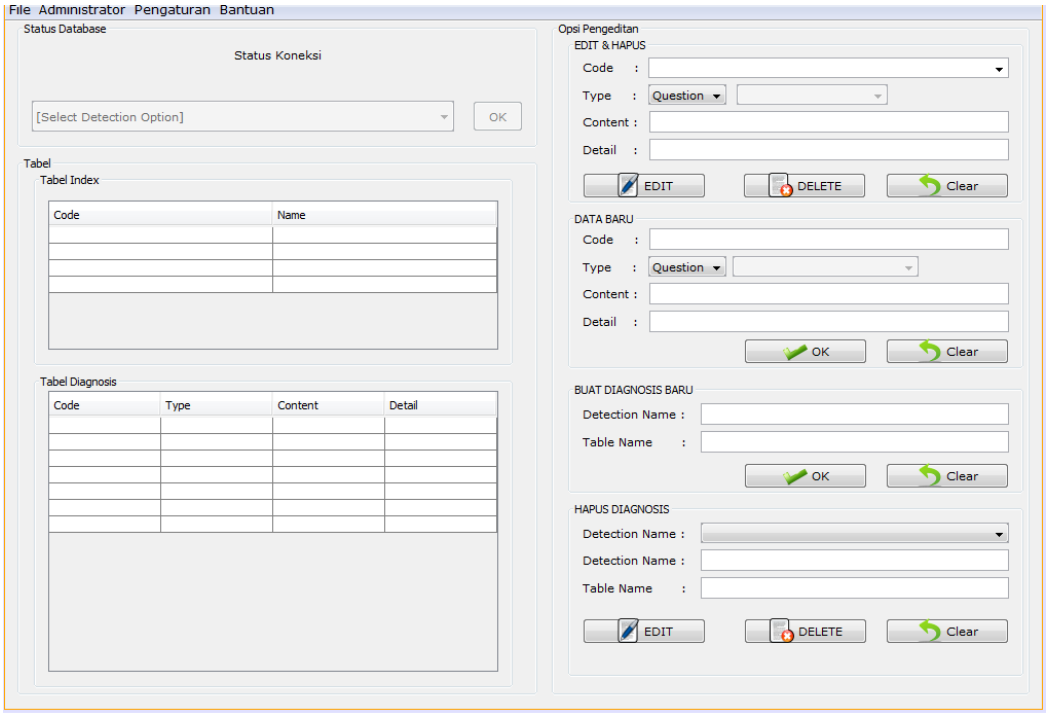
Berikut ini merupakan hasil desain antar muka (*graphical user interface*) Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Basic Admin Mode :



Gambar 17. Desain Antar Muka : Basic Admin Mode

3) *desain antar muka halaman advanced admin mode*

Berikut ini merupakan hasil desain antar muka (*graphical user interface*) Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer bagian Advanced Admin Mode :



Gambar 18. Desain Antar Muka : Advanced Admin Mode

c. Penulisan Program (*Coding*)

Setelah proses desain antar muka (*graphical user interface*) selesai, proses selanutnya adalah proses penulisan program dalam bahasa pemrogram Java. Proses penulisan program Aplikasi Diagnosis Kerusakan Komputer dilakukan dengan Netbeans IDE.

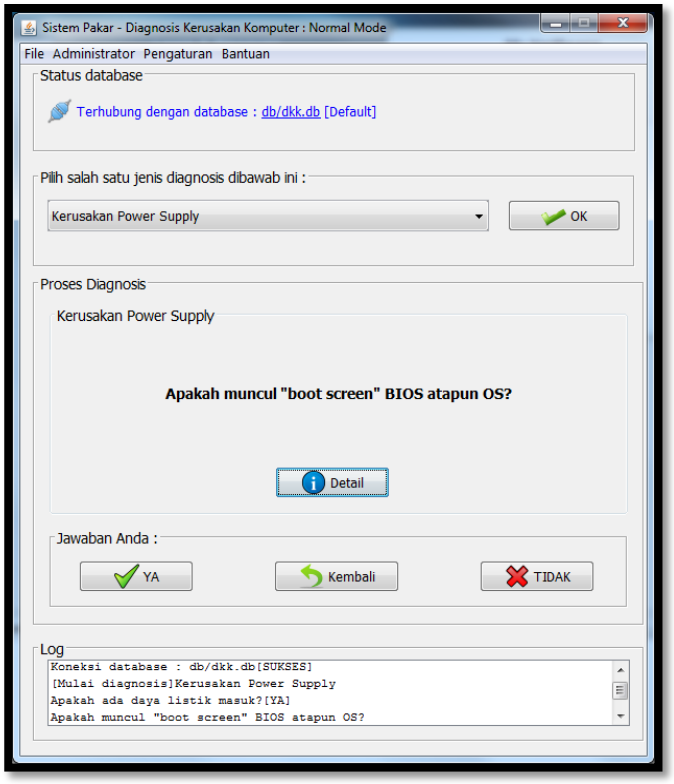
Dalam pemrogram berorientasi objek, hal utama yang perlu diperhatikan dalam coding adalah bagaimana rancangan class dan object yang digunakan. Berikut merupakan gambaran hierarki *package* dan *class* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang disajikan dalam bentuk tabel :

Tabel 17. Hierarki Package & Class Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer

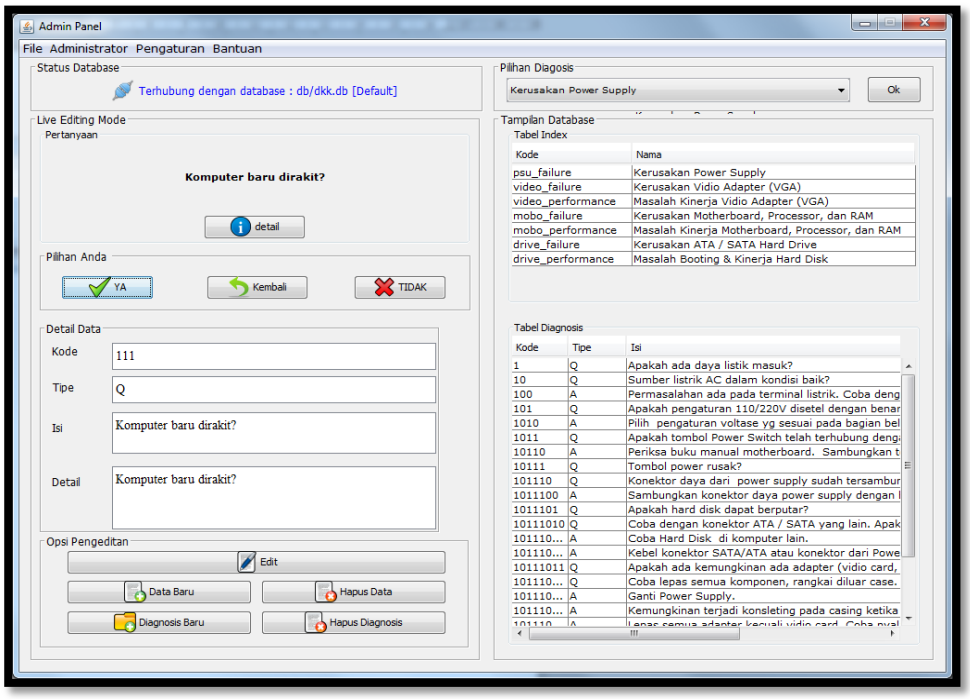
Package / Class	Deskripsi
dkk	*Package* utama Aplikasi SIstem Pakar Diagnosis Kerusakan Komputer (DKK).
Main	*Class* utama yang akan dipanggil ketika program dijalankan. Kemudian *class* ini akan memanggil class *NormalMode*.
Connect2Sqlite	*Class* untuk koneksi dengan database (Sqlite). Class ini berisi *method – method* untuk membaca dan memanipulasi data pada database. *Class* ini membutuhkan *library sqlitejdbc-v056*.
Normal	*Class* yang merupakan *subclass* dari *Connect2Sqlite*. Class ini berisi *method – method* yang digunakan pada *Normal Mode*.
Admin	*Class* yang merupakan *subclass* dari *Connect2Sqlite*. *Class* ini berisi method – metod yang digunakan pada Basic Admin Mode dan *Advanced Admin Mode*.
dkk.gui	*Package* yang berisi *class – class* yang menampilkan antar muka (*user interface*).
NormalMode	Antar muka *Normal Mode*.
BasicAdminMode	Antar muka *Basic Admin Mode*.
AdvancedAdminMode	Antar muka *Advanced Admin Mode*.
Help	Antar Muka halaman *Help* (Panduan)
About	Antar muka halaman *About* (Tentang Aplikasi)
dkk.image	Package ini berisi file – file berupa gambar (images) yang digunakan dalam antar muka.

d. Hasil pengembangan aplikasi.

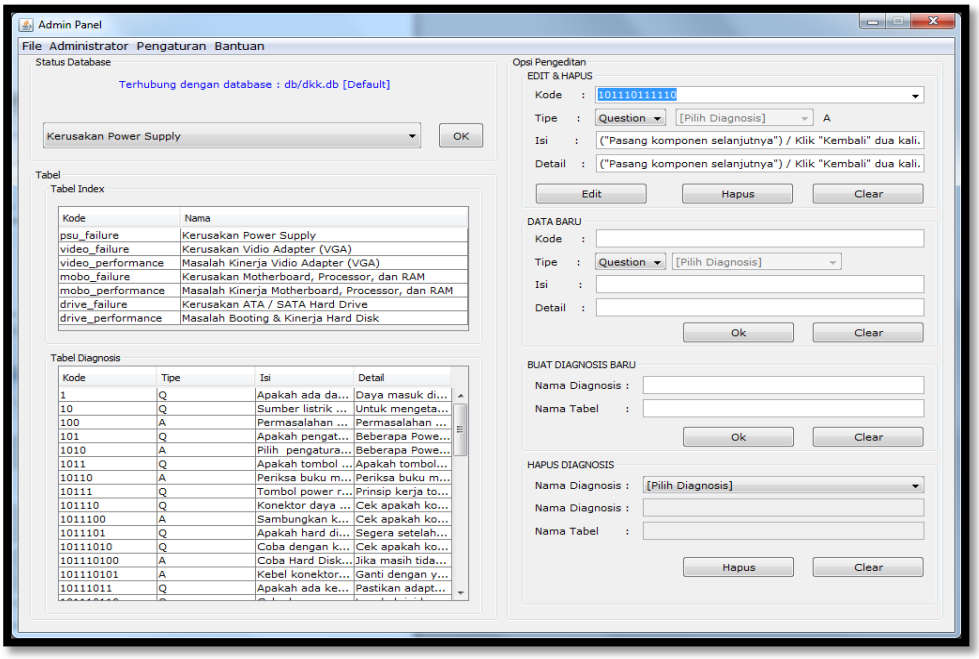
Setelah proses desain antar muka dan coding selesai, Aplikasi Sistem Pakar Diagnosis Kerusakan sudah bisa untuk dijalankan secara *executable* melalui file distribusi **.jar*. Berikut adalah beberapa screenshoot Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



Gambar 19. Screenshoot Aplikasi : Normal Mode



Gambar 20. Screenshoot Aplikasi : Basic Admin Mode



Gambar 21. Screenshoot Aplikasi : Advanced Admin Mode

5. Penyebaran (*Deployment*)

Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer sebagai produk dari pengembangan perangkat lunak yang telah dilakukan kemudian disebarkan agar dapat digunakan oleh pengguna. Penyebaran dilakukan secara *offline* dan *online*. Penyebaran secara offline dilakukan di SMK Muhammadiyah 2 Yogyakaryta, yakni kepada Siswa Kelas XI Jurusan Teknik Komputer dan Jaringan (TKJ) yang selanjutnya akan menjadi subjek penelitian pada pengujian faktor kualitas *usability* aplikasi sistem pakar tersebut. Penyebaran secara *online* dilakukan dengan mengunggah (*upload*) aplikasi pada web server dan membuat halaman khusus di internet.

B. Analisis Kualitas Perangkat Lunak

Setelah proses pengembangan perangkat lunak selesai, proses selanjutnya adalah proses analisis kualitas perangkat lunak. Dalam penelitian ini terbatas pada beberapa faktor kualitas perangkat lunak yaitu : *correctness*, *functionality*, *portability*, dan *usability*.

1. Analisis Faktor Kualitas *Correctness*

Fakor kualitas *correctness* dapat diukur dengan analisisis *defect per KLOC* (cacat / *error* pada setiap *KLOC/Kilo Line of Code*). Untuk mendapatkan nilai error/*KLOC*, diperlukan penghitungan jumlah *Kilo Lines of Code* (*KLOC*), kemudian dilakukan perhitungan jumlah *error* pada *source code* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan *FindBugs*.

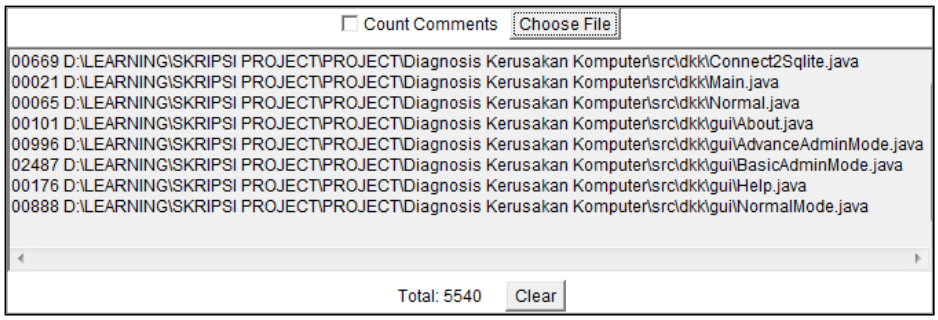
a. Menghitung Jumlah Lines Of Code (LOC)

Untuk menghitung jumlah *lines of code* (*LOC*), penulis menggunakan tool *Lines of Code Counter*. Tool ini dibuat dengan bahasa pemrograman java oleh John Rossi dan terdaftar dalam *Project Java.Net*.

Penghitungan *lines of code* (*LOC*) dilakukan pada source code Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Berikut adalah daftar file yang dihitung jumlah *LOC*-nya :

1. dkk/Main.java
2. dkk/Connect2Sqlite.java
3. dkk/Normal.java
4. dkk/Admin.java
5. dkk/gui/NormalMode.java
6. dkk/gui/BasicAdminMode.java
7. dkk/gui/AdvancedAdminMode.java
8. dkk/gui/About.java
9. dkk/gui/Help.java

Gambar berikut merupakan *screenshoot* hasil penghitungan *lines of code* (*LOC*) file source code Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



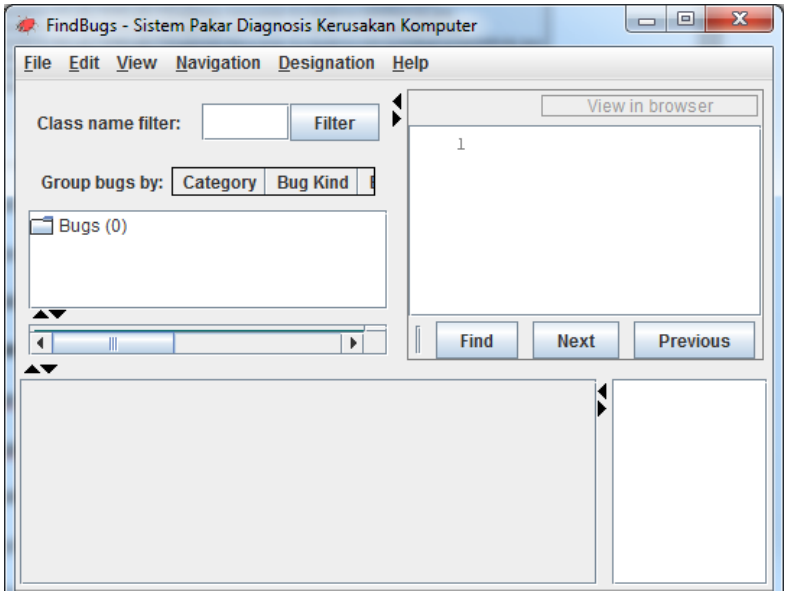
Gambar 22. Penghitungan Jumlah LOC

Screenshoot tersebut menjunjukan bahwa jumlah *lines of code* (*LOC*) *source code* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah **5540 *LOC*** = **5.54 *KLOC*** (***Kilo lines of code***).

b. Penghitungaan jumlah *error*

Perhitungan jumlah error pada source code Aplikasi Sistem Pakar Diagnosis Kerusakan komputer dilakukan menggunakan *FindBugs*. *FindBugs* merupakan *freeware tools* yang dikembangkan oleh *The University of Maryland*. *FindBugs* mampu mendeteksi *bugs (error)* dalam beberapa jenis kategori. Dalam penelitian ini akan digunakan dua kategori yaitu kategori *bugs* yang berkaitan dengan analisis faktor kualitas *correctness* : *correctness* dan *multithreaded correctness*.

Berikut adalah screenshoot penghitungan jumlah *error* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dengan *FindBugs* :



Gambar 23. Penghitungan Jumlah Error menggunakan *FindBugs*

Screenshoot tersebut menunjukan bahwa jumlah error untuk pengujian fakor kualitas correctness Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan *FindBugs* adalah *0*.

c. Perbandingan hasil pengujian dengan standard yang telah ditentukan

Dari hasil pengujian sebelumnya, didapatkan bahwa nilai *Error/KLOC* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah 0 *Error/KLOC.* Aplikasi akan dikatakan **LOLOS** pengujian jika jumlah error <= (lebih sedikit atau sama dengan) standar yang digunakan. Sebaliknya, aplikasi akan dikatatakan **GAGAL** jika jumlah error melebihi standar yang digunakan.

Tabel 18. Perbandingan Hasil Pengujian Faktor Kualitas Correctnes dengan Standar yang Digunakan

Nama Standar	Nilai Standar (*Error* / KLOC)	Hasil Pengujian Aplikasi	Keterangan
Industry Average	1 – 25	$\frac{0}{5.54} = 0$	**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.
Microsoft Application	0.5		**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.

Tabel tersebut menujukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas correctness baik dari Standar *Industry Average* maupun *Microsoft Application*.

2. Analisis Faktor Kualitas *Functionality*

Faktor kualitas *functionality* diuji dengan melakukan tes pada setiap fungsi yang terdapat pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Hasil pengujian kemudian dibandingkan dengan standar *functioality* yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo.*

Berikut ini adalah tabel (Tabel 19 dan Tabel 20) rangkuman test case pengujian faktor kualitas *functionality* pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer pada fungsi primer dan fungsi sekunder (Hasil *test case* lengkap dilampirkan) :

Tabel 19. Rangkuman *Test Case* Fungsi Primer

No	Nama Tes	Lolos / Gagal
1	NormalModeTest1	Lolos
2	NormalModeTest2	Lolos
3	NormalModeTest3	Lolos
4	NormalModeTest4	Lolos
5	NormalModeTest5	Lolos
6	NormalModeTest6	Lolos
7	NormalModeTest7	Lolos
8	NormalModeTest8	Lolos
9	LoginTest1	Lolos
10	LoginTest2	Lolos
11	BasicAdminModeTest1	Lolos
12	BasicAdminModeTest2	Lolos
13	BasicAdminModeTest3	Lolos
14	BasicAdminModeTest4	Lolos
15	BasicAdminModeTest5	Lolos
16	BasicAdminModeTest6	Lolos
17	BasicAdminModeTest7	Lolos
18	BasicAdminModeTest8	Lolos
19	BasicAdminModeTest9	Lolos
20	BasicAdminModeTest10	Lolos
21	AdvancedAdminModeTest1	Lolos
22	AdvancedAdminModeTest2	Lolos
23	AdvancedAdminModeTest3	Lolos
24	AdvancedAdminModeTest4	Lolos
25	AdvancedAdminModeTest5	Lolos

Tabel 20. Rangkuman Test Case Fungsi Pendukung.

No	Nama Tes	Lolos / Gagal
1	ContributingFunctionalityTest1	Lolos
2	ContributingFunctionalityTest2	Lolos
3	ContributingFunctionalityTest3	Lolos
4	ContributingFunctionalityTest4	Lolos
5	ContributingFunctionalityTest5	Lolos
6	ContributingFunctionalityTest6	Lolos

Hasil *test case* kemudian dibandingkan dengan standar *functionality* dalam program *Microsoft Certification Logo*.

Tabel 21. Perbandingan Hasil Pengujian Faktor Kualitas Functionality dengan

Kriteria Lolos	Kriteria Gagal	Hasil Pengujian	Keterangan
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.	Semua fungsi primer dan fungsi pendukung berjalan dengan baik.	**Lolos**

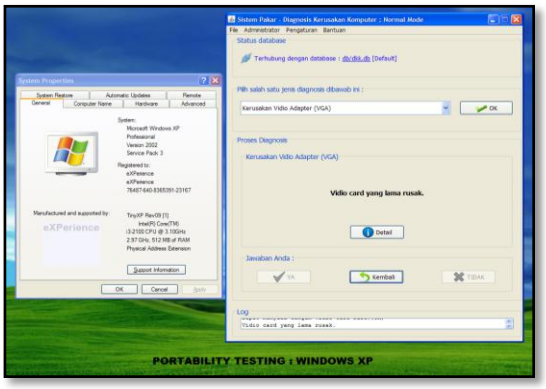
Tabel tersebut menunjukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer **lolos** pengujian faktor kualitas *functionality*, atau dapat dikatakan bahwa aplikasi yang dikembangkan memenuhi faktor kualitas *functionality*.

3. Analisis Faktor Kualitas *Portability*

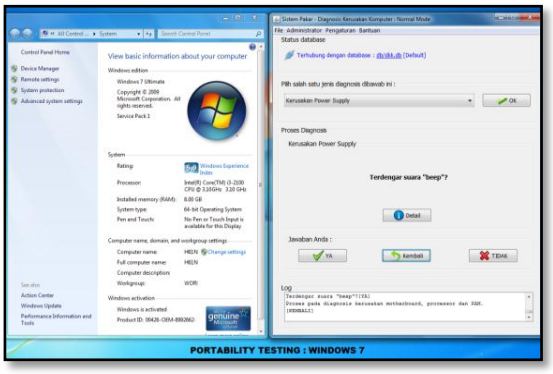
Pengujian faktor kualitas *portability* dilakukan dengan melakukan pengujian aplikasi pada beberapa sistem operasi yang berbeda. Berikut adalah rangkuman *test case* (*test case* lengkap dilampirkan) dan screenshoot pengujian faktor kualitas usability Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer

Tabel 22. Rangkuman *Test Case* Faktor Kualitas *Usability*

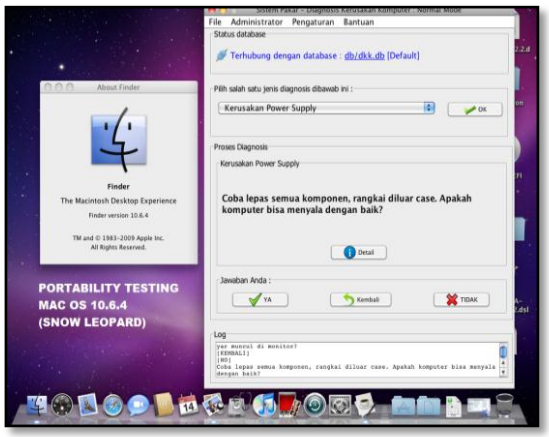
No	Sistem Operasi	Hasil Pengujian	Lolos / Gagal
1	Windows XP	Aplikasi berjalan dengan baik.	Lolos
2	Windows 7	Aplikasi berjalan dengan baik.	Lolos
3	Linux : Ubuntu	Aplikasi berjalan dengan baik.	Lolos
4	Linux : OpenSuse	Aplikasi berjalan dengan baik.	Lolos
5	Mac OS X Snow Leopard	Aplikasi berjalan dengan baik.	Lolos
6	Mac OS X Lion	Aplikasi berjalan dengan baik.	Lolos



Gambar 24. Screenshoot Pengujian Portability : Windows Xp

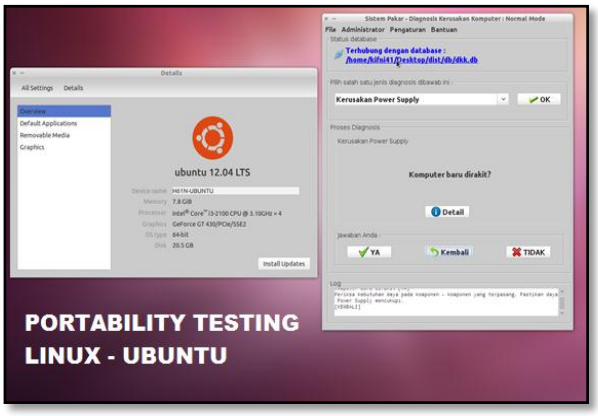


Gambar 25. Screenshoot Pengujian Portability : Windows 7

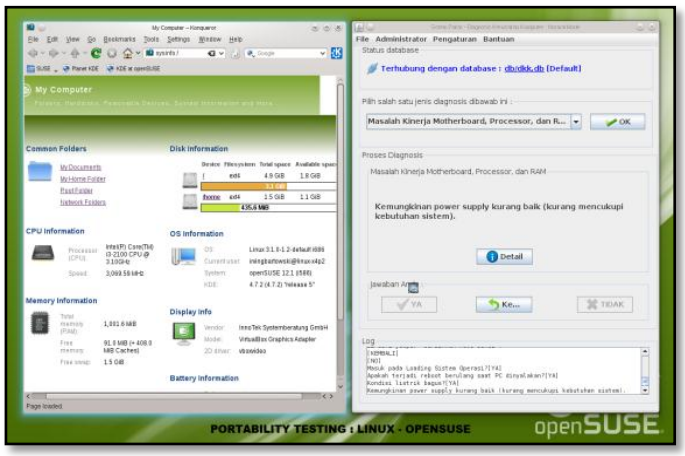


Gambar 26. Screenshoot Pengujian Portability : Mac OS X Snow Leopard

Gambar 27. Screenshoot Pengujian Portability : Mac OS X Lion



Gambar 28. Screenshoot Pengujian Portability : Ubuntu



Gambar 29. Screenshoot Pengujian Portability : Opensuse

Tabel dan gambar tersebut menunjukan bahwa Aplikasi SIstem Pakar Diagnosis Kerusakan Komputer dapat berjalan dengan baik pada setiap sistem operasi yang diujikan sehingga dapat diambil kesimpulan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas *portability*.

4. Analisis Faktor Kualitas *Usability*

Pengujian faktor kualitas usability dilakukan dengan menggunakan metode kuesioner. Kuesioner yang digunakan mengacu pada *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis . Kuesioner diberikan kepada siswa kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta sebanyak 55 siswa. Berikut adalah tabel jawaban responden terhadap tiap pertanyaan dalam kuesioner.

Tabel 23. Tabel Jawaban Responden Terhadap Pertanyaan Kuesioner *Usability*

Pertanyaan	Jawaban Responden				
	Sangat Setuju	**Setuju**	**Ragu - Ragu**	**Tidak Setuju**	**Sangat tidak setuju**
Pertanyaan 1	19	35	1	0	0
Pertanyaan 2	23	31	1	0	0
Pertanyaan 3	4	43	8	0	0
Pertanyaan 4	8	22	23	2	0
Pertanyaan 5	5	41	9	0	0
Pertanyaan 6	16	34	4	1	0
Pertanyaan 7	13	34	8	0	0
Pertanyaan 8	8	34	13	0	0
Pertanyaan 9	10	36	9	0	0
Pertanyaan 10	17	33	5	0	0
Pertanyaan 11	6	43	5	1	0
Pertanyaan 12	12	35	8	0	0
Pertanyaan 13	12	39	3	0	0
Pertanyaan 14	7	42	6	0	0
Pertanyaan 15	8	39	8	0	0
Pertanyaan 16	4	38	12	0	1
Pertanyaan 17	3	44	7	1	0
Pertanyaan 18	3	34	18	0	0
Pertanyaan 19	10	40	4	1	0
Jumlah	**188**	**697**	**152**	**6**	**1**

Data yang dihasilkan dari kuisioner meruapakan data yang bersifat kuantitatif. Data tersebut dapat dikonversi ke dalam data kualitatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Tabel 24. Konversi Jawaban Item Kuesioner menjadi Nilai Kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Berikut perhitungan jumlah skor yang didapat dari hasil kuesioner :

- Jumlah jawaban “Sangat Setuju” = 189 x 5 = 945
- Jumlah jawaban “Setuju” = 701 x 4 = 2804
- Jumlah jawaban “Ragu - Ragu” = 148 x 3 = 444
- Jumlah jawaban “Tidak Setuju” = 7 x 2 = 14
- Jumlah jawaban “Sangat Tidak Setuju” = 0 x 1 = 0
- **Jumlah Total = 4203**

Skor yang didapatkan pada tiap hasil kuesioner tersebut kemudian diambil nilai rata - rata. Nilai rata – rata tersebut kemudian dijumlahkan.

Dengan jumlah responden sebanyak 55 orang maka dapat dihitung nilai tertinggi dan nilai terendah sebagai berikut :

1. Nilai tertinggi = 55 x 19 x 5 = **5225** , dengan asumsi semua responden memberi jawaban sangat setuju pada setiap item kuesioner.
2. Nilai terendah = 55 x 19 x 1 = **1045**, dengan asumsi semua responden memberi jawaban “sangat tidak setuju” pada setiap item kuesioner.

Dari data tersbut data tersebut, kemudian dapat disusun kategori penilaian kuesioner berdasarkan perhitungan interval kelas.

1) Menghitung Jumlah Kelas

K = 1 + 3,3 log n

= 1 + (3,3 x log 55)

= 1 + 4,88 = 5,88 ≈ 5

(dibulatkan menjadi 5 agar jumlah kelas sama dengan jumlah pilihan jawaban pada kuesioner)

2) Menghitung Rentang Data

Rentang Data = (Data terbesar – Data terkecil) + 1

= 5225 – 1045

= 4180

3) Menghitung Panjang Kelas

Panjang Kelas = Rentang Data / Jumlah Kelas

= 4180 / 5

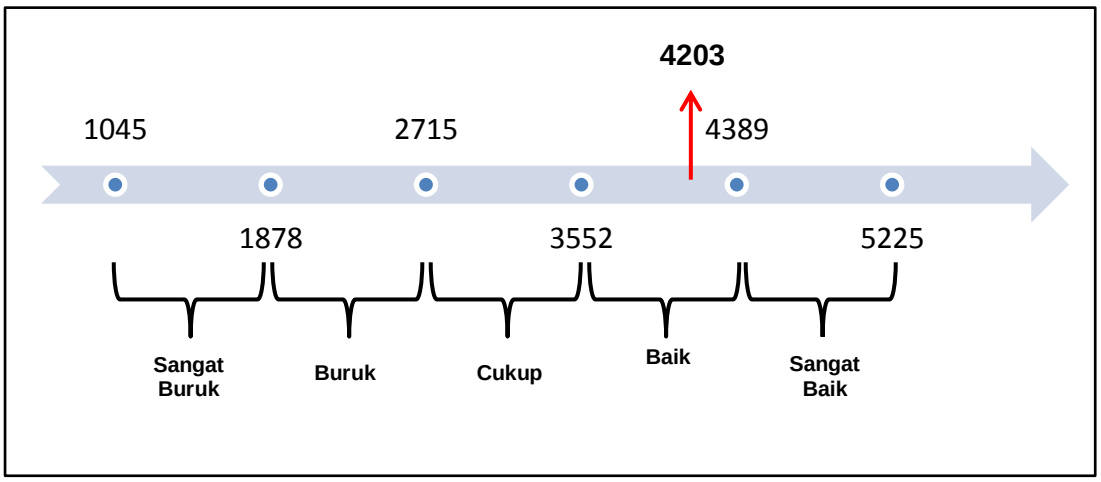
= 836

Dengan data tersebut, kemudian disusun kategorisasi penilain faktor kualitas *usability* bedasarkan interval nilai kuesioner.

Tabel 25 Kategori Penilaian Faktor Kualitas *Usability*

Interval Nilai	Ketegori
1045 - 1877	Sangat Buruk
1878- 2714	Buruk
2715- 3551	Cukup
3552- 4388	Baik
4389- 5225	Sangat Baik

Gambar dibawah ini menujukan perbandingan nilai total yang didapat dari hasil kuesioner dengan interval kategori penilaian faktor kualitas usability :



Gambar 30. Perbandingan Nilai Hasil Kuesioner dengan Ketegorisasi Penilaian Faktor Kualitas Usability

Gambar tersebut menujukan bahwa hasil pengujian kuisioner berada dalam interval 3552 – 4338 dan termasuk dalam kategori **Baik**.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya dapat diambil beberapa kesimpulan sebagai berikut :

1. Hasil dari pengembangan perangkat lunak adalah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang terdiri dari tiga mode yaitu *Normal Mode*, *Basic Admin Mode* dan *Advanced Admin Mode*. Aplikasi tersebut dapat dijalankan secara langsung di komputer dengan sistem operasi yang mendukung *Java Runtime Environment* (*JRE*).
2. Nilai error/KLOC Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah 0 dan dapat disimpulkan bahwa aplikasi memenuhi standar faktor kualitas correctness.
3. Seluruh fungsi primer dan fungsi pendukung Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer berjalan dengan sebagaimana mestinya dan dapat disimpulkan bahwa aplikasi telah memenuhi standar faktor kualitas *functionality*.
4. Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dapat berjalan dengan baik pada semua sistem operasi yang diujikan dan dapat disimpulkan bahwa aplikasi telah memenuhi standar faktor kualitas *portability*.
5. Hasil analisis faktor kualitas *usability* menunjukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer masuk dalam kriteria “Baik” dalam faktor kualitas *usability*.

B. Saran

1. Dalam pembuatan knowledge-base Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini Penulis hanya menggunakan 7 jenis diagnosis dari 17 jenis diagnosis yang terdapat dalam buku "*Computer Failure with Diagnostic Flowchart*". Penelitian selanjutnya dapat mengembangkan knowledge-base dengan menambahkan jenis diagnosis yang belum dibuat dalam penelitian ini.
2. Dalam penelitian ini Penulis hanya melakukan pengujian pada faktor kualitas *correctness*, *functionality*, *portability*, dan *usability*. Penelitian selanjutnya dapat dapat dilakukan pengujian pada faktor kualitas yang lain.

DAFTAR PUSTAKA

Bach, J. (2005). *General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo.* Dipetik Maret 3, 2012, dari Satisfice, Inc: http://www.satisfice.com/tools/procedure.pdf

Agarwal, B. B., Tayal, S. P., & Gupta, M. (2010). *Software Engineering and Testing.* Sudbury: Jones and Bartlett Publishers.

Allen, G., & Owens, M. (2010). *The Definitive Guide to SQLite.* New York: Appress.

Avestro, J. (2007). *Pengenalan Pemrograman Java 1.* (F. Thamura, E. Subiyantoro, C. K. Ratih, R. N.S., & M. K. Mufida, Penerj.) Jakarta: J.E.N.I.

Dharwiyanti, S. (2006, 8 25). *Pengantar Unified Modeling Language (UML).* Dipetik 6 17, 2012, dari IlmuKomputer.Com: http://ilmukomputer.org/2006/08/25/pengantar-uml/

Hartati, S., & Iswanti, S. (2008). *Sistem Pakar & Pengembangannya.* Yogyakarta: Graha Ilmu.

Hartati, S., & Iswanti, S. (2008). *Sistem Pakar & Pengembangannya.* Yogyakarta: Graha Ilmu.

Hass, A. M. (2008). *Guide to Advanced Software Testing.* Norwood: Artech House.

McConnell, S. C. (2004). *Code Complete.* Redmond : Microsoft Press.

Pressman, R. (2010). *Software Engineering : A Practitioner's Approach* (7 ed.). New York: McGraw Hill.

Pugh, B. (2011). *FindBugs™.* Dipetik Maret 22, 2012, dari FindBugs Bug Descriptions: http://findbugs.sourceforge.net/bugDescriptions.html

Rosenthal, M. (2010). *Computer Rapair and Diagnostic Flowchart.* Massachusetts: Foner Books.

Sugiyono. (2009). *Metode Penelitian Pendidikan.* Bandung: Alfabeta.

Sugiyono. (2010). *Statistika untuk Penelitian.* Bandung: Alfabeta.

Sutojo, T., Mulyanto, E., & Suhartono, V. (2011). *Kecerdasan Buatan.* Yogyakarta: Andi.

Vandegriend, B. (2009). *Why You Should Be Using FindBugs.* Dipetik Maret 19, 2012, dari Professional Software Development: http://www.basilv.com/psd/blog/2009/why-you-should-be-using-findbugs

Worldometers. (2011, September 8). *Computers sold in the world this year.* Dipetik Januari 13, 2012, dari Worldometers: http://www.worldometers.info/computers/

LAMPIRAN

1. Surat Keputusan Pengangkatan Pembimbing

2. Surat Permohonan Ijin Penelitian

3. Surat Keterangan Ijin Penelitian

4. Surat Keterangan Telah Melaksanakan Penelitian

5. Test Case Pengujian Faktor Kualitas Functionality

6. Test Case Pengujian Faktor Kualitas Portability

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PAKAR DIAGNOSIS KERUSAKAN KOMPUTER

Oleh : Kifni Taufik Darmawan

Program Studi Pendidikan Teknik Informatika

Universitas Negeri Yogyakarta

td.kifni@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan bahasa pemrograman Java dan database SQLite dan melakukan analisis kualitas pada aplikasi yang dikembangkan, khususnya pada faktor kualitas *correctness*, *functionality*, *portability*, dan *usability*.

Pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan kaidah rekayasa perangkat lunak (*software engineering*) yaitu dimulai dari proses perencanaan (*planning*), modeling (*perancangan*), konstruksi (*construction*), dan penyebaran (*deployment*). Analisis faktor kualitas correctness dilakukan dengan perhitungan jumlah error / KLOC. Analisis faktor kualitas *functionality* dilakukan dengan pengujian setiap fungsi aplikasi. Analisis faktor kualitas *portability* dilakukan dengan percobaan penjalanan aplikasi pada beberapa sistem operasi yang berbeda. Analisis faktor kualitas *usability* dilakukan dengan metode kuesioner dengan responden Siswa Kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta.

Hasil pengembangan aplikasi yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dalam bentuk file *runnable*. Hasil analisis kualitas menunjukan bahwa aplikasi yang dikembangkan memenuhi semua standar faktor kualitas yang diujikan yaitu *correctness*, *functionality*, *portability*, dan *usability*.

Kata kunci : sistem pakar, java, sqlite, *software quality*, *correctness*, *functionality*, *portability*, *usability*.

I. PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi komputer diikuti pula dengan meningkatnya jumlah pengguna komputer di dunia. Seiring dengan meningkatnya jumlah pengguna komputer, permasalahan kerusakan komputer menjadi masalah yang cukup pelik. Banyak sekali dana yang dikeluarkan untuk memperbaiki kerusakan

komputer, padahal kerusakan komputer yang terjadi belum tentu rumit dan dapat diperbaiki secara mandiri.

Sistem Pakar (*Expert System*) sebagai salah satu hasil dari perkembangan ilmu komputer, khususnya di bidang kecerdasan buatan (*Artificial Intelegence*), dapat memberikan solusi untuk mengatasi masalah tersebut. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para pakar/ahli. Penulis menganggap bahwa diperlukan sebuah aplikasi sistem pakar yang dapat membenatu pengguna komputer dalam mendiagnosis kerusakan komputer dan membantu pengguna dalam memperbaikinya

Seperti halnya perangkat lunak lain, Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini dibuat berdasarkan kaidah rekayasa perangkat lunak (*software engineering*) mulai dari proses awal hingga akhir. Pada akhirnya, aplikasi sistem pakar yang dibuat diharapkan memenuhi standar kualitas perangkat lunak dalam kaidah rakayasa perangkat lunak (*software engineering*). Kualitas perangkat lunak ditentukan oleh berbagai faktor. Beberapa ahli maupun organisasi telah merumuskan kriteria – keriteria pengujian kualias perangkat lunak. McCall, Richards, dan Walter merumuskan kriteria – kriteria untuk melakukan pengujian kualitas perangkat lunak yang terdiri dari beberapa faktor – faktor kualitas yaitu : *maintainability, flexibility, testability, portability, reusability, interoperability, correctness, reliability, usability, integrity, dan efficiency*. Selain itu, *International Standard Organization* (ISO) juga mengeluarkan standar ISO-9126 yang terdiri dari enam faktor kualitas yaitu *functionality, reliability, usability, efficiency, maintainability*, dan *portability*.

Berdasarkan latar belakang tersebut, melalui penelitian ini Penulis bermaksud untuk mengembangkan Aplikasi Sistem Pakar untuk Diagnosis Kerusakan Komputer dan melakukan analisis kualitas aplikasi tersebut. Aplikasi sistem pakar yang akan dikembangakan akan difokuskan untuk diagnosis kerusakan komputer desktop dimana komponen – komponenya dapat dilepas, diganti, dan dipasang kembali dengan mudah. Aplikasi tersebut akan dikembangkan menggunakan Bahasa Pemrograman Java dan Database SQlite.

Dalam kaitanya dengan pengujian kualitas perangkat lunak, aplikasi tersebut akan diuji pada beberapa faktor kualitas penrangkat lunak yaitu : correctness, functionality, portability,dan *usability*.

II. KAJIAN TEORI

A. Deskripsi Teori

1. Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan Buatan (Artificial Intelegence) adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia Saat ini kecerdasan buatan merupakan cabang ilmu komputer yang selalu dikembangkan, dalam implementasinya kecerdasan buatan yang banyak ditemui saat ini dibagi dalam beberapa bidang sebagai beberapa bidang antara lain : Sistem Pakar (*Expert System*); Pengenalan Ucapan (*Speech Recognition*); *Game Playing*, Pengolahan Bahasa Alami (*Natural Language Processing*),dan Logika Fuzzy (*Fuzzy Logic*) (Hartati & Iswanti, 2008).

2. Sistem Pakar (*Expert System*)

Menurut Sutojo, Mulyanto dan Suhartono (2011), Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu masalah. Sedangkan menurut Giarratano dan Riley, Sistem Pakar merupakan salah satu cabang dari Kecerdasan Buatan (*Artificial Intelligence*) yang menggunakan pengetahuan – pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu (Hartati & Iswanti, 2008).

Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar (expert) harus bisa melakukan hal – hal yang dapat dikerjalan oleh seorang pakar. ntuk membangun sistem yang seperti itu maka sebuah sistem pakar harus memiliki komponen – komponen sebagai berikut :

1. antar muka pengguna (user interface) : merupakan mekanisme yang digunakan oleh pengguna untuk dan sistem pakar untuk berkomunikasi (Hartati & Iswanti, 2008).

2. basis pengetahuan (knowledge base) : merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. (Hartati & Iswanti, 2008).
3. mesin inferensi (inference engine) : sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan (Sutojo, Mulyanto, & Suhartono, 2011).
4. memori kerja (working memory) : merupakan bagian dari sistem pakar yang menyimpan fakta – fakta yang diperoleh saat dilakukan proses konsultasi (Hartati & Iswanti, 2008).

3. Java

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Avestro, 2007).

Bahasa pemrograman Java memiliki berbagai keunggulan, keunggulan inilah yang merupakan faktor mengapa penulis menggunakan Java untuk membangun aplikasi sistem pakar dalam penelitian ini. Berikut adalah keunggulan – keunggulan Java menurut white paper resmi dari SUN : Sederhana , Berorientasi objek (*object oriented*) , Dapat didistribusi dengan mudah,Robust , Aman,*Architecture Neutral,Portable ,Multithreaded,*Dinamis

4. SQLite

Menurut Allen dan Owens (2010), SQLite merupakan sebuah RDBMS(Relational Database Management System) yang bersifat *embedded* dan *opensource*. Sampai saat ini SQlite dikenal sebagai RDBMS dengan tingkat portabilitas yang tinggi, mudah digunakan, ringkas, dan reliabel. Hal inilah yang mendorong penulis untuk memilih SQlite dalam membangun Aplikasi Sistem Pakar untuk Mendiagnosis Kerusakan Komputer.

5. Kualitas Perangkat Lunak (Software Quality)

Agarwal, Tayal dan Gupta dalam bukunya (2010) menjelaskan bahwa kualitas perangkat lunak merupakan kesesuaian terhadap persyaratan fungsional dan kinerja secara eksplisit, standar pengembangan yang terdokumentasi secara eksplisit, dan karakteristik implisit yang diharapkan dari perangkat lunak yang dikembangkan secara profesional.

Teori – teori tentang kualitas perangkat lunak telah di kemukakan oleh beberapa ahli. Teori –teori tersebut antara lain :

a. McCall quality factors

McCall, Richards, dan Walter merumuskan serangkaian faktor – faktor yang menunjukan kualitas perangkat lunak antara lain : *Correctness; Reliability Usability*; *Integerity*; *Efficiency*; *Maintainability; Flexibility; Testability; Portability; Reusability; Interoperability*.

b. ISO 9126 quality factors

International Standard Organization (ISO) mengembangkan Standar ISO 9126 yang mengidentifikasikan enam faktor kualitas yang menentukan kualitas suatu perangkat lunak (Pressman, 2010). Faktor – faktor kualitas tersebut antara lain : *Functionality; Reliability; Usability; Efficiency; Maintainability ; Portability.*

Karena keterbatasan, dalam penelitian ini Penulis hanya akan melakukan pengujian pada faktor kualitas *correctness, functionality, portability, usability* dan *efficiency*.

6. Faktor Kualitas *Correctness*

Pressman (2010) menjelaskan bahawa correctness merupakan faktor kualitas yang menunjukan tingkat bagaimana perangkat lunak menjalankan fungsi yang dibutuhkanya. Fakor kualitas *correctness* dapat diukur dengan analisisis *defect per KLOC* (cacat / *error* pada setiap *KLOC/Kilo Line of Code*). Dalam pengembangan perangkat lunak, terjadinya cacat / *error* adalah hal yang wajar, dan hampir semua developer perangkat lunak pasti menemuinya dalam pengembangan perangkat lunak baik itu yang sederhana sampai yang sangat komplesk.

McConnel (McConnell, 2004) menjelaskan bahwa kemungkinan error yang dapat ditemukan dalam sebuah project tergantung pada kualitas pengembangan perangkat lunak yang dilakukan. Semakin baik kualitas pengembangan perangkat lunak, semakin kecil ditemukan error dalam project tersebut. Berikut adalah beberapa rentang kemungkinan *error* tersebut :

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

Basil Vandegriend (2009) dalam tulisanya menyebut *FindBugs* sebagai tools yang cocok digunakan developer Java untuk keperluan analisis kualitas *source code* Java. *FindBugs* merupakan *freeware tools* yang dikembangkan oleh *The University of Maryland*. Dalam website resmi *FindBugs* (2011) dijelaskan bahwa *FindBugs* mengkategorikan jenis bugs menjadi beberapa kategori yaitu : *bad practice*, *correctness*, *multithreaded correctness*, *experimental*, *internationalization*, *malicious code vulnerability*, *performance*, *security* , *dodgy code*. Dalam hal ini, terdapat dua buah kategori *bugs* yang berkaitan dengan analisis faktor kualitas *correctness* yaitu : ***correctness*** dan ***multithreaded correctness***.

7. Faktor Kualitas *Functionality*

Functionality merupakan faktor kualitas yang menunjukan tingkat kemampuan menyediakan fungsi – fungsi yang diharapkan sehingga dapat memberikan kepuasan kepada pengguna (Pressman, 2010). Faktor kualitas *functionality* dapat diuji dengan analisis fungsionalitas dari setiap komponen dari suatu perangkat lunak..

Dalam kaitanya dengan standar yang digunakan untuk menentukan apakah sebuah perangkat lunak lolos dalam pengujian faktor kualitas functionality. James Bach (2005) memberikan gambaran bagaimana suatu perangkat lunak dapat dikatakan memenuhi faktor kualitas *functionality* dalam program *Windows Logo Certification.*

Tabel 1. Keirteria Lolos / Gagal pada program Windows Logo Certification

Kriteria Lolos	**Kriteria Gagal**
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

8. Faktor Kualitas *Portability*

Faktor kualitas *portability* menggambarkan kemampuan perangkat lunak untuk dapat dipindah dan dijalankan di lingkungan yang berbeda, dalam kaitanya dengan penelitian ini adalah pada komputer dengan spesifikasi *hardware* maupun *operating system* yang bebeda – beda (Pressman, 2010).

Bahasa pemrograman java merupakan bahasa dengan keunggulan pada aspek *portability* dan *architecture neutral.* Avestro (2007) menjelaskan bahwa aplikasi yang dibuat dengan *b*ahasa pemrograman java dapat berjalan di berbagai *platform* berbeda, bahkan tanpa perlu adanya proses *recompile,* karena Java bersifat independen dan tidak terikat dengan salah satu *platform*. Untuk menjalankan aplikasi java pada komputer diperlukan *Java Runtime Environment* (JRE). *Java Runtime Environment* tersedia secara gratis berbagai sistem operasi : *Windows*, *Linux*, *Macintosh*, dan *Solaris*. Pengujian faktor kualitas portability dapat dilakukan dengan mencoba aplikasi java pada pada *environtment* yang berbeda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda.

9. Faktor Kualitas *Usability*

Agarwal, Tayal, dan Gupta (2010) menjelaskan bahwa usability merupakan faktor kualitas perangkat lunak yang menunjukan kapabilitas untuk dapat dimengerti, dipahami dan digunakan oleh pengguna.

Standar ISO 9126 mengkategorikan usability sebagai faktor kualitas nonfungsional. Usability berkaitan langsung dengan bagaimana sebuah perangkat lunak digunakan oleh pengguna. Standar ISO 9126 membagi faktor kualitas

usability menjadi beberapa subfaktor yaitu *understandability*, *learnability*, *operability* dan *attractiveness*. (Hass, 2008).

Anne Mette Jonassen Hass menjelaskan (2008) menjelaskan bahwa faktor kualitas usability dapat diuji dengan beberapa metode Survey atau Kuisioner. Metode Survey atau kuisioner digunakan untuk menganalisa faktor kualitas usability dari sisi subjektif pengguna. Pertanyaan – pertanyaan yang digunakan dalam kuisioner harus mencerminkan presepsi pengguna terhadap perangkat lunak yang dikembangkan (Hass, 2008).

III. METODE PENELITIAN

A. Model Penelitian

Jenis *penelitian* yang digunakan di dalam pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini adalah jenis penelitian riset dan pengembangan *(research and development).* Penelitian ini bertujuan untuk mengembangkan suatu produk yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer kemudian melakukan analisis kualitas produk tersebut. Analisis kualitas pada penelitian ini difokuskan pada empat faktor kualitas perangkat lunak yaitu *functionality*, *correctness*, *portability*, dan *usability*.

B. Desain Penelitian

1. Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak akan dilakukan dengan kaidah rekayasa perangkat lunak. Berikut tahap – tahap pengembangan perangkat lunak dalam penelitian ini : **Perencanaan (*planning*); Perancangan (*modeling*); Konstruksi (*construction*); Penyebaran (*deployment*)**

2. Analisis Kualitas Perangkat Lunak

1. Faktor kualitas *correctness* diuji dengan analisis error per kilo line of codes (KLOC), yang akan dibandingkan dengan standard error per kilo line of codes pada industry average dan Microsoft Application.
2. Faktor kualitas functionality diuji dengan pengujian pada setiap fungsi pada aplikasi yang dibuat. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi pada aplikasi berkerja sebagai mana mestinya.

3. Faktor kualitas portability dianalisis dengan melakukan pengujian aplikasi pada beberapa environtment yang berbeda, dalam hal ini komputer dengan system operasi yang berbeda – beda.
4. Faktor kualitas usability dikaji dari penilaian pengguna akhir (end user) yang didapat melalui kuisioner. Kuisioner yang digunakan mengacu pada Computer System Usability Questionnaire yang dipublikasi oleh J.R. Lewis.

C. Subjek Penelitian

Subjek dalam penelitian pada analisis faktor kualitas functionality, correctness, dan portability adalah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer, sedangkan subjek pada analisis faktor kualitas usability adalah 55 siswa kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta sebagai pengguna akhir (*end user*).

D. Instrumen Penelitian

Dalam penelitian ini dibutuhkan beberapa instrument yang digunakan mulai dari proses pengembangan perangkat lunak dan proses analisis kualitas perangkat lunak. Instrumen yang digunakan pada penelitian ini antara lain : *Java Development Kit* (JDK); *Netbeans IDE*; *SQLite Manager* ; *Test case*; *Lines of Code Counter*; *FindBugs*; Beberapa Macam Sistem Operasi : *Windows 7, Windows XP, Ubuntu, OpenSuse, dan Mac OS X Lion*; Kuesioner (mengacu pada *Computer System Usability Questionnaire* yang dipublikasi oleh J.R. Lewis).

E. Teknik Analisis Data

1. Analisis Faktor Kualitas *Functionality*

Pengujian faktor kaulitas functionality dilakukan dengan melakukan tes pada setiap fungsi perangkat lunak. Berkaitan dengan standar yang digunakan dalam menentukan apakah perangkat lunak telah memenuhi syarat faktor kualitas functionality, penulis menggunakan standar functioality yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo*.

Tabel 2. Standar kriteria faktor kualitas *functionality* dalam *Microsoft Certification Logo* (Bach, 2005)

Kriteria Lolos	**Kriteria Gagal**
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

2. Analisis Faktor Kualitas *Correctness*

Faktor kualitas correctness dianalisa dengan menghitung jumlah *error* tiap *kilo lines of code (KLOC)*. Jumlah *lines of code* dapat dihitung menggunakan *Lines of Code Counter, tools* yang dikembangkan oleh John Roshi . Sedangkan jumlah error dalam suatu perangkat lunak, dalam hal ini aplikasi java, dapat dihitung dengan FindBugs. Jumlah *error / KLOC* yang didapatkan dalam pengujian kemudian dibandingkan dengan standar *error / KLOC* pada *industry average* dan standar *Microsoft Application*.

3. Analisis Faktor Kualitas *Portability*

Pengujian faktor kualitas *portability* pada penelitian difokuskan untuk menjawab pertanyaan apakah perangkat lunak yang dikembangkan dapat berjalan sebagaimana mestinya pada sistem yang berbeda – beda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda. Berdasarkan acuan bahwa bahasa pemrograman java merupakan bahasa pemrograman dengan tingkat portability yang baik. Penulis menyusun standar pada perangkat lunak yang dikembangkan untuk mementukan apakah perangkat lunak yang dikembangkan lolos atau gagal dalam pengujian faktor kualitas portability.

Tabel 3. Kriteria lolos / gagal pengujian faktor kualitas portability

Kriteria Lolos	**Kriteria Gagal**
Perangkat lunak dapat berjalan sebagaimana mestinya pada setiap sistem yang diujikan.	Paling tidak ada satu sistem dimana perangkat lunak tidak dapat berjalan sebagaimana mestinya.

4. Analisis Faktor Kualitas *Usability*

Pengujian faktor kualitas usability dilakukan dengan menggunakan metode kuesioner. Data yang dihasilkan dari kuesioner tersebut merupakan gambaran pendapat atau presepsi pengguna perangkat lunak, dalam hal ini yang berkaitan dengan faktor kualitas *usability* perangkat lunak yang dikembangkan. Data yang dihasilkan dari kuisioner meruapakan data yang bersifat kuantitatif. Data tersebut dapat dikonversi ke dalam data kualitatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Menurut Sugiyono (2009), Skala Likert digunakan untuk mengukur sikap, pendapat atau presepsi seseorang atau kelompok terhadap sesuatu, dalam hal pendapat pengguna terhadap perangkat lunak yang dikembangkan. Dalam kaitanya dengan kuesioner yang digunakan yaitu, *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis, terdapat 5 macam jawaban dalam setiap item keusioner. Data tersebut diberi skor sebagai berikut :

Tabel 4. Konversi jawaban item kuesioner ke dalam nilai kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Jumlah nilai yang didapat dari hasil konversi jawaban kuisioner ke dalam nilai kuantitatif kemudian dibandingkan dengan kategorisasi penilaian tersebut.

IV. PEMBAHASAN

A. Pengembangan Perangkat Lunak

Salah satu tujuan dari penelitian ini adalah untuk mengembangkan produk perangkat lunak yaitu Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Proses pengembangan perangkat lunak dalam penelitian ini didasarkan pada kaidah rekayasa perangkat lunak (*software engineering*). Proses pengembangan perangkat lunak dalam penelitian ini dibagi menjadi beberapa bagian yaitu : perencanaan (*planning*), perancangan (*modeling*), konstruksi (*construction*), dan penyebaran (*deployment*).

1. Perencanaan (*Planning*)

Produk yang dikembangkan dalam penelitian ini adalah Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Aplikasi ini akan dibuat dengan bahasa pemrograman Java dan dirancang untuk dijalankan pada komputer dengan sistem operasi yang dapat mendukung *Java Runtime Environment*.

Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang dikembangkan menpunya dua jenis mode berdasarkan user yang menggunakan, antara lain :

1. Normal Mode : Normal mode adalah mode awal ketika aplikasi dijalankan, dan tidak memerlukan login untuk menggunakanya.
2. Admin Mode : Mode ini adalah mode administrator yang membutuhkan login untuk dapat menggunakanya. Mode admin dibagi menjadi dua macam : Basic Admin Mode & Advance Admin Mode.

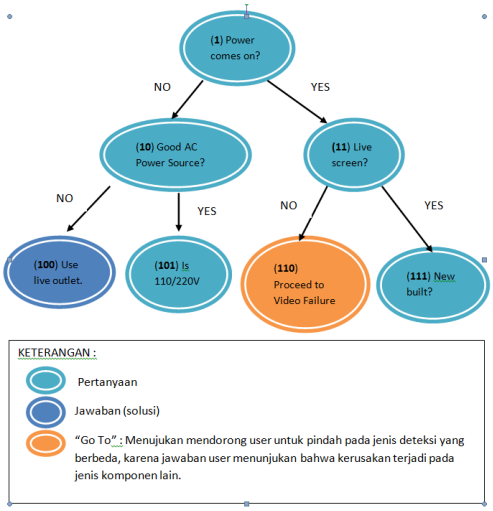
2. Perancangan (*Modeling*)

Proses perancangan (modeling) aplikasi Sistem Pakar Diagnosis Komputer meliputi perancangan *knowledge base*, *database*, *UML* (*Unified Modeling Language*), dan *user inteface* (antar muka).

a. Perancangan *knowledge base* dan database

Knowlegde-base (Basis Pengatahuan) yang digunakan dalam Aplikasi Sistem Pakar Pendiagnosis Kerusakan Komputer berdasar pada Buku karangan Mossis Rosenthal yang berjudul *Computer Failure with Diagnostic Flowchart*. Dalam bukunya tersebut Morris Rosenthal (2010) menjelaskan 17 macam diagnosis kerusakan komputer. Secara umum isi dalam buku tersebut lebih difokuskan untuk diagnosis pada PC (*Personal Computer*) dengan *form factor* ATX (jenis umum pada mayoritas PC pertengahan 1990 sampai sekarang) (Rosenthal, 2010). Karena berbagai keterbatasan, Penulis hanya menggunakan 7 jenis diagnosis yaitu : Kerusakan Power Supply; Kerusakan Vidio Adapter (VGA); Masalah Kinerja Vidio Adapter (VGA); Kerusakan Motherboard, Processor, dan RAM; Masalah Kinerja Motherboard, Processor, dan RAM; Kerusakan ATA / SATA Hard Drive; Masalah Booting & Kinerja Hard Disk

Berikut adalah gambar Struktur Knowledge-Base Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



Gambar 1. Struktur *Knowledge Base*

Setiap data pada *knowledge base* dikodekan untuk memudahkan dalam menampilkanya pada aplikasi. Data pertama pada suatu tabel diagnosis dimulai dengan kode "**1**", kemudian kode pada data sesudahnya ditambah karakter "**0**" atau "**1**", karakter "**0**" menujukan bahwa pertanyan sebelumnya dijawab "*TIDAK*" dan karakter "**1**" menunjukan bahwa pertanyaan sebelumnya dijawab "***YA***". Selain itu terdapat juga data berupa detail dari pertanyaan atau jawaban yang berfungsi untuk memberi penjelasan lebih lanjut.Dalam aplikasi sistem pakar ini, terdapat lebih dari satu macam jenis diagnosis. Tiap diagnosis dan data – datanya akan disimpan dalam tabel. Untuk itu dalam desain database yang akan digunakan terdapat dua jenis tabel yaitu :

1) *table index (table_index).*

Dalam tabel ini disimpan data tentang jenis diagnosis yang terdapat pada sistem pakar. Field *table_code* menunjukan nama tabel dari suatu diagnosis. Sedangkan Field table_title menunjukan nama (judul) suatu diagnosis.

Tabel 5. Struktur Tabel Index

Field	Type	Null	Key	Default
table_code	CHAR	No	UNIQUE	-
table_title	VARCHAR	No	-	-

2) *tabel diagnosis.*

Tabel jenis ini, adalah tabel dimana data – data (pertanyaan, jawaban, dan go_to) dalam suatu diagnosis disimpan. Data berupa pertanyaan disimbolkan

dengan karakter "**Q**", data berupa jawaban disimbolkan dengan karakter "**A**", dan data berupa go_to disimbolkan dengan "**G.[table_code]**" (contoh : *G.psu_failure*).

Satu diagnosis menggunakan satu buah tabel. Jadi jumlah tabel jenis ini adalah sama dengan jumlah diagnosis yang tersedia dalam sistem.

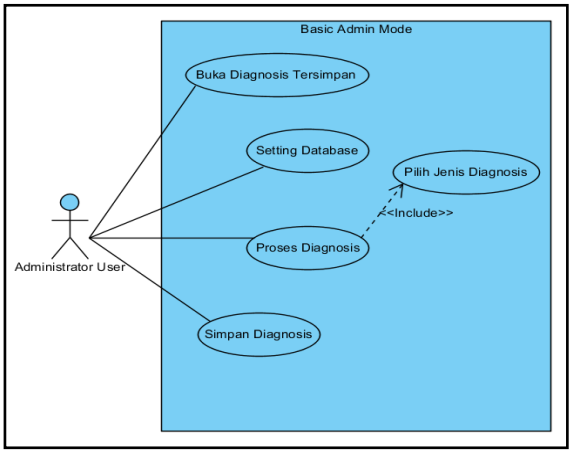
Tabel 6. Struktur Tabel Diagnosis

Field	Type	Null	Key	Default
code	VARCHAR	No	UNIQUE	-
type	VARCHAR	No	-	-
content	TEXT	No	-	-
detail	TEXT	No	-	-

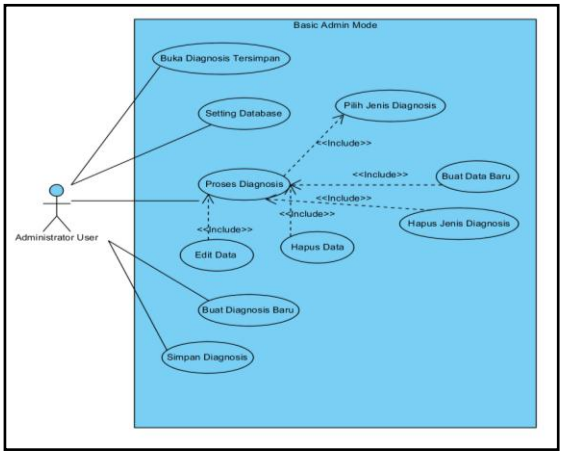
b. Perancangan UML (unified modeling language)

Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar dalam industri untuk visualisasi dan dokumentasi sistem perangkat lunak (software). UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti, 2006). Rancangan sebuah perangkat lunak didefinisikan dalam diagram – diagram dalam UML yaitu : *use case diagram* , *class diagram*, *statechart diagram, activity diagram*, *sequence diagram, collaboration diagram*, *component diagram,* dan *deployment diagram*.

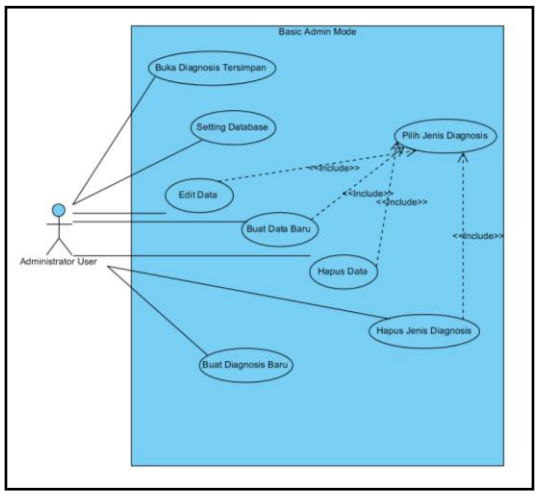
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem (Dharwiyanti, 2006). Use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dibagi menjadi tiga macam yaitu pada *Normal Mode*, *Basic Admin Mode*, dan *Advanced Admin Mode*. Berikut ini merupakan use case diagram Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :



Gambar 2. Use Case : Normal Mode



Gambar 3. Use Case : Basic Admin Mode



Gambar 4. Use Case : Advanced Admin Mode

3. Konstruksi (*Construction*)

Tahap ini merupakan impelentasi dari rancangan yang telah dibuat. Secara umum pada tahap ini dilakukan entry data knowledge base pada database *SQLite* dan kemudian pembuatan aplikasi dengan bahasa pemrograman Java menggunakan *Netbeans IDE*. Pembuatan aplikasi secara umum terbagi menjadi dua proses utama yaitu desain antar muka (*graphical user interface*) dan pengkodean (*coding*).

a. Entry data knowledge-base pada database sqlite

b. Desain Antar Muka (Graphical User Interface)

Proses desain antar muka (*graphical user interface*) Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan bantuan *Netbeans Swing GUI Builder*. Hal itu dilakukan untuk memudahkan proses desain karena sebagian besar komponen yang digunakan dalam antar muka Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah komponen *Swing* (*javax.swing.*)*.

c. Penulisan Program (*Coding*)

Setelah proses desain antar muka (*graphical user interface*) selesai, proses selanutnya adalah proses penulisan program dalam bahasa pemrogram Java. Proses penulisan program Aplikasi Diagnosis Kerusakan Komputer dilakukan dengan Netbeans IDE.

Dalam pemrogram berorientasi objek, hal utama yang perlu diperhatikan dalam coding adalah bagaimana rancangan class dan object yang digunakan. Berikut merupakan gambaran hierarki *package* dan *class* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer yang disajikan dalam bentuk tabel :

Tabel 7. Hierarki Package & Class Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer

Package / Class	Deskripsi
dkk	*Package* utama Aplikasi SIstem Pakar Diagnosis Kerusakan Komputer (DKK).
Main	*Class* utama yang akan dipanggil ketika program dijalankan. Kemudian *class* ini akan memanggil class *NormalMode*.
Connect2Sqlite	*Class* untuk koneksi dengan database (Sqlite). Class ini berisi *method – method* untuk membaca dan memanipulasi

	data pada database. *Class* ini membutuhkan *library sqlitejdbc-v056*.
Normal	*Class* yang merupakan *subclass* dari *Connect2Sqlite*. Class ini berisi *method – method* yang digunakan pada *Normal Mode*.
Admin	*Class* yang merupakan *subclass* dari *Connect2Sqlite*. *Class* ini berisi method – metod yang digunakan pada Basic Admin Mode dan *Advanced Admin Mode*.
dkk.gui	*Package* yang berisi *class – class* yang menampilkan antar muka (*user interface*).
NormalMode	Antar muka *Normal Mode*.
BasicAdminMode	Antar muka *Basic Admin Mode*.
AdvancedAdminMode	Antar muka *Advanced Admin Mode*.
Help	Antar Muka halaman *Help* (Panduan)
About	Antar muka halaman *About* (Tentang Aplikasi)
dkk.image	Package ini berisi file – file berupa gambar (images) yang digunakan dalam antar muka.

d. hasil pengembangan aplikasi.

Setelah proses desain antar muka dan coding selesai, Aplikasi Sistem Pakar Diagnosis Kerusakan sudah bisa untuk dijalankan secara *executable* melalui file distribusi **.jar*. Berikut adalah beberapa screenshoot Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer :

Gambar 5. Screenshoot Aplikasi

4. Penyebaran (*Deployment*)

Penyebaran dilakukan secara *offline* dan *online*. Penyebaran secara offline dilakukan di SMK Muhammadiyah 2 Yogyakaryta, yakni kepada Siswa Kelas XI Jurusan Teknik Komputer dan Jaringan (TKJ) yang selanjutnya akan menjadi subjek penelitian pada pengujian faktor kualitas *usability* aplikasi sistem pakar tersebut. Penyebaran secara *online* dilakukan dengan mengunggah (*upload*) aplikasi pada web server dan membuat halaman khusus di internet.

B. Analisis Kualitas Perangkat Lunak

1. Analisis Faktor Kualitas *Correctness*

Faktor kualitas *correctness* dapat diukur dengan analisisis *defect per KLOC* (cacat / *error* pada setiap *KLOC/Kilo Line of Code*). Untuk mendapatkan nilai error/*KLOC*, diperlukan penghitungan jumlah *Kilo Lines of Code* (*KLOC*),

kemudian dilakukan perhitungan jumlah *error* pada *source code* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan *FindBugs*.

a. Menghitung Jumlah Lines Of Code (LOC)

Hasil perhitungan *lines of code* (*LOC*) *source code* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah **5540** ***LOC*** = **5.54** ***KLOC*** (***Kilo lines of code***).

b. Penghitungaan jumlah *error*

Hasil perhitungan error menggunakan FindBugs menunjukan bahwa jumlah error untuk pengujian fakor kualitas correctness Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer menggunakan *FindBugs* adalah *0*.

c. Perbandingan hasil pengujian dengan standard yang telah ditentukan

Dari hasil pengujian sebelumnya, didapatkan bahwa nilai *Error/KLOC* Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer adalah 0 *Error/KLOC*. Aplikasi akan dikatakan **LOLOS** pengujian jika jumlah error <= (lebih sedikit atau sama dengan) standar yang digunakan. Sebaliknya, aplikasi akan dikatatakan **GAGAL** jika jumlah error melebihi standar yang digunakan.

Tabel 8. Perbandingan Hasil Pengujian Faktor Kualitas Correctnes dengan Standar yang Digunakan

Nama Standar	Nilai Standar (*Error* / KLOC)	Hasil Pengujian Aplikasi	Keterangan
Industry Average	1 – 25	$\frac{0}{5.54} = 0$	**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.
Microsoft Application	0.5		**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.

Tabel tersebut menujukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas correctness baik dari Standar *Industry Average* maupun *Microsoft Application*.

2. Analisis Faktor Kualitas Functionality

Faktor kualitas *functionality* diuji dengan melakukan tes pada setiap fungsi yang terdapat pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer. Hasil

pengujian kemudian dibandingkan dengan standar *functioality* yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo.*

Tabel 9. Perbandingan Hasil Pengujian Faktor Kualitas Functionality dengan

Kriteria Lolos	Kriteria Gagal	Hasil Pengujian	Keterangan
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.	Semua fungsi primer dan fungsi pendukung berjalan dengan baik.	**Lolos**

Tabel tersebut menunjukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer **lolos** pengujian faktor kualitas *functionality*, atau dapat dikatakan bahwa aplikasi yang dikembangkan memenuhi faktor kualitas *functionality*.

3. Analisis Faktor Kualitas Portability

Pengujian faktor kualitas usability dilakukan dengan melakukan pengujian aplikasi pada beberapa sistem operasi yang berbeda. Berikut adalah rangkuman *test case* (test case lengkap dilampirkan) dan screenshoot pengujian faktor kualitas usability Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer

Tabel 10. Rangkuman *Test Case* Faktor Kualitas *Usability*

No	Sistem Operasi	Hasil Pengujian	Lolos / Gagal
1	Windows XP	Aplikasi berjalan dengan baik.	Lolos
2	Windows 7	Aplikasi berjalan dengan baik.	Lolos
3	Linux : Ubuntu	Aplikasi berjalan dengan baik.	Lolos
4	Linux : OpenSuse	Aplikasi berjalan dengan baik.	Lolos
5	Mac OS X Snow Leopard	Aplikasi berjalan dengan baik.	Lolos
6	Mac OS X Lion	Aplikasi berjalan dengan baik.	Lolos

Tabel tersebut menunjukan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dapat berjalan dengan baik pada setiap sistem operasi yang diujikan sehingga dapat diambil kesimpulan bahwa Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas *portability*.

4. Analisis Faktor Kualitas Usability

Pengujian faktor kualitas usability dilakukan dengan menggunakan metode kuesioner. Kuesioner yang digunakan mengacu pada *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis . Kuesioner diberikan kepada siswa kelas XI TKJ SMK Muhammadiyah 2 Yogyakarta sebanyak 55 siswa. Berikut adalah tabel jawaban responden terhadap tiap pertanyaan dalam kuesioner.

Tabel 11. Tabel Jawaban Responden Terhadap Pertanyaan Kuesioner *Usability*

Pertanyaan	Jawaban Responden				
	Sangat Setuju	Setuju	Ragu - Ragu	Tidak Setuju	Sangat tidak setuju
Pertanyaan 1	19	35	1	0	0
Pertanyaan 2	23	31	1	0	0
Pertanyaan 3	4	43	8	0	0
Pertanyaan 4	8	22	23	2	0
Pertanyaan 5	5	41	9	0	0
Pertanyaan 6	16	34	4	1	0
Pertanyaan 7	13	34	8	0	0
Pertanyaan 8	8	34	13	0	0
Pertanyaan 9	10	36	9	0	0
Pertanyaan 10	17	33	5	0	0
Pertanyaan 11	6	43	5	1	0
Pertanyaan 12	12	35	8	0	0
Pertanyaan 13	12	39	3	0	0
Pertanyaan 14	7	42	6	0	0
Pertanyaan 15	8	39	8	0	0
Pertanyaan 16	4	38	12	0	1
Pertanyaan 17	3	44	7	1	0
Pertanyaan 18	3	34	18	0	0
Pertanyaan 19	10	40	4	1	0
Jumlah	**188**	**697**	**152**	**6**	**1**

Data yang dihasilkan dari kuisioner meruapakan data yang bersifat kuantitatif. Data tersebut dapat dikonversi ke dalam data kualitatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Tabel 12. Konversi Jawaban Item Kuesioner menjadi Nilai Kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Berikut perhitungan jumlah skor yang didapat dari hasil kuesioner :

- Jumlah jawaban "Sangat Setuju" = 189 x 5 = 945
- Jumlah jawaban "Setuju" = 701 x 4 = 2804
- Jumlah jawaban "Ragu - Ragu" = 148 x 3 = 444
- Jumlah jawaban "Tidak Setuju" = 7 x 2 = 14
- Jumlah jawaban "Sangat Tidak Setuju" = 0 x 1 = 0
- **Jumlah Total = 4203**

Skor yang didapatkan pada tiap hasil kuesioner tersebut kemudian diambil nilai rata - rata. Nilai rata – rata tersebut kemudian dijumlahkan.

Dengan jumlah responden sebanyak 55 orang maka dapat dihitung nilai tertinggi dan nilai terendah sebagai berikut :

1) Nilai tertinggi = 55 x 19 x 5 = **5225** , dengan asumsi semua responden memberi jawaban sangat setuju pada setiap item kuesioner.
2) Nilai terendah = 55 x 19 x 1 = **1045**, dengan asumsi semua responden memberi jawaban "sangat tidak setuju" pada setiap item kuesioner.

Dari data tersbut data tersebut, kemudian dapat disusun kategori penilaian kuesioner berdasarkan perhitungan interval kelas.

Tabel 13 Kategori Penilaian Faktor Kualitas *Usability*

Interval Nilai	Ketegori
1045 - 1877	Sangat Buruk
1878- 2714	Buruk
2715- 3551	Cukup
3552- 4388	Baik
4389- 5225	Sangat Baik

Tabel tersebut menujukan bahwa hasil pengujian kuisioner berada dalam interval 3552 – 4338 dan termasuk dalam kategori **Baik**.

V. KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan sebelumnya dapat diambil beberapa kesimpulan sebagai berikut :

1. Pengembangan Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer dilakukan dengan kaidah rekayasa perangkat lunak (software engineering) yaitu dimulai dari proses perencanaan (planning), perancangan (modeling), konstruksi (construction), dan penyebaran (deployment).
2. Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas correctness.
3. Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas functionality.
4. Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas portability.
5. Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer memenuhi standar faktor kualitas usability.

DAFTAR PUSTAKA

Bach, J. (2005). *General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo.* Dipetik Maret 3, 2012, dari Satisfice, Inc: http://www.satisfice.com/tools/procedure.pdf

Agarwal, B. B., Tayal, S. P., & Gupta, M. (2010). *Software Engineering and Testing.* Sudbury: Jones and Bartlett Publishers.

Allen, G., & Owens, M. (2010). *The Definitive Guide to SQLite.* New York: Appress.

Avestro, J. (2007). *Pengenalan Pemrograman Java 1.* (F. Thamura, E. Subiyantoro, C. K. Ratih, R. N.S., & M. K. Mufida, Penerj.) Jakarta: J.E.N.I.

Dharwiyanti, S. (2006, 8 25). *Pengantar Unified Modeling Language (UML).* Dipetik 6 17, 2012, dari IlmuKomputer.Com: http://ilmukomputer.org/2006/08/25/pengantar-uml/

Hartati, S., & Iswanti, S. (2008). *Sistem Pakar & Pengembangannya.* Yogyakarta: Graha Ilmu.

Hass, A. M. (2008). *Guide to Advanced Software Testing.* Norwood: Artech House.

McConnell, S. C. (2004). *Code Complete.* Redmond : Microsoft Press.

Pressman, R. (2010). *Software Engineering : A Practitioner's Approach* (7 ed.). New York: McGraw Hill.

Pugh, B. (2011). *FindBugs™.* Dipetik Maret 22, 2012, dari FindBugs Bug Descriptions: http://findbugs.sourceforge.net/bugDescriptions.html

Rosenthal, M. (2010). *Computer Rapair and Diagnostic Flowchart.* Massachusetts: Foner Books.

Sugiyono. (2009). *Metode Penelitian Pendidikan.* Bandung: Alfabeta.

Sutojo, T., Mulyanto, E., & Suhartono, V. (2011). *Kecerdasan Buatan.* Yogyakarta: Andi.

Vandegriend, B. (2009). *Why You Should Be Using FindBugs.* Dipetik Maret 19, 2012, dari Professional Software Development: http://www.basilv.com/psd/blog/2009/why-you-should-be-using-findbugs

Worldometers. (2011, September 8). *Computers sold in the world this year.* Dipetik Januari 13, 2012, dari Worldometers: Worldometers