



TUGAS AKHIR - KI141502

**RANCANG BANGUN KAKAS BANTU
PENGUKURAN KUALITAS PERANGKAT LUNAK
MENGUNAKAN METODE FUZZY**

**MOHAMMAD APRIALDI RIZKY PRATAMA
NRP 5111 100 071**

**Dosen Pembimbing I
Dr. Ir. Siti Rochimah, M.T.**

**Dosen Pembimbing II
Rizky Januar Akbar, S.Kom., M.Eng.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



UNDERGRADUATE THESIS - KI141502

**DESIGN AND IMPLEMENTATION OF A
SOFTWARE QUALITY MEASUREMENT TOOL
USING FUZZY METHOD**

**MOHAMMAD APRIALDI RIZKY PRATAMA
NRP 5111 100 071**

**Supervisor I
Dr. Ir. Siti Rochimah, M.T.**

**Supervisor II
Rizky Januar Akbar, S.Kom., M.Eng.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015**

LEMBAR PENGESAHAN

RANCANG BANGUN KAKAS BANTU PENGUKURAN KUALITAS PERANGKAT LUNAK MENGGUNAKAN METODE FUZZY

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

MOHAMMAD APRIALDI RIZKY PRATAMA
NRP. 5111 100 071

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Ir. Siti Rochimah., M.T.
NIP: 19681002 199403 2 001 (Pembimbing 1)
2. Rizky Januar Akbar, S.Kom., M.Eng.
NIP: 19870103 201404 1 001 (Pembimbing 2)

SURABAYA
JULI, 2015

RANCANG BANGUN KAKAS BANTU PENGUKURAN KUALITAS PERANGKAT LUNAK MENGGUNAKAN METODE FUZZY

Nama : Mohammad Aprialdi Rizky Pratama
NRP : 5111100071
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Dr. Ir. Siti Rochimah, M.T.
Dosen Pembimbing II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstrak

Perangkat lunak merupakan salah satu bentuk produk yang dihasilkan melalui serangkaian proses yang bersifat abstrak sehingga membuat sebagian orang mengalami kesulitan dalam menentukan kualitasnya terutama bagi pengembang perangkat lunak tersebut. Padahal, kualitas perangkat lunak sering berdampak cukup besar bagi tim pengembang tersebut. Terlebih lagi, kualitas bersifat subjektif sehingga sulit untuk dinyatakan dalam bentuk angka.

Proses penghitungan kualitas perangkat lunak membutuhkan proses yang kompleks. Dalam tugas akhir ini, model kualitas ISO/IEC 9126 yang merupakan standar baku kualitas perangkat lunak. Model kualitas ini terdiri dari beberapa aspek dalam terbagi dalam beberapa karakteristik. Proses ini juga menggunakan metode fuzzy untuk menguantifikasikan nilai kualitas yang bersifat subjektif. Fuzzy juga digunakan untuk menghitung nilai kualitas dengan menggunakan model kualitas tersebut.

Untuk mempermudah proses pengukuran kualitas perangkat lunak, dibuatlah aplikasi kakas bantu yang memanfaatkan kedua metode di atas. Kakas bantu dirancang dengan menambahkan beberapa fungsionalitas yang dapat membantu proses pengukuran kualitas. Proses penghitungan nilai

kualitas menggunakan metode fuzzy dilakukan dengan mengelompokkan nilai aspek berdasarkan kriteria tertentu, melakukan fuzzifikasi yang merupakan proses konversi nilai kategori menjadi nilai fuzzysets, menghitung nilai aspek, dan defuzzifikasi untuk mendapatkan nilai kualitas yang sebenarnya.

Hasil uji coba perangkat lunak menunjukkan bahwa perangkat lunak telah memenuhi fungsionalitas yang dibutuhkan. Nilai yang dihasilkan oleh kakas bantu terbukti valid dan nilai kebergunaan aplikasi mencapai 0,81. Hal ini membuktikan bahwa aplikasi bisa digunakan tim pengembang dalam mengukur kualitas perangkat lunak. Penggunaan metode fuzzy terbukti memberikan kemudahan dalam proses penilaian kualitas perangkat lunak dengan data kualitatif.

Kata kunci: *Abstract Syntax Tree, Fuzzy, Java, ISO/IEC 9126, Kualitas Perangkat Lunak*

DESIGN AND IMPLEMENTATION OF A SOFTWARE QUALITY MEASUREMENT TOOL USING FUZZY METHOD

Name : Mohammad Aprialdi Rizky Pratama
NRP : 5111100071
Department : Informatics, FTIf, ITS
Supervisor I : Dr. Ir. Siti Rochimah, M.T.
Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstract

Software is a form of products produced by a series of processes. One particular characteristic of software is that the software is intangible, which makes several people find it hard to assess the quality of software, especially the developer itself. However, the quality of software makes a huge impact for the developer sometimes. Furthermore quality is a subjective aspect, so it's difficult to be declared with a crisp value.

Process of measuring quality of software requires a quite complex process. This thesis uses ISO/IEC 9126 quality model which is the international standard for quality of software. This process also uses fuzzy method for quantifying the value of software quality that somewhat seems very subjective, depends on the people. The method is also used to compute the quality value of software by applying the quality model, fuzzy multiplication and fuzzy addition.

In order to ease the software quality measurement process, this application for software quality measurement developed adopting model quality ISO/IEC 9126 and fuzzy method. The application designed to carry on several functionalities that help the developer to assess the quality of software. The process of calculating the software quality with the use of fuzzy method is done by applying some steps, grouping aspect's value based on

particular criteria then converting this value into fuzzysets, fuzzification, then computing aspect's value, and lastly defuzzifying the fuzzyset to get the crisp value of software quality.

This software has been tested and show that software has met the functionalities required in the design phase. Validity of the value computed by this tools also has been tested and complied with the value resulted from manual analysis and software has satisfied the required usability with a score 0.81. It proves that this developed application can be used for measuring the quality of a software. The usage of fuzzy method give an evidence that it simplify the process of software quality measurement with qualitative data.

Keywords: Abstract Syntax Tree, Fuzzy, ISO/IEC 9126, Java, Software Quality.

KATA PENGANTAR

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga tugas akhir berjudul “Rancang Bangun Kakas Bantu Pengukuran Kualitas Perangkat Lunak Menggunakan Metode Fuzzy” ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Keluarga penulis, Mama Nenah Hunaenah, Papa Riswandy, Kak Risna, Kak Rini, Mas Trie, Mas Adi, dan keponakan-keponakan, Ayska, Mila, Alya yang telah memberikan dukungan moral dan material serta doa yang tak terhingga untuk penulis.
2. Ibu Siti Rohimah selaku dosen pembimbing 1 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Rizky Januar Akbar selaku dosen pembimbing 2 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu yang tidak ternilai kepada penulis.
5. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis
6. Temen-teman terdekat penulis, Eko Adhi Wiyono, Faikha Rizqi Aziza, Astarti Larasati, yang sudah memberikan

semangat dan mau menjadi tempat untuk berkeluh kesah bagi penulis.

7. Teman-teman angkatan 2011 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
8. Serta semua pihak yang turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2015

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER.....	xxvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan	3
1.5. Manfaat.....	4
1.6. Metodologi	4
1.7. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA	9
2.1 ISO/IEC 9126 – Product Quality.....	9
2.2 Fuzzy Multi Kriteria.....	19
2.3 Abstract Syntax Tree.....	20
2.4 Pola Visitor.....	24
2.5 Eclipse	26
2.6 Kompleksitas Siklomatik	27
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	29

3.1. Analisis.....	29
3.1.1. Analisis Permasalahan.....	29
3.1.2. Aktor.....	30
3.1.3. Kasus Penggunaan.....	30
3.1.4. Spesifikasi Kebutuhan Perangkat Lunak.....	45
3.2. Perancangan Sistem.....	47
3.2.1. Perancangan Basis Data.....	48
3.2.2. Perancangan Diagram Kelas.....	53
3.2.3. Perancangan Pengukuran Kualitas.....	57
3.2.4. Perancangan Antarmuka Pengguna.....	65
BAB IV IMPLEMENTASI.....	91
4.1 Lingkungan Implementasi.....	91
4.2 Implementasi.....	93
BAB V PENGUJIAN DAN EVALUASI.....	139
5.1 Lingkungan Uji Coba.....	139
5.2 Skenario Uji Coba.....	139
5.2.1 Pengujian Fungsionalitas.....	140
5.2.2 Pengujian Validitas.....	159
5.2.3 Pengujian Kebergunaan.....	161
5.3 Evaluasi Pengujian.....	161
5.3.1 Evaluasi Pengujian Fungsionalitas.....	161
5.3.2 Evaluasi Pengujian Validitas.....	162
5.3.3 Evaluasi Pengujian Kebergunaan.....	163
BAB VI KESIMPULAN DAN SARAN.....	167
6.1 Kesimpulan.....	167
6.2 Saran.....	168

DAFTAR PUSTAKA.....	169
LAMPIRAN A	171
LAMPIRAN B	183
LAMPIRAN C	195
LAMPIRAN D	207
LAMPIRAN E.....	209
LAMPIRAN F	213
LAMPIRAN G	223
BIODATA PENULIS.....	239

DAFTAR TABEL

Tabel 2. 1 Metrik untuk ISO/IEC 9126	11
Tabel 2. 2 Contoh Fuzifikasi untuk Kompleksitas Siklomatik....	19
Tabel 2. 3 <i>Statement</i> Penambahan Kompleksitas Siklomatik dalam Bahasa Pemrograman Java	27
Tabel 3. 1 Daftar Kode Diagram Kasus Penggunaan.....	31
Tabel 3. 2 Spesifikasi Kasus Penggunaan Mengunggah Kode Sumber.....	32
Tabel 3. 3 Spesifikasi Kasus Penggunaan Mengisi Keterangan Terkait Perangkat Lunak	35
Tabel 3. 4 Spesifikasi Kasus Penggunaan Melihat Daftar Perangkat Lunak.....	38
Tabel 3. 5 Spesifikasi Kasus Penggunaan Mengisi Kuesioner Subjektif	39
Tabel 3. 6 Spesifikasi Kasus Penggunaan Melihat Detail Penilaian Kualitas Perangkat Lunak.....	42
Tabel 3. 7 Spesifikasi Kasus Penggunaan Melihat Detail Penilaian Kualitas Perangkat Lunak.....	43
Tabel 3. 8 Daftar Kebutuhan Fungsional Perangkat Lunak	47
Tabel 3. 9 Spesifikasi Basis Data	51
Tabel 3. 10 Nilai dan Rumus yang Digunakan untuk Metrik yang Didapatkan Diluar Kode Sumber	59
Tabel 3. 11 Cara Ekstraksi Kode Sumber.....	63
Tabel 3. 12 <i>Fuzzy</i> Traingular untuk Pemingkatan Metrik	64
Tabel 3. 13 <i>Fuzzy</i> Traingular untuk Bobot Metrik	64
Tabel 3. 14 Spesifikasi Atribut Antar Muka Tampilan Utama....	66
Tabel 3. 15 Spesifikasi Atribut Antar Muka Halaman Mengunggah Perangkat Lunak.....	68
Tabel 3. 16 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Rating Numerik	70
Tabel 3. 17 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Rating Pilihan Ganda.....	72
Tabel 3. 18 Spesifikasi Atribut Antar Muka Tampilan Halaman Daftar Perangkat Lunak.....	74

Tabel 3. 19 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Rating Subjektif	76
Tabel 3. 20 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Fungsionalitas	78
Tabel 3. 21 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Efisiensi	80
Tabel 3. 22 Spesifikasi Atribut Antar Muka Halaman Pertanyaan Bobot Keterawatan	82
Tabel 3. 23 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Keterawatan 2	85
Tabel 3. 24 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Portabilitas	87
Tabel 3. 25 Spesifikasi Atribut Antar Muka Halaman Menampilkan Detail Penilaian Perangkat Lunak	89
Tabel 4. 1 Tabel Implementasi Pola Visitor untuk Menghitung Kompleksitas Siklomatik.....	126
Tabel 5. 1 Pengujian Fitur Mengunggah Perangkat Lunak Skenario 1	140
Tabel 5. 2 Pengujian Fitur Mengunggah Perangkat Lunak Skenario 2	142
Tabel 5. 3 Pengujian Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1.....	143
Tabel 5. 4 Pengujian Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 2.....	146
Tabel 5. 5 Pengujian Fitur Melihat Daftar Perangkat Lunak.....	148
Tabel 5. 6 Pengujian Fitur Mengisi Kuesioner Subjektif	150
Tabel 5. 7 Pengujian Fitur Melihat Detail Penilaian Perangkat Lunak.....	155
Tabel 5. 8 Pengujian Fitur Menampilkan Laporan Penilaian Perangkat Lunak.....	157
Tabel 5. 9 Hasil Penghitungan Kualitas Perangkat Lunak Menggunakan Kakas Bantu.....	160
Tabel 5. 10 Hasil Ekstraksi Kode Sumber Menggunakan Kakas Bantu	161
Tabel 5. 11 Rangkuman Hasil Pengujian Fungsionalitas	162

Tabel 5. 12 Hasil Pengujian Validitas	162
Tabel 5. 13 Hasil Ekstraksi Kode Sumber Secara Manual	163
Tabel 5. 14 Rangkuman Hasil Pengujian Kebergunaan	164
Tabel A. 1 Hasil Ekstraksi Kode Sumber	171
Tabel A. 2 Isian Data Keterangan Perangkat Lunak	171
Tabel A. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas .	173
Tabel A. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas.....	173
Tabel A. 5 Data Kasus Uji Coba Karakteristik Efisiensi.....	174
Tabel A. 6 Hasil Penghitungan Manual Karakteristik Efisiensi	174
Tabel A. 7 Data Kasus Uji Coba Submetrik Kemampuan Tim.	175
Tabel A. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim	175
Tabel A. 9 Data Kasus Uji Coba Metrik <i>Skill</i>	176
Tabel A. 10 Hasil Penghitungan Manual Metrik <i>Skill</i>	176
Tabel A. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer	176
Tabel A. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer.....	176
Tabel A. 13 Data Kasus Uji Coba Karakteristik <i>Maintainability</i>	177
Tabel A. 14 Hasil Penghitungan Manual Karakteristik <i>Maintainability</i>	178
Tabel A. 15 Data Kasus Uji Coba Karakteristik Portabilitas	179
Tabel A. 16 Hasil Penghitungan Manual Karakteristik Portabilitas	179
Tabel A. 17 Data Kasus Uji Coba Karakteristik	179
Tabel A. 18 Hasil Penghitungan Manual Karakteristik.....	180
Tabel A. 19 Data Kasus Uji Coba Nilai Kualitas	182
Tabel A. 20 Hasil Penghitungan Manual Nilai Kualitas	182
Tabel A. 21 Hasil Penghitungan Kualitas Perangkat Lunak	182
Tabel B. 1 Hasil Ekstraksi Kode Sumber.....	183
Tabel B. 2 Isian Data Keterangan Perangkat Lunak.....	183
Tabel B. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas .	185

Tabel B. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas.....	185
Tabel B. 5 Data Kasus Uji Coba Karakteristik Efisiensi.....	186
Tabel B. 6 Hasil Penghitungan Manual Karakteristik Efisiensi	186
Tabel B. 7 Data Kasus Uji Coba Submetrik Kemampuan tim ..	187
Tabel B. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim	187
Tabel B. 9 Data Kasus Uji Coba Metrik Kemampuan	188
Tabel B. 10 Hasil Penghitungan Manual Metrik Kemampuan..	188
Tabel B. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer	188
Tabel B. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer.....	188
Tabel B. 13 Data Kasus Uji Coba Karakteristik <i>Maintainability</i>	189
Tabel B. 14 Hasil Penghitungan Manual Karakteristik <i>Maintainability</i>	190
Tabel B. 15 Data Kasus Uji Coba Karakteristik Portabilitas.....	191
Tabel B. 16 Hasil Penghitungan Manual Karakteristik Portabilitas	191
Tabel B. 17 Data Kasus Uji Coba Karakteristik.....	191
Tabel B. 18 Hasil Penghitungan Manual Karakteristik Portabilitas	192
Tabel B. 19 Data Kasus Uji Coba Nilai Kualitas	194
Tabel B. 20 Hasil Penghitungan Manual Nilai Kualitas	194
Tabel B. 21 Hasil Penghitungan Kualitas Perangkat Lunak.....	194
Tabel C. 1 Hasil Ekstraksi Kode Sumber.....	195
Tabel C. 2 Isian data keterangan Perangkat Lunak	195
Tabel C. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas .	196
Tabel C. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas.....	197
Tabel C. 5 Data Kasus Uji Coba Karakteristik Efisiensi.....	198
Tabel C. 6 Hasil Penghitungan Manual Karakteristik Efisiensi	198
Tabel C. 7 Data Kasus Uji Coba Submetrik Kemampuan tim ..	199

Tabel C. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim	199
Tabel C. 9 Data Kasus Uji Coba Metrik <i>Skill</i>	200
Tabel C. 10 Hasil Penghitungan Manual Metrik <i>Skill</i>	200
Tabel C. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer	200
Tabel C. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer.....	200
Tabel C. 13 Data Kasus Uji Coba Karakteristik <i>Maintainability</i>	201
Tabel C. 14 Hasil Penghitungan Manual Karakteristik <i>Maintainability</i>	202
Tabel C. 15 Data Kasus Uji Coba Karakteristik Portabilitas.....	203
Tabel C. 16 Hasil Penghitungan Manual Karakteristik Portabilitas	203
Tabel C. 17 Data Kasus Uji Coba Karakteristik.....	204
Tabel C. 18 Hasil Penghitungan Manual Karakteristik Portabilitas	205
Tabel C. 19 Data Kasus Uji Coba Nilai Kualitas	206
Tabel C. 20 Hasil Penghitungan Manual Nilai Kualitas	206
Tabel C. 21 Hasil Penghitungan Kualitas Perangkat Lunak.....	206
Tabel E. 1 Penghitungan Uji Kebergunaan Aspek Kemudahan Penggunaan Aplikasi.....	209
Tabel E. 2 Penghitungan Uji Kebergunaan Aspek Kemudahan Aplikasi Untuk Dipelajari.....	209
Tabel E. 3 Penghitungan Uji Kebergunaan Aspek Penggunaan Bahasa	210
Tabel E. 4 Penghitungan Uji Kebergunaan Aspek Kemampuan Kakas Bantu untuk Menentukan Kualitas Perangkat Lunak	210
Tabel E. 5 Penghitungan Uji Kebergunaan Aspek Kemampuan Kakas Bantu untuk Mengetahui Karakteristik yang berpengaruh terhadap Kualitas	211
Tabel E. 6 Penghitungan Uji Kebergunaan Aspek Kebermanfaatan Aplikasi	211

Tabel E. 7 Penghitungan Uji Kebergunaan Aspek Performa Aplikasi	211
---	-----

DAFTAR GAMBAR

Gambar 2. 1 <i>Pseudocode</i> Algoritma <i>Euclidean</i>	21
Gambar 2. 2 Contoh AST untuk Algoritma <i>Euclidean</i>	22
Gambar 2. 3 Alur Kerja AST	23
Gambar 2. 4 Struktur <i>Node</i> pada AST.....	24
Gambar 2. 5 Contoh <i>Visitor Pattern</i>	25
Gambar 3. 1 Diagram Kasus Penggunaan	31
Gambar 3. 2 Diagram Urutan Mengunggah Perangkat Lunak	33
Gambar 3. 3 Diagram Urutan Mengunggah Perangkat Lunak (Kejadian Alternatif)	33
Gambar 3. 4 Diagram Aktivitas Mengunggah Kode Sumber.....	34
Gambar 3. 5 Diagram Urutan Mengisi Keterangan Terkait Perangkat Lunak.....	36
Gambar 3. 6 Diagram Urutan Mengisi Keterangan Terkait Perangkat Lunak (Kejadian Alternatif)	36
Gambar 3. 7 Diagram Aktivitas Mengisi Keterangan Terkait Perangkat Lunak.....	37
Gambar 3. 8 Diagram Urutan Melihat Daftar Perangkat Lunak	38
Gambar 3. 9 Diagram Aktivitas Melihat Daftar Perangkat Lunak	39
Gambar 3. 10 Diagram Urutan Mengisi Kuesioner Subjektif.....	40
Gambar 3. 11 Diagram Aktivitas Mengisi Kuesioner Subjektif.	41
Gambar 3. 12 Diagram Urutan Melihat Detail Penilaian Kualitas Perangkat Lunak.....	42
Gambar 3. 13 Diagram Aktivitas Melihat Detail Penilaian Kualitas Perangkat Lunak.....	43
Gambar 3. 14 Diagram Urutan Melihat Laporan Kualitas Perangkat Lunak.....	44
Gambar 3. 15 Diagram Aktivitas Melihat Detail Penilaian Kualitas Perangkat Lunak.....	45
Gambar 3. 16 Diagram Arsitektur Sistem	46
Gambar 3. 17 Pemodelan Data Konseptual.....	49
Gambar 3. 18 Pemodelan Data Fisik.....	50
Gambar 3. 19 Diagram Kelas Lapisan Antarmuka.....	54

Gambar 3. 20 Diagram Kelas Lapisan Kontrol	55
Gambar 3. 21 Diagram Kelas Lapisan Model	56
Gambar 3. 22 Diagram Kelas Lapisan Util	57
Gambar 3. 23 Diagram Alir Aplikasi	62
Gambar 3. 24 Rancangan Tampilan Utama.....	66
Gambar 3. 25 Rancangan Tampilan Halaman Mengunggah Perangkat Lunak.....	67
Gambar 3. 26 Rancangan Halaman Menampilkan Pertanyaan Isian	69
Gambar 3. 27 Rancangan Tampilan Halaman Menampilkan Pertanyaan Rating Objektif	71
Gambar 3. 28 Rancangan Tampilan Halaman Daftar Perangkat Lunak.....	74
Gambar 3. 29 Rancangan Halaman Menampilkan Pertanyaan Rating Subjektif.....	75
Gambar 3. 30 Rancangan Halaman Menampilkan Pertanyaan Bobot Fungsionalitas.....	77
Gambar 3. 31 Rancangan Halaman Menampilkan Pertanyaan Bobot Efisiensi	79
Gambar 3. 32 Rancangan Halaman Pertanyaan Bobot Keterawatan	81
Gambar 3. 33 Rancangan Halaman Menampilkan Pertanyaan Bobot Keterawatan 2	84
Gambar 3. 34 Rancangan Halaman Menampilkan Pertanyaan Bobot Portabilitas	86
Gambar 3. 35 Rancangan Tampilan Detail Penilaian Perangkat Lunak.....	88
Gambar 3. 36 Rancangan Laporan Penilaian Kualitas Perangkat Lunak.....	90
Gambar 4. 1 Syntax SQL Menampilkan Nilai Karakteristik.....	136
Gambar 4. 2 Syntax SQL Menampilkan Nilai Subkarakteristik	137
Gambar 5. 1 Tampilan Hasil Pengujian Fitur Mengunggah Perangkat Lunak Skenario 1.....	141
Gambar 5. 2 Penyimpanan Data dalam Basis Data	142

Gambar 5. 3 Hasil Uji Fitur Mengunggah Perangkat Lunak Skenario 2.....	143
Gambar 5. 4 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1 (1).....	145
Gambar 5. 5 Penyimpanan Data Kuesioner Rating Perangkat Lunak pada Basis Data.....	145
Gambar 5. 6 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1 (2).....	146
Gambar 5. 7 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 2.....	148
Gambar 5. 8 Hasil Uji Fitur Menampilkan Daftar Perangkat Lunak	150
Gambar 5. 9 Hasil Uji Fitur Mengisi Kuesioner Rating Subjektif Perangkat Lunak.....	152
Gambar 5. 10 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (1).....	152
Gambar 5. 11 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (2).....	153
Gambar 5. 12 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (3).....	153
Gambar 5. 13 Penyimpanan Data Penilaian dalam Basis Data .	154
Gambar 5. 14 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (4).....	154
Gambar 5. 15 Penyimpanan Data Jawaban Rating Subjektif pada Basis Data.....	154
Gambar 5. 16 Penyimpanan Data Jawaban Kuesioner Bobot pada Basis Data.....	155
Gambar 5. 17 Hasil Uji Fitur Menampilkan Detail Penilaian Perangkat Lunak.....	157
Gambar 5. 18 Hasil Uji Fitur Laporan Penilaian Perangkat Lunak	159
Gambar 5. 19 Grafik Perbandingan Hasil Penghitungan Kualitas Perangkat Lunak Uji Coba	163
Gambar 5. 20 Grafik Perbandingan Skor Atribut Kebergunaan Kakas Bantu	165

Gambar D. 1 Lembar Kuesioner Pengujian Kebergunaan Perangkat Lunak.....	207
Gambar F. 1 Lembar Jawaban Kuesioner 1	213
Gambar F. 2 Lembar Jawaban Kuesioner 2	214
Gambar F. 3 Lembar Jawaban Kuesioner 3	215
Gambar F. 4 Lembar Jawaban Kuesioner 4	216
Gambar F. 5 Lembar Jawaban Kuesioner 5	217
Gambar F. 6 Lembar Jawaban Kuesioner 6	218
Gambar F. 7 Lembar Jawaban Kuesioner 7	219
Gambar F. 8 Lembar Jawaban Kuesioner 8	220
Gambar F. 9 Lembar Jawaban Kuesioner 9	221
Gambar F. 10 Lembar Jawaban Kuesioner 10	222

DAFTAR KODE SUMBER

Kode Sumber 4. 1 Pengecekan Validitas Isian.....	98
Kode Sumber 4. 2 Implementasi Pemanggilan Laporan	99
Kode Sumber 4. 3 Implementasi <i>Method</i> GetScore pada Kelas Karakteristik	101
Kode Sumber 4. 4 Implementasi Pemanggilan Nilai Bobot pada Kelas Karakteristik	102
Kode Sumber 4. 5 Implementasi <i>Method</i> GetChild pada Kelas Karakteristik	103
Kode Sumber 4. 6 Implementasi <i>Method</i> HasBobot pada Kelas Metrik.....	104
Kode Sumber 4. 7 Implementasi <i>Method</i> GetChild pada Kelas Metrik.....	105
Kode Sumber 4. 8 Implementasi <i>Method</i> GetSource pada Kelas Metrik.....	106
Kode Sumber 4. 9 Implementasi <i>Method</i> GetScore pada Kelas Karakteristik	107
Kode Sumber 4. 10 Implementasi Pemanggilan Nilai Bobot pada Kelas Metrik.....	107
Kode Sumber 4. 11 Implementasi <i>Method</i> GetScore pada Kelas Karakteristik.....	108
Kode Sumber 4. 12 Implementasi <i>Method</i> GenerateQM	109
Kode Sumber 4. 13 Implementasi <i>Method</i> InsertRatingToDB ..	110
Kode Sumber 4. 14 Implementasi <i>Method</i> InsertBobotToDB ..	111
Kode Sumber 4. 15 Implementasi <i>Method</i> InsertToDB	112
Kode Sumber 4. 16 Implementasi <i>Method</i> CountScore	113
Kode Sumber 4. 17 Implementasi <i>Method</i> InsertToDB	114
Kode Sumber 4. 18 Implementasi <i>Method</i> InsertToDB pada Kelas PerangkatLunak.....	115
Kode Sumber 4. 19 Implementasi <i>Method</i> GetScoreRating pada Kelas Subkarakteristik.....	116
Kode Sumber 4. 20 Implementasi Pemanggilan Nilai Bobot pada Kelas Subkarakteristik.....	117

Kode Sumber 4. 21 Implementasi <i>Method</i> GetChild pada Kelas Subkarakteristik.....	118
Kode Sumber 4. 22 Implementasi <i>Method</i> HasBobot untuk Kelas Subsubmetrik.....	119
Kode Sumber 4. 23 Implementasi <i>Method</i> HasBobot untuk Kelas Subsubmetrik.....	120
Kode Sumber 4. 24 Implementasi <i>Method</i> HasBobot untuk Kelas Submetrik	121
Kode Sumber 4. 25 Implementasi <i>Method</i> HasBobot untuk Kelas Submetrik	121
Kode Sumber 4. 26 Implementasi <i>Method</i> GetScoreRating pada Kelas SubsubMetrik	123
Kode Sumber 4. 27 Implementasi <i>Method</i> HasBobot untuk Kelas Subsubmetrik.....	124
Kode Sumber 4. 28 Implementasi <i>Method</i> ParseFileInDir.....	124
Kode Sumber 4. 29 Implementasi <i>Method</i> Parse	125
Kode Sumber 4. 30 Implementasi <i>Method</i> GetLineCount	126
Kode Sumber 4. 31 Implementasi VisitorCC.....	128
Kode Sumber 4. 32 Implementasi VisitStatic	129
Kode Sumber 4. 33 Implementasi Kelas CrispValue	131
Kode Sumber 4. 34 Implementasi Kelas Unzip	133
Kode Sumber 4. 35 Implementasi Kelas OlahMetrik.....	136
Kode Sumber G. 1 Kode Sumber Kelasreport.java.....	224
Kode Sumber G. 2 Kode Sumber Client.java.....	227
Kode Sumber G. 3 Kode Sumber Reduksi.java	229

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Perangkat lunak merupakan suatu produk yang dihasilkan melalui proses tertentu. Perangkat lunak dibangun oleh pengembang/pemrogram dengan tujuan untuk memudahkan aktivitas manusia. Sebagian manusia memang merasakan manfaat dari perangkat lunak walaupun wujud nyata dari perangkat lunak tersebut tidak bisa ditangkap. Perangkat lunak merupakan suatu produk yang tidak terlihat, sehingga tidak bisa dinilai secara langsung bagaimana kondisi perangkat lunak yang dibangun dan bagaimana kualitas dari perangkat lunak tersebut. Berbeda dengan barang lain seperti peralatan elektronik, produk *garment*, dan sebagainya yang bisa dinilai secara langsung kualitasnya, perangkat lunak memerlukan cara khusus untuk menghasilkan dan menilai kualitas produk.

Kualitas perangkat lunak menjadi sangat penting mengingat kondisi tersebut, terutama bagi pengembang/pemrogram, perusahaan yang bergerak dalam bidang teknologi dan informasi, ataupun perusahaan yang menggunakan perangkat lunak untuk mendukung proses bisnisnya. Kualitas perangkat lunak akan berpengaruh terhadap kepuasan pengguna yang nantinya akan berdampak pada reputasi suatu perusahaan di mata masyarakat [1]. Dua hal tersebut akan berpangkal pada suatu alasan utama, yaitu biaya. Semakin baik reputasi perusahaan, maka pendapatan perusahaan akan juga semakin tinggi. Sebaliknya apabila perangkat lunak yang dihasilkan tidak berkualitas, maka perusahaan harus melakukan perbaikan terhadap perangkat lunak

tersebut, bahkan perlu membayar ganti rugi apabila ternyata perangkat lunak tersebut mengakibatkan kerugian bagi pengguna. Oleh karena itu, pengecekan kualitas perangkat lunak merupakan hal yang perlu dilakukan sebelum perangkat lunak dipasarkan.

Kualitas merupakan suatu hal yang sulit untuk diukur, karena kualitas bersifat subjektif. Oleh karena itu untuk mengevaluasi kualitas perangkat lunak diperlukan suatu metode khusus untuk mengubah nilai subjektif tersebut. Dalam tugas akhir ini, digunakan *fuzzy method* untuk menguantifikasi parameter-parameter yang digunakan untuk mengukur kualitas perangkat lunak. Logika *fuzzy* merupakan bentuk dari logika banyak-nilai yang memungkinkan suatu kebenaran memiliki nilai antara 0 hingga 1 [2], tidak seperti nilai biner yang hanya memiliki dua kemungkinan nilai. Dengan menggunakan *fuzzy*, evaluasi kualitas perangkat lunak bisa menjadi lebih objektif karena kemungkinan nilai yang lebih banyak.

Parameter-parameter yang digunakan dalam tugas akhir ini menggunakan model kualitas ISO/IEC 9126. ISO/IEC 9126 merupakan suatu standar internasional untuk mengevaluasi perangkat lunak [3]. Dalam ISO/IEC 9126, karakteristik-karakteristik yang dimiliki oleh suatu perangkat lunak adalah fungsionalitas, kehandalan, dapat digunakan, efisiensi, dapat dirawat, dan bersifat portabel. Selanjutnya keenam karakter tersebut dipecah lagi ke dalam beberapa subkarakteristik yang secara langsung akan berpengaruh pada kualitas perangkat lunak. Selanjutnya perangkat lunak akan dievaluasi sesuai dengan karakteristik dan subkarakteristik pada ISO/IEC 9126.

Hasil dari tugas akhir ini adalah kakas bantu yang akan dibangun dapat mengevaluasi kualitas dari suatu perangkat lunak dan dapat mengkuantifikasikan kualitas ke dalam satuan nilai angka. Nantinya kakas bantu ini bisa digunakan oleh pengembang/pemrogram untuk menguji kualitas produk perangkat lunak yang telah dibangun sebelum diluncurkan ke pengguna.

1.2. Rumusan Permasalahan

Rumusan permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut.

1. Bagaimana cara mengekstraksi fitur-fitur dari kode sumber suatu perangkat lunak?
2. Bagaimana cara melakukan penilaian kualitas menggunakan model kualitas ISO/IEC 9162 dan metode fuzzy?
3. Bagaimana cara pembuatan aplikasi kaka bantu yang dapat mengevaluasi kualitas perangkat lunak dengan menggunakan model kualitas ISO/IEC 9126 dan metode fuzzy menggunakan bahasa pemrograman Java?
4. Bagaimana hasil evaluasi kaka bantu pengukuran kualitas perangkat lunak?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, antara lain sebagai berikut.

1. Aplikasi berbasis *desktop* dengan bahasa pemrograman Java dan basis data yang digunakan adalah H2.
2. Perangkat lunak yang dapat diukur adalah perangkat lunak berbasis bahasa pemrograman Java.
3. Metode yang digunakan untuk mengevaluasi kualitas perangkat lunak adalah metode *fuzzy* multi kriteria.
4. Karakteristik, subkarakteristik, dan metrik yang dievaluasi dari perangkat lunak terpapar pada Tabel 2.1.

1.4. Tujuan

Tujuan dalam pembuatan tugas akhir ini antara lain sebagai berikut.

1. Mengetahui cara mengekstraksi fitur kode sumber dari suatu perangkat lunak.

2. Mengetahui langkah-langkah penilaian kualitas perangkat lunak menggunakan model kualitas ISO/IEC 9216 dan metode *fuzzy*.
3. Menghasilkan aplikasi kakas bantu yang dapat mengevaluasi kualitas perangkat lunak yang menggunakan model kualitas ISO/IEC 9126 dan metode *fuzzy* dalam bahasa Java
4. Mendapatkan hasil evaluasi kakas bantu pengukuran kualitas perangkat lunak yang dibangun.

1.5. Manfaat

Dengan dibangunnya kakas bantu pengukuran kualitas perangkat lunak dengan metode *fuzzy*, pengembang/pemrogram dari suatu perangkat lunak dapat mengukur kualitas perangkat lunak yang dibangun sebelum diberikan kepada pengguna.

1.6. Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir.

Tahap awal pengerjaan tugas akhir adalah penyusunan proposal. Proposal ini mengajukan gagasan mengenai “Rancang Bangun Kakas Bantu Pengukuran Kualitas perangkat lunak Menggunakan Metode Fuzzy”.

2. Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur yang diperlukan untuk perancangan perangkat lunak yang akan dibuat. Informasi didapatkan dari buku acuan, internet, maupun materi-materi lain yang berhubungan dengan metode yang digunakan dalam pengerjaan tugas akhir ini. Studi literatur yang digunakan untuk menunjang pengerjaan tugas akhir ini yaitu Model Kualitas ISO/IEC

9126, *Fuzzy Triangular, Abstract Syntax Tree, Visitor Pattern*, Eclipse IDE, dan Kompleksitas Siklomatik.

3. Analisis dan Desain Perangkat Lunak

Pada tahap ini dilakukan analisis dengan mempelajari literatur mengenai proses penghitungan kualitas perangkat lunak dengan membaca buku, jurnal ilmiah, serta artikel internet. Selanjutnya melakukan desain perangkat lunak berdasarkan hasil analisis untuk mendapatkan model kakas bantu pengukuran kualitas perangkat lunak. Analisis yang dilakukan adalah analisis kebutuhan sistem dan perancangan yang dilakukan adalah perancangan basis data, diagram kelas, dan antarmuka.

4. Implementasi Perangkat Lunak

Tahap implementasi ini merupakan tahap untuk membangun “Rancang Bangun Kakas Bantu Pengukuran Kualitas Perangkat Lunak Menggunakan Metode Fuzzy” yang diajukan. Kakas bantu yang digunakan untuk mengimplementasikan aplikasi tersebut adalah IDE Eclipse dengan bahasa pemrograman Java.

5. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian dengan melakukan pengecekan fungsionalitas sistem untuk mencoba kesesuaian aplikasi dengan rancangan dan desain perangkat lunak yang telah dibuat. Tahap ini juga mencari ketidaksesuaian yang ada pada program, agar dapat dilakukan perbaikan dan penyempurnaan di kemudian hari.

6. Penyusunan Buku Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat

1.7. Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

BAB III. ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Bab ini berisi proses implementasi dari setiap kelas pada semua modul.

BAB V. PENGUJIAN DAN EVALUASI

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan metode yang diajukan pada pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 ISO/IEC 9126 – Product Quality

ISO/IEC 9126 *Software engineering-Product Quality* merupakan sebuah standar internasional untuk evaluasi kualitas perangkat lunak [3]. Fungsi utama dari standar ISO/IEC 9126 adalah untuk menunjukkan beberapa kesalahan manusia yang sering terjadi yang dapat berpengaruh negatif terhadap penyampaian dan persepsi terhadap proyek pengembangan perangkat lunak. Dengan menjelaskan dan membuat kesepakatan terhadap prioritas proyek dan selanjutnya mengonversikan prioritas abstrak ke suatu nilai terukur, ISO/IEC 9126 mencoba untuk membangun kesamaan pemahaman pada tujuan proyek [3].

Standar ISO/IEC terdiri dari empat bagian, yaitu:

- *quality model*;
- *external metrics*;
- *internal metrics*; dan
- *quality in use metrics*.

Model kualitas yang digunakan pada kakas bantu ini adalah sebagai berikut [3]:

- Fungsionalitas yaitu atribut-atribut yang digunakan pada keberadaan fungsional sistem dan sistem perangkat lunak:
 - a. *suitability*;

- b. accuracy;*
 - c. interoperability;*
 - d. security;*
 - e. customizability; dan*
 - f. functionality Compliance.*
- Efisiensi yaitu atribut-atribut yang digunakan pada hubungan antara performa perangkat lunak dan sumber daya yang digunakan dalam kondisi tertentu:
 - a. time behavior;*
 - b. resource Utilization;*
 - c. efficiency Compliance; dan*
 - d. scalability..*
- Tingkat keterawatan yaitu atribut-atribut yang digunakan untuk menjelaskan usaha yang diperlukan untuk melakukan perubahan terhadap perangkat lunak:
 - a. analyzability;*
 - b. changeability;*
 - c. testability;*
 - d. maintainability compliance; dan*
 - e. trackability.*
- Portabilitas yaitu atribut-atribut yang digunakan untuk menjelaskan kemampuan perangkat lunak untuk berjalan dari satu lingkungan ke lingkungan lain:
 - a. installability;*
 - b. adaptability;*
 - c. co-existence; dan*
 - d. portability compliance.*

Lebih lanjut, karakteristik dan subkarakteristik tersebut dipecah ke dalam metrik yang lebih objektif seperti pada Tabel 2.1.

Tabel 2. 1 Metrik untuk ISO/IEC 9126

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
1	Fungsionalitas	Kesesuaian	Persentase operasi yang sesuai	-	-
		Akurasi	Persentase operasi yang memenuhi akurasi yang dibutuhkan	-	-
			Pemenuhan terhadap presisi yang dibutuhkan	-	-
		<i>Interoperability</i>	Basis data yang digunakan	-	-
		Keamanan	Persentase pengontrolan	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
			akses yang disediakan		
			Tingkat keketatan akses	-	-
		Pemenuhan fungsionalitas	Pemenuhan terhadap standar fungsionalitas	-	-
		Kustomabilitas	Tingkat kustomabilitas	-	-
2	Efisiensi	Waktu	Jumlah variabel global	-	-
			Tipe <i>translator</i>	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
			Kemampuan prosesor	-	-
		Penggunaan sumber daya	Persentase CPU yang tidak terpakai	-	-
			Dukungan perangkat lunak terhadap sumber daya luar	-	-
		Pemenuhan efisiensi	Pemenuhan terhadap standar fungsionalitas	-	-
		Skalabilitas	Dukungan untuk beberapa pengguna	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
3	Keterawatan	<i>Analyzability</i>	KLOC	-	-
			Kemampuan	Kemampuan Teknis	-
				Pengalaman industri organisasi	-
				Kemampuan tim pengembang	Rata-rata kualitas anggota pengembang
					Kerjasama antara tim pengembang
					Performa keseluruhan tim pengembang

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
			Pengalaman manajer	Pengalaman manajer proyek dalam bisnis IT	-
				Pengalaman manajer proyek sebagai manajer	-
			Jumlah KLOC per anggota tim	-	-
			Ketersediaan dokumentasi	-	-
			<i>Cyclomatic complexity</i>	-	-
			Jumlah versi yang telah dirilis	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
			Level <i>Capability Maturity Model (CMM)</i>	-	-
		<i>Changeability</i>	Persentase jumlah fitur yang dapat diubah	-	-
		<i>Testability</i>	Ketersediaan jumlah kasus uji coba yang cukup	-	-
		Pemenuhan <i>Maintainability</i>	Pemenuhan terhadap standar <i>Maintainability</i>	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
		<i>Track-ability</i>	Ketersediaan sistem penelusuran fungsional dan perilaku sistem	-	-
			Kemudahan dalam penelusuran versi lama perangkat lunak	-	-
4	Portabilitas	Kemampuan adaptasi	Kompatibilitas pada sistem operasi lain	-	-
			Penggunaan <i>tools</i> dasar	-	-
			Jumlah <i>package</i> yang	-	-

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik
			harus diinstall terlebih dulu		
		<i>Installability</i>	Jumlah <i>package</i> yang harus diinstall terlebih dulu selain OS	-	-
		Pemenuhan <i>portability</i>	Pemenuhan terhadap standar <i>portability</i>	-	-

2.2 Fuzzy Multi Kriteria

Dalam kasus bantu ini, teknik yang digunakan adalah *fuzzy* triangular [3]. Nilai *fuzzy* direpresentasikan sebagai fungsi keanggotaan *fuzzy*. Dalam paper ini digunakan juga proses penghitungan untuk menentukan nilai subkarakteristik maupun karakteristik, proses ini dinamai sebagai operasi *fuzzy*. Operasi *fuzzy* yang digunakan ada dua, yaitu:

- Perkalian *fuzzy*: misalkan (a,b,c) dan (x,y,z) adalah dua himpunan *fuzzy* triangular, maka perkalian *fuzzy* didefinisikan sebagai berikut.

$$(a,b,c) \times (x,y,z) = (a \times x, b \times y, c \times z) \quad (2.1)$$

- Penjumlahan *fuzzy*: misalkan (a,b,c) dan (x,y,z) adalah dua himpunan *fuzzy* triangular, maka penjumlahan *fuzzy* didefinisikan sebagai berikut.

$$(a,b,c) \times (x,y,z) = [\max(a,x), \max(b,y), \max(c,z)] \quad (2.2)$$

Fuzzification merupakan suatu proses konversi kejadian nyata ke dalam nilai himpunan *fuzzy*. Nilai fuzifikasi untuk masing-masing aspek lebih detail dapat dilihat pada rujukan [3]. Nilai *fuzzy* didapatkan berdasarkan suatu kriteria tertentu. Tabel 2.2 memberikan contoh untuk fuzifikasi terhadap aspek kompleksitas siklomatik.

Tabel 2. 2 Contoh Fuzifikasi untuk Kompleksitas Siklomatik

Kompleksitas Siklomatik	Nilai <i>Fuzzy</i>
0-5	Sangat Tinggi
6-10	Tinggi
11-20	Sedang
21-50	Rendah

Kompleksitas Siklomatik	Nilai <i>Fuzzy</i>
>50	Sangat Rendah

Defuzzification merupakan suatu proses konversi nilai *fuzzy* ke dalam nilai yang sebenarnya. Rumus yang digunakan pada aplikasi ini adalah rumus *centroid* [4] dengan definisi sebagai berikut.

$$\frac{\int \mu(z).z dz}{\int \mu(z)dz} \quad (2.3)$$

2.3 Abstract Syntax Tree

Pohon Sintaks Abstrak (disingkat AST, *Abstract Syntax Tree*) adalah sebuah pohon yang merepresentasikan sintaks abstrak dari suatu kode sumber yang ditulis dalam suatu bahasa pemrograman [4]. Dalam pohon ini, setiap titik merepresentasikan suatu konstruk yang terjadi pada kode sumber dan variabel atau konstanta direpresentasikan sebagai daun [5].

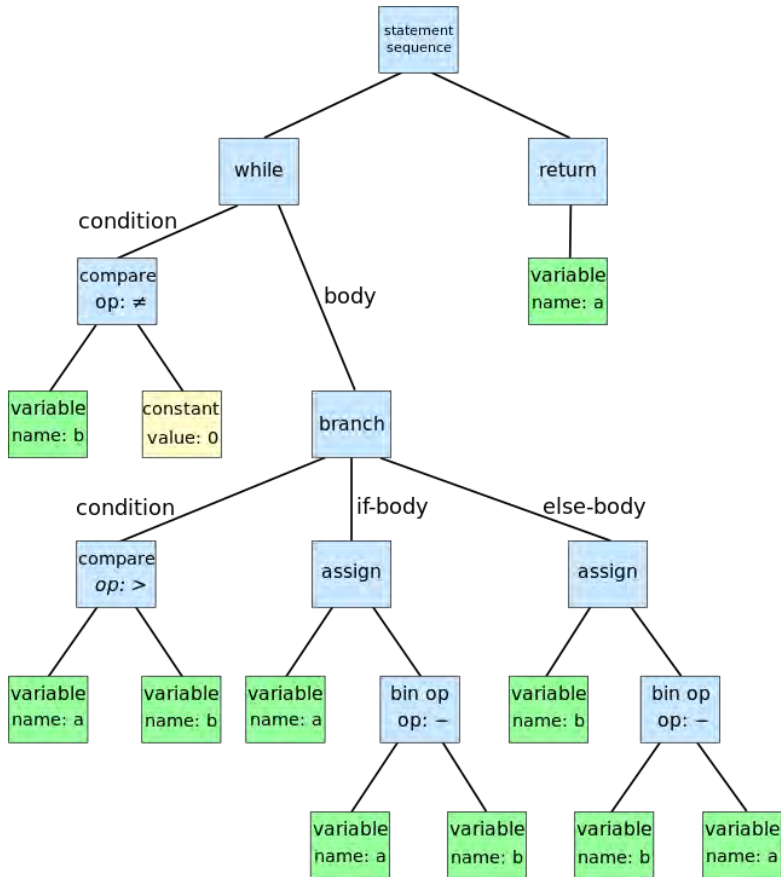
Abstract Syntax Tree (AST) menangkap struktur penting dari sebuah kode program dan merepresentasikannya dalam bentuk *tree*. Struktur kode program tersebut dibentuk menjadi sebuah *node* dengan tipe *node* yang spesifik. AST berbeda dari *concrete syntax tree* karena AST tidak menangkap detail sintaks seperti koma untuk pemisahan argumen.

AST merupakan kerangka basis untuk semua alat pada IDE Eclipse, termasuk *refactoring*, *Quick Fix*, dan *Quick Assist*. AST memetakan kode sumber Java standar ke dalam bentuk pohon. Bentuk pohon ini tergolong lebih mudah dan handal untuk dianalisis dan dimodifikasi secara manual daripada dengan kode sumber yang berbasis teks standar. Pengembang juga dapat memodifikasi model pohon pada AST dan otomatis akan terjadi perubahan yang diinginkan juga pada kode sumber.

Semua kelas yang berhubungan dengan AST terdapat pada paket *org.eclipse.jdt.core.dom* dari *plug-in org.eclipse.jdt.core*. Gambar 2.1 menjelaskan *pseudocode* untuk algoritma *Euclidean*. Gambar 2.2 menjelaskan AST yang dihasilkan dari *pseudocode* algoritma *Euclidean* dan struktur *node* penyusunnya.[6]. Dari Gambar 2.2 dapat dilihat bahwa AST yang dihasilkan merepresentasikan seluruh elemen penyusun suatu kode sumber, baik itu variabel ataupun *statement*. Dari sinilah kita bisa mengekstraksi fitur dari kode sumber

```
while b ≠ 0
  if a > b
    a := a - b
  else
    b := b - a
  return a
```

Gambar 2.1 Pseudocode Algoritma *Euclidean*



Gambar 2. 2 Contoh AST untuk Algoritma Euclidean

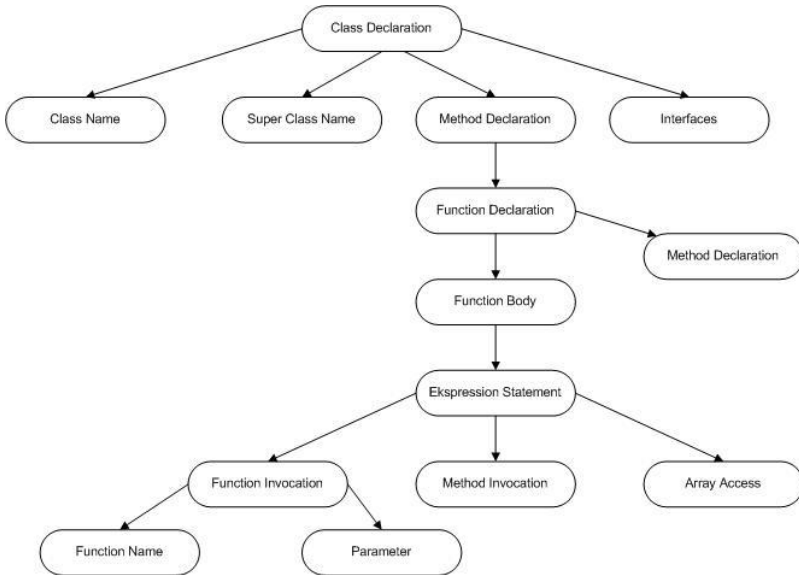


Gambar 2. 3 Alur Kerja AST

Ilustrasi alur kerja dari AST ini dapat diilustrasikan pada Gambar 2.3. Dari Gambar 2.3 dapat dijabarkan beberapa proses yang dilalui sebagai berikut:

- a. Kode Java : Siapkan kode sumber Java yang akan di-*parsing*. Kode sumber bisa berbentuk *file* Java atau berupa *array of char* yang berisi kode Java;
- b. *Parse* : Kode sumber yang telah didefinisikan pada langkah diparsing. Pada langkah ini, proses penguraian telah disediakan dan didefinisikan oleh kelas *org.eclipse.JDT.core.dom.ASTParser*;
- c. Hasil dari langkah b berupa model pohon. Model pohon ini merupakan representasi penuh dari kode sumber Java pada langkah a;
- d. Memanipulasi AST dapat dilakukan dengan dua cara, yaitu:
 - i. dengan mengubah AST secara langsung; dan
 - ii. dengan mencatat perubahan pada protokol yang terpisah. Protokol ini ditangani oleh *ASTRewrite*.
- e. Jika terjadi perubahan, maka perubahan tersebut harus diimplementasikan pada kode sumber Java yang didefinisikan pada langkah a; dan
- f. *IDocument* merupakan pembungkus untuk kode sumber Java pada langkah a dan dibutuhkan pada langkah e.

Struktur model pohon AST untuk kode program Java dapat dilihat pada Gambar 2.4. Pada gambar tersebut dapat dilihat komponen-komponen penyusun AST dimulai dari *ClassDeclaration* sebagai *rootNode*.



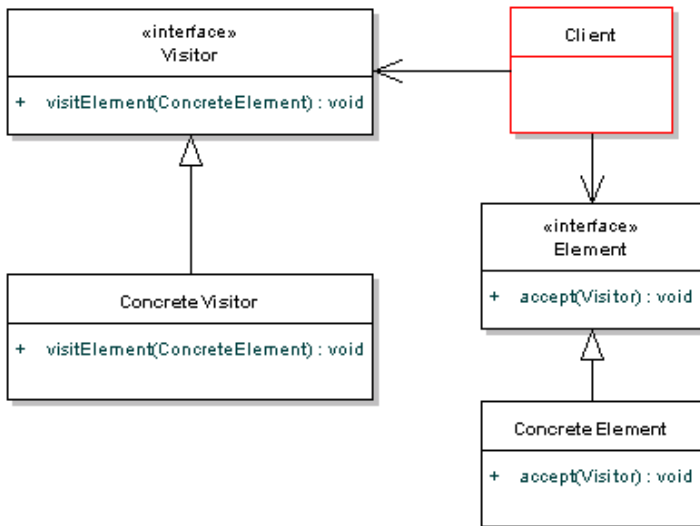
Gambar 2. 4 Struktur *Node* pada AST

2.4 Pola Visitor

Pola *visitor* dalam pemrograman berbasis objek merupakan suatu cara untuk memisahkan algoritma dari suatu struktur objek dimana algoritma tersebut dioperasikan [6]. Sehingga dengan menerapkan pola *visitor* ini, bisa ditambahkan suatu operasi baru kepada struktur objek yang sudah ada tanpa memodifikasi struktur tersebut. Terlebih dulu, perlu dibuat suatu kelas *visitor* yang mengimplementasikan semua fungsi virtual. Sehingga bahasa yang mendukung untuk dibuatkan suatu pola

visitor adalah bahasa yang memungkinkan terjadinya *method overloading* [6].

Untuk dapat mengakses setiap *node* dalam AST, *visitor pattern* harus digunakan. Setiap AST dilengkapi dengan sebuah kelas abstrak yang dibangun berdasarkan *visitor pattern*. Kelas ini dapat digunakan sebagai dasar kelas *visitor* yang akan digunakan untuk menelusuri AST. Kelas ini menelusuri AST secara rekursif, menjalankan abstrak *method visit* untuk setiap *node* dari AST[7]. Terdapat beberapa *method* yang dapat digunakan untuk menelusuri *node* dari AST antara lain *preVisit*, *visit*, *endVisit*, dan *postVisit*. *Method preVisit* dipanggil ketika sebuah *node* dari AST akan dikunjungi, *visit* dipanggil ketika *node* dikunjungi, *endVisit* dipanggil ketika *node* dan *node-node* dibawahnya selesai dikunjungi, dan *postVisit* dipanggil setelah *method endVisit* dipanggil. Berikut ini adalah contoh suatu pola *visitor* [5].



Gambar 2. 5 Contoh Visitor Pattern

Gambar 2.5 menjelaskan salah satu contoh dari pola *visitor*. Kelas *interface* *visitor* berisikan suatu *method* *visitElement* untuk struktur objek *ConcreteElement*. Sementara itu, *ConcreteVector* mengimplementasikan *method* yang didefinisikan pada kelas *interface* *visitor*. Kelas *interface* *Element* mendefinisikan *method* *accept* yang memungkinkan *visitor* untuk menjalankan suatu operasi pada elemen tersebut [5].

2.5 Eclipse

Eclipse merupakan suatu *Integrated Development Environment* (IDE) yang bersifat ekstensibel. Tujuan utamanya ialah memberikan layanan untuk mengatur suatu kumpulan alat yang saling bekerjasama dalam mendukung tugas-tugas pemrograman. Bahasa pemrograman utama Eclipse ialah Java.

2.5.1 Java Development Tool (JDT)

Java Development Tool (JDT) merupakan suatu kumpulan *plug-in* yang memungkinkan adanya penambahan kemampuan IDE Java pada IDE Eclipse. *Plug-in* JDT dapat dikategorikan ke dalam beberapa *plug-in* yang lebih terspesialisasi, diantaranya adalah JDT Core.

2.5.2 JDT Core

JDT Core merupakan *plug-in* yang memberikan layanan sebagai berikut ini [9].

- Java Builder
- Java Model yang memungkinkan API untuk melakukan perubahan pada Java Element Tree. Java Element Tree merupakan suatu pohon yang menggambarkan proyek dari sudut pandang elemen Java. Elemen yang didukung adalah *compilation unit*, *classes*, *fragments*, dan lain sebagainya.
- Pencarian arsitektur terindeks yang digunakan untuk pencarian, penghitungan hirarki tipe, dan refaktor.
- Dukungan evaluasi

2.6 Kompleksitas Siklomatik

Kompleksitas siklomatik adalah metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks uji coba berbasis jalur, nilai yang dihitung untuk kompleksitas siklomatik menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali. Jalur independen adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.

Kompleksitas siklomatik dihitung dengan menggunakan graf kontrol alir (CFG) dari suatu program, dengan *node* yang merupakan suatu sekumpulan kode program yang bersifat atomik dan *edge* menghubungkan dua *node* apabila *node* tersebut dieksekusi secara berkelanjutan. Kompleksitas siklomatik dapat diaplikasikan ke dalam elemen-elemen lain yang lebih kecil, seperti *method*, *class*, dan lain-lain.

Untuk penghitungan kompleksitas siklomatik dalam kode sumber berbahasa Java, dilakukan penambahan untuk setiap *statement* yang ditemukan dalam kode sumber. Statement tersebut terdapat pada Tabel 2.3 [9].

Tabel 2. 3 Statement Penambahan Kompleksitas Siklomatik dalam Bahasa Pemrograman Java

Kategori	Penambahan
<i>Returns</i>	Setiap <i>return</i> dalam <i>method</i>
<i>Selections</i>	<i>If-Else</i> , <i>Switch-Case</i> , <i>Default</i>
Perulangan	<i>For</i> , <i>While</i> , <i>Do-While</i> , <i>Break</i> , <i>Continue</i>
Operator	&&, , ? :
<i>Exceptions</i>	<i>Catch</i> , <i>Finally</i> , <i>Throw</i>

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1. Analisis Permasalahan

Permasalahan utama yang diangkat dalam pembuatan tugas akhir ini adalah bagaimana mengukur kualitas perangkat lunak berbahasa Java dengan menggunakan bahasa Java. Permasalahan kedua yaitu bagaimana cara membangun *Abstract Syntax Tree* untuk aplikasi tersebut.

Abstract Syntax Tree digunakan untuk mendapatkan struktur dan elemen dari suatu kode sumber. Pada tugas akhir ini difokuskan pada bahasa pemrograman Java karena JDT mendukung pembuatan AST untuk bahasa pemrograman Java. AST yang dibangun merepresentasikan elemen-elemen untuk satu kode sumber `.java`. Dengan menggunakan AST, kita bisa mengekstraksi nilai-nilai implisit yang terkandung di dalam suatu kode sumber, untuk tugas akhir ini nilai ekstraksi yang diambil adalah **jumlah variabel global** dan **jumlah kompleksitas siklomatik**.

Sistem yang akan dibuat yaitu berupa kakas bantu. Kakas bantu ini menghitung kualitas perangkat lunak dengan masukan berupa kode sumber aplikasi tersebut dan kuesioner yang diisi oleh pengembang aplikasi. Kakas akan menganalisis dan mengekstraksi fitur-fitur dari kode sumber perangkat lunak dan menguantitatifkan nilai-nilai subjektif dari kuesioner. Kakas akan menampilkan nilai kualitas suatu aplikasi dalam bentuk angka dengan rentang nilai 0 hingga 1.

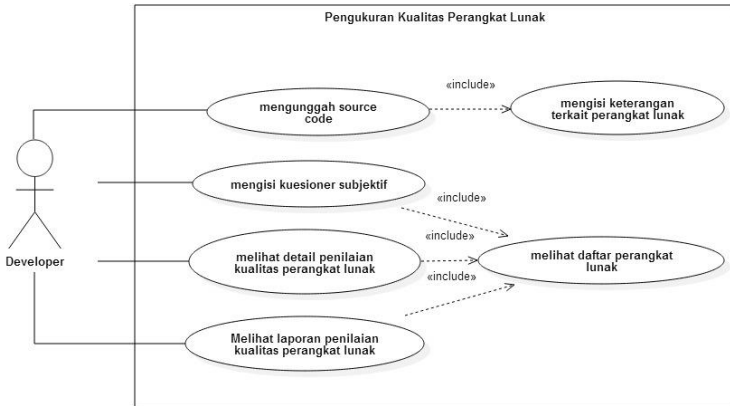
Diharapkan dengan adanya kakas bantu ini, pengguna dapat dengan lebih mudah menilai mutu dari perangkat lunak yang telah dikembangkan, sehingga dapat dilakukan penanganan lebih lanjut.

3.1.2. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Aktor yang terdapat pada sistem ini hanya memiliki sebuah peran yaitu sebagai pengguna. Pengguna perangkat ini adalah pengembang aplikasi *desktop* berbahasa Java.

3.1.3. Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah diagram kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.1. Tabel 3.1 berisi penjelasan dari setiap kasus penggunaan yang tertera pada Gambar 3.1.



Gambar 3. 1 Diagram Kasus Penggunaan

Tabel 3. 1 Daftar Kode Diagram Kasus Penggunaan

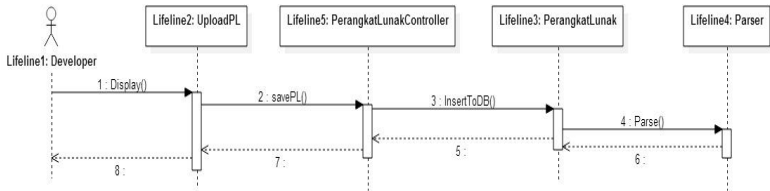
Kode Kasus Penggunaan	Nama
UC-0001	Mengunggah kode sumber
UC-0002	Mengisi keterangan terkait perangkat lunak
UC-0003	Melihat daftar perangkat lunak
UC-0004	Mengisi kuesioner subjektif
UC-0005	Melihat detail penilaian kualitas perangkat lunak
UC-0006	Menampilkan laporan penilaian perangkat lunak

3.1.3.1. *Mengunggah Kode Sumber*

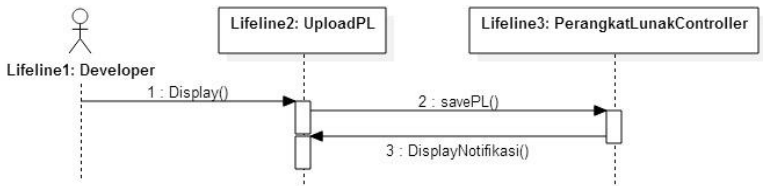
Pada kasus penggunaan ini, sistem menerima masukan berupa kode sumber. Setelah itu, sistem akan melakukan analisa terhadap kode sumber tersebut untuk mengekstraksi beberapa fitur. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.2. Diagram aktivitas kejadian normal dan kejadian alternatif dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.2, Gambar 3.3 dan Gambar 3.4.

Tabel 3. 2 Spesifikasi Kasus Penggunaan Mengunggah Kode Sumber

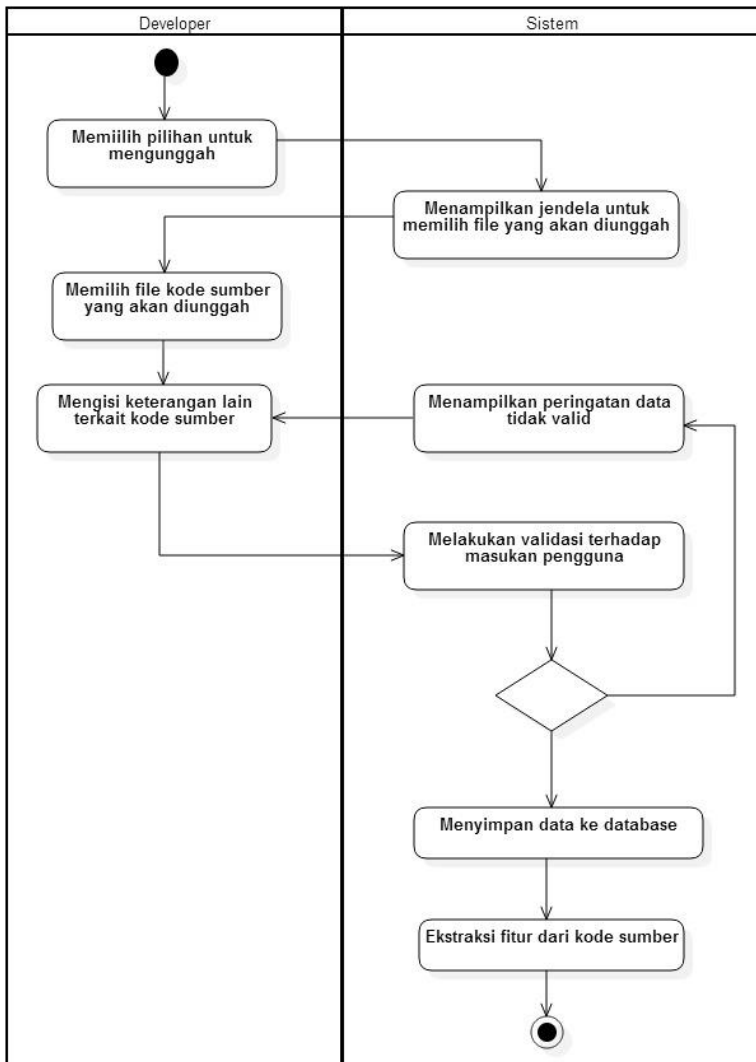
Nama	Mengunggah Kode Sumber
Kode	UC-0001
Deskripsi	Mengunggah kode sumber dari perangkat lunak yang akan dinilai kualitasnya dan menyimpannya ke dalam basis data. Sehingga dapat dilakukan analisa lebih lanjut
Tipe	Fungsional
Pemicu	Pengguna memilih menu untuk mengunggah kode sumber
Aktor	Pengguna
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan untuk mengunggah. 2. Sistem menampilkan jendela untuk memilih <i>file</i> yang akan diunggah. 3. Pengguna memilih <i>file</i> kode sumber yang akan diunggah. 4. Pengguna mengisi keterangan lain terkait dengan kode sumber. 5. Sistem melakukan validasi terhadap masukan pengguna <p>A5. Data masukan pengguna tidak valid</p> <ol style="list-style-type: none"> 6. Sistem menganalisa kode sumber 7. Sistem menyimpan data ke basis data
- Kejadian Alternatif	<p>A5. Data masukan pengguna tidak valid</p> <p>A5.1 Sistem menampilkan pesan bahwa data masukan tidak valid</p>
Kondisi Akhir	Kode sumber tersimpan ke dalam basis data



Gambar 3. 2 Diagram Urutan Mengunggah Perangkat Lunak



Gambar 3. 3 Diagram Urutan Mengunggah Perangkat Lunak (Kejadian Alternatif)



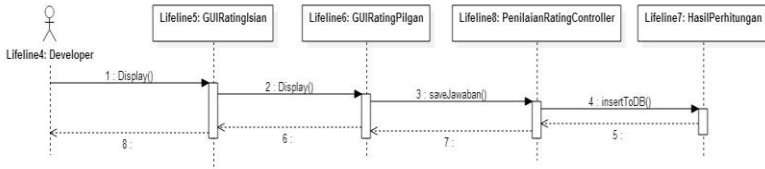
Gambar 3. 4 Diagram Aktivitas Mengunggah Kode Sumber

3.1.3.2. Mengisi Keterangan Terkait Perangkat Lunak

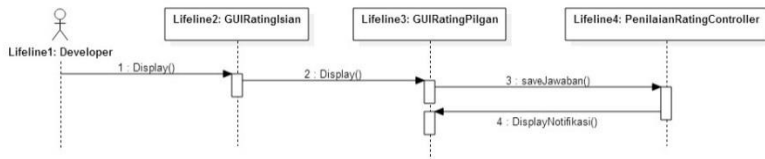
Setelah pengguna mengunggah perangkat lunak ke sistem, sistem akan menampilkan halaman yang berisi keterangan-keterangan terkait perangkat lunak yang harus diisi oleh pengguna. Kemudian, sistem akan menyimpan jawaban tersebut ke dalam basis data. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.3. Diagram aktivitas kejadian normal, kejadian alternatif dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.5, Gambar 3.6 dan Gambar 3.7.

Tabel 3. 3 Spesifikasi Kasus Penggunaan Mengisi Keterangan Terkait Perangkat Lunak

Nama	Mengisi Keterangan Terkait Perangkat Lunak
Kode	UC-0002
Deskripsi	Use case ini digunakan untuk mendapatkan informasi terkait dengan perangkat lunak yang akan dinilai kualitasnya, seperti basis data yang digunakan, dsb.
Tipe	Fungsional
Pemicu	Pengguna memilih pilihan untuk menyimpan perangkat lunak
Aktor	Pengguna
Kondisi Awal	Perangkat lunak berhasil disimpan
Aliran: - Kejadian Normal	1. Pengguna memilih pilihan untuk menyimpan perangkat lunak. 2. Sistem menampilkan pertanyaan terkait perangkat lunak. 3. Pengguna menjawab pertanyaan. 4. Sistem mengecek validitas jawaban. A4. Data yang dimasukkan pengguna tidak valid 5. Sistem menyimpan data ke basis data.
- Kejadian Alternatif	A4. Data yang dimasukkan pengguna tidak valid A4.1 Sistem akan menampilkan peringatan bahwa data yang dimasukkan tidak valid
Kondisi Akhir	Jawaban pertanyaan berhasil disimpan



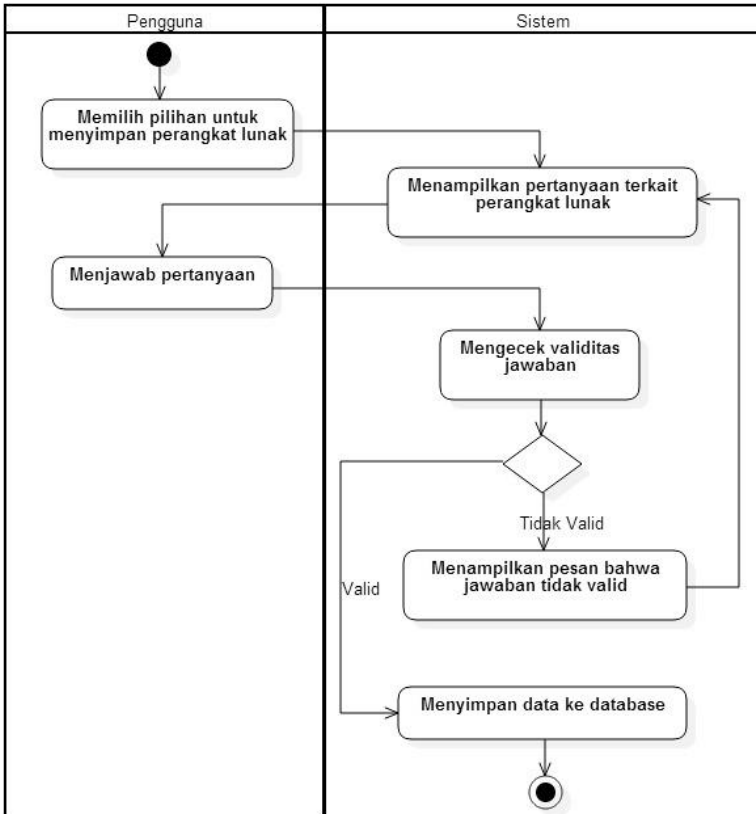
Gambar 3. 5 Diagram Urutan Mengisi Keterangan Terkait Perangkat Lunak



Gambar 3. 6 Diagram Urutan Mengisi Keterangan Terkait Perangkat Lunak (Kejadian Alternatif)

3.1.3.3. *Melihat Daftar Perangkat Lunak*

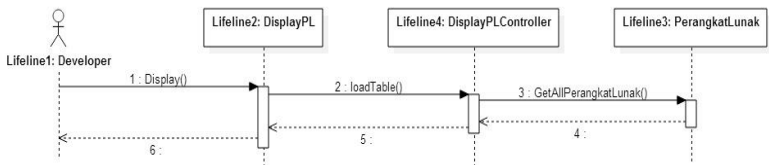
Sistem dapat menampilkan daftar perangkat lunak yang disimpan di dalam basis data sistem dalam bentuk tabel pada halaman kakas bantu. Sistem akan menampilkan data-data terkait perangkat lunak, seperti nama perangkat lunak, deskripsi, dan rata-rata nilai kualitas, dan kategori kualitas perangkat lunak berdasarkan nilai kualitasnya. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.4. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.8 dan Gambar 3.9.



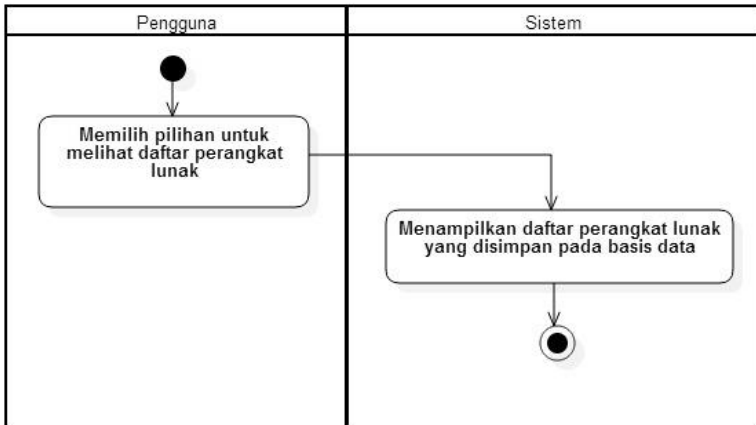
Gambar 3. 7 Diagram Aktivitas Mengisi Keterangan Terkait Perangkat Lunak

Tabel 3. 4 Spesifikasi Kasus Penggunaan Melihat Daftar Perangkat Lunak

Nama	Melihat Daftar Perangkat Lunak
Kode	UC-0003
Deskripsi	Menampilkan daftar perangkat lunak yang disimpan oleh sistem dalam basis data
Tipe	Fungsional
Pemicu	Pengguna memilih menu untuk melihat daftar perangkat lunak
Aktor	Pengguna
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan untuk melihat daftar perangkat lunak 2. Sistem menampilkan daftar perangkat lunak yang disimpan pada basis data.
- Kejadian Alternatif	-
Kondisi Akhir	Detail penilaian ditampilkan ke layar



Gambar 3. 8 Diagram Urutan Melihat Daftar Perangkat Lunak



Gambar 3. 9 Diagram Aktivitas Melihat Daftar Perangkat Lunak

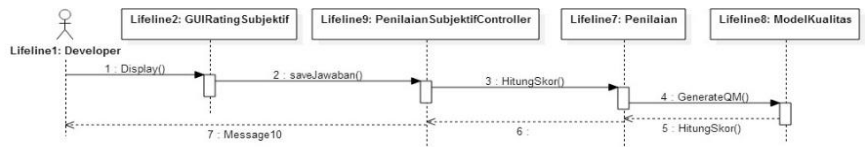
3.1.3.4. Mengisi Kuesioner Subjektif

Untuk melakukan penilaian, sistem menampilkan daftar pertanyaan subjektif yang harus diisi oleh pengguna. Sistem menerima masukan berupa perangkat lunak yang dipilih untuk dinilai. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.5. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.10 dan Gambar 3.11.

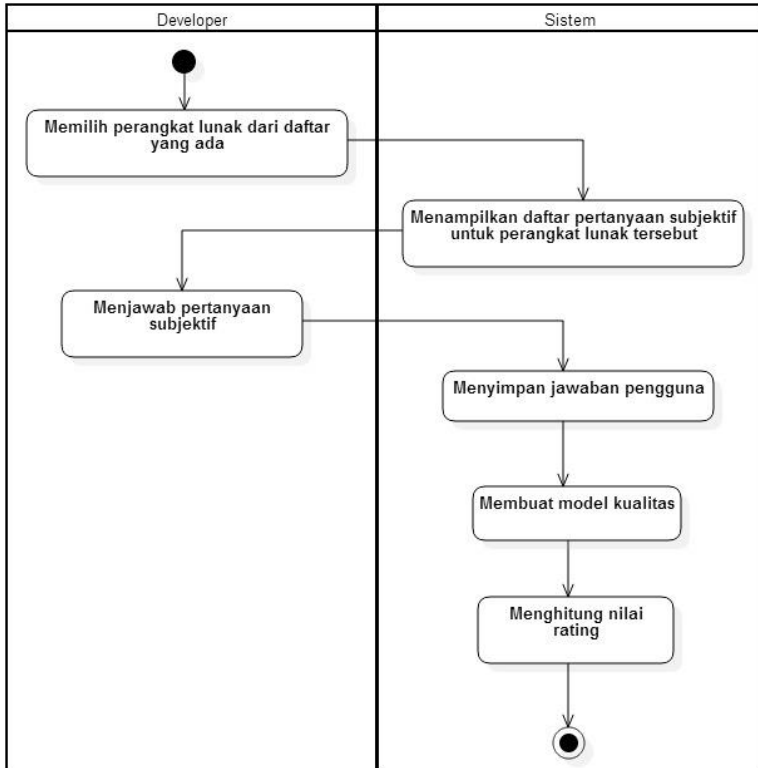
Tabel 3. 5 Spesifikasi Kasus Penggunaan Mengisi Kuesioner Subjektif

Nama	Mengisi Kuesioner Subjektif
Kode	UC-0004
Deskripsi	Selain dari kode sumber, untuk menilai kualitas perangkat lunak diperlukan jawaban subjektif dari pengguna terkait dengan penting tidaknya suatu aspek dalam menentukan kualitas perangkat lunak. Use case ini akan menyimpan jawaban subjektif dari pengguna.
Tipe	Fungsional

Pemicu	Pengguna memilih menu untuk mengunggah kode sumber
Aktor	Pengguna
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih perangkat lunak dari daftar yang ada. 2. Sistem menampilkan daftar pertanyaan subjektif untuk perangkat lunak tersebut. 3. Pengguna menjawab pertanyaan subjektif. 4. Sistem menyimpan jawaban dari pengguna. 5. Sistem membuat model kualitas 6. Sistem menghitung nilai rating.
- Kejadian Alternatif	-
Kondisi Akhir	Detail penilaian ditampilkan ke layar



Gambar 3. 10 Diagram Urutan Mengisi Kuesioner Subjektif



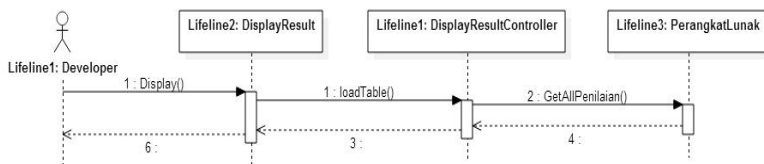
Gambar 3. 11 Diagram Aktivitas Mengisi Kuesioner Subjektif

3.1.3.5. *Melihat Detail Penilaian Kualitas Perangkat Lunak*

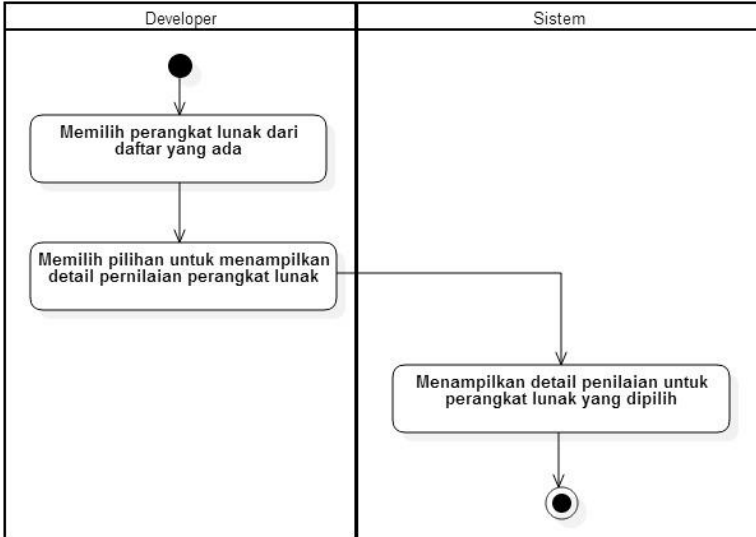
Sistem dapat menampilkan detail penilaian kualitas suatu perangkat lunak. Sistem menerima masukan berupa perangkat lunak yang dipilih untuk ditampilkan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.6. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.12 dan Gambar 3.13.

**Tabel 3. 6 Spesifikasi Kasus Penggunaan Melihat Detail
Penilaian Kualitas Perangkat Lunak**

Nama	Melihat Detail Penilaian Kualitas Perangkat Lunak
Kode	UC-0005
Deskripsi	Menampilkan hasil penghitungan kualitas perangkat lunak dan detail penilaian yang telah dilakukan untuk aplikasi tersebut.
Tipe	Fungsional
Pemicu	Pengguna memilih pilihan untuk melihat detail penilaian
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan daftar perangkat lunak yang disimpan oleh sistem
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih perangkat lunak dari daftar yang disediakan 2. Pengguna memilih pilihan untuk melihat hasil penilaian. 3. Sistem menampilkan detail penilaian untuk perangkat lunak yang dipilih.
- Kejadian Alternatif	-
Kondisi Akhir	Detail penilaian ditampilkan ke layar



**Gambar 3. 12 Diagram Urutan Melihat Detail Penilaian
Kualitas Perangkat Lunak**



Gambar 3. 13 Diagram Aktivitas Melihat Detail Penilaian Kualitas Perangkat Lunak

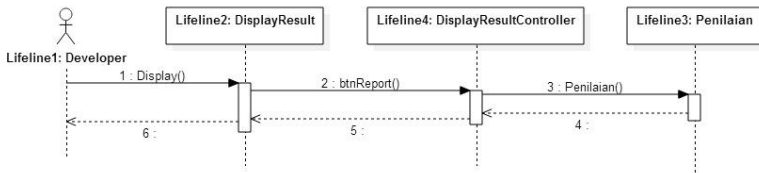
3.1.3.6. Menampilkan Laporan Penilaian Perangkat Lunak

Sistem dapat menampilkan detail penilaian kualitas suatu perangkat lunak. Sistem menerima masukan berupa perangkat lunak yang dipilih untuk ditampilkan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.7. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini dapat dilihat pada Gambar 3.14 dan Gambar 3.15.

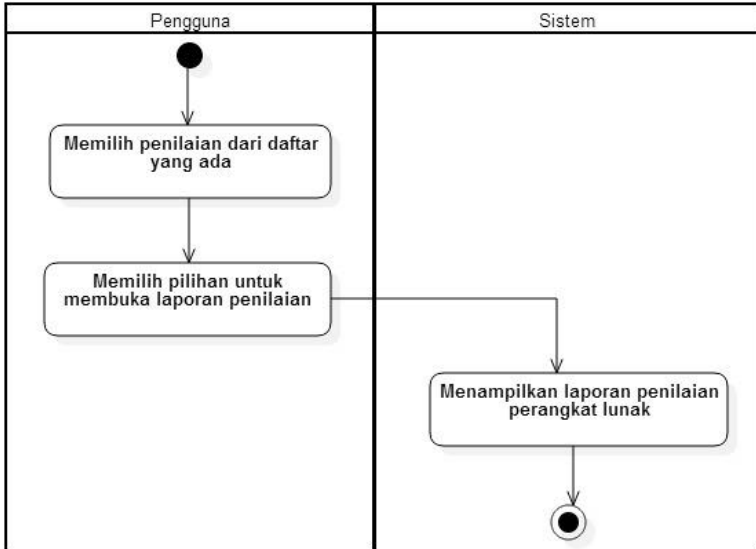
Tabel 3. 7 Spesifikasi Kasus Penggunaan Melihat Detail Penilaian Kualitas Perangkat Lunak

Nama	Menampilkan laporan penilaian perangkat lunak
Kode	UC-0006

Deskripsi	Menampilkan hasil penghitungan kualitas perangkat lunak dan detail penilaian yang telah dilakukan untuk aplikasi tersebut.
Tipe	Fungsional
Pemicu	Pengguna memilih pilihan untuk melihat detail penilaian
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan daftar penilaian yang tersedia untuk suatu perangkat lunak
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih penilaian dari daftar yang ada 2. Pengguna memilih pilihan untuk melihat laporan hasil penilaian perangkat lunak 3. Sistem menampilkan laporan untuk penilaian perangkat lunak yang dipilih.
- Kejadian Alternatif	-
Kondisi Akhir	Detail penilaian ditampilkan ke layar



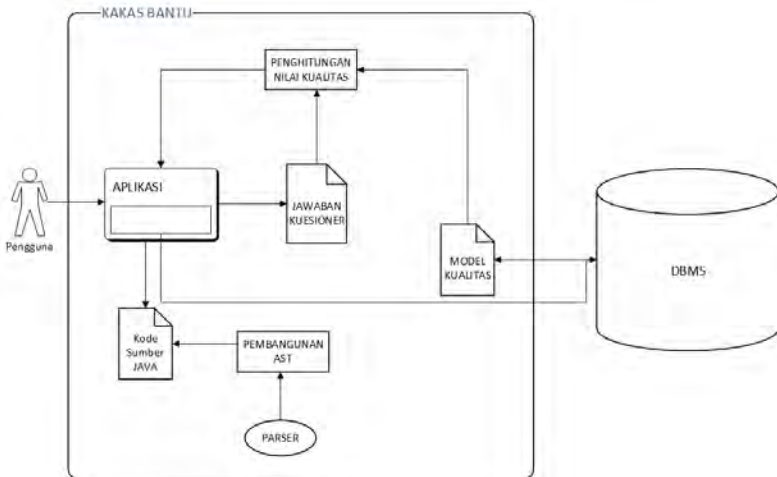
Gambar 3. 14 Diagram Urutan Melihat Laporan Kualitas Perangkat Lunak



Gambar 3. 15 Diagram Aktivitas Melihat Detail Penilaian Kualitas Perangkat Lunak

3.1.4. Spesifikasi Kebutuhan Perangkat Lunak

Bagian ini berisi semua kebutuhan perangkat lunak yang diuraikan secara rinci dalam bentuk diagram kasus, diagram urutan, dan diagram aktivitas. Masing-masing diagram menjelaskan perilaku atau sifat dari kaskas bantu ini. Kebutuhan perangkat lunak dalam kaskas bantu ini mencakup kebutuhan fungsional saja. Pada bab ini juga dijelaskan tentang spesifikasi terperinci pada masing-masing kebutuhan fungsional. Rincian spesifikasi dari kasus penggunaan disajikan dalam bentuk tabel. Diagram arsitektur dari sistem digambarkan pada Gambar 3.16.



Gambar 3. 16 Diagram Arsitektur Sistem

Gambar 3.16 menjelaskan bahwa kakas bantu yang dibangun merupakan suatu sistem berbasis *desktop*. Sehingga seluruh elemen sistem berada di dalam satu komputer. Secara umum, ada dua bagian penting dari arsitektur ini. Bagian pertama merupakan aplikasi, yang berisikan data dan elemen-elemen terkait kakas bantu tersebut, seperti *file* kode sumber, kode sumber untuk menjalankan proses penghitungan kualitas perangkat lunak, dan lain-lain. Bagian yang kedua adalah DBMS. DBMS menyimpan data-data yang bersifat permanen dan dapat digunakan sewaktu-waktu, seperti model kualitas, fuzifikasi, dan sebagainya.

3.1.4.1. *Kebutuhan Fungsional*

Kebutuhan fungsional berisi proses-proses yang harus dimiliki sistem. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan dan reaksi sistem terhadap masukan atau pada situasi tertentu. Daftar kebutuhan fungsional dapat dilihat pada Tabel 3.8 berikut.

Tabel 3. 8 Daftar Kebutuhan Fungsional Perangkat Lunak

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Menyimpan kode sumber	Sistem dapat menyimpan perangkat lunak yang akan dinilai kualitasnya untuk dilakukan analisis.
F-0002	Menyimpan informasi lain terkait perangkat lunak	Sistem dapat menyimpan informasi lain terkait perangkat lunak sebagai elemen penilaian kualitas perangkat lunak.
F-0003	Mengekstraksi fitur dari kode sumber	Sistem dapat mengekstraksi fitur dari kode sumber secara otomatis.
F-0004	Menampilkan daftar perangkat lunak	Sistem dapat menampilkan daftar perangkat lunak yang disimpan oleh sistem.
F-0005	Menyimpan jawaban subjektif	Sistem dapat menyimpan jawaban subjektif dari pengguna sebagai elemen penilaian kualitas perangkat lunak.
F-0006	Menghitung kualitas perangkat secara otomatis	Sistem dapat menghitung nilai kualitas perangkat lunak menggunakan metode <i>fuzzy</i> triangular berdasarkan informasi-informasi yang didapat dari pengguna.
F-0007	Menampilkan detail penilaian suatu perangkat lunak	Sistem dapat menampilkan detail penilaian-penilaian yang telah dilakukan untuk suatu perangkat lunak
F-0008	Menampilkan laporan hasil penilaian perangkat lunak	Sistem dapat menampilkan laporan hasil penilaian suatu perangkat lunak dalam bentuk pdf

3.2. Perancangan Sistem

Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan diagram kelas,

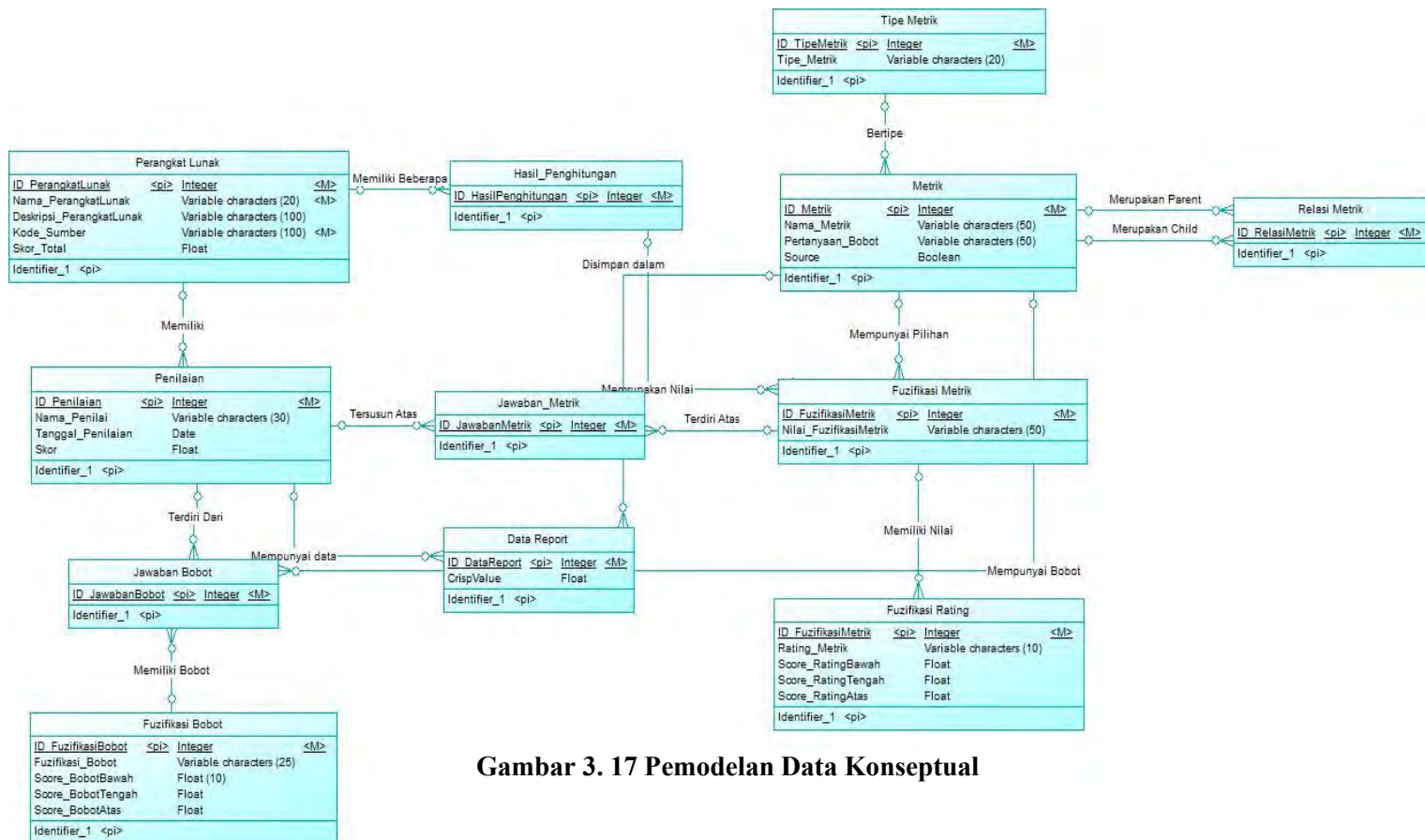
perancangan proses analisis, perancangan antarmuka, dan perancangan basis data.

3.2.1. Perancangan Basis Data

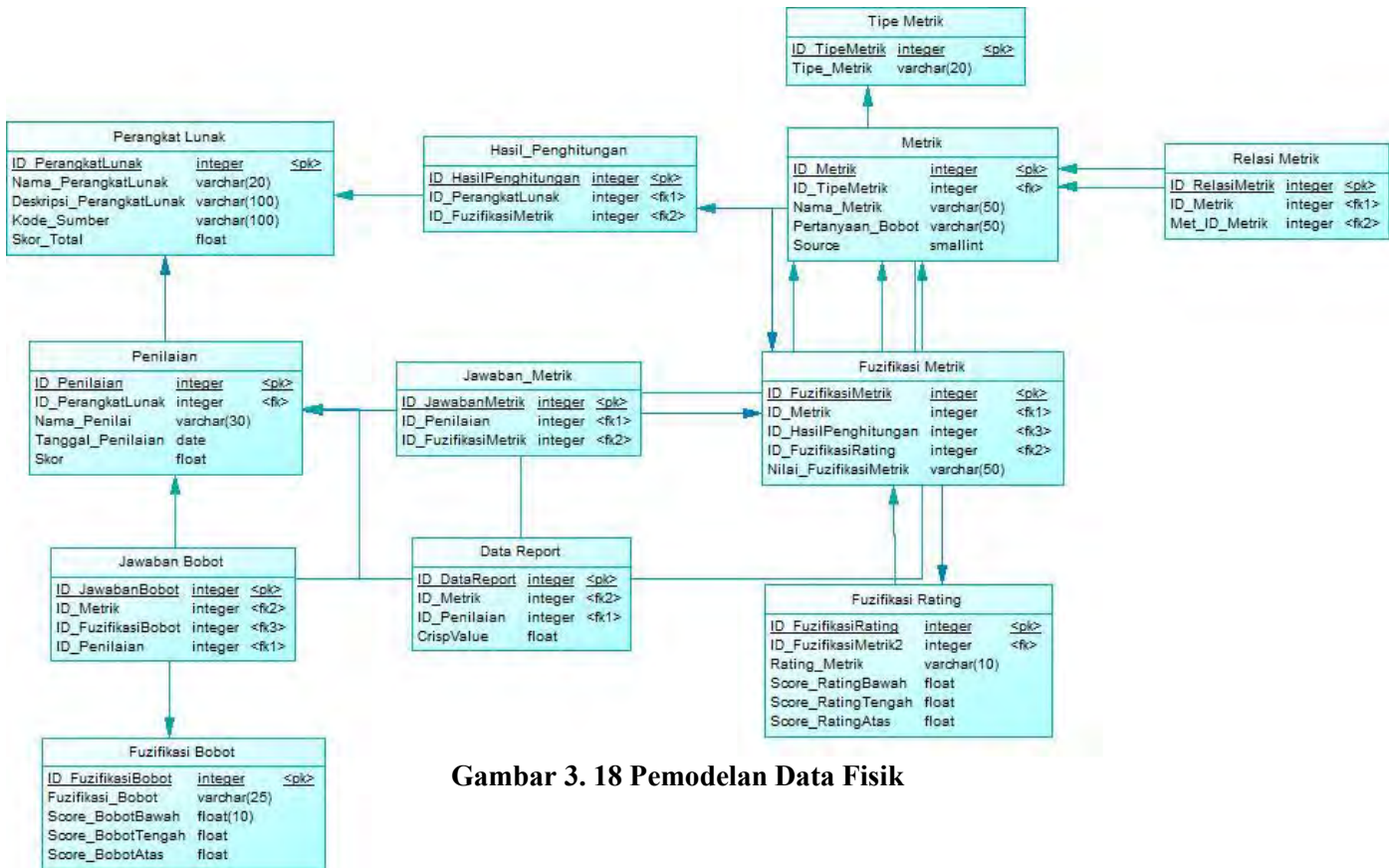
Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut.

Perangkat lunak yang digunakan untuk menyimpan data secara terstruktur dan mengatur bagaimana data tersebut disimpan, diorganisasi, diubah, dan diambil kembali disebut *Relational Database Management System* (RDBMS). Untuk sistem ini, RDBMS yang digunakan adalah H2.

Ilustrasi penyimpanan data dalam basis data ditunjukkan pada Gambar 3.17 dan Gambar 3.18. Gambar 3.17 menjelaskan pemodelan data secara konseptual (CDM) dan Gambar 3.18 menjelaskan pemodelan data secara nyata (PDM). Sementara itu, Tabel 3.9 menjelaskan detail spesifikasi basis data yang digunakan untuk aplikasi ini.



Gambar 3. 17 Pemodelan Data Konseptual



Gambar 3. 18 Pemodelan Data Fisik

Tabel 3. 9 Spesifikasi Basis Data

No	Tabel	Atribut	Tipe Data	Fungsi
1	Perangkat_Lunak	ID_PerangkatLunak	Integer	Menyimpan data perangkat lunak yang diunggah pengguna
		Nama_PerangkatLunak	Varchar(30)	
		Deskripsi_PerangkatLunak	Varchar(100)	
		Kode_Sumber	Varchar(100)	
		Total_Skor	Double	
2	Penilaian	ID_Penilaian	Integer	Menyimpan informasi penilaian kualitas yang pernah dilakukan dalam sistem
		ID_PerangkatLunak	Integer	
		Nama_Penilai	Varchar(30)	
		Tanggal_Penilaian	Date	
		Skor	Double	
3	Metrik	ID_Metrik	Integer	Menyimpan data metrik-metrik yang digunakan untuk penilaian kualitas perangkat lunak
		Nama_Metrik	Varchar(25)	
		Pertanyaan_Bobot	Varchar(100)	
		Source	Boolean	
		ID_TipeMetrik	Integer	
4	Fuzifikasi_Metrik	ID_FuzifikasiMetrik	Integer	Menyimpan nilai pilihan rating metrik
		ID_Metrik	Integer	
		ID_FuzifikasiRating	Integer	
		Fuzifikasi_Metrik	Varchar(30)	
5	Fuzifikasi_Rating	ID_FuzifikasiRating	Integer	Menyimpan nilai pembobotan
		Rating_Metrik	Varchar(10)	

No	Tabel	Atribut	Tipe Data	Fungsi
		Score_RatingBawah	Double	untuk pilihan rating metrik
		Score_RatingTengah	Double	
		Score_RatingAtas	Double	
6	Hasil_Penghitungan	ID_HasilPenghitungan	Integer	Menyimpan hasil penghitungan, informasi perangkat lunak, dan hasil ekstraksi dari kode sumber
		ID_PerangkatLunak	Integer	
		ID_FuzifikasiMetrik	Integer	
7	Fuzifikasi_Bobot	ID_FuzifikasiBobot	Integer	Menyimpan nilai pilihan bobot metrik
		Fuzifikasi_Bobot	Varchar(10)	
		Score_BobotBawah	Double	
		Score_BobotTengah	Double	
		Score_BobotAtas	Double	
8	Jawaban_Bobot	ID_JawabanBobot	Integer	Menyimpan jawaban kuesioner bobot metrik dari pengguna
		ID_Metrik	Integer	
		ID_FuzifikasiBobot	Integer	
		ID_Penilaian	Integer	
9	Jawaban_Metrik	ID_JawabanMetrik	Integer	Menyimpan jawaban kuesioner metrik subjektif dari pengguna
		ID_Penilaian	Integer	
		ID_FuzifikasiMetrik	Integer	
10	TipeMetrik	ID_TipeMetrik	Integer	

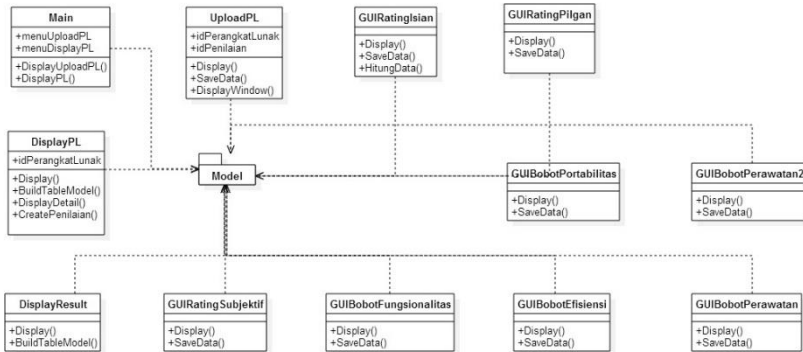
No	Tabel	Atribut	Tipe Data	Fungsi
		Tipe_Metrik	Varchar(20)	Menyimpan tingkatan metrik pada model kualitas
11	Relasi_Metrik	ID_RelasiMetrik	Integer	Menyimpan relasi antar metrik pada model kualitas
		ID_MetrikParent	Integer	
		ID_MetrikChild	Integer	
12	Data_Report	ID_DataReport	Integer	Menyimpan data-data yang akan digunakan dalam laporan penilaian
		ID_Penilaian	Integer	
		ID_Metrik	Integer	
		ID_Skor	S\Double	

3.2.2. Perancangan Diagram Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Pada subbab ini, hubungan dan perilaku antar kelas digambarkan dengan lebih jelas. Tiga lapisan pada arsitektur ini terdiri dari lapisan antarmuka, kontrol, dan data. Lapisan kontrol merupakan penghubung antara lapisan antarmuka dengan lapisan data. Subbab ini dibagi menjadi tiga bagian, yaitu diagram kelas untuk lapisan antarmuka, kontrol, dan data.

3.2.2.1. Diagram Kelas Lapisan Antarmuka

Diagram kelas lapisan antarmuka dapat dilihat pada Gambar 3.17. Terdapat dua belas kelas penyusun lapisan ini. Kelas-kelas pada lapisan ini merupakan kelas-kelas yang berupa elemen `JInternalFrame` yang berfungsi sebagai antarmuka dalam bahasa pemrograman Java. Masing-masing kelas ini berhubungan dengan kelas lainnya sesuai dengan urutan pemanggilan program. Selain itu, beberapa kelas pada lapisan ini juga berhubungan dengan kelas-kelas lain yang ada pada lapisan Object, seperti kelas penilaian, perangkat lunak yang berhubungan dengan kelas UploadPL.

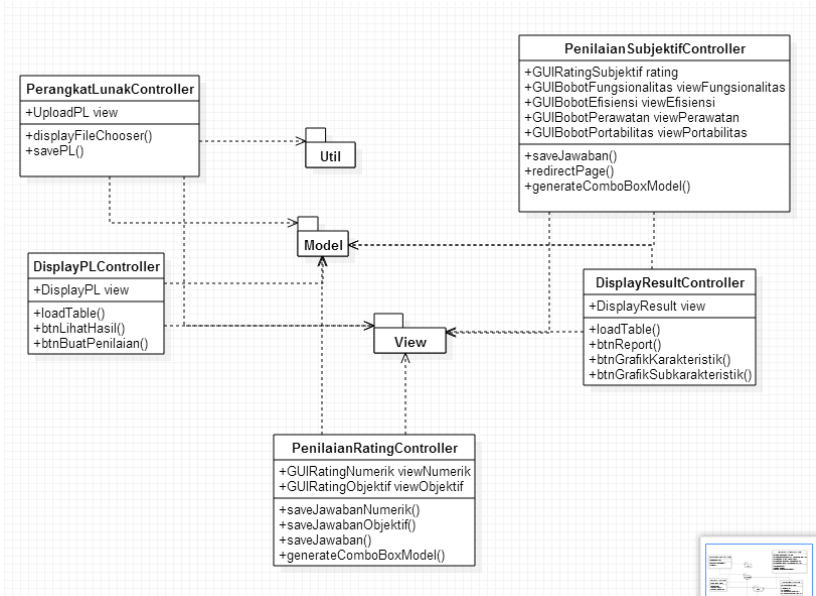


Gambar 3. 19 Diagram Kelas Lapisan Antarmuka

3.2.2.2. Diagram Kelas Lapisan Kontrol

Kelas-kelas penyusun lapisan kontrol dapat dilihat pada Gambar 3.20. Kelas-kelas tersebut antara lain kelas *PerangkatLunakController*, *PenilaianRatingController*, *DisplayPLController*, *DisplayResultController*, dan *PenilaianSubjektifController*. Kelas-kelas pada lapisan ini berfungsi sebagai pengatur kelas-kelas pada lapisan antarmuka dan sebagai penghubung antara kelas antarmuka dan representasi data, sekaligus untuk memisahkan kode sumber untuk pengolahan data dan untuk

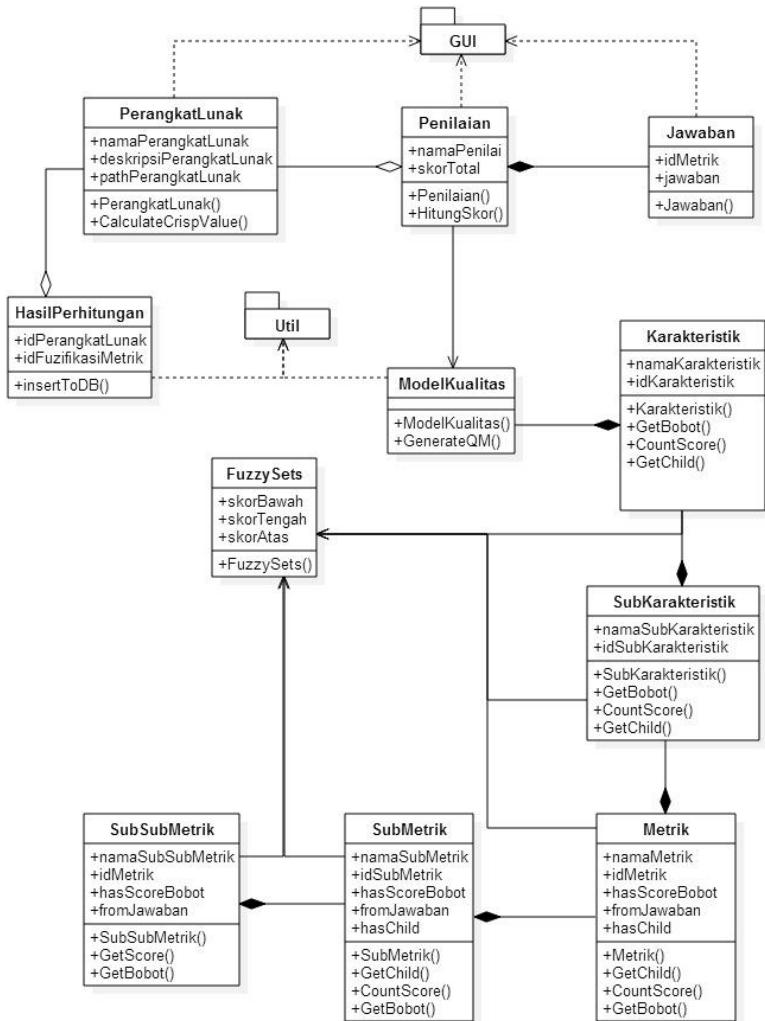
antarmuka yang nantinya akan berpengaruh pada *reusability* dari suatu kode sumber.



Gambar 3. 20 Diagram Kelas Lapisan Kontrol

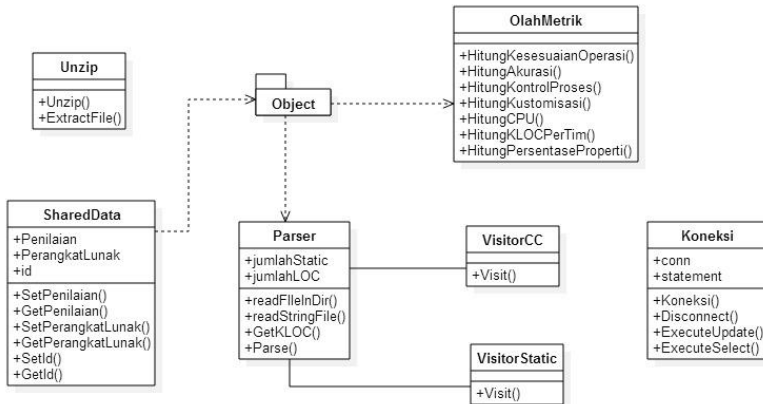
3.2.2.3. Diagram Kelas Lapisan Model

Kelas-kelas penyusun lapisan model dapat dilihat pada Gambar 3.21. Kelas-kelas tersebut antara lain kelas PerangkatLunak, Penilaian, Jawaban, ModelKualitas, dan Metrik. Kelas-kelas pada lapisan ini berfungsi sebagai representasi data, sekaligus untuk mengolah dan menyimpan data ke dalam basis data. Untuk merepresentasikan data, beberapa kelas memiliki hubungan dengan kelas lain sesuai dengan derajat hubungannya seperti kelas Penilaian dan PerangkatLunak yang memiliki hubungan agregasi, dan lain sebagainya.



Gambar 3. 21 Diagram Kelas Lapisan Model

3.2.2.4. Diagram Kelas Lapisan Util



Gambar 3. 22 Diagram Kelas Lapisan Util

Gambar 3.22. menunjukkan diagram untuk kelas lapisan Util. Kelas-kelas pada lapisan ini merupakan kelas pembantu untuk mengolah dan menjalankan sistem. Kelas-kelas yang berada pada lapisan ini contohnya adalah kelas Parser, kelas Koneksi, kelas SharedData, dan lain sebagainya. Kelas Parser merupakan kelas yang akan menghasilkan AST untuk perangkat lunak yang disimpan oleh pengguna, sementara itu kelas SharedData merupakan kelas yang berfungsi sebagai *Session* untuk menyimpan berbagai macam data yang dibutuhkan selama proses penilaian.

3.2.3. Perancangan Pengukuran Kualitas

Pada bagian ini akan dijelaskan proses penghitungan nilai kualitas perangkat lunak yang dikembangkan menggunakan bahasa pemrograman Java. Proses ini memanfaatkan menggunakan logika fuzzy, model kualitas ISO/IEC 9126 dan AST untuk mendapatkan nilai kualitas suatu perangkat lunak.

Kebutuhan perangkat lunak yang semakin tinggi memicu para pengembang/pemrogram, perusahaan yang bergerak di bidang IT

untuk membangun suatu perangkat lunak yang berkualitas. Namun, untuk mengukur kualitas perangkat lunak bukanlah perkara mudah. perangkat lunak merupakan produk yang tidak berwujud, sehingga perlu cara khusus untuk menilai kualitas perangkat lunak.

Untuk menilai kualitas perangkat lunak, digunakanlah suatu standar yang akan digunakan sebagai metrik. Dalam kasus ini, digunakanlah standar ISO/IEC 9126 dengan 6 karakteristik perangkat lunak yang berkualitas. Masing-masing karakteristik tersebut dipecah ke dalam subkarakteristik-subkarakteristik yang lebih bersifat spesifik. Lampiran H menjelaskan subkarakteristik dan metriknya beserta sumber ekstraksinya.

Subkarakteristik tersebut bersifat tiga macam, dapat diekstraksi dari kode sumber, berasal dari parameter lain di luar kode sumber, dan berasal dari subjektif pengembang/pemrogram. Subkarakteristik yang diekstrak dari kode sumber merupakan atribut-atribut teknis dari suatu perangkat lunak, seperti panjang kilokode (disingkat KLOC, *Kilo Lines of Code*), kompleksitas siklomatik, dan lain sebagainya. Metrik yang didapatkan dari parameter lain di luar kode sumber, contohnya adalah persentase operasi yang sesuai, persentase jumlah CPU yang tidak terpakai, dsb. Sementara itu, subkarakteristik yang bersifat subjektif merupakan atribut-atribut dari perangkat lunak yang bersifat non-teknis, seperti kemampuan tim pengembang, pengalaman dan kemampuan manajer proyek, dan lain sebagainya. Ketiga macam subkarakteristik ini akan berpengaruh terhadap kualitas suatu perangkat lunak.

Untuk mendapatkan atribut-atribut dari kode sumber untuk suatu perangkat lunak, pengguna perlu memasukkan kode sumber ke dalam kakas bantu ini. Setelah itu, kode sumber yang dimasukkan tersebut akan dibaca dan diekstraksi atribut-atributnya menggunakan metode Pohon Sintaks Abstrak (AST) dan pola *Visitor*. Atribut-atribut yang didapatkan dari proses ekstraksi kode sumber tersebut selanjutnya akan digunakan untuk mengukur subkarakteristik dalam kualitas perangkat lunak. Sementara itu, untuk mendapatkan nilai dari

subkarakteristik yang bersifat subjektif, dilakukan proses pengisian kuesioner langsung terhadap pengembang aplikasi yang bersangkutan.

Sementara itu, untuk mendapatkan metrik yang di luar kode sumber, diperlukan suatu kuesioner isian yang nantinya akan diolah oleh aplikasi menjadi satu nilai yang mewakili metrik tersebut dan digunakan dalam fungsi keanggotaan *fuzzy*. Tabel 3.10 menjelaskan nilai-nilai yang diperlukan untuk mendapatkan nilai untuk suatu metrik.

Tabel 3. 10 Nilai dan Rumus yang Digunakan untuk Metrik yang Didapatkan Diluar Kode Sumber

Metrik	Nilai yang dimasukkan oleh pengguna	Rumus
Persentase operasi yang sesuai	Jumlah operasi yang tidak sesuai	$1 - \left(\frac{\text{Jumlah operasi yang tidak sesuai}}{\text{Jumlah operasi yang disediakan}} \right)$
	Jumlah operasi yang disediakan	
Persentase operasi yang akurasiya terpenuhi	Jumlah operasi yang akurasiya terpenuhi	$\frac{\text{Jumlah operasi yang akurasiya terpenuhi}}{\text{Jumlah operasi yang disediakan}} \times 100\%$
	Jumlah operasi yang disediakan	
Persentase pengontrolan akses yang disediakan	Jumlah pengontrolan akses yang disediakan	$\frac{\text{Jumlah pengontrolan akses disediakan}}{\text{Jumlah pengontrolan akses yang dibutuhkan}}$
	Jumlah pengontrolan akses yang dibutuhkan	

Metrik	Nilai yang dimasukkan oleh pengguna	Rumus
Tingkat fitur yang dapat dikustomisasi	Jumlah fitur yang dapat dikustomisasi	$1 - \left(\frac{1}{\text{Jumlah fitur yang dapat dikustomisasi}} \right)$
Persentase CPU yang tidak digunakan	Penggunaan CPU untuk menjalankan perangkat lunak	$1 - \left(\frac{\text{Penggunaan CPU}}{100} \right)$
Jumlah KLOC per anggota tim	Jumlah anggota tim	$\frac{\text{Jumlah KLOC}}{\text{Jumlah anggota tim}}$
Persentase <i>property</i> perangkat lunak yang dapat dikustomisasi	Jumlah perangkat lunak perangkat lunak yang dapat dikustomisasi	$\frac{\text{Jumlah } \textit{property} \text{ yang dapat dikustomisasi}}{\text{Jumlah } \textit{property} \text{ perangkat lunak}}$
	Jumlah <i>property</i> perangkat lunak	

Supaya proses evaluasi kualitas perangkat lunak dapat terkuantifikasi dengan baik, digunakanlah metode logika *fuzzy*. Setiap nilai yang mungkin dihasilkan dari hasil ekstraksi kode sumber diklasifikasikan ke dalam suatu nilai, tergantung pada fungsi keanggotaan yang berlaku untuk suatu subkarakteristik. Nilai yang dihasilkan dari fungsi keanggotaan *fuzzy* ini berupa nilai angka, dan nilai inilah yang disebut sebagai pemeringkatan *fuzzy*. Untuk menghitung kualitas perangkat lunak, kakas bantu memerlukan satu nilai lain yang nantinya disebut pembobotan *fuzzy*. Pembobotan *fuzzy* merupakan nilai yang merepresentasikan tingkat pengaruh suatu karakteristik atau subkarakteristik terhadap kualitas perangkat lunak menurut sudut pandang pengguna. Untuk mendapatkan nilai dari suatu

subkarakteristik, dilakukan proses perkalian *dot matrix* antara pemeringkatan *fuzzy* dan pembobotan *fuzzy*. Proses tersebut dilakukan mulai dari tingkat subkarakteristik, karakteristik, hingga dihasilkannya nilai akhir yang disebut sebagai peringkat *net*. Nilai ini bukanlah nilai akhir, sehingga untuk mendapatkan nilai yang sebenarnya dilakukan proses *defuzzification*. Nilai dari proses inilah yang merepresentasikan kualitas suatu perangkat lunak.

Secara umum, aliran kerja yang dilakukan kakas bantu untuk mengevaluasi kualitas perangkat lunak tampak pada Gambar 3.23. Penjelasan dari masing-masing proses tersebut adalah sebagai berikut.

a. *Input* kode sumber dan kuesioner

Pengguna memasukkan kode sumber dari perangkat lunak yang akan dievaluasi kualitasnya, dan pengguna mengisi kuesioner-kuesioner terkait dengan perangkat lunak tersebut. Selain itu, pengguna juga akan memasukkan nilai pembobotan *fuzzy*.

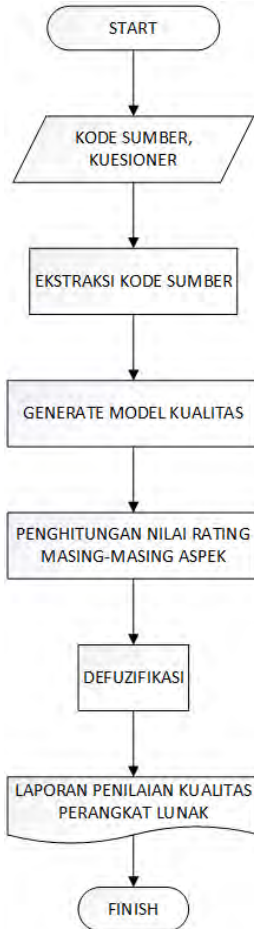
b. Pembacaan kode sumber dan hasil kuesioner pengguna

Setelah pengguna memasukkan kode sumber, aplikasi akan membaca kode sumber untuk diproses selanjutnya. Selain itu, aplikasi akan membaca jawaban kuesioner yang diisi oleh pengguna. Kuesioner terdiri dari tiga jenis, kuesioner untuk mengisi nilai subkarakteristik subjektif, kuesioner untuk mengisi nilai subkarakteristik objektif diluar kode sumber dan kuesioner untuk mendapatkan nilai pembobotan *fuzzy*. Hasil kuesioner yang dimasukkan pengguna langsung disimpan ke dalam basis data, sehingga dapat dipanggil kembali sewaktu-waktu. Data kuesioner pengguna disimpan dalam bentuk *fuzzy triangular*.

c. Ekstraksi atribut-atribut kode sumber

Dalam proses pembacaan kode sumber, didapatkan pula informasi-informasi yang tersimpan didalamnya. Dalam kakas bantu ini, fitur yang diekstrak dari kode sumber adalah panjang kilokode (KLOC), jumlah variabel global, dan jumlah kompleksitas siklomatik. Cara ekstraksi

fitur kode sumber dapat dilihat pada Tabel 3.11 di bawah ini.



Gambar 3. 23 Diagram Alir Aplikasi

Tabel 3. 11 Cara Ekstraksi Kode Sumber

Fitur	Cara Ekstraksi
KLOC	Menghitung jumlah baris pada kode program tanpa baris kosong dan <i>comment</i>
Jumlah variabel global	Menghitung jumlah variabel statis dalam kode program
Kompleksitas siklomatik	Menghitung jumlah <i>statement</i> pada subbab 2.6 dalam kode sumber

d. Pembangunan Model Kualitas

Hasil ekstraksi kode sumber dan jawaban kuesioner yang telah diisi oleh pengguna selanjutnya dipetakan ke dalam subkarakteristik-subkarakteristik sesuai dengan standar ISO/IEC 9126. Hal ini dilakukan untuk mendapatkan parameter mana saja yang mempengaruhi suatu subkarakteristik kualitas perangkat lunak. Parameter-parameter ini nantinya akan direpresentasikan ke dalam beberapa kelas sesuai dengan tingkatan levelnya seperti pada Lampiran H, yaitu kelas Karakteristik, Subkarakteristik, Metrik, Submetrik, dan Subsubmetrik.

e. Penghitungan nilai rating

Jawaban kuesioner yang dimasukkan oleh pengguna disimpan dalam basis data dalam bentuk sudah terfuzifikasi, dimana setiap masing-masing jawaban untuk suatu parameter memiliki nilai rating dan fuzzysets masing-masing. Begitu juga dengan nilai bobot yang dimasukkan. Tabel 3.12 dan Tabel 3.13 menjelaskan fuzzysets untuk masing-masing rating dan bobot yang digunakan dalam penghitungan kualitas menggunakan kakas bantu ini.

Tabel 3. 12 Fuzzy Traingular untuk Pemingkatan Metrik

<i>Rating</i>	<i>Fuzzy Rating</i>
Sangat Rendah	(0.0, 0.1, 0.3)
Rendah	(0.1, 0.3, 0.5)
Cukup	(0.3, 0.5, 0.7)
Tinggi	(0.5, 0.7, 0.9)
Sangat Tinggi	(0.7, 0.9, 1.0)

Tabel 3. 13 Fuzzy Traingular untuk Bobot Metrik

<i>Bobot</i>	<i>Fuzzy Bobot</i>
Sangat Rendah	(0.0, 0.0, 0.25)
Rendah	(0.0, 0.25, 0.5)
Cukup	(0.25, 0.5, 0.75)
Tinggi	(0.5, 0.75, 1.0)
Sangat Tinggi	(0.75, 1.0, 1.0)

Beberapa sub karakteristik akan menggambarkan suatu karakteristik, Untuk mendapatkan nilai karakteristik, aplikasi akan menggabungkan nilai dari subkarakteristik. Pada proses ini aplikasi akan membaca pembobotan *fuzzy* yang dimasukkan oleh pengguna melalui kuesioner dan akan melakukan proses penghitungan nilai rating suatu aspek. Jika suatu aspek tersebut sudah tidak memiliki aspek-aspek lain dibawahnya, maka nilai rating akan didapat dari jawaban

yang disimpan dalam basis data. Sebaliknya, maka akan dilakukan proses penghitungan dengan persamaan 3.1

$$rating = \sum ri \times wi \quad (3.1)$$

Sementara itu, penghitungan nilai bobot untuk suatu aspek tergantung dari metrik tersebut juga. Jika dalam suatu aspek induk hanya terdapat satu aspek, maka nilai bobot diset default, yaitu (1,1,1). Setelah nilai dari karakteristik-karakteristik didapatkan, selanjutnya aplikasi kaka bantu akan menghitung nilai *net quality* secara keseluruhan untuk perangkat lunak yang dimasukkan. Pada proses ini aplikasi juga akan membaca pembobotan *fuzzy* yang dimasukkan oleh pengguna melalui kuesioner.

f. *Defuzzification*

Untuk mendapatkan nilai kualitas yang sebenarnya, perlu dilakukan proses *defuzzification*. *Defuzzification* dilakukan dengan memasukkan nilai *net quality* yang berupa matriks ke dalam suatu rumus. Rumus yang digunakan dapat dilihat pada persamaan 2.3 pada bab sebelumnya.

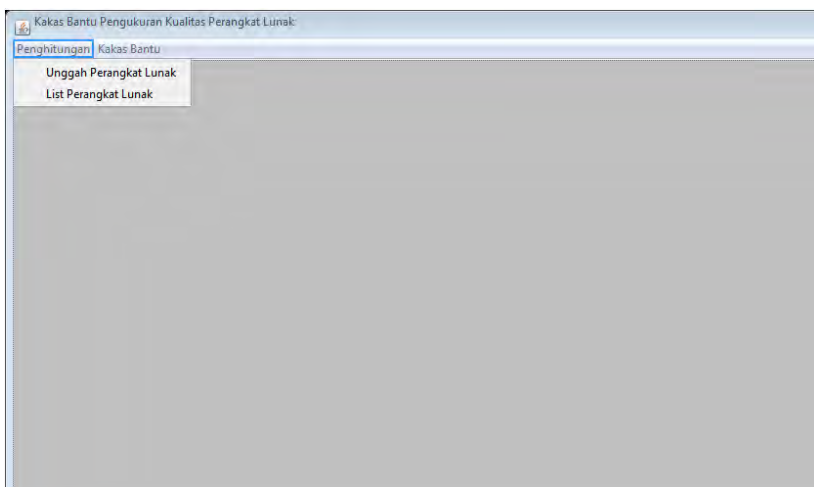
g. Menampilkan nilai kualitas perangkat lunak

Sebagai luaran dari kaka bantu ini, nilai kualitas yang ditampilkan merupakan nilai yang didapat dari proses-proses yang sebelumnya telah dilakukan. Nilai yang dikeluarkan adalah berupa angka dimana nilai tersebut merupakan hasil dari proses *defuzzification*. Nilai kualitas perangkat lunak ditampilkan dalam halaman berupa tabel atau dapat diunduh dalam bentuk laporan berformat pdf.

3.2.4. Perancangan Antarmuka Pengguna

Bagian ini membahas rancangan tampilan antar muka pada sistem. Pada sistem ini terdapat satu tampilan utama yaitu tampilan pada bagian view dari IDE Eclipse.

3.2.4.1. Tampilan Halaman Utama



Gambar 3. 24 Rancangan Tampilan Utama

Halaman ini merupakan tampilan utama yang muncul ketika sistem pertama kali dijalankan. Pada halaman ini terdapat menu bar yang didalamnya terdapat pilihan-pilihan menu yang dapat diakses oleh pengguna. Pengguna dapat memilih untuk menjalankan fungsional sistem dengan cara memilih salah satu menu yang tersedia. Selain itu, halaman ini merupakan halaman induk dari semua halaman yang ada, karena semua fungsional sistem akan dilakukan melalui suatu jendela *InternalFrame* yang akan muncul pada elemen *DesktopPane* pada halaman ini. Rancangan tampilan dapat dilihat pada Gambar 3.24.

Tabel 3. 14 Spesifikasi Atribut Antar Muka Tampilan Utama

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	desktopPane	jDesktopPane	Sebagai elemen dasar bagi jInternalFrame untuk muncul di layar	<i>jInternalFrame</i>

2	menuUploadPL	MenuItem	Pilihan aksi untuk menampilkan halaman mengunggah perangkat lunak	<i>Event Clicked</i>
3	menuListPL	MenuItem	Pilihan aksi untuk menampilkan halaman daftar perangkat lunak	<i>Event Clicked</i>

3.2.4.2. *Halaman Mengunggah Perangkat Lunak*

Gambar 3. 25 Rancangan Tampilan Halaman Mengunggah Perangkat Lunak

Halaman ini merupakan tampilan yang muncul saat pengguna memilih menu untuk mengunggah perangkat lunak. Pada halaman ini terdapat tiga *textbox* yang digunakan untuk memasukkan keterangan perangkat lunak yang diunggah. Selain itu, pada halaman ini terdapat satu tombol yang digunakan untuk membuka jendela baru untuk memilih *file* kode sumber mana yang akan diunggah. Saat tombol simpan ditekan, maka sistem akan menyimpan informasi yang

dimasukkan oleh pengguna pada halaman ini ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.25.

Tabel 3. 15 Spesifikasi Atribut Antar Muka Halaman Mengunggah Perangkat Lunak

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	txtNamaPL	<i>Textbox</i>	Lokasi pengisian nama perangkat lunak	<i>Integer</i>
2	txtDeskripsi	<i>Textbox</i>	Lokasi pengisian deskripsi perangkat lunak	<i>Integer</i>
3	txtKodeSumber	<i>Textbox</i>	Lokasi untuk menampilkan path direktori kode sumber	<i>Integer</i>
4	btnPilih	<i>Button</i>	Pilihan aksi untuk menampilkan jendela untuk memilih kode sumber	<i>Event Clicked</i>
5	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan perangkat lunak	<i>Event Clicked</i>

3.2.4.3. Halaman Menampilkan Pertanyaan Rating Numerik

The screenshot shows a web form titled "DATA KETERANGAN PERANGKAT LUNAK". Below the title, there is a bold instruction: "Isi data-data berikut sesuai dengan perangkat lunak yang anda masukkan". The form contains ten rows, each with a text label on the left and a rectangular input field on the right. The labels are: "Jumlah operasi perangkat lunak yang tidak sesuai dengan kebutuhan", "Jumlah operasi perangkat lunak yang disediakan", "Jumlah operasi perangkat lunak yang akurasiya terpenuhi", "Jumlah pengontrolan akses yang disediakan perangkat lunak", "Jumlah pengontrolan akses yang dibutuhkan perangkat lunak", "Jumlah fitur perangkat lunak yang dapat dikustomisasi", "Persentase CPU yang digunakan untuk menjalankan perangkat lunak", "Jumlah anggota dalam tim pengembang perangkat lunak", "Jumlah property perangkat lunak yang dapat dikustomisasi", and "Jumlah property perangkat lunak". At the bottom of the form, there are three elements: a "Reset" button on the left, the text "1/2" in the center, and a "Next" button on the right. A small asterisk with the text "*arahkan kursor ke label untuk hint" is located at the bottom right corner of the form area.

Gambar 3. 26 Rancangan Halaman Menampilkan Pertanyaan Isian

Halaman ini merupakan tampilan yang muncul ketika pengguna berhasil mengunggah kode sumber ke dalam sistem. Halaman ini digunakan untuk mendapatkan keterangan-keterangan terkait perangkat lunak berupa nilai-nilai angka melalui 10 *textbox* yang tersedia dengan keterangan sesuai dengan label yang ada di sisi kirinya. Terdapat dua tombol pada halaman ini, yaitu tombol Reset untuk menghapus isi *textbox* secara bersamaan dan tombol Submit untuk menyimpan nilai-nilai tersebut untuk diolah lebih lanjut. Rancangan tampilan dapat dilihat pada Gambar 3.26.

**Tabel 3. 16 Spesifikasi Atribut Antar Muka Halaman
Menampilkan Pertanyaan Rating Numerik**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	txtOperasiTidakSesuai	<i>Textbox</i>	Lokasi pengisian jumlah operasi yang tidak sesuai	<i>Integer</i>
2	txtOperasiTersebut	<i>Textbox</i>	Lokasi pengisian jumlah operasi yang disediakan	<i>Integer</i>
3	txtAkurasi	<i>Textbox</i>	Lokasi pengisian jumlah operasi yang akurasinya memenuhi	<i>Integer</i>
4	txtAksesDisediakan	<i>Textbox</i>	Lokasi pengisian jumlah pengontrolan akses yang disediakan	<i>Integer</i>
5	txtAksesDibutuhkan	<i>Textbox</i>	Lokasi pengisian jumlah pengontrolan akses yang dibutuhkan	<i>Integer</i>
6	txtFiturKustomisasi	<i>Textbox</i>	Lokasi pengisian jumlah fitur yang dapat dikustomisasi	<i>Integer</i>
7	txtPenggunaanCPU	<i>Textbox</i>	Lokasi pengisian persentase penggunaan CPU saat aplikasi berjalan	<i>Integer</i>
8	txtAnggotaTim	<i>Textbox</i>	Lokasi pengisian jumlah anggota tim pengembang	<i>Integer</i>
9	txtPropertiKustom	<i>Textbox</i>	Lokasi pengisian jumlah properti perangkat lunak yang dapat dikustomisasi	<i>Integer</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
10	txtProperti	<i>Textbox</i>	Lokasi pengisian jumlah properti perangkat lunak	<i>Integer</i>
11	btnReset	<i>Button</i>	Pilihan aksi untuk mereset semua <i>textbox</i> pada halaman	<i>Event Clicked</i>
12	btnSubmit	<i>Button</i>	Pilihan aksi untuk mengolah data dan menyimpannya ke dalam basis data	<i>Event Clicked</i>

3.2.4.4. Tampilan Halaman Menampilkan Pertanyaan Rating Objektif

ASPEK OBJEKTIF PERANGKAT LUNAK

i data berikut sesuai dengan perangkat lunak yang diinputkan!

Basis data yang digunakan	<input type="text"/>	Pengalaman manajer proyek sebagai manajer proyek	<input type="text"/>
Spesifikasi prosesor software berjalan	<input type="text"/>	Ketersediaan sistem tracking fungsional dan perilaku sistem	<input type="text"/>
Jumlah versi yang telah dirilis	<input type="text"/>	Kompatibilitas pada sistem operasi lain	<input type="text"/>
Kemampuan teknis	<input type="text"/>	Pengalaman manajer proyek dalam bisnis IT	<input type="text"/>
Penggunaan tools dasar	<input type="text"/>	Jumlah package yang harus diinstall sebelumnya	<input type="text"/>
Pengalaman industri	<input type="text"/>	Jumlah package selain OS yang harus diinstall sebelumnya	<input type="text"/>
Ketersediaan dokumentasi	<input type="text"/>	Support untuk beberapa pengguna	<input type="text"/>
Level CMM	<input type="text"/>	Support perangkat lunak terhadap sumber daya luar	<input type="text"/>
		Tingkat keketatan akses	<input type="text"/>

2/2

*arahkan kursor ke lab

Gambar 3. 27 Rancangan Tampilan Halaman Menampilkan Pertanyaan Rating Objektif

Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai pada halaman sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan keterangan-keterangan terkait perangkat lunak yang berupa pilihan-pilihan yang

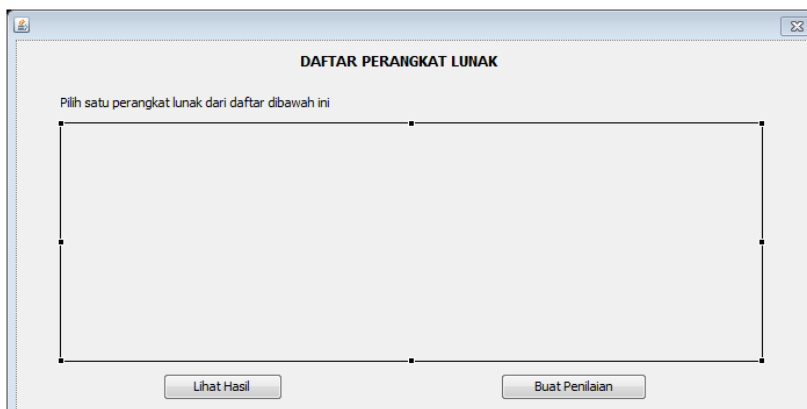
tersedia pada sistem melalui 17 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol Submit untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.27.

Tabel 3. 17 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Rating Pilihan Ganda

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboDB	<i>DropDownList</i>	Menampilkan pilihan basis data yang digunakan	<i>ListItem</i>
2	ComboKeketaan	<i>DropDownList</i>	Menampilkan pilihan keketatan akses	<i>ListItem</i>
3	ComboVersi	<i>DropDownList</i>	Menampilkan pilihan jumlahversi yang telah dirilis	<i>ListItem</i>
4	ComboProcessor	<i>DropDownList</i>	Menampilkan pilihan jenis prosesor minimal untuk menjalankan perangkat lunak	<i>ListItem</i>
5	ComboTeknis	<i>DropDownList</i>	Menampilkan pilihan kemampuan teknis yang dimiliki tim pengembang	<i>ListItem</i>
6	ComboTools	<i>DropDownList</i>	Menampilkan pilihan penggunaan tools dasar	<i>ListItem</i>
7	ComboIndustri	<i>DropDownList</i>	Menampilkan pilihan lama pengalaman industri	<i>ListItem</i>
8	ComboDokumentasi	<i>DropDownList</i>	Menampilkan pilihan ketersediaan dokumentasi	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
9	ComboCMM	<i>DropDownList</i>	Menampilkan pilihan level CMM	<i>ListItem</i>
10	ComboProyek	<i>DropDownList</i>	Menampilkan pilihan lama pengalaman proyek	<i>ListItem</i>
11	ComboTracking	<i>DropDownList</i>	Menampilkan pilihan ketersediaan sistem penelusuran	<i>ListItem</i>
12	ComboOS	<i>DropDownList</i>	Menampilkan pilihan kompatibilitas terhadap OS lain	<i>ListItem</i>
13	ComboIT	<i>DropDownList</i>	Menampilkan pilihan lama pengalaman di dunia IT	<i>ListItem</i>
14	ComboPackage OS	<i>DropDownList</i>	Menampilkan pilihan jumlah kebutuhan <i>package</i>	<i>ListItem</i>
15	ComboPackage	<i>DropDownList</i>	Menampilkan pilihan jumlah kebutuhan <i>package</i> selain OS	<i>ListItem</i>
16	ComboSupport	<i>DropDownList</i>	Menampilkan pilihan ketersediaan dukungan untuk berbagai pengguna	<i>ListItem</i>
17	ComboResource	<i>DropDownList</i>	Menampilkan pilihan ketersediaan dukungan untuk resource luar	<i>ListItem</i>
18	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan keterangan dari pengguna	<i>Event Clicked</i>

3.2.4.5. Halaman Tampilan Menampilkan Daftar Perangkat Lunak



Gambar 3. 28 Rancangan Tampilan Halaman Daftar Perangkat Lunak

Halaman ini merupakan tampilan yang muncul ketika pengguna memilih menu menampilkan daftar perangkat lunak pada halaman utama. Pada halaman ini terdapat tabel yang berfungsi untuk menampilkan data perangkat lunak yang disimpan oleh sistem. Pengguna dapat memilih untuk melihat daftar penilaian kualitas untuk perangkat lunak tertentu dengan menekan tombol yang ada di sebelah kiri bawah. Selain itu, pengguna juga dapat memilih untuk melakukan penilaian subjektif dengan menekan tombol yang ada di sebelah kanan bawah. Rancangan tampilan dapat dilihat pada Gambar 3.28.

Tabel 3. 18 Spesifikasi Atribut Antar Muka Tampilan Halaman Daftar Perangkat Lunak

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	table	<i>jTable</i>	Menampilkan data daftar perangkat lunak yang tersimpan dalam sistem	<i>TableModel</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
2	btnLihatHasil	button	Pilihan aksi untuk menampilkan detail penilaian kualitas perangkat lunak	Event Clicked
3	btnPenilaian	Button	Pilihan aksi untuk melakukan penilaian subjektif	Event Clicked

3.2.4.6. Halaman Menampilkan Pertanyaan Rating Subjektif

Gambar 3. 29 Rancangan Halaman Menampilkan Pertanyaan Rating Subjektif

Halaman ini merupakan tampilan yang muncul ketika pengguna memilih untuk melakukan penilaian kualitas terhadap perangkat lunak yang telah dipilih pada halaman sebelumnya. Halaman ini digunakan untuk mendapatkan keterangan-keterangan terkait perangkat lunak yang berupa pilihan-pilihan yang tersedia pada sistem melalui 10 *buttongroup* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. Informasi yang diisikan oleh

pengguna pada halaman ini bersifat subjektif, sehingga antara satu pengguna dengan pengguna lain bisa saja memiliki jawaban yang berbeda pada satu aspek. Terdapat tombol *Submit* untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.29.

Tabel 3. 19 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Rating Subjektif

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	btnPresisi	<i>ButtonGroup</i>	Menampilkan pilihan pemenuhan presisi yang dibutuhkan	<i>RadioItem</i>
2	btnFungsionalitas	<i>ButtonGroup</i>	Menampilkan pilihan pemenuhan standar fungsionalitas	<i>RadioItem</i>
3	btnUjiCoba	<i>ButtonGroup</i>	Menampilkan pilihan ketersediaan jumlah kasus uji coba yang cukup	<i>RadioItem</i>
4	btnEfisiensi	<i>ButtonGroup</i>	Menampilkan pilihan pemenuhan standar fungsionalitas	<i>RadioItem</i>
5	btnPortability	<i>ButtonGroup</i>	Menampilkan pilihan pemenuhan standar portabilitas	<i>RadioItem</i>
6	btnMaintainability	<i>ButtonGroup</i>	Menampilkan pilihan pemenuhan standar <i>maintainability</i>	<i>RadioItem</i>
7	btnKerjasama	<i>ButtonGroup</i>	Menampilkan pilihan kerjasama antar tim pengembang	<i>RadioItem</i>
8	btnKualitas	<i>ButtonGroup</i>	Menampilkan pilihan kualitas individu tim pengembang	<i>RadioItem</i>
9	btnPerforma	<i>ButtonGroup</i>	Menampilkan pilihan performa tim pengembang	<i>RadioItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
10	btnTracking	<i>ButtonGroup</i>	Menampilkan pilihan kemudahan dalam penelusuran sistem terdahulu	<i>RadioItem</i>
11	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan jawaban	<i>Event Clicked</i>

3.2.4.7. Halaman Menampilkan Pertanyaan Bobot Fungsionalitas

KUESIONER BOBOT FUNGSIONALITAS

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

Persentase operasi yang memenuhi akurasi yang dibutuhkan

Pemenuhan terhadap presisi yang dibutuhkan

Persentase pengontrolan akses yang disediakan

Tingkat keketatan akses

Kesesuaian dengan kebutuhan

Akurasi

Interoperability

Keamanan

Pemenuhan terhadap standar fungsionalitas

Fungsionalitas

Customizability

<<<First <<Prev 2/6 Next >> Last>>>

*arahkan kursor ke label untuk hint

Gambar 3. 30 Rancangan Halaman Menampilkan Pertanyaan Bobot Fungsionalitas

Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai pada halaman sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan bobot aspek fungsionalitas terhadap kualitas perangkat lunak melalui 11 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini

dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol *Submit* untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.30.

Tabel 3. 20 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Fungsionalitas

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboPersentaseOperasi	<i>DropDownList</i>	Menampilkan pilihan bobot untuk persentase operasi	<i>ListItem</i>
2	ComboPresisi	<i>DropDownList</i>	Menampilkan pilihan bobot untuk pemenuhan presisi	<i>ListItem</i>
3	ComboPengontrolanAkses	<i>DropDownList</i>	Menampilkan pilihan bobot untuk pengontrolan akses	<i>ListItem</i>
4	ComboKeketatanAkses	<i>DropDownList</i>	Menampilkan pilihan bobot untuk tingkat keketatan akses	<i>ListItem</i>
5	ComboKesesuaian	<i>DropDownList</i>	Menampilkan pilihan bobot untuk kesesuaian dengan kebutuhan	<i>ListItem</i>
6	ComboAkurasi	<i>DropDownList</i>	Menampilkan pilihan bobot untuk akurasi perangkat lunak	<i>ListItem</i>
7	ComboInteroperability	<i>DropDownList</i>	Menampilkan pilihan bobot untuk interoperabilitas	<i>ListItem</i>
8	ComboKeamanan	<i>DropDownList</i>	Menampilkan pilihan bobot untuk keamanan perangkat lunak	<i>ListItem</i>
9	ComboPemenuhan	<i>DropDownList</i>	Menampilkan pilihan bobot untuk pemenuhan standar fungsionalitas	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
10	ComboFungsionalitas	<i>DropDownList</i>	Menampilkan pilihan bobot untuk fungsionalitas perangkat lunak	<i>ListItem</i>
11	ComboKustomabilitas	<i>DropDownList</i>	Menampilkan pilihan bobot untuk kustomabilitas perangkat lunak	<i>ListItem</i>
12	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan jawaban pertanyaan	<i>Event Clicked</i>

3.2.4.8. Halaman Menampilkan Pertanyaan Bobot Efisiensi

KUESIONER BOBOT EFISIENSI

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

Jumlah variabel global	<input type="text"/>
Tipe translator	<input type="text"/>
Kemampuan minimal prosesor untuk menjalankan perangkat lunak	<input type="text"/>
Persentase CPU yang tidak terpakai	<input type="text"/>
Support perangkat lunak terhadap sumber daya luar	<input type="text"/>
Waktu	<input type="text"/>
Penggunaan sumber daya	<input type="text"/>
Pemenuhan terhadap standar efisiensi	<input type="text"/>
Skalabilitas	<input type="text"/>
Efisiensi	<input type="text"/>

<<<First <<Prev 3/6 Next>>> Last>>>

*arahkan kursor ke label untuk hint

Gambar 3. 31 Rancangan Halaman Menampilkan Pertanyaan Bobot Efisiensi

Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai bobot fungsionalitas pada halaman

sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan bobot aspek efisiensi perangkat lunak terhadap kualitas perangkat lunak melalui 10 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol Submit untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.31.

Tabel 3. 21 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Efisiensi

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboVariabel	<i>DropDownList</i>	Menampilkan pilihan bobot untuk jumlah variabel global	<i>ListItem</i>
2	ComboTranslator	<i>DropDownList</i>	Menampilkan pilihan bobot untuk tipe translator	<i>ListItem</i>
3	ComboProcessor	<i>DropDownList</i>	Menampilkan pilihan bobot untuk spesifikasi minimal processor	<i>ListItem</i>
4	ComboCPU	<i>DropDownList</i>	Menampilkan pilihan bobot untuk persentase CPU yang tidak terpakai	<i>ListItem</i>
5	ComboResource	<i>DropDownList</i>	Menampilkan pilihan bobot untuk dukungan perangkat lunak terhadap sumber daya luar	<i>ListItem</i>
6	ComboWaktu	<i>DropDownList</i>	Menampilkan pilihan bobot untuk waktu yang dibutuhkan perangkat lunak	<i>ListItem</i>
7	ComboSumber Daya	<i>DropDownList</i>	Menampilkan pilihan bobot penggunaan sumber daya	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
8	ComboPemenuhan	<i>DropDownList</i>	Menampilkan pilihan bobot untuk pemenuhan standar efisiensi	<i>ListItem</i>
9	ComboSkalabilitas	<i>DropDownList</i>	Menampilkan pilihan bobot untuk skalabilitas perangkat lunak	<i>ListItem</i>
10	ComboEfisiensi	<i>DropDownList</i>	Menampilkan pilihan bobot untuk efisiensi perangkat lunak	<i>ListItem</i>
11	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan jawaban pertanyaan	<i>Event Clicked</i>

3.2.4.9. Halaman Menampilkan Pertanyaan Bobot Keterawatan

KUESIONER BOBOT PERAWATAN

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

KLOC	<input type="text"/>	Changeability	<input type="text"/>
Skill	<input type="text"/>	Testability	<input type="text"/>
Skill teknis	<input type="text"/>	Trackability	<input type="text"/>
Skill organisasi	<input type="text"/>	Kompleksitas sikdomatik	<input type="text"/>
Skill tim	<input type="text"/>	Level CMM	<input type="text"/>
Analyzability	<input type="text"/>	Abstraksi	<input type="text"/>
Maintainability	<input type="text"/>		

<<<First <<Prev 4/6 Next >> Last>>>

*arahkan kursor ke label untuk hint

Gambar 3. 32 Rancangan Halaman Pertanyaan Bobot Keterawatan

Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai bobot efisiensi pada halaman sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan bobot aspek keterawatan perangkat lunak terhadap

kualitas perangkat lunak melalui 13 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol Submit untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.32.

**Tabel 3. 22 Spesifikasi Atribut Antar Muka Halaman
Pertanyaan Bobot Keterawatan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboLOC	<i>DropDownList</i>	Menampilkan pilihan bobot panjang <i>Lines Of Code</i>	<i>ListItem</i>
2	ComboSkill	<i>DropDownList</i>	Menampilkan pilihan bobot jumlah kemampuan yang dimiliki tim pengembang	<i>ListItem</i>
3	ComboSTeknis	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan teknis tim pengembang	<i>ListItem</i>
4	ComboSOrganisasi	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan organisasi	<i>ListItem</i>
5	ComboSTim	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan keseluruhan tim pengembang	<i>ListItem</i>
6	ComboAnalyzability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>analyzability</i> perangkat lunak	<i>ListItem</i>
7	ComboChangeability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>changeability</i> perangkat lunak	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
8	ComboTestability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>testability</i>	<i>ListItem</i>
9	ComboTrackability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>trackability</i>	<i>ListItem</i>
10	ComboKompleksitas	<i>DropDownList</i>	Menampilkan pilihan bobot jumlah kompleksitas siklomatik perangkat lunak	<i>ListItem</i>
11	ComboCMM	<i>DropDownList</i>	Menampilkan pilihan bobot level CMM	<i>ListItem</i>
12	ComboAbstraksi	<i>DropDownList</i>	Menampilkan pilihan bobot jenis abstraksi	<i>ListItem</i>
13	ComboMaintainability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>maintainability</i>	<i>ListItem</i>
14	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan jawaban pertanyaan	<i>Event Clicked</i>

3.2.4.10. Halaman Menampilkan Pertanyaan Bobot Keterawatan 2

KUESIONER BOBOT PERAWATAN

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

Pengalaman manajer proyek sebagai manajer proyek	<input type="text"/>
Jumlah KLOC per anggota tim	<input type="text"/>
Ketersediaan dokumentasi	<input type="text"/>
Pemenuhan maintainability	<input type="text"/>
Jumlah versi yang telah dirilis	<input type="text"/>
Ketersediaan sistem tracking fungsional dan perilaku sistem	<input type="text"/>
Kemudahan dalam tracking versi lama perangkat lunak	<input type="text"/>
Pengalaman manajer proyek dalam bisnis IT	<input type="text"/>
Pengalaman manajer	<input type="text"/>
SOB anggota tim pengembang	<input type="text"/>
Kerjasama antar tim pengembang	<input type="text"/>
Performa keseluruhan tim pengembang	<input type="text"/>

<<<First <<Prev 5/6 Next >> Last>>>

*arahkan kursor ke label untuk hint

Gambar 3. 33 Rancangan Halaman Menampilkan Pertanyaan Bobot Keterawatan 2

Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai bobot keterawatan pada halaman sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan lanjutan bobot aspek perawatam terhadap kualitas perangkat lunak melalui 13 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol Submit untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.33.

**Tabel 3. 23 Spesifikasi Atribut Antar Muka Halaman
Menampilkan Pertanyaan Bobot Keterawatan 2**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboManajer	<i>DropDownList</i>	Menampilkan pilihan bobot pengalaman pamanjer sebagai manajer proyek	<i>ListItem</i>
2	ComboKLOCTim	<i>DropDownList</i>	Menampilkan pilihan bobot jumlah KLOC per anggota tim	<i>ListItem</i>
4	ComboDokumentasi	<i>DropDownList</i>	Menampilkan pilihan bobot ketersediaan dokumentasi untuk perangkat lunak	<i>ListItem</i>
5	ComboPemenuhan	<i>DropDownList</i>	Menampilkan pilihan bobot pemenuhan perangkat lunak terhadap standar <i>maintainability</i>	<i>ListItem</i>
6	ComboVersi	<i>DropDownList</i>	Menampilkan pilihan bobot jumlah versi yang telah dirilis	<i>ListItem</i>
7	ComboTrackingTersedia	<i>DropDownList</i>	Menampilkan pilihan bobot ketersediaan penelusuran sistem terdahulu	<i>ListItem</i>
8	ComboTrackingMudah	<i>DropDownList</i>	Menampilkan pilihan bobot kemudahan penelusuran sistem terdahulu	<i>ListItem</i>
9	ComboManajerBisnis	<i>DropDownList</i>	Menampilkan pilihan bobot pengalaman manajer dalam bisnis IT	<i>ListItem</i>
10	ComboPengalaman	<i>DropDownList</i>	Menampilkan pilihan bobot pengalaman manajer terhadap kualitas perangkat lunak	<i>ListItem</i>
11	ComboKualitas	<i>DropDownList</i>	Menampilkan pilihan bobot rata-rata kualitas tim pengembang	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
12	ComboKerjasama	<i>DropDownList</i>	Menampilkan pilihan bobot kerjasama antara tim pengembang	<i>ListItem</i>
13	ComboPerforma	<i>DropDownList</i>	Menampilkan pilihan bobot performa keseluruhan tim pengembang	<i>ListItem</i>
14	btnSubmit	<i>Button</i>	Pilihan aksi untuk menyimpan jawaban pertanyaan	<i>Event Clicked</i>

3.2.4.11. Halaman Menampilkan Pertanyaan Bobot Portabilitas

KUESIONER BOBOT PORTABILITAS

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

Kompatibilitas pada sistem operasi lain

Penggunaan tools dasar

Jumlah package yang harus diinstall sebelumnya

Kemampuan adaptasi perangkat lunak

Installability

Pemenuhan portability

Portabilitas

<<<First <<Prev 6/6

*arahkan kursor ke label untuk hint

Gambar 3. 34 Rancangan Halaman Menampilkan Pertanyaan Bobot Portabilitas

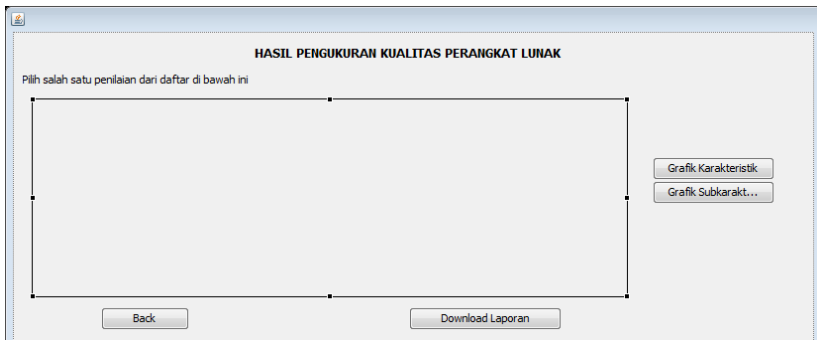
Halaman ini merupakan tampilan yang muncul ketika sistem berhasil menyimpan nilai-nilai bobot keterawatan pada halaman sebelumnya ke dalam basis data. Halaman ini digunakan untuk mendapatkan bobot aspek portabilitas perangkat lunak terhadap kualitas perangkat lunak melalui 7 *DropDownList* yang tersedia dengan kerangan sesuai dengan label yang ada di sisi kirinya. List nilai yang ada pada *DropDownList* ini dihasilkan secara dinamis sesuai dengan data yang ada pada basis data. Terdapat tombol Submit untuk menyimpan nilai-nilai tersebut untuk disimpan ke dalam basis data. Rancangan tampilan dapat dilihat pada Gambar 3.34.

Tabel 3. 24 Spesifikasi Atribut Antar Muka Halaman Menampilkan Pertanyaan Bobot Portabilitas

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	ComboKompatibilitas	<i>DropDownList</i>	Menampilkan pilihan bobot kompatibilitas terhadap OS	<i>ListItem</i>
2	ComboToolsDasar	<i>DropDownList</i>	Menampilkan pilihan bobot penggunaan tools dasar	<i>ListItem</i>
3	ComboPaket	<i>DropDownList</i>	Menampilkan pilihan bobot kebutuhan <i>package</i>	<i>ListItem</i>
4	ComboAdaptasi	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan adaptasi perangkat lunak	<i>ListItem</i>
5	ComboInstallability	<i>DropDownList</i>	Menampilkan pilihan bobot kemampuan <i>installability</i> perangkat lunak	<i>ListItem</i>
6	ComboPemenuhan	<i>DropDownList</i>	Menampilkan pilihan bobot pemenuhan standar portabilitas	<i>ListItem</i>
7	ComboPortabilitas	<i>DropDownList</i>	Menampilkan pilihan bobot portabilitas perangkat lunak	<i>ListItem</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
8	btnSubmit	Button	Pilihan aksi untuk menyimpan jawaban pertanyaan	Event Clicked

3.2.4.12. Halaman Menampilkan Detail Penilaian Perangkat Lunak



Gambar 3. 35 Rancangan Tampilan Detail Penilaian Perangkat Lunak

Halaman ini merupakan tampilan yang muncul ketika pengguna melakukan pilihan untuk melihat detail hasil pengukuran kualitas perangkat lunak. Pada halaman ini terdapat tabel yang akan menampilkan daftar penilaian kualitas yang telah dilakukan untuk suatu perangkat lunak. Rancangan tampilan dapat dilihat pada Gambar 3.35.

**Tabel 3. 25 Spesifikasi Atribut Antar Muka Halaman
Menampilkan Detail Penilaian Perangkat Lunak**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	table	jTable	Menampilkan detail penilaian kualitas untuk suatu perangkat lunak yang dipilih	<i>TableModel</i>
2	btnBack	<i>button</i>	Menutup halaman dan kembali ke halaman daftar perangkat lunak	<i>Event Clicked</i>
3	btnLaporan	<i>button</i>	Pilihan aksi bagi pengguna untuk menampilkan laporan hasil penilaian suatu perangkat lunak	<i>Event Clicked</i>
4	btnGrafikKarakteristik	<i>button</i>	Pilihan aksi bagi pengguna untuk menampilkan skor karakteristik dalam bentuk grafik	<i>Event Clicked</i>
5	btnGrafikSubkarakteristik	<i>button</i>	Pilihan aksi bagi pengguna untuk menampilkan skor subkarakteristik dalam bentuk grafik	<i>Event Clicked</i>

3.2.4.13. Laporan Penilaian Perangkat Lunak

Laporan ini merupakan laporan yang akan menampilkan data penilaian perangkat lunak. Pada laporan ini terdapat dua tabel yang masing-masing digunakan untuk menampilkan data aspek Karakteristik dan aspek Subkarakteristik. Rancangan tampilan laporan dapat dilihat pada Gambar 3.36.

<div style="text-align: center;"> Hasil Pengukuran Kualitas Perangkat Lunak </div>	
Nama Perangkat Lunak	SF
Penilai	SF(NAMA_PENILAI)
Tanggal Penilaian	SF(TANGGAL_PENILAIAN)
Total Skor Penilaian	SF(SKOR)
Hasil Penghitungan Subkarakteristik	
	
Hasil Penghitungan Karakteristik	
	
Grafik Hasil Penilaian Subkarakteristik	
	

Gambar 3. 36 Rancangan Laporan Penilaian Kualitas Perangkat Lunak

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan lingkungan dimana sistem ini dibangun, dimana akan dijelaskan mengenai kebutuhan perangkat yang diperlukan untuk membangun sistem ini. Lingkungan implementasi dibagi menjadi dua, yaitu lingkungan implementasi terhadap perangkat keras dan lingkungan implementasi terhadap perangkat lunak.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak ini adalah komputer/*notebook* dengan spesifikasi sebagai berikut:

- Tipe : *Notebook* Sony Vaio SV Series,
- Processor : Intel(R) Core(TM) i5-3210M CPU @ 2.50 GHz (4CPUs) ~2.50 GHz
- Memori : 4096 MB DDR2

4.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem antara lain sebagai berikut.

1. Sistem Operasi Windows
Sistem operasi yang digunakan adalah Microsoft Windows 7 Home Ultimate.
2. Eclipse IDE

Kakas bantu yang digunakan dalam mengembangkan sistem ini adalah Eclipse IDE versi Luna.

3. Java Development Kit (JDK)

Untuk menjalankan suatu program berbahasa Java, diperlukan suatu lingkungan pemrograman yang disebut dengan JDK. JDK terdiri dari lingkungan eksekusi program yang berada diatas sistem operasi. JDK yang digunakan adalah JDK versi 1.7

4. Java Development Tools (JDT)

Untuk membuat AST dari suatu kode sumber, dibutuhkan pustaka JDT. Pustaka JDT merupakan pustaka untuk menghasilkan AST dari suatu kode sumber perangkat lunak. Versi yang digunakan adalah versi 3.5.2

5. JasperReport

Untuk menghasilkan laporan penilaian perangkat lunak, dibutuhkan pustaka JasperReport. Pustaka JasperReport merupakan salah satu pustaka untuk menampilkan laporan dalam bahasa Java. Versi JasperReport yang digunakan adalah 6.0.4

6. ApacheMath

Untuk mendapatkan *crisp value* dari suatu nilai kualitas perangkat lunak, dibutuhkan pustaka ApacheMath. Pustaka ApacheMath digunakan untuk menghitung nilai hasil dari operasi integral. Versi ApacheMath yang digunakan adalah versi 3.5

7. WindowBuilder

Untuk mendesain rancangan antarmuka dalam IDE Eclipse, dibutuhkan pustaka WindowBuilder. Pustaka ini menyediakan elemen-elemen yang biasa digunakan untuk membuat aplikasi GUI dalam bahasa Java, seperti swing, applet, dan lain sebagainya. Versi yang digunakan adalah versi 1.7.0

8. JFreeChart

JfreeChart merupakan suatu pustaka yang digunakan untuk menampilkan diagram dalam bahasa pemrograman Java. Diagram ini nantinya akan digunakan untuk menampilkan diagram skor

penilaian aspek perangkat lunak yang diambil dari basis data. JfreeChart yang digunakan adalah JfreeChart versi 1.0.19

4.2 Implementasi

Lingkungan implementasi merupakan lingkungan dimana sistem ini dibangun, dimana akan dijelaskan mengenai kebutuhan perangkat yang diperlukan untuk membangun sistem ini. Lingkungan implementasi dibagi menjadi dua, yaitu lingkungan implementasi terhadap perangkat keras dan lingkungan implementasi terhadap perangkat lunak.

4.2.1 Implementasi Lapisan Antarmuka Pengguna

Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem. Pada bagian ini akan dijelaskan implementasi dari rancangan lapisan antarmuka.

4.2.1.1 Kelas Main

Kelas ini merupakan kelas yang pertama kali muncul pada saat kaskas bantu dijalankan. Kelas ini akan menjalankan fungsinya sesuai dengan perintah yang dimasukkan oleh pengguna. Kelas ini akan memanggil kelas UploadPL ketika pengguna memilih menu unggah perangkat lunak dan akan memanggil kelas DisplayPL ketika pengguna memilih menu daftar perangkat lunak.

4.2.1.2 Kelas UploadPL

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan perangkat lunak dan kode sumbernya untuk disimpan di dalam sistem. Antarmuka yang ditampilkan adalah implementasi dari fitur pengunggahan perangkat lunak. Kelas ini berhubungan dengan kelas PerangkatLunak yang berada di lapisan object yang berguna untuk mengelola data yang dimasukkan.

4.2.1.3 Kelas GUIRatingIsian

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.4 Kelas GUIRatingPilgan

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan yang berupa pertanyaan bersifat pilihan ganda. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.5 Kelas GUIRatingSubjektif

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating subjektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.6 Kelas GUIBobotFungsionalitas

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner bobot aspek fungsionalitas terhadap penilaian perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner bobot perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.7 Kelas GUIBobotEfisiensi

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner bobot aspek efisiensi terhadap penilaian perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner bobot perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.8 Kelas GUIBobotKeterawatan

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating bobot aspek keterawatan penilaian perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner bobot perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.9 Kelas GUIBobotKeterawatan2

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.10 Kelas GUIBobotPortabilitas

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.11 Kelas DisplayPL

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.1.12 Kelas DisplayResult

Kelas ini merupakan kelas yang digunakan oleh pengguna untuk memasukkan kuesioner rating objektif terkait dengan perangkat lunak yang dimasukkan. Antarmuka yang ditampilkan adalah implementasi dari fitur pengisian kuesioner rating perangkat lunak. Kelas ini berhubungan dengan kelas Penilaian yang berada di lapisan object yang berguna untuk mengelola data masukan dari pengguna melalui halaman ini.

4.2.2 Implementasi Lapisan Kontrol

Lapisan kontrol merupakan lapisan yang bertugas untuk mengontrol jalannya aplikasi dan data. Pada bagian ini akan dijelaskan implementasi dari rancangan kelas pada lapisan model.

4.2.2.1 Kelas PerangkatLunakController

Kelas ini merupakan kelas kontrol yang bertugas untuk mengatur kelas antarmuka UploadPL. Kelas ini akan menjalankan perintah-perintah *logic* yang digunakan dalam kelas UploadPL, seperti memanggil halaman lain, menampilkan jendela untuk memilih *file* kode sumber, dan sebagainya.

4.2.2.2 Kelas PenilaianRatingController

Kelas ini merupakan kelas kontrol yang bertugas untuk mengatur fungsionalitas perangkat lunak, yaitu mengisi kuesioner rating. Kelas ini berisikan kode-kode untuk mengontrol dua halaman antarmuka, yaitu kelas GUIRatingNumerik dan kelas

GUIRatingObjektif yang sama-sama digunakan untuk mengisi kuesioner objektif. Kelas ini akan menjalankan perintah-perintah *logic* yang digunakan dalam kedua kelas tersebut, seperti memanggil halaman lain, menyimpan jawaban sementara dari pengguna, menampilkan data untuk elemen *JComboBox*, dan sebagainya.

Dalam kelas ini juga terdapat satu *method* yang digunakan untuk melakukan validasi terhadap data yang dimasukkan oleh pengguna dan akan menampilkan pesan melalui kaskas bantu bahwa data yang dimasukkan tidak valid. *Method* yang melakukan fungsionalitas ini adalah *method* *IsValid*. Implementasi *method* ini dapat dilihat pada Kode Sumber 4.1

```
1 public Notifikasi isValid(int tdSesuai, int operasi, int
2 jumlahAkurasi, int kontrolSedia, int kontrolButuh, int
3 jumlahTim, int kustom, int properti)
4 {
5     String message="";
6     boolean valid=true;
7
8     if(operasi==0)
9     {
10         valid=false;
11         message="Jumlah operasi yang dibutuhkan
12 tidak boleh 0";
13     }
14     else if(tdSesuai>operasi||jumlahAkurasi>operasi)
15     {
16         valid=false;
17         message="Jumlah operasi yang dibutuhkan
18 tidak boleh kurang dari jumlah operasi tidak sesuai dan
19 jumlah akurasi";
20     }
21     else if(kontrolSedia>kontrolButuh)
22     {
23         valid=false;
24         message="Jumlah pengontrolan akses yang
25 dibutuhkan tidak boleh kurang dari jumlah pengontrolan akses
26 yang disediakan";
27     }
28     else if(jumlahTim==0)
29     {
30         valid=false;
31         message="Jumlah anggota tim tidak boleh 0";
32     }
33 }
```

```

33         else if(kustom>properti)
34         {
35             valid=false;
36             message="Jumlah properti perangkat lunak
37 tidak boleh kurang dari jumlah customizable properti";
38         }
39
40         Notifikasi n=new Notifikasi(valid, message);
41         return n;
42     }
43

```

Kode Sumber 4. 1 Pengecekan Validitas Isian

4.2.2.3 Kelas DisplayPLController

Kelas ini merupakan kelas kontrol yang bertugas untuk mengatur kelas antarmuka DisplayPL. Kelas ini akan menjalankan perintah-perintah *logic* yang digunakan dalam kelas DisplayPL, seperti memanggil halaman lain, mengisi tabel dengan data-data yang didapat dari basis data, dan sebagainya.

4.2.2.4 Kelas PenilaianSubjektifController

Kelas ini merupakan kelas kontrol yang bertugas untuk mengatur fungsionalitas perangkat lunak, yaitu mengisi kuesioner subjektif. Kelas ini berisikan kode-kode untuk mengontrol beberapa kelas antarmuka, yaitu kelas GUIRatingSubjektif, kelas GUIBototFungsionalitas, GUIBototEfisiensi, GUIBototKeterawatan, GUIBototKeterawatan2, dan GUIBototPortabilitas yang sama-sama digunakan untuk mengisi kuesioner subjektif. Kelas ini akan menjalankan perintah-perintah *logic* yang digunakan dalam kelas-kelas tersebut, seperti memanggil halaman lain, menyimpan jawaban sementara dari pengguna, menampilkan data untuk elemen *JComboBox*, dan sebagainya.

4.2.2.5 Kelas DisplayResultController

Kelas ini merupakan kelas kontrol yang bertugas untuk mengatur kelas antarmuka DisplayPL. Kelas ini akan menjalankan perintah-perintah *logic* yang digunakan dalam kelas DisplayPL,

seperti memanggil halaman lain, mengisi tabel dengan data-data yang didapat dari basis data, dan sebagainya.

Selain itu, di kelas ini terdapat satu *method* yang menangani pemanggilan laporan penilaian perangkat lunak dan memunculkannya kepada pengguna. *Method* yang melakukan hal ini adalah *method* `buttonLaporanActionPerformed`. Implementasi *method* tersebut dapat dilihat pada Kode Sumber 4.2.

```
1 public void actionPerformed(ActionEvent arg0) {
2     Koneksi koneksi=new Koneksi();
3     int selectedRowIndex = table.getSelectedRow();
4     int selectedId = (Integer)
5     table.getModel().getValueAt(selectedRowIndex, 0);
6     Map parametersMap = new HashMap();
7
8     parametersMap.put("GidPenilaian",selectedId);
9     try
10    {
11        JasperReport jasperReport =
12        JasperCompileManager.compileReport("ReportKualitas.jrxml");
13        JasperPrint jasperPrint =
14        JasperFillManager.fillReport(jasperReport,parametersMap,
15        koneksi.getConnection());
16        JasperViewer.viewReport(jasperPrint);
17        koneksi.Disconnect();
18    }catch(NullPointerException e)
19    {
20        JOptionPane.showMessageDialog(null, "Harap pilih
21        Perangkat Lunak!");
22    }
23    catch(Exception e)
24    {
25        System.out.println("Terjadi kesalahan");
26    }
27 }
28 }
```

Kode Sumber 4. 2 Implementasi Pemanggilan Laporan

4.2.3 Implementasi Lapisan Model

Lapisan model merupakan lapisan yang bertugas untuk merepresentasikan dan mengolah data. Pada bagian ini akan dijelaskan implementasi dari rancangan kelas pada lapisan model.

4.2.3.1 Kelas ComboItem

Kelas ini merupakan kelas yang digunakan dalam elemen JComboBox sebagai item elemen pilihan, karena JComboBox tidak menyediakan objek item sebagai pilihan nilai. Dalam kelas ini, terdapat konstruktor, variabel key dan value. dan *method* getValue dan getKey. *Method* getValue adalah *method* yang digunakan untuk mendapatkan nilai variabel value untuk objek ComboItem. Selain itu, dalam kelas ini terdapat *method* toString yang merupakan overriding *method* yang digunakan untuk menampilkan variabel value.

4.2.3.2 Kelas Jawaban

Kelas ini merupakan kelas yang menyimpan data jawaban yang dimasukkan oleh pengguna melalui antarmuka kakas bantu untuk diproses lebih lanjut. Di dalam kelas ini, terdapat satu konstruktor dengan parameter idJawaban berupa integer dan nilai jawaban yang berupa string. Kelas ini dipanggil oleh kelas-kelas di lapisan antarmuka seperti kelas GUIRatingPilgan, GUIRatingSubjektif, dan sebagainya.

4.2.3.3 Kelas Karakteristik

Kelas ini merupakan representasi dari elemen model kualitas level pertama, yaitu karakteristik. Sistem akan membuat objek dari kelas ini ketika objek dari kelas ModelKualitas dibuat dan akan diisi dengan data metrik dan jawaban rating dan bobot untuk metrik dan idpenilaian tertentu. Kelas ini memiliki satu buat objek List yang berisi objek-objek Subkarakteristik sesuai dengan model kualitas yang digunakan.

Kelas Karakteristik tersusun atas satu buah konstruktor dengan parameter idPenilaian dan *method* GetScore yang digunakan untuk menghitung rating dari karakteristik dengan cara menghitung *sum of product* dari nilai bobot dan rating dari kelas Subkarakteristik yang disimpan pada List. Nilai bobot dari kelas Karakteristik didapatkan dari tabel jawaban bobot yang ada pada basis data.

Implementasi *method* GetScore lebih lengkapnya tampak seperti pada Kode Sumber 4.3 di bawah ini.

```
1 public FuzzySets GetScoreRating()
2 {
3     double skorBawah=0, skorTengah=0, skorAtas=0;
4     for(int i=0; i<this.listChild.size(); i++)
5     {
6         Subkarakteristik sk=this.listChild.get(i);
7         FuzzySets skorBobot=sk.skorBobot;
8         FuzzySets skorRating=sk.skorRating;
9         skorBawah=Math.max(skorBawah,
10 skorBobot.scoreBawah*skorRating.scoreBawah);
11
12         skorTengah=Math.max(skorTengah,skorBobot.score
13 Tengah*skorRating.scoreTengah);
14
15         skorAtas=Math.max(skorAtas,skorBobot.scoreAta
16 s*skorRating.scoreAtas);
17     }
18
19     FuzzySets result=new FuzzySets(skorBawah, skorTengah,
20 skorAtas);
21     return result;
22 }
```

Kode Sumber 4. 3 Implementasi *Method* GetScore pada Kelas Karakteristik

Sementara itu, untuk mendapatkan nilai bobot yang disimpan pada basis data, sistem akan melakukan kueri untuk memanggil nilai bobot dan mengubahnya ke dalam suatu objek FuzzySets. Nilai bobot karakteristik akan digunakan dalam proses penghitungan nilai rating keseluruhan. Kode Sumber 4.4 menerangkan implementasi pemanggilan data bobot dari basis data. Kode sumber ini akan dijalankan di dalam konstruktor, pada saat objek tersebut dibuat.

```
1 try {
2     ResultSet rs1=koneksi.executeSelect("SELECT
3 FB.SCORE_BOBOTBAWAH, FB.SCORE_BOBOTTENGAH, FB.SCORE_BOBOTATAS
4 "+ "FROM FUZIFIKASI_BOBOT FB, METRIK M, JAWABAN_BOBOT JB,
5 PENILAIAN P " + "WHERE M.ID_METRIK=JB.ID_METRIK " + "AND
6 FB.ID_FUZIFIKASIBOBOT=JB.ID_FUZIFIKASIBOBOT "+ "AND
```

```

7 P.ID_PENILAIAN=JB.ID_PENILAIAN "+ "AND
8 P.ID_PENILAIAN="+idPenilaian + "AND
9 M.ID_METRIK="+this.idMetrik);
10     while(rs1.next())
11     {
12         double
13 score_bawah=rs1.getDouble("SCORE_BOBOTBAWAH");
14         double
15 score_tengah=rs1.getDouble("SCORE_BOBOTTENGAH");
16         double
17 score_atas=rs1.getDouble("SCORE_BOBOTATAS");
18         skorBotot=new FuzzySets(score_bawah,
19 score_tengah, score_atas);
20     }
21
22     koneksi.Disconnect();
23 } catch (SQLException e) {
24     // TODO Auto-generated catch block
25     e.printStackTrace();
26 }
27

```

Kode Sumber 4. 4 Implementasi Pemanggilan Nilai Bobot pada Kelas Karakteristik

Untuk memanggil semua aspek-aspek subkarakteristik di bawahnya, kelas Karakteristik memiliki satu *method* yang berfungsi untuk memanggil data-data aspek subkarakteristik dan membuat objek Subkarakteristik sesuai dengan data pada basis data. *Method* tersebut adalah *method* GetChild. Nantinya, objek-objek Subkarakteristik ini akan disimpan dalam suatu objek list. Implementasi *method* GetScore pada kelas Karakteristik dapat dilihat pada Kode Sumber 4.5.

```

1 public void GetChild(int idPenilaian)
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeQuery("SELECT
6 M2.ID_METRIK" + " FROM METRIK M1, METRIK M2, RELASI_METRIK"
7 + " WHERE M1.ID_METRIK=RELASI_METRIK.ID_METRIKPARENT AND
8 M2.ID_METRIK=RELASI_METRIK.ID_METRIKCHILD" + " AND
9 M1.ID_METRIK="+this.idMetrik);
10
11         while(rs.next())
12         {

```

```

13         Subkarakteristik child=new
14 Subkarakteristik(rs.getInt("ID_METRIK"), idPenilaian);
15             this.listChild.add(child);
16         }
17         koneksi.Disconnect();
18     } catch (SQLException e) {
19         // TODO Auto-generated catch block
20         e.printStackTrace();
21     }
22 }
23

```

Kode Sumber 4. 5 Implementasi *Method* GetChild pada Kelas Karakteristik

4.2.3.4 Kelas Metrik

Kelas ini merupakan representasi dari elemen model kualitas level ketiga, yaitu metrik. Sistem akan membuat objek dari kelas ini ketika objek dari kelas ModelKualitas dibuat dan akan diisi dengan data metrik dan jawaban rating dan bobot untuk metrik dan idpenilaian tertentu. Sesuai dengan model kualitas yang digunakan, kelas ini memiliki List objek Submetrik yang akan terisi apabila metrik tersebut memang memiliki submetrik pada model kualitas.

Kelas Metrik terdiri dari satu konstruktor, *method* HasBobot, *method* HasChild, *method* GetSource, dan *method* GetScore. *Method* HasBobot merupakan *method* yang digunakan untuk mengecek apakah metrik tersebut memiliki nilai bobot ataukah tidak, karena tidak semua metrik memiliki nilai bobot. Jika *method* ini mengembalikan nilai *false*, maka bobot akan diset nilai *default*, yaitu (1,1,1). *Method* ini mendapatkan informasi tersebut dengan cara memanggil *method* HasChild. *Method* HasChild merupakan *method* yang akan mengecek apakah metrik tersebut memiliki objek Submetrik di bawahnya dengan cara mengecek jumlah objek kelas Submetrik yang terdapat pada List Submetrik yang disimpan oleh kelas ini. Implementasi kedua *method* ini dapat dilihat pada Kode Sumber 4.6 dan Kode Sumber 4.7.

```

1   public void HasBobot()
2   {
3       Koneksi koneksi=new Koneksi();

```

```

4         try {
5             ResultSet rs=koneksi.executeSelect("SELECT
6 PERTANYAAN_BOBOT "
7             + "FROM METRIK "
8             + "WHERE ID_METRIK="+this.idMetrik);
9
10            while(rs.next())
11            {
12
13            if(rs.getString("PERTANYAAN_BOBOT")== "")
14            {
15
16            this.hasScoreBobot=false;
17
18            System.out.println("nggak punya bobot");
19            }
20
21            else
22            {
23
24            this.hasScoreBobot=true;
25
26            System.out.println("punya bobot");
27            }
28            }
29            koneksi.Disconnect();
30            } catch (SQLException e) {
31                // TODO Auto-generated catch block
32                e.printStackTrace();
33            }
34        }

```

Kode Sumer 4. 6 Implementasi *Method* HasBobot pada Kelas Metrik

```

1 public void GetChild(int idPenilaian)
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 M2.ID_METRIK" + " FROM METRIK M1, METRIK M2, RELASI_METRIK"
7 + " WHERE M1.ID_METRIK=RELASI_METRIK.ID_METRIKPARENT AND
8 M2.ID_METRIK=RELASI_METRIK.ID_METRIKCHILD" + " AND
9 M1.ID_METRIK="+this.idMetrik);
10
11        while(rs.next())
12        {

```

```

13         Submetrik child=new
14 Submetrik(rs.getInt("ID_METRIK"), idPenilaian);
15             this.listChild.add(child);
16         }
17         koneksi.Disconnect();
18     } catch (SQLException e) {
19         // TODO Auto-generated catch block
20         e.printStackTrace();
21     }
22
23     if(this.listChild.size()==0)
24         this.hasChild=false;
25     else
26         this.hasChild=true;
27 }
28

```

Kode Sumber 4. 7 Implementasi *Method* GetChild pada Kelas Metrik

Method GetSource merupakan *method* yang akan mengecek dari tabel manakah nilai rating suatu perangkat dapat dipanggil kembali. *Method* ini akan mengecek nilai yang ada pada field *source* pada tabel perangkat lunak di basis data. Sementara itu, *method* GetScore merupakan *method* yang digunakan untuk menghitung nilai skor untuk metrik tersebut. Cara yang dilakukan tergantung dengan nilai HasChild. Jika hasChild bernilai benar, maka *method* akan menghitung *sum of product* dari nilai bobot dan rating dari kelas Submetrik yang disimpan pada List. Sebaliknya, *method* akan mendapatkan nilai rating sesuai dengan yang tersimpan di basis data. Kode sumber untuk *method* GetSource dan GetScore dapat dilihat pada Kode Sumber 4.8 dan 4.9 berikut ini.

```

1 public void GetSource()
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 SOURCE "
7         + "FROM METRIK "
8         + "WHERE ID_METRIK="+this.idMetrik);
9
10        while(rs.next())
11

```

```

12         {
13             if(rs.getInt("SOURCE")==0)
14             {
15
16                 this.fromJawaban=true;
17             }
18
19             else
20             {
21                 this.fromJawaban=false;
22             }
23         }
24         koneksi.Disconnect();
25     } catch (SQLException e) {
26         // TODO Auto-generated catch block
27         e.printStackTrace();
28     }
29 }
30
31

```

Kode Sumber 4. 8 Implementasi *Method* GetSource pada Kelas Metrik

```

1 public FuzzySets GetScoreRating()
2 {
3     double skorBawah=0, skorTengah=0, skorAtas=0;
4     for(int i=0; i<this.listChild.size(); i++)
5     {
6         Subkarakteristik sk=this.listChild.get(i);
7         FuzzySets skorBobot=sk.skorBobot;
8         FuzzySets skorRating=sk.skorRating;
9         skorBawah=Math.max(skorBawah,
10 skorBobot.scoreBawah*skorRating.scoreBawah);
11
12         skorTengah=Math.max(skorTengah,skorBobot.score
13 Tengah*skorRating.scoreTengah);
14
15         skorAtas=Math.max(skorAtas,skorBobot.scoreAta
16 s*skorRating.scoreAtas);
17     }
18
19     FuzzySets result=new FuzzySets(skorBawah, skorTengah,
20 skorAtas);
21     return result;
22 }

```


Kode Sumber 4. 9 Implementasi *Method* GetScore pada Kelas Karakteristik

Sementara itu, untuk mendapatkan nilai bobot yang disimpan pada basis data, sistem akan melakukan kueri untuk memanggil nilai bobot dan mengubahnya ke dalam suatu objek FuzzySets. Nilai bobot karakteristik akan digunakan dalam proses penghitungan nilai rating Subkarakteristik. Kode Sumber 4.10 menerangkan implementasi pemanggilan data bobot dari basis data. Kode sumber ini akan dijalankan di dalam konstruktor, pada saat objek tersebut dibuat.

```
1  try {
2      ResultSet rs1=koneksi.executeSelect("SELECT
3  FB.SCORE_BOBOTBAWAH, FB.SCORE_BOBOTTENGAH, FB.SCORE_BOBOTATAS
4  "+ "FROM FUZIFIKASI_BOBOT FB, METRIK M, JAWABAN_BOBOT JB,
5  PENILAIAN P " + "WHERE M.ID_METRIK=JB.ID_METRIK "+ "AND
6  FB.ID_FUZIFIKASIBOBOT=JB.ID_FUZIFIKASIBOBOT "+ "AND
7  P.ID_PENILAIAN=JB.ID_PENILAIAN "+ "AND
8  P.ID_PENILAIAN="+idPenilaian + "AND
9  M.ID_METRIK="+this.idMetrik);
10     while(rs1.next())
11     {
12         double
13 score_bawah=rs1.getDouble("SCORE_BOBOTBAWAH");
14         double
15 score_tengah=rs1.getDouble("SCORE_BOBOTTENGAH");
16         double
17 score_atas=rs1.getDouble("SCORE_BOBOTATAS");
18         skorBobot=new FuzzySets(score_bawah,
19 score_tengah, score_atas);
20     }
21
22     koneksi.Disconnect();
23 } catch (SQLException e) {
24     // TODO Auto-generated catch block
25     e.printStackTrace();
26 }
27 }
```

Kode Sumber 4. 10 Implementasi Pemanggilan Nilai Bobot pada Kelas Metrik

4.2.3.5 Kelas ModelKualitas

Kelas ini merupakan kelas yang merepresentasikan model kualitas ISO/IEC 9126 yang digunakan untuk menilai kualitas suatu perangkat lunak. Kelas ModelKualitas akan dipanggil pada saat proses penilaian akan dimulai. Kelas ini berisikan satu konstruktor yang akan membuat model kualitas dengan membuat objek-objek dari kelas Karakteristik.

Selain itu, di kelas ini terdapat *method* GenerateQM yang akan melakukan proses pembuatan model kualitas yang akan digunakan dalam penilaian kualitas ini. Karena pembuatan model dilakukan secara iteratif, pada Kelas ModelKualitas hanya akan dihasilkan beberapa objek kelas Karakteristik yang merupakan aspek tertinggi dari model kualitas ini. Selain itu, dan terdapat pula *method* GetScoreRating yang akan menghitung nilai kualitas dari perangkat lunak secara keseluruhan. Kode sumber yang digunakan pada *method* GenerateQM dan GetScoreRating berturut-turut dapat dilihat di Kode Sumber 4.11 dan 4.12.

```
1 public FuzzySets GetScoreRating()
2 {
3     double skorBawah=0, skorTengah=0, skorAtas=0;
4     for(int i=0; i<this.listChild.size(); i++)
5     {
6         Karakteristik k=this.listChild.get(i);
7         FuzzySets skorBobot=k.skorBobot;
8         FuzzySets skorRating=k.skorRating;
9         skorBawah=Math.max(skorBawah,
10 skorBobot.scoreBawah*skorRating.scoreBawah);
11
12         skorTengah=Math.max(skorTengah,skorBobot.score
13 Tengah*skorRating.scoreTengah);
14
15         skorAtas=Math.max(skorAtas,skorBobot.scoreAta
16 s*skorRating.scoreAtas);
17     }
18
19     FuzzySets result=new FuzzySets(skorBawah, skorTengah,
20 skorAtas);
21     return result;
22 }
```

Kode Sumber 4. 11 Implementasi *Method* GetScore pada Kelas Karakteristik

```

1 public void GenerateQM(int idPenilaian)
2 {
3     Koneksi koneksi = new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 ID_METRIK" + " FROM METRIK WHERE ID_TIPEMETRIK=1");
7         while(rs.next())
8         {
9             Karakteristik kr= new
10 Karakteristik(rs.getInt("ID_METRIK"), idPenilaian);
11             this.karakteristik.add(kr);
12         }
13         koneksi.Disconnect();
14     } catch (SQLException e) {
15         // TODO Auto-generated catch block
16         e.printStackTrace();
17     }
18 }
19

```

Kode Sumber 4. 12 Implementasi *Method* GenerateQM

Method CountScore akan menghitung skor kualitas dengan cara mendapatkan *sum of product* nilai bobot dan rating yang dimiliki oleh masing-masing objek kelas Karakteristik yang telah dibuat sebelumnya.

4.2.3.6 Kelas Penilaian

Kelas penilaian merupakan kelas yang merepresentasikan objek penilaian kualitas perangkat lunak. Kelas ini digunakan untuk menyimpan jawaban kuesioner rating metrik yang bersifat subjektif dan ditampilkan di halaman GUIRatingSubjektif dan jawaban bobot aspek kualitas perangkat lunak. Kelas ini berisikan atribut idPenilaian, idPerangkatLunak, namaPenilai, dan dua List objek Jawaban yang masing-masing digunakan untuk menyimpan jawaban rating dan jawaban bobot aspek kualitas perangkat lunak.

Selain atribut-atribut tersebut, di dalam kelas penilaian juga terdapat *method* untuk menyimpan data jawaban tersebut ke basis data. *Method* yang menjalankan fungsi ini adalah *method* SaveRatingToDB dan *method* saveBobotToDB. *Method*

SaveRatingToDB digunakan untuk menyimpan jawaban rating subjektif ke dalam basis data, sementara *method* SaveBobotToDB digunakan untuk menyimpan jawaban bobot ke dalam basis data. Informasi terkait penilaian itu sendiri disimpan ke dalam basis data dengan menjalankan *method* InsertToDB. Implementasi dari ketiga *method* ini dapat dilihat pada Kode Sumber 4.13, Kode Sumber 4.14 dan Kode Sumber 4.15 berikut ini.

```

1  public void insertRatingToDB()
2  {
3      int idJawaban=0;
4      for(int i=0; i<this.jawabanRating.size(); i++)
5      {
6          Jawaban jwb=this.jawabanRating.get(i);
7          try {
8              Koneksi koneksi=new Koneksi();
9              ResultSet
10 rs=koneksi.executeSelect("SELECT COUNT(*)+1 AS ID"
11 + " FROM JAWABAN_METRIK");
12              while(rs.next())
13              {
14                  idJawaban=rs.getInt("ID");
15              }
16
17                  int
18 result=koneksi.executeUpdate("INSERT INTO JAWABAN_METRIK
19 VALUES(" + idJawaban+", "+ this.getIdPenilaian()+", "
20 + jwb.jawaban+"");
21              koneksi.Disconnect();
22          } catch (SQLException e) {
23              // TODO Auto-generated catch block
24              e.printStackTrace();
25          }
26      }
27  }
28

```

Kode Sumber 4. 13 Implementasi *Method* InsertRatingToDB

```

1  public void insertBobotToDB()
2  {
3      int idJawaban=0;
4      for(int i=0; i<this.jawabanBobot.size(); i++)
5      {
6          Jawaban jwb=this.jawabanBobot.get(i);
7          try {

```

```

8         Koneksi koneksi=new Koneksi();
9         ResultSet
10 rs=koneksi.executeSelect("SELECT COUNT(*)+1 AS ID"
11 + " FROM JAWABAN_BOBOT");
12         while(rs.next())
13         {
14             idJawaban=rs.getInt("ID");
15         }
16
17         int
18 result=koneksi.executeUpdate("INSERT INTO JAWABAN_BOBOT
19 VALUES("+ idJawaban+", "
20 + this.getIdPenilaian()+","+ jwb.idMetrik+", "
21 + jwb.jawaban+"");
22         koneksi.Disconnect();
23     } catch (SQLException e) {
24         // TODO Auto-generated catch block
25         e.printStackTrace();
26     }
27 }
28
29 }
30

```

Kode Sumber 4. 14 Implementasi *Method* InsertBototToDB

```

1 public void insertToDB()
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 COUNT(*)+1 AS ID" + " FROM PENILAIAN");
7         while(rs.next())
8         {
9             this.setIdPenilaian(rs.getInt("ID"));
10        }
11
12        int result=koneksi.executeUpdate("INSERT
13 INTO PENILAIAN (ID_PENILAIAN, NAMA_PENILAI,
14 TANGGAL_PENILAIAN, ID_PERANGKATLUNAK) VALUES("
15 + this.getIdPenilaian()+"," + this.namaPenilai
16 + ",CURRENT_TIMESTAMP," + this.idPerangkatLunak+"");
17        koneksi.Disconnect();
18    } catch (SQLException e) {
19        // TODO Auto-generated catch block
20        e.printStackTrace();
21    }
22    insertBototToDB();
23

```

24	insertRatingToDB();
25	}

Kode Sumber 4. 15 Implementasi *Method InsertToDB*

Kelas Penilaian juga mengandung *method* yang digunakan untuk menghitung skor akhir dari suatu penilaian yang telah dilakukan pengguna. *Method* tersebut adalah *method* CountScore. *Method* ini akan memanggil kelas ModelKualitas untuk membuat model kualitas dan akan menghitung *crisp value* dengan menggunakan kelas CrispValue yang berada pada lapisan Util. Setelah proses penghitungan selesai, kelas ini akan menjalankan *method* saveToDB yang berfungsi untuk melakukan perubahan pada data penilaian dan perangkat lunak pada basis data. Perubahan data pada tabel penilaian hanya akan mengisikan atribut skor untuk idPenilaian tertentu dengan nilai yang didapatkan dari *method* GetCrispValue. Sementara itu perubahan data pada tabel perangkat lunak dilakukan dengan cara memperbarui nilai atribut skor pada tabel perangkat lunak dengan idPerangkatLunak tertentu dengan nilai rata-rata penilaian yang telah dilakukan untuk perangkat lunak tersebut. Kode sumber untuk *method* ini dapat dilihat pada Kode Sumber 4.16 berikut ini.

```

1 public void CountScore()
2 {
3     ModelKualitas mk=new
4 ModelKualitas(this.idPenilaian);
5     FuzzySets nilai=mk.GetScore();
6     CrispValue cv=new CrispValue(nilai);
7     System.out.println("Nilai akhir:
8 "+nilai.scoreBawah+" "+nilai.scoreTengah+"
9 "+nilai.scoreAtas);
10    double crispValue=cv.getCrispValue();
11    System.out.println("Skor kualitas perangkat lunak:
12 "+crispValue);
13
14    int jumlahPenilaian=0;
15    try {
16        Koneksi koneksi=new Koneksi();
17        int result=koneksi.executeUpdate("UPDATE
18 PENILAIAN SET" + " SKOR=" +crispValue + " WHERE
19 ID_PENILAIAN="+this.idPenilaian);
20

```

```

21         ResultSet rs=koneksi.executeSelect("SELECT
22 COUNT(*) AS JUMLAH" + " FROM PENILAIAN" + " WHERE
23 ID_PERANGKATLUNAK="+this.idPerangkatLunak);
24         while(rs.next())
25         {
26             jumlahPenilaian=rs.getInt("JUMLAH");
27         }
28
29         if(jumlahPenilaian==1)
30         {
31             int p=koneksi.executeUpdate("UPDATE
32 PERANGKAT_LUNAK SET" + " TOTAL_SKOR=" +crispValue
33 + " WHERE ID_PERANGKATLUNAK="+this.idPerangkatLunak);
34         }
35         else
36         {
37             double average=0;
38             ResultSet
39 rst=koneksi.executeSelect("SELECT AVG(SKOR) AS RATA"
40 + " FROM PENILAIAN" + " WHERE
41 ID_PERANGKATLUNAK="+this.idPerangkatLunak);
42             while(rst.next())
43             {
44                 average=rst.getDouble("RATA");
45             }
46
47             int p=koneksi.executeUpdate("UPDATE
48 PERANGKAT_LUNAK SET" + " TOTAL_SKOR=" +average
49 + " WHERE ID_PERANGKATLUNAK="+this.idPerangkatLunak);
50         }
51         koneksi.Disconnect();
52     } catch (SQLException e) {
53         // TODO Auto-generated catch block
54         e.printStackTrace();
55     }
56 }
57
58 }

```

Kode Sumber 4. 16 Implementasi *Method* CountScore

4.2.3.7 Kelas PenilaianRating

Kelas ini merupakan kelas yang digunakan untuk menyimpan data-data terkait pengisian keterangan/kuesioner rating objektif suatu perangkat lunak. Berbeda dengan kelas Penilaian, untuk satu perangkat lunak hanya terdapat satu penilaianrating.Kelas PenilaianRating memiliki hubungan agregasi dengan kelas Jawaban,

yang digunakan untuk menyimpan data-data jawaban PenilaianRating yang dimasukkan oleh pengguna.

Kelas ini tersusun atas konstruktor, suatu List objek kelas Jawaban, dan atribut idPerangkatLunak dan idPenilaianRating. Sesuai dengan fungsinya, di dalam kelas ini juga terdapat *method* SaveToDB yang digunakan untuk menyimpan data penilaian rating ke dalam basis data. Implementasi *method* SaveToDB dapat dilihat pada Kode Sumber 4.17 di bawah ini

```
1 public void insertToDB()
2 {
3
4     for(int i=0; i<this.jawabanRating.size(); i++)
5     {
6         Jawaban jwb=this.jawabanRating.get(i);
7         try {
8             Koneksi koneksi=new Koneksi();
9             ResultSet
10 rs=koneksi.executeSelect("SELECT COUNT(*)+1 AS ID"
11 + " FROM HASIL_PENGHITUNGAN");
12             while(rs.next())
13             {
14                 this.idPenilaian=rs.getInt("ID");
15             }
16
17             int
18 result=koneksi.executeUpdate("INSERT INTO HASIL_PENGHITUNGAN
19 VALUES("+ this.idPenilaian+", "+ this.idPerangkatLunak+", "
20 + jwb.jawaban+"");
21             koneksi.Disconnect();
22         } catch (SQLException e) {
23             // TODO Auto-generated catch block
24             e.printStackTrace();
25         }
26     }
27 }
```

Kode Sumber 4. 17 Implementasi *Method* InsertToDB

4.2.3.8 Kelas PerangkatLunak

Kelas ini adalah kelas yang berperan untuk menyimpan data suatu perangkat lunak. Kelas ini merepresentasikan tabel perangkat lunak yang ada pada basis data, Di dalam kelas PerangkatLunak

terdapat satu konstruktor yang digunakan untuk membuat objek PerangkatLunak. Selain itu, operasi penyimpanan data perangkat lunak juga dilakukan oleh kelas PerangkatLunak, yaitu oleh *method* SaveToDB. Kode sumber yang digunakan untuk pengimplementasian *method* SaveToDB dapat dilihat di Kode Sumber 4.18 berikut ini.

```
1 public void insertToDB()
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         int result=koneksi.executeUpdate("INSERT
6 INTO PERANGKAT_LUNAK VALUES(" + this.idPerangkatLunak+",
7 + ""+this.namaPL+", " + ""+this.deskripsiPL+",
8 + ""+this.pathPL+", "+ this.crispValue+"");
9         koneksi.Disconnect();
10    } catch (SQLException e) {
11        // TODO Auto-generated catch block
12        e.printStackTrace();
13    }
14 }
```

Kode Sumber 4. 18 Implementasi *Method* InsertToDB pada Kelas PerangkatLunak

4.2.3.9 Kelas Subkarakteristik

Kelas ini merupakan representasi dari elemen model kualitas level kedua, yaitu subkarakteristik. Sistem akan membuat objek dari kelas ini ketika objek dari kelas ModelKualitas dibuat dan akan diisi dengan data subkarakteristik dan jawaban rating dan bobot untuk metrik dan idpenilaian tertentu. Kelas ini memiliki satu buah objek List yang berisi objek-objek kelas metrik sesuai dengan model kualitas yang digunakan.

Kelas Subkarakteristik terdiri dari satu buah konstruktor dengan parameter idPenilaian dan *method* GetScore yang digunakan untuk menghitung rating dari kelas subkarakteristik dengan cara menghitung *sum of product* dari nilai bobot dan rating dari kelas metrik yang disimpan pada suatu List di kelas ini. Nilai bobot dari kelas Subkarakteristik didapatkan dari tabel jawaban bobot yang ada pada basis data. Implementasi *method* GetScore lebih lengkapnya tampak seperti pada Kode Sumber 4.19 di bawah ini.

```

1 public FuzzySets GetScoreRating()
2 {
3     double skorBawah=0, skorTengah=0, skorAtas=0;
4     for(int i=0; i<this.listChild.size(); i++)
5     {
6         Metrik m=this.listChild.get(i);
7         System.out.println(i+" "+m.idMetrik);
8         FuzzySets skorBobot=m.skorBobot;
9         FuzzySets skorRating=m.skorRating;
10        skorBawah=Math.max(skorBawah,
11 skorBobot.scoreBawah*skorRating.scoreBawah);
12        skorTengah=Math.max(skorTengah,skorBobot.scoreTengah*skorRating.scoreTengah);
13        skorAtas=Math.max(skorAtas,skorBobot.scoreAtas*skorRating.scoreAtas);
14    }
15    FuzzySets result=new FuzzySets(skorBawah, skorTengah,
16 skorAtas);
17    return result;
18 }

```

Kode Sumber 4. 19 Implementasi *Method* GetScoreRating pada Kelas Subkarakteristik

Sementara itu, untuk mendapatkan nilai bobot yang disimpan pada basis data, sistem akan melakukan kueri untuk memanggil nilai bobot dan mengubahnya ke dalam suatu objek FuzzySets. Nilai bobot karakteristik akan digunakan dalam proses penghitungan nilai rating karakteristik. Kode Sumber 4.20 menerangkan implementasi pemanggilan data bobot dari basis data. Kode sumber ini akan dijalankan di dalam konstruktor, pada saat objek tersebut dibuat.

```

1 try {
2     ResultSet rs1=koneksi.executeSelect("SELECT
3 FB.SCORE_BOBOTBAWAH, FB.SCORE_BOBOTTENGAH, FB.SCORE_BOBOTATAS
4 "+ "FROM FUZIFIKASI_BOBOT FB, METRIK M, JAWABAN_BOBOT JB,
5 PENILAIAN P " + "WHERE M.ID_METRIK=JB.ID_METRIK "+ "AND
6 FB.ID_FUZIFIKASIBOBOT=JB.ID_FUZIFIKASIBOBOT "+ "AND
7 P.ID_PENILAIAN=JB.ID_PENILAIAN "+ "AND
8 P.ID_PENILAIAN="+idPenilaian + "AND
9 M.ID_METRIK="+this.idMetrik);
10     while(rs1.next())
11     {

```

```

12         double
13         score_bawah=rs1.getDouble("SCORE_BOBOTBAWAH");
14         double
15         score_tengah=rs1.getDouble("SCORE_BOBOTTEGAH");
16         double
17         score_atas=rs1.getDouble("SCORE_BOBOTATAS");
18         skorRobot=new FuzzySets(score_bawah,
19         score_tengah, score_atas);
20     }
21
22     koneksi.Disconnect();
23 } catch (SQLException e) {
24     // TODO Auto-generated catch block
25     e.printStackTrace();
26 }
27

```

Kode Sumber 4. 20 Implementasi Pemanggilan Nilai Bobot pada Kelas Subkarakteristik

Untuk memanggil semua aspek-aspek subkarakteristik di bawahnya, kelas Subkarakteristik memiliki satu *method* yang berfungsi untuk memanggil data-data aspek metrik dan membuat objek Metrik sesuai dengan data pada basis data. *Method* tersebut adalah *method* GetChild. Nantinya, objek-objek Metrik ini akan disimpan dalam suatu objek list. Implementasi *method* GetScore pada kelas Subkarakteristik dapat dilihat pada Kode Sumber 4.21.

```

1  public void GetChild(int idPenilaian)
2  {
3      Koneksi koneksi=new Koneksi();
4      try {
5          ResultSet rs=koneksi.executeSelect("SELECT
6  M2.ID_METRIK" + " FROM METRIK M1, METRIK M2, RELASI_METRIK"
7  + " WHERE M1.ID_METRIK=RELASI_METRIK.ID_METRIKPARENT AND
8  M2.ID_METRIK=RELASI_METRIK.ID_METRIKCHILD" + " AND
9  M1.ID_METRIK="+this.idMetrik);
10
11         while(rs.next())
12         {
13             Subkarakteristik child=new
14             Subkarakteristik(rs.getInt("ID_METRIK"), idPenilaian);
15             this.listChild.add(child);
16         }
17         koneksi.Disconnect();

```

```

18         } catch (SQLException e) {
19             // TODO Auto-generated catch block
20             e.printStackTrace();
21         }
22     }
23

```

Kode Sumber 4. 21 Implementasi *Method* GetChild pada Kelas Subkarakteristik

4.2.3.10 Kelas Submetrik

Kelas ini merupakan representasi dari elemen model kualitas level ketiga, yaitu metrik. Sistem akan membuat objek dari kelas ini ketika objek dari kelas ModelKualitas dibuat dan akan diisi dengan data metrik dan jawaban rating dan bobot untuk metrik dan idpenilaian tertentu. Sesuai dengan model kualitas yang digunakan, kelas ini memiliki List objek Submetrik yang akan terisi apabila metrik tersebut memang memiliki submetrik pada model kualitas.

Kelas Submetrik terdiri dari satu konstruktor, *method* HasBobot, *method* HasChild, *method* GetSource, dan *method* GetScore. *Method* HasBobot merupakan *method* yang digunakan untuk mengecek apakah metrik tersebut memiliki nilai bobot ataukah tidak, karena tidak semua metrik memiliki nilai bobot. Jika *method* ini mengembalikan nilai *false*, maka bobot akan diset nilai *default*, yaitu (1,1,1). *Method* ini mendapatkan informasi tersebut dengan cara memanggil *method* HasChild. *Method* HasChild merupakan *method* yang akan mengecek apakah metrik tersebut memiliki objek Submetrik di bawahnya dengan cara mengecek jumlah objek kelas Submetrik yang terdapat pada List Submetrik yang disimpan oleh kelas ini. Implementasi kedua *method* ini dapat dilihat pada Kode Sumber 4.22 dan Kode Sumber 4.23.

```

1 public void HasBobot()
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 PERTANYAAN_BOBOT " + "FROM METRIK " + "WHERE
7 ID_METRIK="+this.idMetrik);
8

```

```

9         while(rs.next())
10        {
11            if(rs.getString("PERTANYAAN_BOBOT")== "")
12                this.hasScoreBobot=false;
13                else this.hasScoreBobot=true;
14        }
15        koneksi.Disconnect();
16    } catch (SQLException e) {
17        // TODO Auto-generated catch block
18        e.printStackTrace();
19    }
20 }
21 if(this.hasScoreBobot)
22 {
23
24     rs=koneksi.executeQuery("SELECT FB.SCORE_BOBOTBAWAH,
25 FB.SCORE_BOBOTTENGAH, FB.SCORE_BOBOTATAS " + "FROM
26 FUZIFIKASI_BOBOT FB, METRIK M, JAWABAN_BOBOT JB, PENILAIAN P "
27 + "WHERE M.ID_METRIK=JB.ID_METRIK " + "AND
28 FB.ID_FUZIFIKASIBOBOT=JB.ID_FUZIFIKASIBOBOT "+ "AND
29 P.ID_PENILAIAN=JB.ID_PENILAIAN " + "AND
30 P.ID_PENILAIAN="+idPenilaian + "AND
31 M.ID_METRIK="+this.idMetrik);
32     while(rs.next())
33     {
34         double
35         score_bawah=rs.getDouble("SCORE_BOBOTBAWAH");
36         double
37         score_tengah=rs.getDouble("SCORE_BOBOTTENGAH");
38         double
39         score_atas=rs.getDouble("SCORE_BOBOTATAS");
40         skorBobot=new FuzzySets(score_bawah,
41 score_tengah, score_atas);
42     }
43 }
44 else
45     skorBobot=new FuzzySets(1,1,1);
46
47

```

Kode Sumber 4. 22 Implementasi *Method* HasBobot untuk Kelas Subsubmetrik

```

1 public void GetChild(int idPenilaian)
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5

```

```

6         ResultSet rs=koneksi.executeSelect("SELECT
7 M2.ID_METRIK" + " FROM METRIK M1, METRIK M2, RELASI_METRIK"
8 + " WHERE M1.ID_METRIK=RELASI_METRIK.ID_METRIKPARENT AND
9 M2.ID_METRIK=RELASI_METRIK.ID_METRIKCHILD" + " AND
10 M1.ID_METRIK="+this.idMetrik);
11
12         while(rs.next())
13         {
14             Subsubmetrik child=new
15 Subsubmetrik(rs.getInt("ID_METRIK"), idPenilaian);
16             this.listChild.add(child);
17         }
18         koneksi.Disconnect();
19     } catch (SQLException e) {
20         // TODO Auto-generated catch block
21         e.printStackTrace();
22     }
23 }
24 if(this.listChild.isEmpty())
25 {
26     hasChild=false;
27 }
28 else
29     hasChild=true;

```

Kode Sumber 4. 23 Implementasi *Method* HasBobot untuk Kelas Subsubmetrik

Method GetSource merupakan *method* yang akan mengecek dari tabel manakah nilai rating suatu perangkat dapat dipanggil kembali. *Method* ini akan mengecek nilai yang ada pada field *source* pada tabel perangkat lunak di basis data. Sementara itu, *method* GetScore merupakan *method* yang digunakan untuk menghitung nilai skor untuk metrik tersebut. Cara yang dilakukan tergantung dengan nilai HasChild. Jika hasChild bernilai benar, maka *method* akan menghitung *sum of product* dari nilai bobot dan rating dari kelas Submetrik yang disimpan pada List. Sebaliknya, *method* akan mendapatkan nilai rating sesuai dengan yang tersimpan di basis data. Kode sumber untuk *method* GetSource dan GetScore dapat dilihat pada Kode Sumber 4.24 dan Kode Sumber 4.25 berikut ini.

```

1 public void GetSource()
2 {

```

```

3      Koneksi koneksi=new Koneksi();
4      try {
5          ResultSet rs=koneksi.executeSelect("SELECT
6 SOURCE " + "FROM METRIK " + "WHERE ID_METRIK="+this.idMetrik);
7
8          while(rs.next())
9          {
10             if(rs.getInt("SOURCE")==0)
11                 this.fromJawaban=true;
12             else
13                 this.fromJawaban=false;
14         }
15         koneksi.Disconnect();
16     } catch (SQLException e) {
17         // TODO Auto-generated catch block
18         e.printStackTrace();
19     }
20 }
21

```

Kode Sumber 4. 24 Implementasi *Method* HasBotot untuk Kelas Submetrik

```

1  public FuzzySets GetScoreRating()
2  {
3      double skorBawah=0, skorTengah=0, skorAtas=0;
4      for(int i=0; i<this.listChild.size(); i++)
5      {
6          Subsubmetrik ssm=this.listChild.get(i);
7          FuzzySets skorBotot=ssm.skorBotot;
8          FuzzySets skorRating=ssm.skorRating;
9          skorBawah=Math.max(skorBawah,
10 skorBotot.scoreBawah*skorRating.scoreBawah);
11 skorTengah=Math.max(skorTengah,skorBotot.scoreTengah*skorRating.scoreTengah);
12 skorAtas=Math.max(skorAtas,skorBotot.scoreAtas*skorRating.scoreAtas);
13     }
14     FuzzySets result=new FuzzySets(skorBawah, skorTengah,
15 skorAtas);
16     return result;
17 }
18

```

Kode Sumber 4. 25 Implementasi *Method* HasBotot untuk Kelas Submetrik

4.2.3.11 Kelas Subsubmetrik

Kelas ini merupakan representasi dari elemen model kualitas level kedua, yaitu subkarakteristik. Sistem akan membuat objek dari kelas ini ketika objek dari kelas ModelKualitas dibuat dan akan diisi dengan data subkarakteristik dan jawaban rating dan bobot untuk metrik dan idpenilaian tertentu. Kelas ini memiliki satu buah objek List yang berisi objek-objek kelas metrik sesuai dengan model kualitas yang digunakan.

Kelas Submetrik terdiri dari satu konstruktor, *method* HasBobot, *method* GetSource, dan *method* GetScore. *Method* HasBobot merupakan *method* yang digunakan untuk mengecek apakah metrik tersebut memiliki nilai bobot ataukah tidak, karena tidak semua metrik memiliki nilai bobot. Jika *method* ini mengembalikan nilai *false*, maka bobot akan diset nilai *default*, yaitu (1,1,1).

Method GetSource merupakan *method* yang akan mengecek dari tabel manakah nilai rating suatu perangkat dapat dipanggil kembali. *Method* ini akan mengecek nilai yang ada pada field *source* pada tabel perangkat lunak di basis data. Kode sumber untuk *method* GetSource dan HasBobot dapat dilihat pada Kode Sumber 4.26 dan 4.27 berikut ini.

```
1 public void GetSource()
2 {
3     Koneksi koneksi=new Koneksi();
4     try {
5         ResultSet rs=koneksi.executeSelect("SELECT
6 SOURCE " + "FROM METRIK " + "WHERE ID_METRIK="+this.idMetrik);
7
8         while(rs.next())
9         {
10             if(rs.getInt("SOURCE")==0)
11             {
12                 this.fromJawaban=true;
13             }
14
15             else
16             {
17                 this.fromJawaban=false;
18             }
19         }
20     }
21 }
```



```

19         }
20         koneksi.Disconnect();
21     } catch (SQLException e) {
22         // TODO Auto-generated catch block
23         e.printStackTrace();
24     }
25 }

```

Kode Sumber 4. 26 Implementasi *Method* GetScoreRating pada Kelas SubsubMetrik

```

1  public void HasBotot()
2  {
3      Koneksi koneksi=new Koneksi();
4      try {
5          ResultSet rs=koneksi.executeSelect("SELECT
6  PERTANYAAN_BOBOT " + "FROM METRIK " + "WHERE
7  ID_METRIK="+this.idMetrik);
8
9          while(rs.next())
10         {
11             if(rs.getString("PERTANYAAN_BOBOT")== "")
12                 this.hasScoreBotot=false;
13             else this.hasScoreBotot=true;
14         }
15         koneksi.Disconnect();
16     } catch (SQLException e) {
17         // TODO Auto-generated catch block
18         e.printStackTrace();
19     }
20 }
21 if(this.hasScoreBotot)
22 {
23
24     rs=koneksi.executeSelect("SELECT FB.SCORE_BOBOTBAWAH,
25 FB.SCORE_BOBOTTENGAH, FB.SCORE_BOBOTATAS " + "FROM
26 FUZIFIKASI_BOBOT FB, METRIK M, JAWABAN_BOBOT JB, PENILAIAN P "
27 + "WHERE M.ID_METRIK=JB.ID_METRIK " + "AND
28 FB.ID_FUZIFIKASIBOBOT=JB.ID_FUZIFIKASIBOBOT "+ "AND
29 P.ID_PENILAIAN=JB.ID_PENILAIAN " + "AND
30 P.ID_PENILAIAN="+idPenilaian + "AND
31 M.ID_METRIK="+this.idMetrik);
32     while(rs.next())
33     {
34         double
35 score_bawah=rs.getDouble("SCORE_BOBOTBAWAH");
36         double
37 score_tengah=rs.getDouble("SCORE_BOBOTTENGAH");

```

```

38         double
39         score_atas=rs.getDouble("SCORE_BOBOTATAS");
40         skorBobot=new FuzzySets(score_bawah,
41         score_tengah, score_atas);
42     }
43 }
44 else
45     skorBobot=new FuzzySets(1,1,1);
46
47

```

Kode Sumber 4. 27 Implementasi *Method* HasBobot untuk Kelas Subsubmetrik

4.2.4 Implementasi Lapisan Util

Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem. Pada bagian ini akan dijelaskan implementasi dari rancangan lapisan antarmuka.

4.2.4.1 Kelas Parser

Kelas ini merupakan kelas yang digunakan untuk menghasilkan AST untuk suatu kode sumber. Kelas ini menggunakan pustaka JDT untuk menjalankan fungsinya. Dalam kelas ini, terdapat *method* untuk membaca setiap *file* yang ada dalam suatu folder penyimpanan kode sumber yaitu *method* ParseFileInDir. Fungsi tersebut bisa dilihat pada Kode Sumber 4.28 di bawah ini.

```

1     public static void ParseFilesInDir(String dirPath) throws
2     IOException{
3
4         File root = new File(dirPath);
5         File[] files = root.listFiles ( );
6         String filePath = null;
7
8         for (File f : files ) {
9             filePath = f.getAbsolutePath();
10            if(f.isFile()){
11                parse(readFileToString(filePath));
12                jumlahKLOC+=getLineCount(filePath);
13            }
14        }
15    }

```

Kode Sumber 4. 28 Implementasi *Method* ParseFileInDir

Selain fungsi yang telah disebutkan di atas, di dalam kelas Parser ini juga terdapat *method* Parse untuk menghasilkan AST dari suatu kode sumber dengan mengambil parameter yaitu berupa *string of source code*. Dalam *method* ini, versi JLS yang digunakan adalah JLS versi 3. Kode sumber untuk *method* parser bisa dilihat pada kode sumber 4.29 di bawah ini.

```
1 public static void parse(String str) {
2     ASTParser parser = ASTParser.newParser(AST.JLS3);
3     parser.setSource(str.toCharArray());
4     parser.setKind(ASTParser.K_COMPILATION_UNIT);
5
6     final CompilationUnit cu = (CompilationUnit)
7     parser.createAST(null);
8     jumlahCC=VisitorCC.Visit(cu);
9     jumlahStatic=VisitorStatic.Visit(cu);
10 }
```

Kode Sumber 4. 29 Implementasi Method Parse

Selain itu, pada kelas Parser ini, terdapat satu fungsi untuk mendapatkan KLOC. Fungsi ini mengambil paramater berupa kode sumber dalam bentuk string, melakukan proses pemecahan string ke dalam *substring-substring*, dan penghapusan tanda seperti //,/,*,dan \, lalu menyimpan *substring* tersebut ke dalam suatu array. Kode sumber yang digunakan untuk mendeklarasikan *method* ini dapat dilihat di Kode Sumber 4.30 berikut.

```
1 public static double getLineCount(String FilePath) {
2     String code = null;
3     System.out.println(FilePath);
4     try {
5         code = new Scanner(new
6     File(FilePath)).useDelimiter("\\A").next();
7     } catch (FileNotFoundException e) {
8         // TODO Auto-generated catch block
9         e.printStackTrace();
10    }
11    code = Pattern.compile("/\\*.?*\\*/|/.*?$/",
12    Pattern.MULTILINE |
13    Pattern.DOTALL).matcher(code).replaceAll("");
14 }
```

```

15     String[] s = Pattern.compile("\\S.*?$",
16     Pattern.MULTILINE).split(code.trim());
17     System.out.println("KLOC:
18     "+(double)s.length/(double)1000);
19     return((double)s.length/(double)1000);
    }

```

Kode Sumber 4. 30 Implementasi *Method* GetLineCount

4.2.4.2 Kelas Visitor

Kelas ini merupakan kelas yang mengimplementasikan visitor pattern pada sistem ini. Kelas ini digunakan untuk mendapatkan jumlah kompleksitas siklomatik dan jumlah variabel global (statis) dari kode sumber perangkat lunak. Untuk mendapatkan jumlah kompleksitas siklomatik, *method* VisitCC pada kelas ini akan menghitung jumlah *node* dalam AST yang di visit. *Node-node* tersebut bisa dilihat pada Tabel 4.1 di bawah ini dan Kode Sumber 4.31 di bawah ini merupakan kode sumber yang digunakan pada *method* VisitCC.

Tabel 4. 1 Tabel Implementasi Pola Visitor untuk Menghitung Kompleksitas Siklomatik

<i>Statement</i> dalam Java	<i>ASTNode</i>
<i>Return</i>	<i>ReturnStatement</i>
<i>Else</i>	<i>IfStatement</i>
<i>For</i>	<i>ForStatement</i>
<i>While</i>	<i>WhileStatement</i>
<i>Throw</i>	<i>ThrowStatement</i>
<i>Break</i>	<i>BreakStatement</i>
<i>Continue</i>	<i>ContinueStatement</i>
<i>Do</i>	<i>DoStatement</i>
<i>Case</i>	<i>SwitchCaseStatement</i>
<i>Catch</i>	<i>TryStatement</i>
<i>Conditional Expression</i> (?:)	<i>ConditionalExpression</i>
<i>Logic Expression</i> (&&,)	<i>InfixExpression</i>

```

1     public static int Visit(CompilationUnit cu) {
2
3         cu.accept(new ASTVisitor() {

```

```
4
5     public boolean visit (ReturnStatement node) {
6         jumlahStatement++;
7         return true;
8     }
9
10    public boolean visit (IfStatement node) {
11        jumlahStatement++;
12        return true;
13    }
14
15    public boolean visit (ForStatement node) {
16        jumlahStatement++;
17        return true;
18    }
19
20    public boolean visit (WhileStatement node) {
21        jumlahStatement++;
22        return true;
23    }
24
25    public boolean visit (ThrowStatement node) {
26        jumlahStatement++;
27        return true;
28    }
29
30    public boolean visit (BreakStatement node) {
31        jumlahStatement++;
32        return true;
33    }
34
35    public boolean visit (ContinueStatement node) {
36        jumlahStatement++;
37        return true;
38    }
39
40    public boolean visit (DoStatement node) {
41        jumlahStatement++;
42        return true;
43    }
44
45    public boolean visit (SwitchCase node) {
46        jumlahStatement++;
47        return true;
48    }
49
50    public boolean visit (TryStatement node) {
51        jumlahStatement++;
52    }
```

```

53         return true;
54     }
55
56     public boolean visit (ConditionalExpression node) {
57         jumlahStatement++;
58         return true;
59     }
60
61     public boolean visit (InfixExpression node) {
62
63         if(node.getOperator()==InfixExpression.Operator.CONDITIONAL_AND)
64             jumlahStatement++;
65         else
66             if(node.getOperator()==InfixExpression.Operator.CONDITIONAL_OR)
67                 jumlahStatement++;
68                 return true;
69             }
70         });
71         System.out.println("Jumlah kompleksitas siklomatik:
72 "+jumlahStatement);
73         return jumlahStatement;
74     }
75
76
77
78

```

Kode Sumber 4. 31 Implementasi VisitorCC

Sementara itu untuk mendapatkan jumlah variabel statis, terdapat *method* VisitStatic yang akan mengunjungi *node VariableDeclaration* pada AST yang dilewatkan melalui parameter *method* tersebut. Untuk menghindari adanya deklarasi *method* static, *method* ini juga mengunjungi *node MethodDeclaration* namun mengembalikan nilai false sehingga *node* yang lain tidak dikunjungi lebih lanjut. Kode sumber untuk fungsi VisitStatic dapat dilihat pada kode sumber 4.32 berikut ini

```

1     public static int Visit(CompilationUnit cu) {
2         cu.accept(new ASTVisitor() {
3
4             public boolean visit(MethodDeclaration node)
5             {
6                 return false;

```

```

7         }
8
9         public boolean visit(Modifier node)
10        {
11            if(node.isStatic())
12                jumlahStatic++;
13            return true;
14        }
15
16    });
17    System.out.println("Jumlah variabel static:
18    "+jumlahStatic);
19    return jumlahStatic;
20 }

```

Kode Sumber 4. 32 Implementasi VisitStatic

4.2.4.3 Kelas SharedData

Kelas SharedData merupakan kelas yang berperan sebagai session, dimana kelas ini akan menyimpan sejumlah data yang akan digunakan dalam beberapa antarmuka, seperti idPenilaian dan idPerangkatLunak. Karena kelas ini digunakan untuk menyimpan variabel yang bersifat global, semua variabel di kelas ini bertipe statis. Dalam *method* ini, terdapat *method* setter/getter untuk setiap variabel dalam kelas SharedData.

4.2.4.4 Kelas Koneksi

Kelas ini merupakan kelas pembantu yang digunakan untuk mengoneksikan basis data dengan aplikasi perangkat lunak. Pada kelas ini terdapat *method* untuk menjalankan kueri select untuk mendapatkan data dari basis data dan kueri *update* untuk melakukan perubahan data dalam basis data. Selain itu, di dalam kelas ini juga terdapat konstruktor yang akan membuat koneksi dengan basis data dan *method* Disconnect yang akan menghapus koneksi perangkat lunak dengan basis data.

4.2.4.5 Kelas FuzzySets

Kelas ini merupakan kelas pembantu yang digunakan untuk membungkus suatu nilai fuzzy triangular yang terdiri dari nilai bawah, nilai tengah, dan nilai atas. Untuk masing-masing nilai memiliki tipe

double, disesuaikan dengan data asli nilai fuzzy triangularnya. Kelas ini berisikan suatu konstruktor yang mengambil tiga parameter berupa nilai bawah, nilai tengah, dan nilai atas.

4.2.4.6 Kelas CrispValue

Kelas ini merupakan kelas yang digunakan untuk mendapatkan crisp value untuk suatu fuzzy sets. *Crisp value* didapatkan dengan mengaplikasikan persamaan yang telah dijabarkan pada subbab 3.2.2, yaitu menggunakan rumus centroid. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 4.33 di bawah ini.

```
1 public double getCrispValue()
2 {
3     double deltaZ1=this.fs.scoreBawah-
4 this.fs.scoreTengah;
5     double deltaZ2=fs.scoreAtas-fs.scoreTengah;
6     double pengali1=1.00/deltaZ1;
7     double pengali2=1.00/deltaZ2;
8     double kons1=fs.scoreBawah*pengali1;
9     double kons2=fs.scoreAtas*pengali2;
10    double m1=pengali1*-1;
11    double m2=pengali2*-1;
12
13    double crispValue=(GetNominatorIntegralValue(m1,
14 kons1, fs.scoreBawah,
15 fs.scoreTengah)+GetNominatorIntegralValue(m2, kons2,
16 fs.scoreTengah,
17 fs.scoreAtas))/(GetDenominatorIntegralValue(m1, kons1,
18 fs.scoreBawah,
19 fs.scoreTengah)+GetDenominatorIntegralValue(m2, kons2,
20 fs.scoreTengah, fs.scoreAtas));
21    return crispValue;
22 }
23
24 public double GetNominatorIntegralValue(double m, double
25 kons, double a, double b)
26 {
27     TrapezoidIntegrator trapezoid = new
28 TrapezoidIntegrator();
29
30     double[] vector = new double[3];
31     vector[0] = 0;
32     vector[1] = kons;
33     vector[2] = m;
34
35 }
```



```

36         UnivariateFunction uf = (UnivariateFunction)new
37 PolynomialFunction(vector);
38         double j = trapezoid.integrate(10000, uf, a, b);
39         System.out.println("Trapezoid integral : "
40 + j);
41         return j;
42     }
43
44     public double GetDenominatorIntegralValue(double m, double
45 kons, double a, double b)
46     {
47         TrapezoidIntegrator trapezoid = new TrapezoidIntegrator();
48
49         double[] vector = new double[2];
50         vector[0] = kons;
51         vector[1] = m;
52         UnivariateFunction uf = (UnivariateFunction)new
53 PolynomialFunction(vector);
54
55         double j = trapezoid.integrate(10000, uf, a, b);
56         System.out.println("Trapezoid integral : " + j);
57         return j;
58     }
59

```

Kode Sumber 4. 33 Implementasi Kelas CrispValue

Kelas ini terdiri dari 3 *method*, yaitu *GetCrispValue*, *GetNominatorValue*, dan *GetDenominatorValue*. Sementara itu, *GetNominatorValue* merupakan *method* yang digunakan untuk mendapatkan nilai integral dari persamaan yang merupakan pembilang dari rumus centroid dan *method* *GetDenominatorValue* merupakan *method* yang digunakan untuk mendapatkan nilai integral dari persamaan penyebut. Kedua kelas ini dipanggil oleh *method* *GetCrispValue* untuk mendapatkan crisp value dari suatu objek fuzzzsets.

Kelas ini menggimpor beberapa kelas dari pustaka ApacheMath, yaitu kelas *TrapezoidIntegrator* dan *PolynominalValue*. Kelas *PolynominalValue* merupakan kelas yang digunakan untuk membangun persamaan yang akan dilakukan proses integral dengan masukkan berupa array yang menjelaskan konstanta untuk masing-masing suku. Sementara itu kelas *TrapezoidIntegrator* merupakan

kelas yang digunakan untuk menghitung nilai integral menggunakan pendekatan trapezoid.

4.2.4.7 Kelas Unzip

Kelas ini merupakan kelas pembantu yang digunakan untuk mengekstrak *file* zip kode sumber yang dimasukkan oleh pengguna melalui halaman UploadPL. Kelas ini akan mengekstrak *file* ke dalam alamat direktori yang sudah ditentukan dalam sistem secara *hardcode*. Kode sumber untuk proses ekstraksi *file* dalam kelas *Unzip* dapat dilihat pada Kode Sumber 4.34 berikut ini.

```
1 public void unzip(String zipFilePath, String destDirectory)
2 throws IOException {
3     File destDir = new File(destDirectory);
4     if (!destDir.exists()) {
5         destDir.mkdir();
6     }
7     ZipInputStream zipIn = new ZipInputStream(new
8 FileInputStream(zipFilePath));
9     ZipEntry entry = zipIn.getNextEntry();
10
11     while (entry != null) {
12         String filePath = destDirectory + File.separator
13 + entry.getName();
14         if (!entry.isDirectory()) {
15             extractFile(zipIn, filePath);
16         } else {
17             File dir = new File(filePath);
18             dir.mkdir();
19         }
20         zipIn.closeEntry();
21         entry = zipIn.getNextEntry();
22     }
23     zipIn.close();
24 }
25
26 private void extractFile(ZipInputStream zipIn, String
27 filePath) throws IOException {
28     BufferedOutputStream bos = new
29 BufferedOutputStream(new FileOutputStream(filePath));
30     byte[] bytesIn = new byte[BUFFER_SIZE];
31     int read = 0;
32     while ((read = zipIn.read(bytesIn)) != -1) {
33         bos.write(bytesIn, 0, read);
34     }
```

```

35     }
36     bos.close();
    }

```

Kode Sumber 4. 34 Implementasi Kelas Unzip

4.2.4.8 Kelas OlahMetrik

Kelas ini merupakan kelas pembantu yang digunakan untuk melakukan perhitungan nilai-nilai rating dari data angka-angka yang dimasukkan oleh pengguna melalui halaman GUIRatingIsian. Kelas ini akan mengimplementasikan rumus-rumus penghitungan nilai rating seperti yang tertera di Tabel 3.11 pada bab sebelumnya. Kelas OlahMetrik terdiri dari 7 *method* yang masing-masing *method* akan melakukan perhitungan untuk satu metrik dan akan mengembalikan nilai berupa suatu objek kelas Jawaban. Pada *method* ini juga terdapat proses fuzifikasi dengan mencocokkan nilai yang diperoleh dengan data fuzifikasi metrik yang ada pada basis data. Implementasi rumus tersebut pada kelas OlehMetrik dapat dilihat pada KodeSumber 4.35.

```

1  public Jawaban HitungKesesuaianOperasi(int
2  jumlahOperasiSesuai, int jumlahOperasi)
3  {
4      double
5  r=(double)jumlahOperasiSesuai/(double)jumlahOperasi;
6      double result=1-r;
7      String idFuzifikasi=null;
8      if(result<=0.3)
9          idFuzifikasi="1";
10     else if(result<=0.5)
11         idFuzifikasi="2";
12     else if(result<=0.7)
13         idFuzifikasi="3";
14     else if(result<=0.85)
15         idFuzifikasi="4";
16     else
17         idFuzifikasi="5";
18     Jawaban kesesuaian=new Jawaban(1,idFuzifikasi);
19     return kesesuaian;
20 }
21
22 public Jawaban HitungAkurasiOperasi(int jumlahAkurasi, int
23 jumlahOperasi)
24 {
25

```

```

26     double
27     result=(double)jumlahAkurasi/(double)jumlahOperasi;
28     String idFuzifikasi=null;
29     if(result<=0.3)
30         idFuzifikasi="6";
31     else if(result<=0.5)
32         idFuzifikasi="7";
33     else if(result<=0.7)
34         idFuzifikasi="8";
35     else if(result<=0.85)
36         idFuzifikasi="9";
37
38     else
39         idFuzifikasi="10";
40     Jawaban akurasi=new Jawaban(2,idFuzifikasi);
41     return akurasi;
42 }
43
44 public Jawaban HitungKontrolAkses(int jumlahDisediakan, int
45 jumlahDibutuhkan)
46 {
47     double
48     result=(double)jumlahDisediakan/(double)jumlahDibutuhkan;
49     String idFuzifikasi=null;
50     if(result<=0.3)
51         idFuzifikasi="17";
52     else if(result<=0.5)
53         idFuzifikasi="18";
54     else if(result<=0.7)
55         idFuzifikasi="19";
56     else if(result<=0.85)
57         idFuzifikasi="20";
58
59     else
60         idFuzifikasi="21";
61     Jawaban kontrol=new Jawaban(5,idFuzifikasi);
62     return kontrol;
63 }
64
65 public Jawaban HitungKustomisasi(int jumlahKustomisasi)
66 {
67     double result=1.00-(1.00/(double)jumlahKustomisasi);
68     String idFuzifikasi=null;
69     if(result<=0.4)
70         idFuzifikasi="27";
71     else if(result<=0.6)
72         idFuzifikasi="28";
73     else if(result<=0.8)
74         idFuzifikasi="29";
75
76     else

```

```

74         idFuzifikasi="30";
75         Jawaban kustomisasi=new Jawaban(8,idFuzifikasi);
76         return kustomisasi;
77     }
78
79     public Jawaban HitungCPU(int penggunaanCPU)
80     {
81         double result=1.00-(1.00/(double)penggunaanCPU);
82         String idFuzifikasi=null;
83         if(result<=0.2)
84             idFuzifikasi="40";
85         else if(result<=0.4)
86             idFuzifikasi="41";
87         else if(result<=0.6)
88             idFuzifikasi="42";
89         else
90             idFuzifikasi="43";
91         Jawaban cpu=new Jawaban(12,idFuzifikasi);
92         return cpu;
93     }
94
95     public Jawaban HitungKLOCPerTim(double jumlahKLOC, int
96     jumlahTim)
97     {
98         double result=jumlahKLOC/(double)jumlahTim;
99         String idFuzifikasi=null;
100        if(result>5)
101            idFuzifikasi="88";
102        else if(result>3)
103            idFuzifikasi="89";
104        else if(result>1)
105            idFuzifikasi="90";
106        else
107            idFuzifikasi="91";
108        Jawaban kloc=new Jawaban(25,idFuzifikasi);
109        return kloc;
110    }
111
112    public Jawaban HitungPersentaseProperti(int propertiKustom,
113    int jumlahProperti)
114    {
115        double result=(double)propertiKustom/jumlahProperti;
116        String idFuzifikasi=null;
117        if(result<=0.3)
118            idFuzifikasi="112";
119        else if(result<=0.5)
120            idFuzifikasi="111";
121        else if(result<=0.7)
122

```

```

123         idFuzifikasi="110";
124     else if(result<=0.85)
125         idFuzifikasi="109";
126     else
127         idFuzifikasi="108";
128     Jawaban kustomisasi=new Jawaban(30,idFuzifikasi);
129     return kustomisasi;
130 }

```

Kode Sumber 4. 35 Implementasi Kelas OlahMetrik

4.2.5 Implementasi Laporan Penilaian Kualitas Perangkat Lunak

Laporan ini berisikan tentang nilai-nilai aspek kualitas perangkat lunak untuk suatu penilaian yang telah dilakukan. Data-data yang diperlukan dalam laporan ini dipanggil dari tabel-tabel yang ada pada basis data, sehingga diperlukan suatu syntax untuk mendapatkan data tersebut. Syntax SQL yang digunakan dalam pembuatan laporan ini dapat dilihat pada Gambar 4.1 berikut.

```

SELECT M1."NAMA_METRIK",ROUND(DR."SKOR",3) AS SKOR
FROM "DATA_REPORT" DR INNER JOIN "METRIK" M1 ON
DR."ID_METRIK" = M1."ID_METRIK"
WHERE M1.ID_TIPEMETRIK = 1
AND DR.ID_PENILAIAN=$P{idPenilaian}

```

Gambar 4. 1 Syntax SQL Menampilkan Nilai Karakteristik

Syntax yang ada pada Gambar 4.1 merupakan syntax yang digunakan untuk mendapatkan data nilai karakteristik untuk suatu penilaian perangkat lunak. Syntax ini akan memanggil tabel-tabel lain yang berelasi, seperti tabel Data Report, tabel metrik, dan tabel relasi metrik. Id_tipemetrik 1 merupakan data pada tabel tipemetrik dimana id yang bernilai 1 menunjukkan bahwa aspek tersebut merupakan karakteristik Untuk pemanggilan data nilai subkarakteristik secara umum hampir sama, namun dengan sedikit perbedaan pada parameter yang digunakannya. Syntax SQL untuk menampilkan nilai subkarakteristik dapat dilihat pada Gambar 4.2.

```
SELECT M1.NAMA_METRIK, ROUND(DR.SKOR,3) AS SKOR
FROM DATA_REPORT DR, METRIK M1, TIPE_METRIK TM
WHERE DR.ID_METRIK=M1.ID_METRIK
AND DR.ID_PENILAIAN=${idPenilaian}
AND TM.ID_TIPEMETRIK=M1.ID_TIPEMETRIK
AND TM.ID_TIPEMETRIK=2
ORDER BY M1.ID_METRIK ASC
```

Gambar 4. 2 Syntax SQL Menampilkan Nilai Subkarakteristik

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan untuk menguji apakah fungsionalitas aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. Pengujian ini mengacu pada kasus penggunaan pada bab tiga

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan “Rancang Bangun Kakas Bantu Pengukuran Kualitas Perangkat Lunak Menggunakan Metode Fuzzy” pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i5 CPU @ 2.50GHz
 - b. Memori(RAM): 4 GB
 - c. Tipe sistem: Sistem operasi 32-bit
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Ultimate
 - b. Perangkat pengembang: Eclipse Luna IDE

5.2 Skenario Uji Coba

Uji coba pada sistem ini mengacu pada pengujian *blackbox* untuk menguji apakah fungsionalitas sistem telah berjalan sebagaimana mestinya. Pengujian mengacu pada setiap fitur yang telah diimplementasikan. Hasil pengujian dapat dilihat pada Tabel 5.1 hingga Tabel 5.10. Tampilan dari hasil pengujian tiap fitur dapat dilihat pada Gambar 5.1 hingga Gambar 5.12.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.3. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

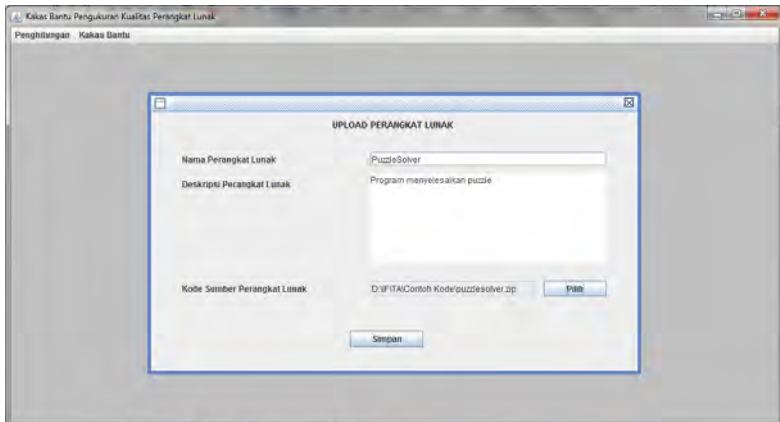
5.2.1.1 Pengujian Fitur Mengunggah Perangkat Lunak

Pengujian fitur mengunggah perangkat lunak merupakan pengujian terhadap kemampuan kakas bantu untuk menyimpan kode sumber perangkat lunak yang dimasukkan pengguna. Pengujian ini juga menguji kemampuan sistem untuk mengekstrak *file* kode sumber berekstensi .zip. Skenario pertama yaitu pengunggahan perangkat lunak dengan alur kejadian normal dan skenario kedua yaitu pengunggahan perangkat lunak dengan alur kejadian alternatif. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.1 dan Tabel 5.2.

Tabel 5. 1 Pengujian Fitur Mengunggah Perangkat Lunak Skenario 1

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur mengunggah perangkat lunak
Tujuan Pengujian	Menguji fitur untuk mengunggah perangkat lunak ke dalam sistem
Skenario	Perangkat lunak yang diunggah adalah perangkat lunak GA-MLP
Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan.

Data Uji	Data yang digunakan adalah data kakas bantu GA-MLP
Langkah Pengujian	Pengguna memilih menu unggah perangkat lunak dari halaman utama
Hasil Yang Diharapkan	Perangkat lunak dapat disimpan oleh sistem.
Hasil Yang Didapat	Data perangkat lunak berhasil disimpan ke dala sistem
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil deteksi dapat dilihat pada Gambar 5.1 dan penyimpanan data pada basis data dapat dilihat pada Gambar 5.2.



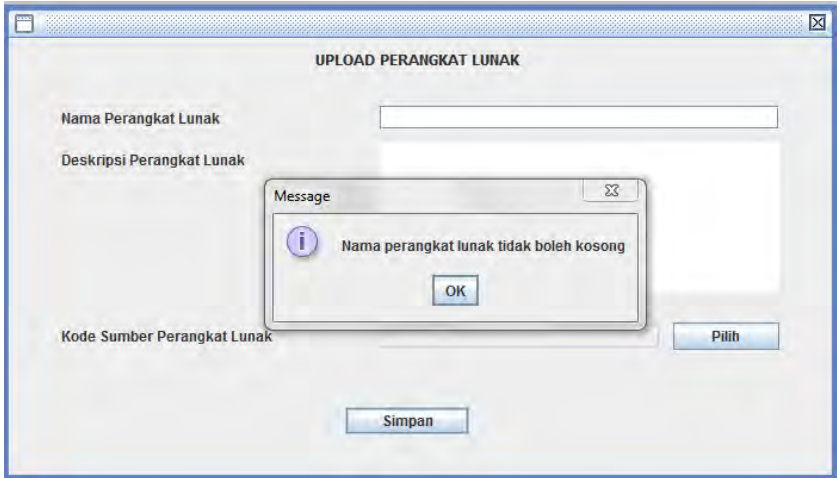
Gambar 5. 1 Tampilan Hasil Pengujian Fitur Mengunggah Perangkat Lunak Skenario 1

ID_PERANGKATLUNAK	NAMA_PERANGKATLUNAK	DESKRIPSI_PERANGKATLUNAK	KODE_SUMBER	TOTAL_SKOR
31	MLP-GA	Implementasi MLP dan GA menggunakan bahasa Java	D:\IFTA\Ekstrak Kode\MLP-GA\	0.0

Tabel 5. 2 Pengujian Fitur Mengunggah Perangkat Lunak

Gambar 5. 2 Penyimpanan Data dalam Basis Data Skenario 2

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur mengunggah perangkat lunak
Tujuan Pengujian	Menguji fitur untuk mengunggah perangkat lunak ke dalam sistem
Skenario	Data yang dimasukkan berupa string kosong
Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan.
Data Uji	Data yang digunakan adalah data kakas bantu GA-MLP
Langkah Pengujian	Pengguna memilih menu unggah perangkat lunak dari halaman utama
Hasil Yang Diharapkan	Perangkat lunak menampilkan pesan bahwa data yang dimasukkan tidak valid
Hasil Yang Didapat	Perangkat lunak menampilkan peringatan
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan program dapat dilihat pada Gambar 5.3



Gambar 5. 3 Hasil Uji Fitur Mengunggah Perangkat Lunak Skenario 2

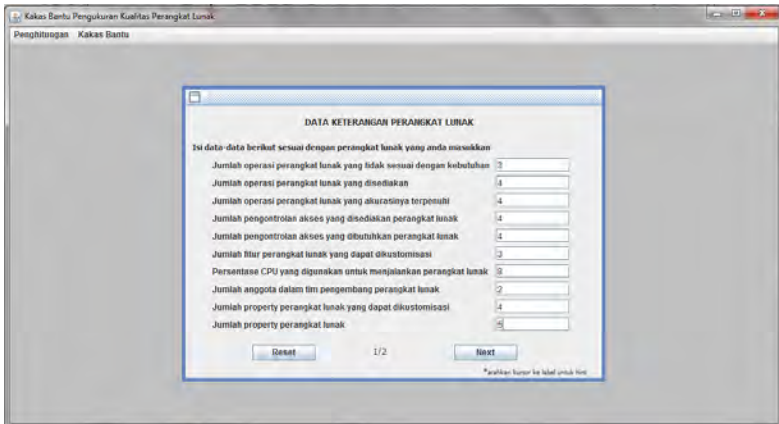
5.2.1.2 Pengujian Fitur Mengisi Kuesioner Rating Perangkat Lunak

Pengujian fitur mengisi kuesioner rating perangkat lunak merupakan pengujian terhadap kemampuan kaskas bantu untuk menyimpan jawaban kuesioner rating perangkat lunak yang dimasukkan pengguna. Pengujian ini juga menguji kemampuan sistem untuk memvalidasi masukan dari pengguna. Pengujian ini dilakukan dengan menggunakan dua skenario. Skenario pertama yaitu pengisian data dengan alur kejadian normal dan skenario kedua yaitu pengisian data dengan alur kejadian alternatif. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.3 dan Tabel 5.4.

Tabel 5. 3 Pengujian Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1

ID	UJ.UC-0002
Referensi Kasus Penggunaan	UC-0002

Nama	Pengujian fitur mengisi kuesioner rating perangkat lunak
Tujuan Pengujian	Menguji fitur untuk mengisi kuesioner rating dari perangkat lunak yang dipilih.
Skenario	Data yang diisikan pada halaman Kuesioner Rating Numerik berupa data angka
Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan, dan pengguna telah berhasil memasukkan perangkat lunak ke dalam sistem.
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP
Langkah Pengujian	Pengguna menjawab pertanyaan-pertanyaan dalam kuesioner rating perangkat lunak
Hasil Yang Diharapkan	Jawaban kuesioner berhasil tersimpan dalam basis data
Hasil Yang Didapat	Kakas bantu berhasil menyimpan jawaban kuesioner perangkat lunak
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan program dapat dilihat pada Gambar 5.4 dan Gambar 5.6 dan tampilan isi basis data dapat dilihat pada Gambar 5.5



Gambar 5. 4 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1 (1)

ID_HASILPENGHITUNGAN	ID_PERANGKATLUNAK	ID_FUZIFIKASIMETRIK
342	31	35
343	31	58
344	31	95
345	31	50
346	31	31
347	31	5
348	31	8
349	31	17
350	31	27
351	31	91
352	31	43
353	31	109
354	31	16
355	31	24
356	31	102
357	31	39
358	31	61
true)	31	127

Gambar 5. 5 Penyimpanan Data Kuesioner Rating Perangkat Lunak pada Basis Data

Gambar 5. 6 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 1 (2)

Tabel 5. 4 Pengujian Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 2

ID	UJ.UC-0003
Referensi Kasus Penggunaan	UC-0002
Nama	Pengujian fitur mengisi kuesioner rating perangkat lunak
Tujuan Pengujian	Menguji fitur untuk mengisi kuesioner rating dari perangkat lunak yang dipilih.
Skenario	Data yang diisikan pada halaman Kuesioner Rating Numerik berupa data non numerik
Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan, dan pengguna telah berhasil memasukkan perangkat lunak ke dalam sistem.
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP

Langkah Pengujian	Sistem menampilkan pesan bahwa data yang dimasukkan tidak valid
Hasil Yang Diharapkan	Jawaban kuesioner berhasil tersimpan dalam basis data
Hasil Yang Didapat	Kakas bantu berhasil menampilkan peringatan bahwa data yang dimasukkan tidak valid
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan program dapat dilihat pada Gambar 5.7

DATA KETERANGAN PERANGKAT LUNAK

Isi data-data berikut sesuai dengan perangkat lunak yang anda masukkan

Jumlah operasi perangkat lunak yang tidak sesuai dengan kebutuhan

Jumlah operasi perangkat lunak yang sesuai dengan kebutuhan

Jumlah operasi perangkat lunak yang tidak sesuai dengan kebutuhan

Jumlah properti perangkat lunak yang dapat dikustomisasi

Jumlah properti perangkat lunak yang dapat dikustomisasi

Jumlah fitur perangkat lunak yang tidak sesuai dengan kebutuhan

Persentase CPU yang digunakan untuk menjalankan perangkat lunak

Jumlah anggota dalam tim pengembang perangkat lunak

Jumlah property perangkat lunak yang dapat dikustomisasi

Jumlah property perangkat lunak

1/2

*arahkan kursor ke label untuk hint.

Gambar 5. 7 Hasil Uji Fitur Mengisi Kuesioner Rating Perangkat Lunak Skenario 2

5.2.1.3 Pengujian Fitur Melihat Daftar Perangkat Lunak

Pengujian fitur melihat daftar perangkat lunak merupakan pengujian terhadap kemampuan kakas bantu untuk menampilkan daftar perangkat lunak yang disimpan dalam sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.5.

Tabel 5. 5 Pengujian Fitur Melihat Daftar Perangkat Lunak

ID	UJ.UC-0004
Referensi Kasus Penggunaan	UC-0003
Nama	Pengujian fitur melihat daftar perangkat lunak

Tujuan Pengujian	Menguji fitur untuk menampilkan daftar perangkat lunak yang disimpan oleh sistem
Skenario	Informasi terkait perangkat lunak ditampilkan dalam bentuk tabel
Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan.
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP
Langkah Pengujian	Pengguna memilih menu list perangkat lunak dari halaman utama.
Hasil Yang Diharapkan	Data perangkat lunak dalam basis data dapat ditampilkan ke aplikasi
Hasil Yang Didapat	Semua data perangkat lunak pada sistem berhasil ditampilkan ke halaman dalam bentuk tabel
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan program dapat dilihat pada Gambar 5.8.

DAFTAR PERANGKAT LUNAK				
Pilih satu perangkat lunak dari daftar dibawah ini				
ID	Perangkat Lunak	Deskripsi	Nilai Kualitas	Keterangan
20	Program1	Program1	0.00	Cukup berkualitas
21	Cobalagi	Coba Fungsionalit...	0.0	Sangat kurang ber...
22	ProgramABC	Program1	0.0	Sangat kurang ber...
23	Word Counter	Menghitung jumlah...	0.0	Sangat kurang ber...
24	Word Counter	Program ini dilaku...	0.0	Sangat kurang ber...
25	FP KK	Implementasi MLP...	0.0	Sangat kurang ber...
26	WordCounter	Aplikasi penghitun...	0.0	Sangat kurang ber...
27	FP KK	Implementasi MLP...	0.0	Sangat kurang ber...
28	MLP-GA	Implementasi Multi...	0.0	Sangat kurang ber...
29	GA-MLP	Implementasi MLP...	0.0	Sangat kurang ber...
30	GA-MLP	Implementasi GA-...	0.0	Sangat kurang ber...
31	MLP-GA	Implementasi MLP...	0.0	Sangat kurang ber...

Lihat Ha... Buat Penilaian

Gambar 5. 8 Hasil Uji Fitur Menampilkan Daftar Perangkat Lunak

5.2.1.4 Pengujian Fitur Mengisi Kuesioner Subjektif

Pengujian fitur mengisi kuesioner subjektif merupakan pengujian terhadap kemampuan kaskas untuk menyimpan jawaban kuesioner subjektif yang dimasukkan pengguna. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6.

Tabel 5. 6 Pengujian Fitur Mengisi Kuesioner Subjektif

ID	UJ.UC-0005
Referensi Kasus Penggunaan	UC-0004
Nama	Pengujian fitur mengisi kuesioner subjektif
Tujuan Pengujian	Menguji fitur untuk mengisi penilaian subjektif dari perangkat lunak yang dipilih.
Skenario	Data yang digunakan adalah data perangkat lunak GA-MLP

Kondisi Awal	Kakas bantu sedang berjalan, tidak ada aplikasi lain yang terhubung dengan basis data yang digunakan, dan pengguna telah memilih salah satu perangkat lunak dari daftar yang ada.
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu perangkat lunak pada daftar yang disediakan. 2. Pengguna menekan tombol buat penilaian 3. Pengguna menjawab pertanyaan-pertanyaan subjektif pada kakas bantu
Hasil Yang Diharapkan	Jawaban pengguna terhadap kuesioner subjektif dapat disimpan oleh sistem
Hasil Yang Didapat	Jawaban kuesioner subjektif berhasil tersimpan ke dalam basis data.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan kode program dapat dilihat pada Gambar 5.9, Gambar 5.10, Gambar 5.11, Gambar 5.12, dan Gambar 5.14. Tampilan basis data dapat dilihat pada Gambar 5.13.

Kakas Bantu Pengukuran Kualitas Perangkat Lunak

Penghitungan Kakas Bantu

KUESIONER RATING SUBJEKTIF

Isi kuesioner subjektif di bawah ini sesuai dengan perangkat lunak yang diada

Pemenuhan terhadap prestasi yang dibutuhkan perangkat lu... Terpenuhi Tidak terpenuhi

Pemenuhan perangkat lunak terhadap standar fungsionalitas Terpenuhi Tidak terpenuhi

Jumlah kasus uji coba yang mencukupi Jumlah menc... Jumlah tidak mencukupi

Pemenuhan perangkat lunak terhadap standar portability Terpenuhi Tidak terpenuhi

Pemenuhan perangkat lunak terhadap standar efisiensi Terpenuhi Tidak terpenuhi

Pemenuhan perangkat lunak terhadap standar maintaiabil... Terpenuhi Tidak terpenuhi

Kerjasama antar tim pengembang perangkat lu... Buruk Cukup Baik Sangat baik

SOB anggota tim pengembang perangkat lunak Buruk Cukup Baik Sangat baik

Performa keseluruhan tim pengembang per... Buruk Cukup Baik Sangat baik

Kemudahan dalam tracking versi lama perangkat L... Sangat mudah dan sangat nyaman Mudah dan nyaman Tidak mudah tetapi nyaman Sulit dan tidak nyaman Sangat sulit dan tidak nyaman

1/6 Next >> Last >>>

*Pilihkan Survey dan label untuk Item.

Gambar 5. 9 Hasil Uji Fitur Mengisi Kuesioner Rating Subjektif Perangkat Lunak

Kakas Bantu Pengukuran Kualitas Perangkat Lunak

Penghitungan Kakas Bantu

KUESIONER BOBOT FUNGSIONALITAS

Seberapa besar pengaruh dari aspek berikut terhadap kualitas perangkat lunak:

Persentase operasi yang memenuhi akurasi yang dibutuhkan

Pemenuhan terhadap prestasi yang dibutuhkan

Persentase pengontrolan akses yang disediakan

Tingkat kekelatras akses

Kesesuaian dengan kebutuhan

Akurasi

Interoperability

Keamanan

Pemenuhan terhadap standar fungsionalitas

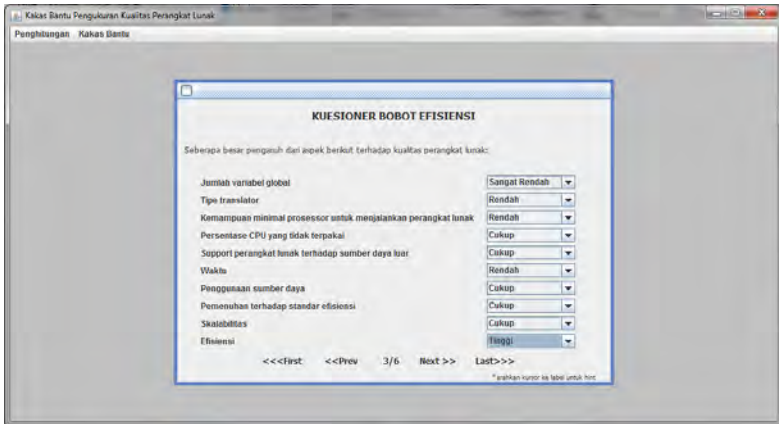
Fungsionalitas

Customizability

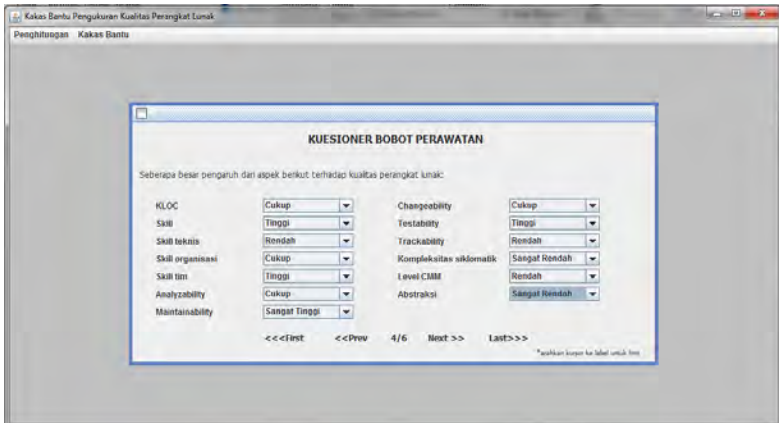
<<<First <<Prev 2/6 Next >> Last>>>

*Pilihkan Survey dan label untuk Item.

Gambar 5. 10 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (1)



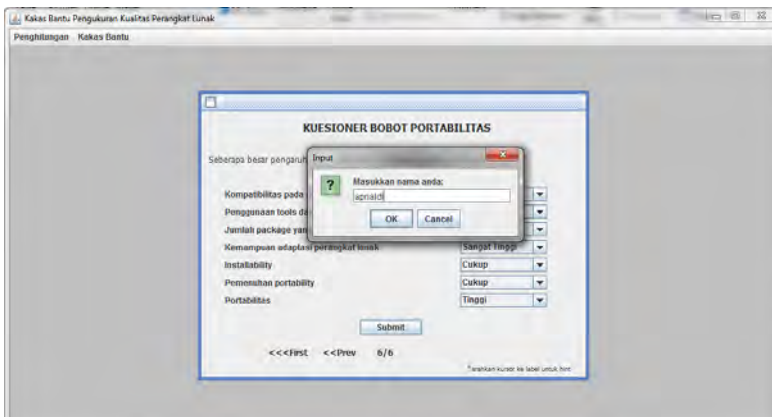
Gambar 5. 11 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (2)



Gambar 5. 12 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (3)

ID_PENILAIAN	ID_PERANGKATLUNAK	NAMA_PENILAI	TANGGAL_PENILAIAN	SKOR
41	31	Aprialdi	2015-05-21	0.6458333611488413

Gambar 5. 14 Penyimpanan Data Penilaian dalam Basis Data



Gambar 5. 13 Hasil Uji Fitur Mengisi Kuesioner Subjektif Perangkat Lunak (4)

ID_JAWABANMETRIK	ID_PENILAIAN	ID_FUZIFIKASIMETRIK
433	41	11
434	41	25
435	41	47
436	41	135
437	41	116
438	41	113
439	41	73
440	41	69
441	41	76
442	41	123

Gambar 5. 15 Penyimpanan Data Jawaban Rating Subjektif pada Basis Data

ID_JAWABANBOBOT	ID_PENILAIAN	ID_METRIK	ID_FUZIFIKASIBOBOT
2268	41	2	5
2269	41	3	4
2270	41	5	1
2271	41	6	1
2272	41	40	4
2273	41	41	5
2274	41	42	2
2275	41	43	1
2276	41	44	4
2277	41	58	4
2278	41	62	3
2279	41	9	4
2280	41	10	2
2281	41	11	2
2282	41	12	3
2283	41	13	3
2284	41	45	4
2285	41	46	2

Gambar 5. 16 Penyimpanan Data Jawaban Kuesioner Bobot pada Basis Data

5.2.1.5 Pengujian Fitur Melihat Detail Penilaian Perangkat Lunak

Pengujian fitur melihat detail penilaian perangkat lunak merupakan pengujian terhadap kemampuan kakas bantu untuk menampilkan detail penilaian yang telah dilakukan berdasarkan perangkat lunak yang dipilih. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7.

Tabel 5. 7 Pengujian Fitur Melihat Detail Penilaian Perangkat Lunak

ID	UJ.UC-0006
Referensi Kasus Penggunaan	UC-0005
Nama	Pengujian fitur melihat detail penilaian perangkat lunak

Tujuan Pengujian	Menguji fitur untuk melihat detail penilaian dari perangkat lunak yang dipilih.
Skenario	Menampilkan detail penilaian perangkat lunak dari daftar yang ada
Kondisi Awal	Kakas bantu dalam keadaan aktif. Perangkat lunak yang akan dilihat detail penilaiannya telah dipilih dari halaman daftar perangkat lunak
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu perangkat lunak dari daftar yang disediakan. 2. Pengguna menekan tombol lihat detail pada halaman
Hasil Yang Diharapkan	Aplikasi menampilkan seluruh data penilaian yang telah dilakukan untuk perangkat lunak yang dipilih
Hasil Yang Didapat	Semua data penilaian suatu perangkat lunak dapat ditampilkan oleh sistem
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan kode program dapat dilihat pada Gambar 5.17.



Gambar 5. 17 Hasil Uji Fitur Menampilkan Detail Penilaian Perangkat Lunak

5.2.1.6 Pengujian Fitur Menampilkan Laporan Penilaian Perangkat Lunak

Pengujian fitur menampilkan laporan penilaian perangkat lunak merupakan pengujian terhadap kemampuan kakas bantu untuk menampilkan laporan berdasarkan penilaian yang dipilih. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8.

Tabel 5. 8 Pengujian Fitur Menampilkan Laporan Penilaian Perangkat Lunak

ID	UJ.UC-0007
Referensi Kasus Penggunaan	UC-0006
Nama	Pengujian fitur menampilkan laporan penilaian perangkat lunak
Tujuan Pengujian	Menguji fitur untuk menampilkan laporan penilaian perangkat lunak untuk penilaian yang terpilih

Skenario	Laporan penilaian ditampilkan ketika user memilih salah satu penilaian.
Kondisi Awal	Kakas bantu dalam keadaan aktif. Perangkat lunak dan penilaian yang akan dilihat telah dipilih oleh pengguna dari halaman sebelumnya
Data Uji	Data yang digunakan adalah data perangkat lunak GA-MLP
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu penilaian pada daftar penilaian perangkat lunak 2. Pengguna menekan tombol lihat laporan
Hasil Yang Diharapkan	Aplikasi dapat menampilkan laporan penilaian perangkat lunak secara benar
Hasil Yang Didapat	Laporan untuk hasil penilaian perangkat lunak berhasil ditampilkan
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan kode program dapat dilihat pada Gambar 5.18.

Hasil Pengukuran Kualitas Perangkat Lunak	
Nama Perangkat Lunak	MLP-GA
Penilai	Aprialdi
Tanggal Penilaian	5/21/15 12:00 AM
Hasil Penghitungan Subkarakteristik	
NAMA SUBKARAKTERISTIK	SKOR
Persentase operasi yang sesuai	0.867
Persentase operasi yang memenuhi	0.5
Pemenuhan terhadap presisi yang	0.7
Basis data yang digunakan	0.3
Persentase pengontrolan akses yang	0.133
Tingkat keketatan akses	0.5
Pemenuhan terhadap standar	0.867
Tingkat kustomabilitas	0.3
Jumlah variabel global	0.867
Tipe translator	0.7
Kemampuan prosesor	0.867
Persentase CPU yang tidak terpakai	0.867
Support perangkat lunak terhadap sumber	0.3

Gambar 5. 18 Hasil Uji Fitur Laporan Penilaian Perangkat Lunak

5.2.2 Pengujian Validitas

Pengujian validitas sistem dilakukan dengan menyiapkan sejumlah kasus uji coba dan melakukan proses penghitungan nilai kualitas dengan penghitungan manual dan penggunaan kaskas bantu. Hasil yang diperoleh dengan kedua cara tersebut nantinya akan dibandingkan untuk mendapatkan nilai kebenaran kaskas bantu dalam penghitungan kualitas perangkat lunak.

Perangkat lunak yang digunakan dalam pengujian validitas ini adalah perangkat lunak MLP-GA, CerdasCermat, dan Pesta yang ketiganya merupakan *final project* dari mata kuliah terkait. Data uji coba yang digunakan dan hasil penghitungan secara manual untuk masing-masing katakteristik, subkarakteristik, dan aspek untuk menguji

validitas kakas bantu dapat dilihat pada Lmpiran A, Lampiran B, dan Lampiran C. Rumus penghitungan yang dilakukan untuk masing-masing aspek juga mengacu pada persamaan 3.1 Jawaban disesuaikan dengan pilihan yang disediakan oleh kakas bantu pada aplikasi. Dan nantinya akan dilakukan proses defuzifikasi dengan formula yang sama, yaitu rumus centroid. Tabel 5.9 menjelaskan hasil penghitungan kualitas tiga perangkat lunak menggunakan kakas bantu.

Selain itu, dalam pengujian validitas ini dilakukan juga proses ekstraksi dari kode sumber secara manual dan menggunakan kakas bantu. Hasil dari kedua metode ini akan dibandingkan dan dilihat validitasnya. Data yang digunakan dalam uji coba ini adalah tiga kode sumber dari tiga perangkat lunak yang diujicobakan dikarenakan banyaknya jumlah *file* kode sumber dari masing-masing perangkat lunak tersebut. Kode sumber perangkat lunak dapat dilihat pada Lampiran G. Tabel 5.10 menjelaskan hasil ekstraksi kode sumber menggunakan kakas bantu.

Fitur yang diekstraksi dari kode sumber adalah jumlah variabel global, panjang KLOC, dan kompleksitas siklomatik. Jumlah variabel global bisa didapatkan secara manual dengan melihat jumlah variabel dengan deklarasi tipe statis. Sementara itu, panjang KLOC didapatkan dengan menghitung jumlah baris *statement* dalam suatu *file* kode sumber dengan menghilangkan baris komentar dan baris kosong. Kompleksitas siklomatik didapatkan secara manual dengan menghitung jumlah *statement* dalam kode sumber seperti yang telah dijelaskan pada subbab 2.6.

Tabel 5. 9 Hasil Penghitungan Kualitas Perangkat Lunak Menggunakan Kakas Bantu

No	Nama Perangkat Lunak	Hasil Penghitungan Kakas Bantu
1	MLP-GA	0,646
2	Cerdas Cermat	0,56
3	Pesta	0,765

Tabel 5. 10 Hasil Ekstraksi Kode Sumber Menggunakan Kakas Bantu

No	Perangkat Lunak	Kode Sumber	KLOC	Kompleksitas Siklomatik	Variabel Statis
1	MLP-GA	Reduksi.java	0,076	10	1
2	Cerdas Cermat	Client.java	0,101	16	0
3	Pesta	Kelasreport.java	0,027	1	0

5.2.3 Pengujian Kebergunaan

Pengujian kebergunaan sistem dilakukan dengan tujuan untuk mengetahui kenyamanan dan kemudahan pengguna dalam menggunakan kakas bantu yang telah dibangun. Uji coba dilakukan oleh 10 orang dari kalangan IT, sehingga penguji sudah terbiasa dengan istilah-istilah yang digunakan dalam kakas bantu. Pengujian kebergunaan dilakukan dengan cara mengajukan sejumlah pertanyaan terkait kebergunaan kakas bantu setelah penguji menggunakan kakas bantu ini sesuai dengan skenario tertulis dalam uji coba ini. Lembar uji coba dan daftar pertanyaan yang diajukan dapat dilihat pada Lampiran D.

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi hasil deteksi.

5.3.1 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.11. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik.

Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 5. 11 Rangkuman Hasil Pengujian Fungsionalitas

ID	Nama	Skenario	Hasil
UJ.UC-0001	Pengujian fitur mengunggah perangkat lunak	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0002	Pengujian fitur mengisi kuesioner rating perangkat lunak	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0003	Pengujian fitur melihat daftar perangkat lunak		Berhasil
UJ.UC-0004	Pengujian fitur mengisi kuesioner subjektif		Berhasil
UJ.UC-0005	Pengujian fitur melihat detail penilaian perangkat lunak		Berhasil
UJ.UC-0006	Pengujian fitur menampilkan laporan penilaian perangkat lunak		Berhasil

5.3.2 Evaluasi Pengujian Validitas

Rangkuman mengenai hasil pengujian validitas yang telah dilakukan dapat dilihat pada Tabel 5.12 dan Tabel 5.13. Dari tabel ini dapat dilihat bahwa kebenaran nilai kakas bantu dalam penghitungan kualitas perangkat lunak terpenuhi. Tabel 5.12 jika direpresentasikan dalam bentuk grafik, akan tampak seperti pada Gambar 5.19.

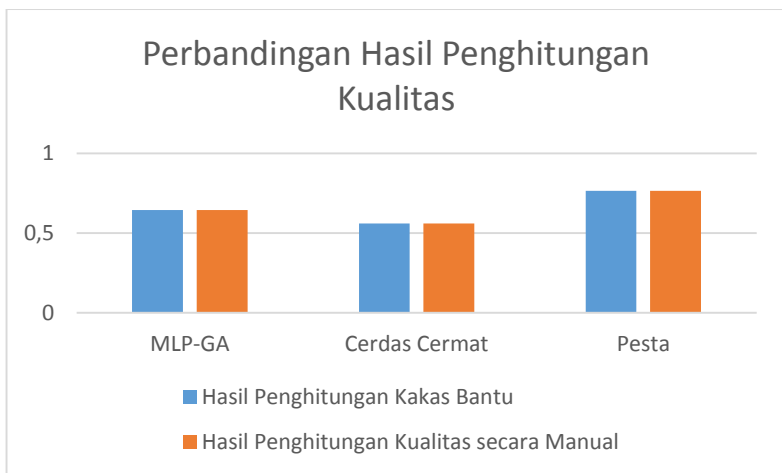
Tabel 5. 12 Hasil Pengujian Validitas

No	Nama Perangkat Lunak	Hasil Penghitungan Kakas Bantu	Hasil Penghitungan Manual
1	MLP-GA	0,646	0,646

2	Cerdas Cermat	0,56	0,560
3	Pesta	0,765	0,765

Tabel 5. 13 Hasil Ekstraksi Kode Sumber Secara Manual

No	Perangkat Lunak	Kode Sumber	KLOC	Kompleksitas Siklomatik	Variabel Statis
1	MLP-GA	Reduksi.java	0,076	10	1
2	Cerdas Cermat	Client.java	0,101	16	0
3	Pesta	Kelasreport.java	0,027	1	0



Gambar 5. 19 Grafik Perbandingan Hasil Penghitungan Kualitas Perangkat Lunak Uji Coba

5.3.3 Evaluasi Pengujian Kebergunaan

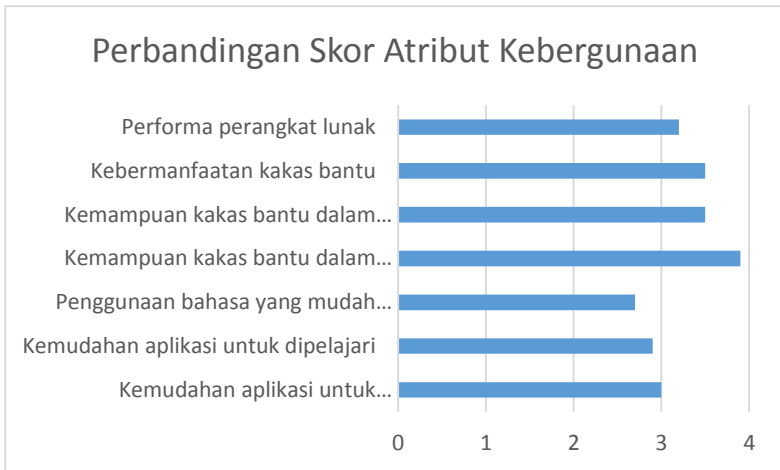
Jawaban dan proses penghitungan nilai masing-masing pertanyaan dapat dilihat pada Lampiran E. Rangkuman mengenai hasil

pengujian kebergunaan yang telah dilakukan dapat dilihat pada Tabel 5.14. Nilai yang dihasilkan dari pengujian kebergunaan ini berkisar antara 1-4, dengan nilai maksimalnya 4 dan minimalnya adalah 1.

Tabel 5. 14 Rangkuman Hasil Pengujian Kebergunaan

No	Atribut	Nilai (1-4)
1	Kemudahan aplikasi untuk digunakan	3
2	Kemudahan aplikasi untuk dipelajari	2,9
3	Penggunaan bahasa yang mudah dipahami	2,7
4	Kemampuan kakas bantu dalam membantu menilai kualitas perangkat lunak	3,9
5	Kemampuan kakas bantu dalam membantu mengetahui karakteristik apakah yang perlu ditingkatkan untuk suatu perangkat lunak	3,5
6	Kebermanfaatan kakas bantu	3,5
7	Performa perangkat lunak	3,2
Rata-rata skor		3,242

Berdasarkan Tabel 5.14, dapat diketahui bahwa nilai rata-rata skor adalah 3,242. Dengan menggunakan skala likert, maka nilai kebergunaan kakas bantu adalah 0,81 dengan kategori sangat baik (dengan menggunakan 4 kategori seperti pada kuesioner). Grafik perbandingan nilai-nilai aspek kebergunaan dapat dilihat pada Gambar 5.20 di bawah ini.



Gambar 5. 20 Grafik Perbandingan Skor Atribut Kebergunaan Kakas Bantu

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap pembuatan sistem “Rancang Bangun Kakas Bantu Pengukuran Kualitas Perangkat Lunak Menggunakan Metode Fuzzy”, diambil kesimpulan sebagai berikut:

1. Untuk mengekstrak fitur kode sumber, AST yang merepresentasikan struktur suatu kode sumber dan visitor pattern yang akan mengunjungi AST yang dibuat digunakan. Algoritma untuk mengekstraksi fitur berbeda-beda, tergantung pada fitur apa yang akan diambil. Dalam tugas akhir ini, fitur yang berhasil didapatkan menggunakan AST adalah jumlah variabel global dan jumlah kompleksitas siklomatik.
2. Proses penilaian kualitas perangkat lunak dengan menggunakan model kualitas ISO/IEC 9126 dan metode fuzzy diawali dengan proses membuat model kualitas yang tersusun dari beberapa aspek dalam lima jenis tingkatan, yaitu karakteristik, subkarakteristik, metrik, submetrik, dan subsubmetrik, yang masing-masing memiliki nilai bobot dan rating dalam bentuk *fuzzyset*. Aspek kualitas pada level tinggi akan menghitung nilai dibawahnya dan pada akhirnya akan dihasilkan nilai kualitas secara keseluruhan.
3. Kakas bantu yang dibangun dapat menilai kualitas perangkat lunak dengan menerapkan langkah-langkah penilaian kualitas yang telah disebutkan. Untuk membangun aplikasi ini dibutuhkan beberapa *library* tambahan, seperti JDT yang

digunakan untuk membuat AST, dan lain sebagainya. Seluruh kebutuhan fungsionalitas telah diimplementasikan dengan menggunakan bahasa pemrograman Java.

4. Hasil evaluasi kakas bantu menunjukkan bahwa semua fungsionalitas sistem yang telah dirancang dapat berjalan dengan baik. Sementara itu, hasil evaluasi validitas perangkat lunak menunjukkan bahwa nilai kualitas yang dihasilkan kakas bantu telah valid dan evaluasi kebergunaan sistem menunjukkan bahwa kebergunaan sistem dapat dikategorikan sangat baik.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem “Rancang Bangun Kakas Bantu Pengukuran Kualitas Perangkat Lunak Menggunakan Metode Fuzzy” pada tugas akhir ini antara lain:

1. Penilaian kualitas yang dilakukan sebaiknya tidak hanya menilai dari sudut pandang pengembang aplikasi saja, tetapi juga penilaian dari *stakeholder* lain, seperti pengguna, manajer proyek. Untuk pengembangan ke depannya, kakas bantu sebaiknya mampu mengakomodasi penilaian dari pihak-pihak selain tim pengembang.
2. Model kualitas yang digunakan pada tugas akhir ini merupakan model kualitas ISO/IEC 9126 yang telah digunakan sejak tahun 1991. Saat ini ISO telah mengeluarkan standar kualitas perangkat terbarunya, yaitu ISO/IEC 25010:2011 pada tahun 2011. Penelitian selanjutnya diharapkan bisa menggunakan standar terbaru ini.

DAFTAR PUSTAKA

- [1] Smartbear, "5 Reasons Why *Software* Quality Matters to your Business," SmartBear, 3rd April 2012. [Online]. Available: <http://blog.smartbear.com/sqc/5-reasons-why-Software-quality-matters-to-your-business/>. [Diakses 25 November 2014].
- [2] ARiSA, "*Software* Quality ISO Standards" ARiSA, 16 October 2008. [Online]. Available: <http://www.arisa.se/compendium/node6.html>. [Diakses 8 November 2014].
- [3] Challa Jagat Sesh, Paul Arindam, Dada Yogesh, Nerella Venkatesh, Srivastava Praveen Ranjan, and Singh Ajit Pratap "Integrated *Software* Quality Evaluation: A Fuzzy Multi-Criteria Approach" *Journal of Information Processing Systems*, vol. 7, no.3, September 2011 pp. 473-518, 2011.
- [4] Eclipse, "Abstract Syntax Tree," Eclipse, 20 November 2006. [Online]. Available: http://eclipse.org/articles/Article-JavaCodeManipulation_AST/index.html. [Diakses 26 April 2015].
- [5] CuningHam & CuningHam Inc., "Abstract Syntax Tree," CuningHam & CuningHam Inc, 4 January 2014. [Online]. Available: <http://c2.com/cgi/wiki?AbstractSyntaxTree>. [Diakses 26 April 2015].
- [6] Tutorialspoint, "Visitor Pattern," Tutorialspoint, 24 November 2014. [Online]. Available: http://www.tutorialspoint.com/design_pattern/visitor_pattern.htm. [Diakses 7 December 2014].
- [7] C. D. Wang, W. A. Appel, L. J. Kom and S. C. Serra, "The Zephy abstract syntax description language," in

Proceedings of the Conference on Domain-Specific Language, Berkeley, 1997.

- [8] DZone, "*Design Patterns Uncovered: The Visitor Pattern*," DZone, 3 September 2010. [Online]. Available: <http://java.dzone.com/articles/design-patterns-visitor>. [Diakses pada 27th November 2014].
- [9] JavaNotes, "Java: Computing Cyclomatic Complexity" JavaNotes, 2007. [Online]. Available: http://www.lepoint.net/principles_and_practices/complexity/complexity-java-method.html [Diakses pada 27 April 2015]

DAFTAR LAMPIRAN

LAMPIRAN A. Penghitungan Kualitas Aplikasi Uji Coba 1 ...	171
LAMPIRAN B. Penghitungan Kualitas Aplikasi Uji Coba 2 ...	183
LAMPIRAN C. Penghitungan Kualitas Aplikasi Uji Coba 3 ...	195
LAMPIRAN D. Lembar Kuesioner Uji Coba Kebergunaan.....	207
LAMPIRAN E. Hasil Pengolahan Jawaban Kuesioner.....	209
LAMPIRAN F. Hasil Jawaban Kuesioner Kebergunaan	213
LAMPIRAN G. Kode Sumber Ekstraksi	223

LAMPIRAN A

Tabel A. 1 Hasil Ekstraksi Kode Sumber

Fitur	Jumlah
Jumlah variabel global	1
KLOC	0.457
Kompleksitas siklomatik	47

Tabel A. 2 Isian Data Keterangan Perangkat Lunak

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
Persentase operasi yang sesuai	Jumlah operasi yang tidak sesuai	0	1
	Jumlah operasi yang disediakan	3	
Persentase operasi yang akurasiya terpenuhi	Jumlah operasi yang akurasiya terpenuhi	2	0.666
	Jumlah operasi yang disediakan	3	

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
Persentase pengontrolan akses yang disediakan	Jumlah pengontrolan akses yang disediakan	0	0
	Jumlah pengontrolan akses yang dibutuhkan	0	
Tingkat fitur yang dapat dikustomisasi	Jumlah fitur yang dapat dikustomisasi	0	0
Persentase CPU yang tidak digunakan	Penggunaan CPU untuk menjalankan perangkat lunak	5	0.95
Jumlah KLOC per anggota tim	Jumlah anggota tim	1	0.457
Persentase <i>property</i> perangkat lunak yang dapat dikustomisasi	Jumlah <i>property</i> perangkat lunak yang dapat dikustomisasi	5	0.714
	Jumlah <i>property</i> perangkat lunak	7	

Tabel A. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Persentase operasi yang sesuai	1	-
Persentase operasi yang memenuhi akurasi yang dibutuhkan	0.666	VH
Pemenuhan presisi yang dibutuhkan	Terpenuhi	H
Basis data yang digunakan	Lainnya	-
Persentase pengontrolan akses yang disediakan	0%	VL
Tingkat keketatan akses	Tanpa pembatasan akses	VL
Pemenuhan terhadap standar fungsionalitas	Terpenuhi	-
Tingkat kustomabilitas	0	-

Tabel A. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Persentase operasi yang sesuai	(0.7, 0.9, 1.0)	-	Kesesuaian	(0.7, 0.9, 1.0)
Persentase operasi yang memenuhi akurasi yang dibutuhkan	(0.3, 0.5, 0.7)	(0.75, 1.0, 1.0)	Akurasi	(0.25, 0.525, 0.9)
Pemenuhan presisi yang dibutuhkan	(0.5, 0.7, 0.9)	(0.5, 0.75, 1.0)		
Basis data yang digunakan	(0.1, 0.3, 0.5)	-	Interoperability	(0.1, 0.3, 0.5)
Persentase pengontrolan akses yang disediakan	(0.0, 0.1, 0.3)	(0.0, 0.0, 0.25)	Keamanan	(0.0, 0.0, 0.125)

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Tingkat keketatan akses	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Pemenuhan terhadap standar fungsionalitas	(0.7, 0.9, 1.0)	-	Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)
Tingkat kustomabilitas	(0.1, 0.3, 0.5)	-	Kustomabilitas	(0.1, 0.3, 0.5)

Tabel A. 5 Data Kasus Uji Coba Karakteristik Efisiensi

Nama Metrik	Jawaban Rating	Jawaban Bobot
Jumlah variabel global	1	H
Tipe translator	Compiler	L
Kemampuan prosessor	Core2Duo	L
Persentase CPU yang tidak terpakai	0.95	M
Dukungan perangkat lunak terhadap sumber daya luar	Tidak tersedia	M
Pemenuhan terhadap standar efisiensi	Tidak terpenuhi	-
Dukungan untuk beberapa pengguna	Tidak tersedia	-

Tabel A. 6 Hasil Penghitungan Manual Karakteristik Efisiensi

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Jumlah variabel global	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	Waktu	(0.35, 0.675, 1.0)
Tipe translator	(0.5, 0.7, 0.9)	(0.0, 0.25, 0.5)		
Kemampuan prosessor	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Persentase CPU yang tidak terpakai	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Penggunaan sumber daya	(0.175, 0.45, 0.75)
Dukungan perangkat lunak terhadap sumber daya luar	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Pemenuhan terhadap standar efisiensi	(0.1, 0.3, 0.5)	-	Pemenuhan efisiensi	(0.1, 0.3, 0.5)
Dukungan untuk beberapa pengguna	(0.1, 0.3, 0.5)	-	Skalabilitas	(0.1, 0.3, 0.5)

Tabel A. 7 Data Kasus Uji Coba Submetrik Kemampuan Tim

Nama Metrik	Jawaban Rating	Jawaban Bobot
<i>Citizenship</i> anggota tim pengembang	Baik	H
Kerjasama antara tim pengembang	Baik	H
Performa keseluruhan tim pengembang	Sangat baik	VH

Tabel A. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim

Nama Subsubmetrik	Rating	Bobot	Nama Submetrik	Nilai Rating
<i>Citizenship</i> anggota tim pengembang	(0.3, 0.5, 0.7)	(0.5, 0.75, 1.0)	Kemampuan tim	(0.525, 0.9, 1.0)
Kerjasama antara tim pengembang	(0.5, 0.7, 0.9)	(0.5, 0.75, 1.0)		
Performa keseluruhan tim pengembang	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)		

Tabel A. 9 Data Kasus Uji Coba Metrik Kemampuan

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kemampuan teknis	3	VH
Kemampuan organisasi	4-7 tahun	M
Kemampuan tim	(0.525, 0.9, 1.0)	VH

Tabel A. 10 Hasil Penghitungan Manual Metrik Kemampuan

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Kemampuan teknis	(0.3, 0.5, 0.7)	(0.75, 1.0, 1.0)	Kemampuan	(0.394, 0.9, 1.0)
Kemampuan organisasi	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Kemampuan tim	(0.525, 0.9, 1.0)	(0.75, 1.0, 1.0)		

Tabel A. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer

Nama Submetrik	Jawaban Rating	Jawaban Bobot
Pengalaman manajer dalam perusahaan perangkat lunak	2-4 tahun	H
Pengalaman manajer sebagai manajer	<2 tahun	VH

Tabel A. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Pengalaman manajer dalam perusahaan perangkat lunak	(0.3, 0.5, 0.7)	(0.5, 0.75, 1.0)	Pengalaman manajer	(0.15, 0.375, 0.7)

Pengalaman manajer sebagai manajer	(0.1, 0.3, 0.5)	(0.75, 1.0, 1.0)		
------------------------------------	-----------------	------------------	--	--

Tabel A. 13 Data Kasus Uji Coba Karakteristik *Maintainability*

Nama Metrik	Jawaban Rating	Jawaban Bobot
KLOC	0.457	M
Abstraksi	Java	L
Kemampuan	(0.394, 0.9, 1.0)	H
Pengalaman manajer	(0.15, 0.375, 0.7)	VL
Panjang KLOC per anggota tim	0.457	M
Ketersediaan dokumentasi	Tidak tersedia	VH
Kompleksitas siklomatik	47	H
Jumlah versi yang dirilis	1	M
Level CMM	1	VL
Persentase jumlah properti yang dapat dikustomisasi	0.714	-
Ketersediaan jumlah kasus uji coba yang cukup	Tersedia	-
Pemenuhan terhadap standar <i>maintainability</i>	Tidak terpenuhi	-
Ketersediaan penelusuran fungsional dan perilaku sistem	Tidak tersedia	H
Kemudahan dalam penelusuran versi lama perangkat lunak	Sangat buruk	M

Tabel A. 14 Hasil Penghitungan Manual Karakteristik *Maintainability*

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
KLOC	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Analyzeability	(0.197, 0.675, 1.0)
Abstraksi	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Kemampuan	(0.394, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Pengalaman manajer	(0.15, 0.375, 0.7)	(0.0, 0.0, 0.25)		
Panjang KLOC per anggota tim	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Ketersediaan dokumentasi	(0.1, 0.3, 0.5)	(0.75, 1.0, 1.0)		
Kompleksitas siklomatik	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)		
Jumlah versi yang dirilis	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Level CMM	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Persentase jumlah properti yang dapat dikustomisasi	(0.3, 0.5, 0.7)	-	<i>Changeability</i>	(0.3, 0.5, 0.7)
Ketersediaan jumlah kasus uji coba yang cukup	(0.7, 0.9, 1.0)	-	<i>Testability</i>	(0.7, 0.9, 1.0)
Pemenuhan terhadap standar <i>maintainability</i>	(0.1, 0.3, 0.5)	-	Pemenuhan <i>maintainability</i>	(0.1, 0.3, 0.5)
Ketersediaan penelusuran fungsional dan perilaku sistem	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)	Trackability	(0.05, 0.225, 0.5)
Kemudahan dalam penelusuran versi lama perangkat lunak	(0.0, 0.1, 0.3)	(0.25, 0.5, 0.75)		

Tabel A. 15 Data Kasus Uji Coba Karakteristik Portabilitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kompatibilitas terhadap sistem operasi lain	Hanya Windows	M
Penggunaan tools dasar	Tidak menggunakan	VL
Jumlah required <i>package</i>	0	M
Jumlah required packahe-nonOS	0	-
Pemenuhan terhadap standar <i>portability</i>	Terpenuhi	-

Tabel A. 16 Hasil Penghitungan Manual Karakteristik Portabilitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Jumlah variabel global	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)	Kemampuan adaptasi	(0.175, 0.45, 0.75)
Tipe translator	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Kemampuan prosessor	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Persentase CPU yang tidak terpakai	(0.7, 0.9, 1.0)	-	Installability	(0.7, 0.9, 1.0)
Dukungan perangkat lunak terhadap sumber daya luar	(0.7, 0.9, 1.0)	-	Pemenuhan <i>portability</i>	(0.7, 0.9, 1.0)

Tabel A. 17 Data Kasus Uji Coba Karakteristik

Nama Subkarakteristik	Jawaban Bobot
Kesesuaian	H
Akurasi	VH

Nama Subkarakteristik	Jawaban Bobot
Interoperability	L
Keamanan	VL
Pemenuhan fungsionalitas	H
Kustomabilitas	M
Waktu	H
Penggunaan sumber daya	M
Pemenuhan efisiensi	VH
Skalabilitas	L
Analyzability	H
<i>Changeability</i>	VL
<i>Testability</i>	VH
Pemenuhan <i>maintainability</i>	M
Trackability	L
Kemampuan adaptasi	L
Installability	VL
Pemenuhan <i>portability</i>	L

Tabel A. 18 Hasil Penghitungan Manual Karakteristik

Nama Subkarakteristik	Rating	Bobot	Nama Karakteristik	Nilai Rating
Kesesuaian	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	Fungsionalitas	(0.35, 0.675, 1.0)
Akurasi	(0.25, 0.525, 0.9)	(0.75, 1.0, 1.0)		

Nama Subkarakteristik	Rating	Bobot	Nama Karakteristik	Nilai Rating
Interoperability	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Keamanan	(0.0, 0.0, 0.25)	(0.0, 0.0, 0.25)		
Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Kustomabilitas	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Waktu	(0.35, 0.675, 1.0)	(0.5, 0.75, 1.0)		
Penggunaan sumber daya	(0.175, 0.45, 0.75)	(0.25, 0.5, 0.75)	Efisiensi	(0.175, 0.506, 1.0)
Pemenuhan efisiensi	(0.1, 0.3, 0.5)	(0.75, 1.0, 1.0)	Keterawatan	(0.525, 0.9, 1.0)
Skalabilitas	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Analyzability	(0.197, 0.675, 1.0)	(0.5, 0.75, 1.0)		
<i>Changeability</i>	(0.3, 0.5, 0.7)	(0.0, 0.0, 0.25)		
<i>Testability</i>	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)		
Pemenuhan <i>maintainability</i>	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Trackability	(0.05, 0.225, 0.5)	(0.0, 0.25, 0.5)		
Kemampuan adaptasi	(0.175, 0.45, 0.75)	(0.0, 0.25, 0.5)		
Installability	(0.7, 0.9, 1.0)	(0.0, 0.0, 0.25)		
Pemenuhan <i>portability</i>	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		

Tabel A. 19 Data Kasus Uji Coba Nilai Kualitas

Nama Subkarakteristik	Jawaban Bobot
Fungsionalitas	H
Efisiensi	M
Keterawatan	H
<i>Portability</i>	VL

Tabel A. 20 Hasil Penghitungan Manual Nilai Kualitas

Nama Subkarakteristik	Rating	Bobot	Nilai Kualitas
Kesesuaian	(0.35, 0.675, 1.0)	(0.5, 0.75, 1.0)	(0.2625, 0.675, 1.0)
Akurasi	(0.175, 0.506, 1.0)	(0.25, 0.5, 0.75)	
Interoperability	(0.525, 0.9, 1.0)	(0.5, 0.75, 1.0)	
Keamanan	(0.0, 0.225, 0.5)	(0.0, 0.0, 0.25)	

Tabel A. 21 Hasil Penghitungan Kualitas Perangkat Lunak

Nilai Kualitas	(0.2625, 0.675, 1.0)
CrispValue	0.6458

LAMPIRAN B

Tabel B. 1 Hasil Ekstraksi Kode Sumber

Fitur	Jumlah
Jumlah variabel global	5
KLOC	0.639
Kompleksitas siklomatik	54

Tabel B. 2 Isian Data Keterangan Perangkat Lunak

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
Persentase operasi yang sesuai	Jumlah operasi yang tidak sesuai	0	1
	Jumlah operasi yang disediakan	3	
Persentase operasi yang akurasiya terpenuhi	Jumlah operasi yang akurasiya terpenuhi	2	0.666
	Jumlah operasi yang disediakan	3	

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
Persentase pengontrolan akses yang disediakan	Jumlah pengontrolan akses yang disediakan	1	1
	Jumlah pengontrolan akses yang dibutuhkan	1	
Tingkat fitur yang dapat dikustomisasi	Jumlah fitur yang dapat dikustomisasi	0	0
Persentase CPU yang tidak digunakan	Penggunaan CPU untuk menjalankan perangkat lunak	4	0.96
Jumlah KLOC per anggota tim	Jumlah anggota tim	4	0.160
Persentase <i>property</i> perangkat lunak yang dapat dikustomisasi	Jumlah <i>property</i> perangkat lunak yang dapat dikustomisasi	2	0.333
	Jumlah <i>property</i> perangkat lunak	6	

Tabel B. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Persentase operasi yang sesuai	100%	-
Persentase operasi yang memenuhi akurasi yang dibutuhkan	66,67%	H
Pemenuhan presisi yang dibutuhkan	Terpenuhi	M
Basis data yang digunakan	Lainnya	-
Persentase pengontrolan akses yang disediakan	100%	H
Tingkat keketatan akses	Tanpa password	M
Pemenuhan terhadap standar fungsionalitas	Terpenuhi	-
Tingkat kustomabilitas	0	-

Tabel B. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Persentase operasi yang sesuai	(0.7, 0.9, 1.0)	-	Kesesuaian	(0.7, 0.9, 1.0)
Persentase operasi yang memenuhi akurasi yang dibutuhkan	(0.3, 0.5, 0.7)	(0.5, 0.75, 1.0)	Akurasi	(0.15, 0.375, 0.7)
Pemenuhan presisi yang dibutuhkan	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Basis data yang digunakan	(0.1, 0.3, 0.5)	-	Interoperability	(0.1, 0.3, 0.5)
Persentase pengontrolan akses yang disediakan	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	Keamanan	(0.35, 0.675, 1.0)
Tingkat keketatan akses	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Pemenuhan terhadap standar fungsionalitas	(0.7, 0.9, 1.0)	-	Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)
Tingkat kustomabilitas	(0.1, 0.3, 0.5)	-	Kustomabilitas	(0.1, 0.3, 0.5)

Tabel B. 5 Data Kasus Uji Coba Karakteristik Efisiensi

Nama Metrik	Jawaban Rating	Jawaban Bobot
Jumlah variabel global	5	M
Tipe translator	Compiler	L
Kemampuan prosessor	Core2Duo	VL
Persentase CPU yang tidak terpakai	0.96	L
Dukungan perangkat lunak terhadap sumber daya luar	Tidak tersedia	VL
Pemenuhan terhadap standar efisiensi	Tidak terpenuhi	-
Dukungan untuk beberapa pengguna	Tidak tersedia	-

Tabel B. 6 Hasil Penghitungan Manual Karakteristik Efisiensi

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Jumlah variabel global	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Waktu	(0.175, 0.45, 0.75)
Tipe translator	(0.5, 0.7, 0.9)	(0.0, 0.25, 0.5)		
Kemampuan prosessor	(0.7, 0.9, 1.0)	(0.0, 0.0, 0.25)		
Persentase CPU yang tidak terpakai	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)	Penggunaan sumber daya	(0.0, 0.225, 0.5)

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Dukungan perangkat lunak terhadap sumber daya luar	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Pemenuhan terhadap standar efisiensi	(0.1, 0.3, 0.5)	-	Pemenuhan efisiensi	(0.1, 0.3, 0.5)
Dukungan untuk beberapa pengguna	(0.1, 0.3, 0.5)	-	Skalabilitas	(0.1, 0.3, 0.5)

Tabel B. 7 Data Kasus Uji Coba Submetrik Kemampuan tim

Nama Metrik	Jawaban Rating	Jawaban Bobot
<i>Citizenship</i> anggota tim pengembang	Cukup	M
Kerjasama antara tim pengembang	Kurang	H
Performa keseluruhan tim pengembang	Kurang	VH

Tabel B. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim

Nama Subsubmetrik	Rating	Bobot	Nama Submetrik	Nilai Rating
<i>Citizenship</i> anggota tim pengembang	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)	Kemampuan tim	(0.075, 0.3, 0.525)
Kerjasama antara tim pengembang	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)		
Performa keseluruhan tim pengembang	(0.1, 0.3, 0.5)	(0.75, 1.0, 1.0)		

Tabel B. 9 Data Kasus Uji Coba Metrik Kemampuan

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kemampuan teknis	5	M
Kemampuan organisasi	2-4 tahun	M
Kemampuan tim	(0.075, 0.3, 0.525)	VH

Tabel B. 10 Hasil Penghitungan Manual Metrik Kemampuan

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Kemampuan teknis	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Kemampuan	(0.175, 0.45, 0.75)
Kemampuan organisasi	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)		
Kemampuan tim	(0.075, 0.3, 0.525)	(0.75, 1.0, 1.0)		

Tabel B. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer

Nama Submetrik	Jawaban Rating	Jawaban Bobot
Pengalaman manajer dalam perusahaan perangkat lunak	2-4 tahun	M
Pengalaman manajer sebagai manajer	<2 tahun	M

Tabel B. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Pengalaman manajer dalam perusahaan perangkat lunak	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)	Pengalaman manajer	(0.075, 0.25, 0.525)

Pengalaman manajer sebagai manajer	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
------------------------------------	-----------------	-------------------	--	--

Tabel B. 13 Data Kasus Uji Coba Karakteristik *Maintainability*

Nama Metrik	Jawaban Rating	Jawaban Bobot
KLOC	0.639	M
Abstraksi	Java	L
Kemampuan	(0.175, 0.45, 0.75)	VH
Pengalaman manajer	(0.075, 0.25, 0.525)	VL
Panjang KLOC per anggota tim	0.160	H
Ketersediaan dokumentasi	Tersedia	M
Kompleksitas siklomatik	54	M
Jumlah versi yang dirilis	1	M
Level CMM	1	VL
Persentase jumlah properti yang dapat dikustomisasi	0.333	-
Ketersediaan jumlah kasus uji coba yang cukup	Tersedia	-
Pemenuhan terhadap standar <i>maintainability</i>	Tidak terpenuhi	-
Ketersediaan penelusuran fungsional dan perilaku sistem	Tersedia	M
Kemudahan dalam penelusuran versi lama perangkat lunak	Sulit tapi nyaman digunakan	M

Tabel B. 14 Hasil Penghitungan Manual Karakteristik *Maintainability*

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
KLOC	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Analyzability	(0.35, 0.675, 1.0)
Abstraksi	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Kemampuan	(0.175, 0.45, 0.75)	(0.75, 1.0, 1.0)		
Pengalaman manajer	(0.075, 0.25, 0.525)	(0.0, 0.0, 0.25)		
Panjang KLOC per anggota tim	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Ketersediaan dokumentasi	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Kompleksitas siklomatik	(0.0, 0.1, 0.3)	(0.25, 0.5, 0.75)		
Jumlah versi yang dirilis	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Level CMM	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Persentase jumlah properti yang dapat dikustomisasi	(0.1, 0.3, 0.5)	-		
Ketersediaan jumlah kasus uji coba yang cukup	(0.7, 0.9, 1.0)	-	<i>Testability</i>	(0.7, 0.9, 1.0)
Pemenuhan terhadap standar <i>maintainability</i>	(0.1, 0.3, 0.5)	-	Pemenuhan <i>maintainability</i>	(0.1, 0.3, 0.5)
Ketersediaan penelusuran fungsional dan perilaku sistem	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Trackability	(0.175, 0.45, 0.75)
Kemudahan dalam penelusuran versi lama perangkat lunak	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)		

Tabel B. 15 Data Kasus Uji Coba Karakteristik Portabilitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kompatibilitas terhadap sistem operasi lain	Hanya Windows	VH
Penggunaan tools dasar	Tidak menggunakan	M
Jumlah required <i>package</i>	0	M
Jumlah required packahe-nonOS	0	-
Pemenuhan terhadap standar <i>portability</i>	Terpenuhi	-

Tabel B. 16 Hasil Penghitungan Manual Karakteristik Portabilitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Kompatibilitas terhadap sistem operasi lain	(0.3, 0.5, 0.7)	(0.75, 1.0, 1.0)	Kemampuan adaptasi	(0.225, 0.5, 0.75)
Penggunaan tools dasar	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Jumlah required <i>package</i>	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Jumlah required packahe-nonOS	(0.7, 0.9, 1.0)	-	Installability	(0.7, 0.9, 1.0)
Pemenuhan terhadap standar <i>portability</i>	(0.7, 0.9, 1.0)	-	Pemenuhan <i>portability</i>	(0.7, 0.9, 1.0)

Tabel B. 17 Data Kasus Uji Coba Karakteristik

Nama Subkarakteristik	Jawaban Bobot
Kesesuaian	H

Nama Subkarakteristik	Jawaban Bobot
Akurasi	L
Interoperability	L
Keamanan	VH
Pemenuhan fungsionalitas	H
Kustomabilitas	L
Waktu	M
Penggunaan sumber daya	L
Pemenuhan efisiensi	VL
Skalabilitas	M
Analyzability	M
<i>Changeability</i>	M
<i>Testability</i>	L
Pemenuhan <i>maintainability</i>	H
Trackability	M
Kemampuan adaptasi	VH
Installability	H
Pemenuhan <i>portability</i>	H

Tabel B. 18 Hasil Penghitungan Manual Karakteristik Portabilitas

Nama Subkarakteristik	Rating	Bobot	Nama Karakteristik	Nilai Rating
Kesesuaian	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	Fungsionalitas	(0.35, 0.675, 1.0)

Nama Subkarakteristik	Rating	Bobot	Nama Karakteristik	Nilai Rating
Akurasi	(0.15, 0.375, 0.7)	(0.0, 0.25, 0.5)		
Interoperability	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Keamanan	(0.35, 0.675, 1.0)	(0.75, 1.0, 1.0)		
Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Kustomabilitas	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Waktu	(0.175, 0.45, 0.75)	(0.25, 0.5, 0.75)	Efisiensi	(0.044, 0.225, 0.562)
Penggunaan sumber daya	(0.0, 0.225, 0.5)	(0.0, 0.25, 0.5)		
Pemenuhan efisiensi	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Skalabilitas	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Analyzability	(0.35, 0.675, 1.0)	(0.25, 0.5, 0.75)		
<i>Changeability</i>	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)	Keterawatan	(0.087, 0.337, 0.75)
<i>Testability</i>	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Pemenuhan <i>maintainability</i>	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)		
Trackability	(0.175, 0.45, 0.75)	(0.25, 0.5, 0.75)		
Kemampuan adaptasi	(0.225, 0.5, 0.75)	(0.75, 1.0, 1.0)		
Installability	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	<i>Portability</i>	(0.35, 0.675, 1.0)
Pemenuhan <i>portability</i>	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)		

Tabel B. 19 Data Kasus Uji Coba Nilai Kualitas

Nama Subkarakteristik	Jawaban Bobot
Fungsionalitas	H
Efisiensi	L
Keterawatan	M
<i>Portability</i>	H

Tabel B. 20 Hasil Penghitungan Manual Nilai Kualitas

Nama Subkarakteristik	Rating	Bobot	Nilai Kualitas
Fungsionalitas	(0.35, 0.675, 1.0)	(0.5, 0.75, 1.0)	(0.175, 0.506, 1.0)
Efisiensi	(0.044, 0.225, 0.562)	(0.0, 0.25, 0.5)	
Keterawatan	(0.087, 0.337, 0.75)	(0.25, 0.5, 0.75)	
<i>Portability</i>	(0.35, 0.675, 1.0)	(0.5, 0.75, 1.0)	

Tabel B. 21 Hasil Penghitungan Kualitas Perangkat Lunak

Nilai Kualitas	(0.175, 0.506, 1.0)
CrispValue	0.5603

LAMPIRAN C

Tabel C. 1 Hasil Ekstraksi Kode Sumber

Fitur	Jumlah
Jumlah variabel global	0
KLOC	0.47
Kompleksitas siklomatik	19

Tabel C. 2 Isian Data Keterangan Perangkat Lunak

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
Persentase operasi yang sesuai	Jumlah operasi yang tidak sesuai	0	1
	Jumlah operasi yang disediakan	5	
Persentase operasi yang akurasinya terpenuhi	Jumlah operasi yang akurasinya terpenuhi	5	1
	Jumlah operasi yang disediakan	5	
Persentase pengontrolan akses yang disediakan	Jumlah pengontrolan akses yang disediakan	0	0

Metrik	Nilai yang dimasukkan oleh pengguna	Nilai	Hasil Penghitungan
	Jumlah pengontrolan akses yang dibutuhkan	0	
Tingkat fitur yang dapat dikustomisasi	Jumlah fitur yang dapat dikustomisasi	0	0
Persentase CPU yang tidak digunakan	Penggunaan CPU untuk menjalankan perangkat lunak	8	0.92
Jumlah KLOC per anggota tim	Jumlah anggota tim	1	0.47
Persentase <i>property Software</i> yang dapat dikustomisasi	Jumlah <i>property</i> perangkat lunak yang dapat dikustomisasi	0	0
	Jumlah <i>property</i> perangkat lunak	2	

Tabel C. 3 Data Kasus Uji Coba Karakteristik Fungsionalitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Persentase operasi yang sesuai	1	-

Nama Metrik	Jawaban Rating	Jawaban Bobot
Persentase operasi yang memenuhi akurasi yang dibutuhkan	1	VH
Pemenuhan presisi yang dibutuhkan	Terpenuhi	VH
Basis data yang digunakan	MySQL	-
Persentase pengontrolan akses yang disediakan	0%	VL
Tingkat keketatan akses	Tanpa pembatasan akses	VL
Pemenuhan terhadap standar fungsionalitas	Terpenuhi	-
Tingkat kustomabilitas	0	-

Tabel C. 4 Hasil Penghitungan Manual Karakteristik Fungsionalitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Persentase operasi yang sesuai	(0.7, 0.9, 1.0)	-	Kesesuaian	(0.7, 0.9, 1.0)
Persentase operasi yang memenuhi akurasi yang dibutuhkan	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)	Akurasi	(0.525, 0.9, 1.0)
Pemenuhan presisi yang dibutuhkan	(0.5, 0.7, 0.9)	(0.5, 0.75, 1.0)		
Basis data yang digunakan	(0.5, 0.7, 0.9)	-	Interoperability	(0.5, 0.7, 0.9)
Persentase pengontrolan akses yang disediakan	(0.0, 0.1, 0.3)	(0.0, 0.0, 0.25)	Keamanan	(0.0, 0.0, 0.125)
Tingkat keketatan akses	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Pemenuhan terhadap standar fungsionalitas	(0.7, 0.9, 1.0)	-	Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)
Tingkat kustomabilitas	(0.1, 0.3, 0.5)	-	Kustomabilitas	(0.1, 0.3, 0.5)

Tabel C. 5 Data Kasus Uji Coba Karakteristik Efisiensi

Nama Metrik	Jawaban Rating	Jawaban Bobot
Jumlah variabel global	0	M
Tipe translator	Compiler	M
Kemampuan prosessor	Core2Duo	M
Persentase CPU yang tidak terpakai	0.92	H
Dukungan perangkat lunak terhadap sumber daya luar	Tidak tersedia	VH
Pemenuhan terhadap standar efisiensi	Tidak terpenuhi	-
Dukungan untuk beberapa pengguna	Tidak tersedia	-

Tabel C. 6 Hasil Penghitungan Manual Karakteristik Efisiensi

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Jumlah variabel global	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)	Waktu	(0.175, 0.45, 0.75)
Tipe translator	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Kemampuan prosessor	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Persentase CPU yang tidak terpakai	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)	Penggunaan sumber daya	(0.35, 0.675, 1.0)

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Dukungan perangkat lunak terhadap sumber daya luar	(0.1, 0.3, 0.5)	(0.75, 1.0, 1.0)		
Pemenuhan terhadap standar efisiensi	(0.1, 0.3, 0.5)	-	Pemenuhan efisiensi	(0.1, 0.3, 0.5)
Dukungan untuk beberapa pengguna	(0.1, 0.3, 0.5)	-	Skalabilitas	(0.1, 0.3, 0.5)

Tabel C. 7 Data Kasus Uji Coba Submetrik Kemampuan tim

Nama Metrik	Jawaban Rating	Jawaban Bobot
<i>Citizenship</i> anggota tim pengembang	Baik	VH
Kerjasama antara tim pengembang	Baik	M
Performa keseluruhan tim pengembang	Baik	VH

Tabel C. 8 Hasil Penghitungan Manual Submetrik Kemampuan Tim

Nama Subsubmetrik	Rating	Bobot	Nama Submetrik	Nilai Rating
<i>Citizenship</i> anggota tim pengembang	(0.5, 0.7, 0.9)	(0.75, 1.0, 1.0)	Kemampuan tim	(0.375, 0.7, 0.9)
Kerjasama antara tim pengembang	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Performa keseluruhan tim pengembang	(0.5, 0.7, 0.9)	(0.75, 1.0, 1.0)		

Tabel C. 9 Data Kasus Uji Coba Metrik Kemampuan

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kemampuan teknis	2	H
Kemampuan organisasi	< 2 tahun	L
Kemampuan tim	(0.375, 0.7, 0.9)	H

Tabel C. 10 Hasil Penghitungan Manual Metrik Kemampuan

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Kemampuan teknis	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)	Kemampuan	(0.188, 0.525, 0.9)
Kemampuan organisasi	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Kemampuan tim	(0.375, 0.7, 0.9)	(0.5, 0.75, 1.0)		

Tabel C. 11 Data Kasus Uji Coba Metrik Pengalaman Manajer

Nama Submetrik	Jawaban Rating	Jawaban Bobot
Pengalaman manajer dalam perusahaan perangkat lunak	<2 tahun	VL
Pengalaman manajer sebagai manajer	<2 tahun	VL

Tabel C. 12 Hasil Penghitungan Manual Metrik Pengalaman Manajer

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Pengalaman manajer dalam perusahaan perangkat lunak	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)	Pengalaman manajer	(0.0, 0.0, 0.125)

Nama Submetrik	Rating	Bobot	Nama Metrik	Nilai Rating
Pengalaman manajer sebagai manajer	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		

Tabel C. 13 Data Kasus Uji Coba Karakteristik *Maintainability*

Nama Metrik	Jawaban Rating	Jawaban Bobot
KLOC	0.47	VH
Abstraksi	Java	L
Kemampuan	(0.188, 0.525, 0.9)	H
Pengalaman manajer	(0.0, 0.0, 0.125)	VL
Panjang KLOC per anggota tim	0.47	M
Ketersediaan dokumentasi	Tersedia	H
Kompleksitas siklomatik	19	M
Jumlah versi yang dirilis	1	L
Level CMM	1	VL
Persentase jumlah properti yang dapat dikustomisasi	0	-
Ketersediaan jumlah kasus uji coba yang cukup	Tersedia	-
Pemenuhan terhadap standar <i>maintainability</i>	Terpenuhi	-
Ketersediaan penelusuran fungsional dan perilaku sistem	Tidak tersedia	H
Kemudahan dalam penelusuran versi lama perangkat lunak	Mudah dan nyaman	H

Tabel C. 14 Hasil Penghitungan Manual Karakteristik *Maintainability*

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
KLOC	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)	Analyzeability	(0.525, 0.9, 1.0)
Abstraksi	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Kemampuan	(0.188, 0.525, 0.9)	(0.5, 0.75, 1.0)		
Pengalaman manajer	(0.0, 0.0, 0.125)	(0.0, 0.0, 0.25)		
Panjang KLOC per anggota tim	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Ketersediaan dokumentasi	(0.7, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Kompleksitas siklomatik	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)		
Jumlah versi yang dirilis	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Level CMM	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.25)		
Persentase jumlah properti yang dapat dikustomisasi	(0.0, 0.1, 0.3)	-	<i>Changeability</i>	(0.0, 0.1, 0.3)
Ketersediaan jumlah kasus uji coba yang cukup	(0.7, 0.9, 1.0)	-	<i>Testability</i>	(0.7, 0.9, 1.0)
Pemenuhan terhadap standar <i>maintainability</i>	(0.7, 0.9, 1.0)	-	Pemenuhan <i>maintainability</i>	(0.7, 0.9, 1.0)
Ketersediaan penelusuran fungsional dan perilaku sistem	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)	Trackability	(0.25, 0.525, 0.9)
Kemudahan dalam penelusuran versi lama perangkat lunak	(0.5, 0.7, 0.9)	(0.5, 0.75, 1.0)		

Tabel C. 15 Data Kasus Uji Coba Karakteristik Portabilitas

Nama Metrik	Jawaban Rating	Jawaban Bobot
Kompatibilitas terhadap sistem operasi lain	Hanya Windows	M
Penggunaan tools dasar	Tidak menggunakan	VL
Jumlah required <i>package</i>	1 <i>package</i> mudah didapatkan	M
Jumlah required packahe-nonOS	1 <i>package</i> mudah didapatkan	-
Pemenuhan terhadap standar <i>portability</i>	Tidak terpenuhi	-

Tabel C. 16 Hasil Penghitungan Manual Karakteristik Portabilitas

Nama Metrik	Rating	Bobot	Nama Subkarakteristik	Nilai Rating
Kompatibilitas terhadap sistem operasi lain	(0.3, 0.5, 0.7)	(0.25, 0.5, 0.75)	Kemampuan adaptasi	(0.125, 0.35, 0.675)
Penggunaan tools dasar	(0.7, 0.9, 1.0)	(0.0, 0.25, 0.5)		
Jumlah required <i>package</i>	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Jumlah required packahe-nonOS	(0.5, 0.7, 0.9)	-	Installability	(0.5, 0.7, 0.9)
Pemenuhan terhadap standar <i>portability</i>	(0.1, 0.3, 0.5)	-	Pemenuhan <i>portability</i>	(0.1, 0.3, 0.5)

Tabel C. 17 Data Kasus Uji Coba Karakteristik

Nama Subkarakteristik	Jawaban Bobot
Kesesuaian	VH
Akurasi	H
Interoperability	M
Keamanan	L
Pemenuhan fungsionalitas	VH
Kustomabilitas	M
Waktu	M
Penggunaan sumber daya	H
Pemenuhan efisiensi	L
Skalabilitas	H
Analyzability	VH
<i>Changeability</i>	M
<i>Testability</i>	VH
Pemenuhan <i>maintainability</i>	M
Trackability	VL
Kemampuan adaptasi	VL
Installability	H
Pemenuhan <i>portability</i>	M

Tabel C. 18 Hasil Penghitungan Manual Karakteristik Portabilitas

Nama Subkarakteristik	Rating	Bobot	Nama Karakteristik	Nilai Rating
Kesesuaian	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)	Fungsionalitas	(0.525, 0.9, 1.0)
Akurasi	(0.525, 0.9, 1.0)	(0.5, 0.75, 1.0)		
Interoperability	(0.5, 0.7, 0.9)	(0.25, 0.5, 0.75)		
Keamanan	(0.0, 0.0, 0.125)	(0.0, 0.25, 0.5)		
Pemenuhan fungsionalitas	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)		
Kustomabilitas	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		
Waktu	(0.175, 0.45, 0.75)	(0.25, 0.5, 0.75)	Efisiensi	(0.175, 0.506, 1.0)
Penggunaan sumber daya	(0.35, 0.675, 1.0)	(0.5, 0.75, 1.0)		
Pemenuhan efisiensi	(0.1, 0.3, 0.5)	(0.0, 0.25, 0.5)		
Skalabilitas	(0.1, 0.3, 0.5)	(0.5, 0.75, 1.0)		
Analyzability	(0.525, 0.9, 1.0)	(0.75, 1.0, 1.0)	Keterawatan	(0.525, 0.9, 1.0)
<i>Changeability</i>	(0.0, 0.1, 0.3)	(0.25, 0.5, 0.75)		
<i>Testability</i>	(0.7, 0.9, 1.0)	(0.75, 1.0, 1.0)		
Pemenuhan <i>maintainability</i>	(0.7, 0.9, 1.0)	(0.25, 0.5, 0.75)		
Trackability	(0.25, 0.525, 0.9)	(0.0, 0.0, 0.25)		
Kemampuan adaptasi	(0.125, 0.35, 0.675)	(0.0, 0.0, 0.25)	<i>Portability</i>	(0.25, 0.525, 0.9)
Installability	(0.5, 0.7, 0.9)	(0.5, 0.75, 1.0)		
Pemenuhan <i>portability</i>	(0.1, 0.3, 0.5)	(0.25, 0.5, 0.75)		

Tabel C. 19 Data Kasus Uji Coba Nilai Kualitas

Nama Subkarakteristik	Jawaban Bobot
Fungsionalitas	VH
Efisiensi	M
Keterawatan	H
<i>Portability</i>	M

Tabel C. 20 Hasil Penghitungan Manual Nilai Kualitas

Nama Subkarakteristik	Rating	Bobot	Nilai Kualitas
Fungsionalitas	(0.525, 0.9, 1.0)	(0.75, 1.0, 1.0)	(0.394, 0.9, 1.0)
Efisiensi	(0.175, 0.506, 1.0)	(0.25, 0.5, 0.75)	
Keterawatan	(0.525, 0.9, 1.0)	(0.5, 0.75, 1.0)	
<i>Portability</i>	(0.25, 0.525, 0.9)	(0.25, 0.5, 0.75)	

Tabel C. 21 Hasil Penghitungan Kualitas Perangkat Lunak

Nilai Kualitas	(0.394, 0.9, 1.0)
CrispValue	0.765

LAMPIRAN D

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji :

Tanggal Pengujian :

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?					
2	Apakah aplikasi yang dibangun mudah dipelajari?					
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?					
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?					
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?					
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?					
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?					

Gambar D. 1 Lembar Kuesioner Pengujian Kebergunaan Perangkat Lunak

(Halaman ini sengaja dikosongkan)

LAMPIRAN E

Tabel E. 1 Penghitungan Uji Kebergunaan Aspek Kemudahan Penggunaan Aplikasi

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	2	20%	8
2	Setuju	3	6	60%	18
3	Tidak Setuju	2	2	20%	4
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					30
Rata-rata					3

Tabel E. 2 Penghitungan Uji Kebergunaan Aspek Kemudahan Aplikasi Untuk Dipelajari

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	2	20%	8
2	Setuju	3	5	50%	15
3	Tidak Setuju	2	3	30%	6
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					29
Rata-rata					2.9

Tabel E. 3 Penghitungan Uji Kebergunaan Aspek Penggunaan Bahasa

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	1	10%	4
2	Setuju	3	6	60%	18
3	Tidak Setuju	2	2	20%	4
4	Sangat Tidak Setuju	1	1	10%	1
Jumlah					27
Rata-rata					2.7

Tabel E. 4 Penghitungan Uji Kebergunaan Aspek Kemampuan Kakas Bantu untuk Menentukan Kualitas Perangkat Lunak

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	9	90%	36
2	Setuju	3	1	10%	3
3	Tidak Setuju	2	0	0%	0
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					39
Rata-rata					3.9

Tabel E. 5 Penghitungan Uji Kebergunaan Aspek Kemampuan Kakas Bantu untuk Mengetahui Karakteristik yang berpengaruh terhadap Kualitas

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	5	50%	20
2	Setuju	3	5	50%	15
3	Tidak Setuju	2	0	0%	0
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					35
Rata-rata					3.5

Tabel E. 6 Penghitungan Uji Kebergunaan Aspek Kebermanfaatan Aplikasi

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	5	50%	20
2	Setuju	3	5	50%	15
3	Tidak Setuju	2	0	0%	0
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					3.5
Rata-rata					3.5

Tabel E. 7 Penghitungan Uji Kebergunaan Aspek Performa Aplikasi

No	Keterangan	Bobot	Responden	Persentase	Skor
1	Sangat Setuju	4	3	30%	12

No	Keterangan	Bobot	Responden	Persentase	Skor
2	Setuju	3	6	60%	18
3	Tidak Setuju	2	1	10%	2
4	Sangat Tidak Setuju	1	0	0%	0
Jumlah					32
Rata-rata					3.2

LAMPIRAN F

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : ~~Ma~~ Vickriana

Tanggal Pengujian : 31 Juni 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?		✓			
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?		✓			
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?	✓				
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 1 Lembar Jawaban Kuesioner 1

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : *Isna Fauzia Rahmah*

Tanggal Pengujian : *06/06/2015*

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?	✓				
2	Apakah aplikasi yang dibangun mudah dipelajari?	✓				
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?	✓				
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?		✓			
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?		✓			
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?	✓				

Gambar F. 2 Lembar Jawaban Kuesioner 2

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : ana pumamasari

Tanggal Pengujian : 31-05-15

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?	✓				
2	Apakah aplikasi yang dibangun mudah dipelajari?	✓				
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?	✓				
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?		✓			
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?	✓				

Gambar F. 3 Lembar Jawaban Kuesioner 3

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : Astari Larwati

Tanggal Pengujian : 27/5/2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?		✓			
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?	✓				
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?	✓				
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 4 Lembar Jawaban Kuesioner 4

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : Eko Adhi Wiyono

Tanggal Pengujian : 29 Mei 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?		✓			
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?	✓				
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?	✓				
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?	✓				

Gambar F. 5 Lembar Jawaban Kuesioner 5

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : Widya Andi P

Tanggal Pengujian : 5 Mei 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?		✓			
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?		✓			
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?	✓				
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?		✓			
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 6 Lembar Jawaban Kuesioner 6

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : Ermas Rangga

Tanggal Pengujian : 6 Juni 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?			✓		
2	Apakah aplikasi yang dibangun mudah dipelajari?			✓		
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?			✓		
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?		✓			
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?	✓				
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 7 Lembar Jawaban Kuesioner 7

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : Luan Tadwini A

Tanggal Pengujian : 31 Mei 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?		✓			
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?		✓			
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?	✓				
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?	✓				
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 8 Lembar Jawaban Kuesioner 8

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : *ghandra*

Tanggal Pengujian : *02.06.15*

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?			✓		
2	Apakah aplikasi yang dibangun mudah dipelajari?			✓		
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?				✓	
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?		✓			
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?		✓			
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?		✓			

Gambar F. 9 Lembar Jawaban Kuesioner 9

KUESIONER PENGUJIAN PERANGKAT LUNAK

Nama Penguji : M. Anes Rahman

Tanggal Pengujian : 6 Juni 2015

No	Pertanyaan	SS	S	TS	STS	Catatan
1	Apakah aplikasi yang dibangun mudah digunakan?		✓			
2	Apakah aplikasi yang dibangun mudah dipelajari?			✓		
3	Apakah anda setuju penggunaan bahasa yang digunakan mudah dimengerti?			✓		
4	Apakah anda setuju aplikasi ini dapat membantu dalam menentukan kualitas perangkat lunak?	✓				
5	Apakah anda setuju aplikasi ini dapat membantu dalam mengetahui karakteristik manakah dari perangkat lunak yang perlu ditingkatkan?		✓			
6	Apakah anda setuju aplikasi ini yang dibangun bermanfaat?		✓			
7	Secara umum, apakah anda setuju bahwa performa perangkat lunak sangat baik?			✓		

Gambar F. 10 Lembar Jawaban Kuesioner 10

LAMPIRAN G

```
1 package satu;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import javax.swing.JOptionPane;
6 import net.sf.jasperreports.engine.JasperCompileManager;
7 import net.sf.jasperreports.engine.JasperFillManager;
8 import net.sf.jasperreports.engine.JasperPrint;
9 import net.sf.jasperreports.engine.JasperReport;
10 import net.sf.jasperreports.engine.design.JasperDesign;
11 import net.sf.jasperreports.engine.xml.JRXmlLoader;
12 import net.sf.jasperreports.view.JasperViewer;
13
14 public class kelasreport{
15
16     koneksi kon = new koneksi();
17     public void report(String fileReport){
18         kon.bukakon();
19
20         try{
21             Map<String,String> param = new
22 HashMap<String,String>();
23             param.put("Report Title", "Report Paket
24 Pesta");
25             JasperDesign design =
26 JRXmlLoader.load(fileReport);
27             JasperReport report =
28 JasperCompileManager.compileReport(design);
29             JasperPrint print =
30 JasperFillManager.fillReport(report,param,kon.kon);
31             //conn adalah nama koneksi pada kelas Koneksi
32 yang berisi Driver Manager,
33 //sesuaikan dengan koneksi masing-masing
34 JasperViewer.viewReport(print,false);
35
36         }catch(Exception x){
37             JOptionPane.showMessageDialog(null,"salah di
38 "+x);
39         }
40     }
41 }
```

```
42
43
```

```
}
```

Kode Sumber G. 1 Kode Sumber Kelasreport.java

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package cerdascermat;
6
7  import java.io.IOException;
8  import java.net.DatagramPacket;
9  import java.net.InetAddress;
10 import java.net.MulticastSocket;
11
12 /**
13  *
14  * @author M.Aprialdi
15  */
16 public class Client implements Runnable{
17     private String name;
18     private int score;
19     private MulticastSocket client;
20     private InetAddress group;
21     private String msg;
22     private int moreTime;
23     private long tStart, tNow;
24
25     public long gettStart() {
26         return tStart;
27     }
28
29     public int getMoreTime() {
30         tNow = System.currentTimeMillis()/1000;
31         moreTime = (int) (tNow - tStart);
32         return moreTime;
33     }
34
35     public Client() throws IOException {
36         client = new MulticastSocket(1222);
37         group = InetAddress.getByName("234.5.6.7");
38         tStart = -1;
39     }
40
41     public String getName() {
42         return name;
43     }
44 }
```

```

45
46     public void setName(String name) {
47         this.name = name;
48     }
49
50     public int getScore() {
51         return score;
52     }
53
54     public void setScore(int score) {
55         this.score = score;
56     }
57
58     public String displaytThis() {
59         return msg;
60     }
61
62     @Override
63     public void run() {
64         try{
65             while(true){
66                 byte buf[] = new byte[1024];
67                 DatagramPacket data = new
68 DatagramPacket(buf,buf.length);
69                 client.receive(data);
70                 String msg2 = new
71 String(data.getData()).trim();
72                 System.out.println(msg2+"this is message");
73                 String[] trans = msg2.split("\\|");
74                 for(String s : trans)
75 System.out.println(s);
76                 if(trans[0].equals("*")){
77                     if(!trans[1].equals("X")){
78                         msg = trans[1];
79                         if(trans[1].indexOf("Host: ")==
80 1)tStart = System.currentTimeMillis()/1000;
81                     }
82                     else{
83                         msg=trans[2];
84                         client.leaveGroup(group);
85                         tStart=-1;
86                     }
87                 } else if(trans[0].equals("@")){
88                     System.out.println("private message");
89                     if(trans[1].equals(name) ||
90 name.equals("PREREGISTER")){
91                         System.out.println("for me");
92                     }
93

```

```

94         if(trans[2].equals("?")) score =
95         score + Integer.parseInt(trans[3]);
96         else if(trans[2].equals(">")){
97             System.out.println("can i
98         register?");
99             if(trans[3].equals("Y")){
100                 this.name=trans[1];
101                 this.score=0;
102                 msg="Welcome
103         "+name+"\nPlease wait...";
104                 score = 0;
105             }
106             else{
107                 client.leaveGroup(group);
108                 msg = "The game has already
109         started\n";
110             }
111         }
112     }
113 }
114 }
115 } catch (IOException e){
116     System.err.println("IOException: " + e);
117 }
118 }
119 }
120
121     public void sendToServer(String msg) throws
122     IOException {System.out.println("about to send");
123         DatagramPacket data = new
124         DatagramPacket(msg.getBytes(),0, msg.length(), group,
125         1222);
126         client.send(data);System.out.println("sent");
127     }
128
129     public void answer(String ans) throws IOException {
130         sendToServer("?|" +name+"|" +ans);
131     }
132
133     public void register(String name2) throws IOException
134     {System.out.println("regis");
135         client.joinGroup(group);
136         sendToServer(">" +name2);
137         name="PREREGISTER";
138         System.out.println("ok");
139     }

```

Kode Sumber G. 2 Kode Sumber Client.java

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5   package fp;
6
7   import java.io.BufferedReader;
8   import java.io.BufferedWriter;
9   import java.io.File;
10  import java.io.FileNotFoundException;
11  import java.io.FileOutputStream;
12  import java.io.FileReader;
13  import java.io.FileWriter;
14  import java.io.IOException;
15  import java.util.Scanner;
16  import java.util.logging.Level;
17  import java.util.logging.Logger;
18
19  /**
20   *
21   * @author M.Aprialdi
22   */
23  public class Reduksi {
24      static int jumlahKolom=39000;
25      public static void main(String[] args) {
26
27          System.out.println("Masukkan jumlah baris: ");
28          Scanner scanBaris=new Scanner(System.in);
29          int jumlahBaris=scanBaris.nextInt();
30
31          System.out.println("Masukkan dimensi akhir: ");
32          Scanner scanReduksi=new Scanner(System.in);
33          int jumlahReduksi=scanReduksi.nextInt();
34
35          int besarPembagi=jumlahKolom/jumlahReduksi;
36
37
38          double sum[]=new double[besarPembagi];
39          double data[][]= new
40  double[jumlahBaris][jumlahKolom];
41          double dataReduksi[][]=new
42  double[jumlahBaris][jumlahReduksi];
43          BufferedReader readerFile=null;
44          try {
45
46
```



```

47         readerFile = new BufferedReader(new
48         FileReader("E:/Semester 6/KK C/DatasetANormalEdit.txt"));
49         } catch (FileNotFoundException ex) {
50
51         Logger.getLogger(Individu.class.getName()).log(Level.SEVERE
52         RE, null, ex);
53         }
54         String line = null;
55
56         try {
57             int i=0;
58             while ((line = readerFile.readLine()) !=
59             null)
60                 {
61                     String[] parts = line.split(",");
62                     for (int j=0; j<jumlahKolom; j++) {
63
64                     data[i][j]=Double.parseDouble(parts[j]);
65                     }
66                     i++;
67                 }
68             } catch (IOException ex) {
69
70         Logger.getLogger(Individu.class.getName()).log(Level.SEVERE
71         RE, null, ex);
72         }
73
74         for(int x=0; x<jumlahBaris; x++)
75             {
76                 for(int i=0; i<jumlahReduksi; i++)
77                     {
78                         sum[i]=0;
79                         for(int j=i*besarPembagi;
80                         j<(i+1)*besarPembagi; j++)
81                             {
82                                 sum[i]+=data[x][j];
83                             }
84                             dataReduksi[x][i]=sum[i]/besarPembagi;
85                         }
86                 }
87             }
88         try{
89             FileWriter fstream = new
90             FileWriter("normalreduksi10.txt");
91             BufferedWriter out = new
92             BufferedWriter(fstream);
93             for(int i=0; i<jumlahBaris; i++)
94                 {
95                     for(int j=0; j<jumlahReduksi; j++)

```

```
96         {
97
98         out.write(String.valueOf(dataReduksi[i][j])+" ");
99             out.flush();
100        }
101        out.write("\n");
102        out.flush();
103    }
104    out.close();
105
106    }catch(Exception e){
107        System.err.println("Error: " +
108    e.getMessage());
109    }
110
111    }
```

Kode Sumber G. 3 Kode Sumber Reduksi.java

(Halaman ini sengaja dikosongkan)

LAMPIRAN H

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
1	Fungsionalitas	Kesesuaian	Persentase operasi yang sesuai	-		Kuesioner isian
		Akurasi	Persentase operasi yang memenuhi akurasi yang dibutuhkan	-		Kuesioner isian
			Pemenuhan terhadap presisi yang dibutuhkan	-		Kuesioner subjektif
		<i>Interoperability</i>	Basis data yang digunakan	-		Kuesioner pilihan ganda

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
		Keamanan	Persentase pengontrolan akses yang disediakan	-		Kuesioner isian
			Tingkat keketatan akses	-		Kuesioner pilihan ganda
		Pemenuhan fungsionalitas	Pemenuhan terhadap standar fungsionalitas	-		Kuesioner subjektif
		Kustomabilitas	Tingkat kustomabilitas	-		Kuesioner isian
2	Efisiensi	Waktu	Jumlah variabel global	-		Kode sumber
			Tipe <i>translator</i>	-		Kode sumber

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
			Kemampuan prosesor	-		Kuesioner pilihan ganda
		Penggunaan sumber daya	Persentase CPU yang tidak terpakai	-		Kuesioner isian
			Dukungan perangkat lunak terhadap sumber daya luar	-		Kuesioner pilihan ganda
		Pemenuhan efisiensi	Pemenuhan terhadap standar fungsionalitas	-		Kuesioner subjektif
		Skalabilitas	Dukungan untuk beberapa pengguna	-		Kuesioner pilihan ganda
3	Keterawatan	<i>Analyzability</i>	KLOC	-		Kode sumber

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
			Kemampuan	Kemampuan Teknis	-	Kuesioner subjektif
				Pengalaman industri organisasi	-	-
				Kemampuan tim pengembang	Rata-rata kualitas anggota pengembang	Kuesioner subjektif
					Kerjasama antara tim pengembang	Kuesioner subjektif
					Performa keseluruhan tim pengembang	Kuesioner subjektif
			Pengalaman manajer	Pengalaman manajer proyek dalam bisnis IT	-	Kuesioner pilihan ganda

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
				Pengalaman manajer proyek sebagai manajer	-	Kuesioner pilihan ganda
			Jumlah KLOC per anggota tim	-	-	Kuesioner isian
			Ketersediaan dokumentasi	-	-	Kuesioner pilihan ganda
			<i>Cyclomatic complexity</i>	-	-	Kode sumber
			Jumlah versi yang telah dirilis	-	-	Kuesioner pilihan ganda
			Level <i>Capability Maturity Model</i> (CMM)	-	-	Kuesioner pilihan ganda
		<i>Changeability</i>	Persentase jumlah fitur	-	-	Kuesioner isian

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
			yang dapat diubah			
		<i>Testability</i>	Ketersediaan jumlah kasus uji coba yang cukup	-	-	Kuesioner subjektif
		Pemenuhan <i>maintainability</i>	Pemenuhan terhadap standar <i>maintainability</i>	-	-	Kuesioner subjektif
		<i>Track-ability</i>	Ketersediaan sistem penelusuran fungsional dan perilaku sistem	-	-	Kuesioner pilihan ganda
			Kemudahan dalam penelusuran	-	-	Kuesioner subjektif

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
			versi lama perangkat lunak			
4	Portabilitas	Kemampuan adaptasi	Kompatibilitas pada sistem operasi lain	-	-	Kuesioner pilihan ganda
			Penggunaan <i>tools</i> dasar	-	-	Kuesioner pilihan ganda
			Jumlah <i>package</i> yang harus diinstall terlebih dulu	-	-	Kuesioner pilihan ganda
		<i>Installability</i>	Jumlah package yang harus diinstall terlebih dulu selain OS	-	-	Kuesioner pilihan ganda

No	Karakteristik	Subkarakteristik	Metrik	SubMetrik	SubsubMetrik	Sumber Ekstraksi
		Pemenuhan <i>portability</i>	Pemenuhan terhadap standar <i>portability</i>	-	-	Kuesioner subjektif

BIODATA PENULIS



Mohammad Aprialdi Rizky Pratama, lahir di Malang pada tanggal 6 April 1994. Penulis menempuh pendidikan mulai dari SDN 009 Samarinda (1999-2001), SDN Sawojajar I Malang (2001-2005), SMPN 3 Malang (2005-2008), SMK Telkom Malang (2008-2011) dan S1 Teknik Informatika ITS (2011-2015).

Di jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak dan memiliki ketertarikan di bidang SQA, basis data, *software evolution*, dan *software maintenance*. Selama masa kuliah, penulis aktif dalam organisasi yang ada di lingkungan kampus ITS yaitu Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) sebagai staf dan kepala departemen kesejahteraan mahasiswa, dan sebagai administrator Laboratorium Pemrograman (LP). Dan penulis juga beberapa kali menjadi asisten dosen, diantaranya Asisten Praktikum mata kuliah Basis Data, Asisten Dosen mata kuliah Perancangan dan Analisis Algoritma, Asisten Dosen mata kuliah Analisis dan Perancangan Sistem, dan Asisten Dosen mata kuliah Kecerdasan Komputasional. Penulis dapat dihubungi melalui surel: mohammad.aprialdi@gmail.com