

SKRIPSI

**PENGUKURAN COUPLING PADA PERANGKAT LUNAK
BERBASIS OBJEK**



Maudy Nur Avianti

NPM: 2013730057

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2017**

UNDERGRADUATE THESIS

**COUPLING MEASUREMENT IN OBJECT ORIENTED
SOFTWARE**



Maudy Nur Avianti

NPM: 2013730057

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2017**

LEMBAR PENGESAHAN



**PENGUKURAN COUPLING PADA PERANGKAT LUNAK
BERBASIS OBJEK**

Maudy Nur Avianti

NPM: 2013730057

Bandung, 12 Desember 2017

Menyetujui,

Pembimbing

Dott. Thomas Anung Basuki

Ketua Tim Penguji

Luciana Abednego, M.T.

Anggota Tim Penguji

Natalia, M.Si.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENGUKURAN COUPLING PADA PERANGKAT LUNAK BERBASIS OBJEK

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 12 Desember 2017



Maudy Nur Avianti
NPM: 2013730057

ABSTRAK

Persaingan ketat antara perusahaan yang bergerak di bidang pengembangan perangkat lunak mengakibatkan produk perangkat lunak yang dihasilkan harus berkualitas. Tren pengembangan perangkat lunak saat ini cenderung berbasis objek. Salah satu ciri dari kualitas perangkat lunak adalah tingkat *coupling* yang rendah. *Coupling* adalah sebuah istilah yang digunakan saat terdapat ketergantungan antara satu modul dengan modul lainnya di sebuah perangkat lunak. Penelitian ini bertujuan untuk mengembangkan perangkat lunak yang mampu mengetahui tingkat *coupling* yang dimiliki oleh suatu perangkat lunak.

Untuk mengukur *coupling*, digunakan metrik yang dicetuskan oleh Chidamber dan Kemerer. Metrik tersebut terdiri atas *Depth of Inheritance Tree (DIT)*, *Number of Children (NOC)*, *Coupling between Object Classes (CBO)*, *Response for a Class (RFC)*, dan *Lack of Cohesion in Method (LCOM)*. Pengukuran metrik tersebut membutuhkan pengenalan struktur dari kode program. ANTLR digunakan sebagai kakas untuk mengenali struktur tersebut, dengan cara mengubah kode program menjadi sebuah *Abstract Syntax Tree (AST)*. Dari AST tersebut, diambil elemen-elemen yang dibutuhkan oleh algoritma metrik.

Berdasarkan pengujian berupa *unit testing* didapat bahwa perangkat lunak berjalan sesuai dengan implementasi. Selain itu dilakukan pengujian terhadap perangkat sejenis dan didapatkan bahwa perilaku perangkat lunak yang dikembangkan serupa dengan perilaku perangkat lunak sejenis. Didapat pula bahwa RFC dikenali sebagai parameter yang berpengaruh dalam pengukuran *coupling*.

Kata-kata kunci: *Coupling*, Perangkat lunak berbasis objek, ANTLR, Chidamber-Kemerer, *Abstract Syntax Tree*, AST

ABSTRACT

Tough competition between companies that engaged in software development affects the software to be in excellent quality. Current trend in software development is object-oriented software. One of the characteristics in a software quality is to have a low level of coupling. Coupling is a term which is used when there are dependencies between one module and another in a software. This study aims to develop a software that is able to find out coupling level from a software.

Metrics proclaimed by Chidamber and Kemerer is used to measure the coupling. The metrics consist of Depth of Inheritance Tree(DIT), Number of Children (NOC), Coupling between Object Classes (CBO), Response for a Class (RFC), and Lack of Cohesion in Method (LCOM). Measurement process needs structure recognizer for recognizing the input. ANTLR is used for recognizing such structure by changing the input into Abstract Syntax Tree (AST). From the AST, the software will collect elements which is needed by the metric algorithm.

Based on unit testing, this metric measurement software works well as implemented. In addition, this metric measurement software is being tested by similar software, and it is found that the metric measurement software behaviour is the same as the similar software behaviour. It is also found that RFC is recognized as influential parameter in coupling measurement.

Keywords: Coupling, Object-oriented software, ANTLR, Chidamber-Kemerer, Abstract Syntax Tree, AST

Dipersembahkan kepada diri sendiri

KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan skripsi ini. Dalam proses penyusunan skripsi, penulis banyak mendapat kesempatan untuk mempelajari hal-hal baru, serta mendapatkan banyak bantuan baik secara langsung maupun tidak langsung dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada seluruh pihak yang telah berperan dalam penyusunan skripsi ini. Secara khusus penulis ingin menyampaikan terima kasih kepada:

1. Orang tua penulis serta adik penulis dan keluarga besar penulis yang selalu mendoakan dan mendukung penulis selama penyusunan skripsi.
2. Bapak Dott. Thomas Anung Basuki selaku dosen pembimbing yang telah memberi arahan dan masukan selama penyusunan skripsi.
3. Ibu Luciana Abednego, M.T dan Ibu Natalia, M.Si selaku dosen penguji yang telah memberikan kritik dan saran untuk skripsi ini.
4. Hasudungan Dimas Nathanael, yang selalu memotivasi, menolong, mendoakan, dan mende-ngarkan keluh kesah penulis. Terima kasih untuk selalu berada di samping penulis sekalipun orang lain menyuruh sebaliknya.
5. Sani Puspaning Naima dan Zeliana Yuanisa. Terima kasih telah selalu ada di saat penulis senang maupun sedih. *You guys really are the bestest bestfriend anyone could ever ask for.*
6. Terima kasih kepada teman-teman penulis semasa berkuliah yaitu Devina Emily, Janice Sella, Clara, Devi Handevi, dan Jacinta Delora.
7. Terakhir, *thank you to the haters in the back. Thank you for selling me short, all I can say is that gives me so much power to turn your ugly remarks into my fuel.*

Bandung, Desember 2017

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Perangkat Lunak	5
2.2 Rekayasa Perangkat Lunak [1]	6
2.3 Paradigma Berbasis Objek	7
2.4 Implementasi Bahasa Java Menggunakan Paradigma Berbasis Objek [2]	9
2.5 Coupling	11
2.6 Jenis Coupling	12
2.7 Metrik <i>Coupling</i>	14
2.8 Graf [3]	16
2.8.1 Lintasan (<i>path</i>)	18
2.8.2 Sirkuit (<i>circuit</i>)	18
2.8.3 Terhubung (<i>connected</i>)	18
2.9 Pohon [4]	19
2.9.1 Terminologi pada Pohon Berakar	19
2.9.2 Penelusuran Pohon Secara <i>Depth-First Search</i>	21
2.10 <i>Compiler</i> [5]	22
2.10.1 Bagian-bagian <i>Compiler</i>	22
2.10.2 Proses Kompilasi	23
2.10.3 BNF [3]	25
2.10.4 EBNF	25
2.11 ANTLR [6]	27
2.11.1 Pemasangan ANTLR pada Sistem Windows	27
2.11.2 <i>Grammar</i> ANTLR	28
2.11.3 Parsing menggunakan ANTLR	29
3 ANALISIS	33
3.1 Deskripsi Masalah	33

3.2	Analisis Hubungan antara Jenis <i>Coupling</i> dan Metrik <i>Coupling</i>	34
3.3	Analisis Kebutuhan Perangkat Lunak	35
3.3.1	Kebutuhan Masukan Perangkat Lunak	35
3.3.2	Kebutuhan Keluaran Perangkat Lunak	36
3.4	Analisis Perangkat Lunak	36
3.4.1	Diagram Kelas	36
3.4.2	Diagram Aktivitas	36
3.4.3	Perangkat Lunak Sejenis	38
4	PERANCANGAN	39
4.1	Perancangan Antarmuka	39
4.2	Perancangan Masukan	41
4.2.1	<i>File</i> Java	41
4.2.2	ANTLR	41
4.2.3	<i>File</i> hasil pembangkitan <i>file grammar</i> Java.g4	41
4.3	Algoritma Pengukuran Nilai <i>Coupling</i>	42
4.3.1	Algoritma <i>Depth of Inheritance Tree</i> (DIT)	42
4.3.2	Algoritma <i>Number of Children</i> (NOC)	44
4.3.3	Algoritma <i>Coupling between Object Classes</i> (CBO)	45
4.3.4	Algoritma <i>Response for a Class</i> (RFC)	46
4.3.5	Algoritma <i>Lack of Cohesion in Methods</i> (LCOM)	47
4.4	Diagram Kelas Rinci	49
4.4.1	Kelas ListenerDIT	51
4.4.2	Kelas ListenerNOC	52
4.4.3	Kelas ListenerCBO	53
4.4.4	Kelas ListenerRFC	55
4.4.5	Kelas ListenerLCOM	58
4.4.6	Kelas Controller	58
5	IMPLEMENTASI DAN PENGUJIAN	65
5.1	Implementasi	65
5.1.1	Lingkungan Perangkat Keras	65
5.1.2	Lingkungan Perangkat Lunak	65
5.1.3	Implementasi Algoritma ke dalam Perangkat Lunak	66
5.2	Pengujian	67
5.2.1	<i>Unit Testing</i>	67
5.2.2	Pengujian terhadap Perangkat Lunak Sejenis	116
6	KESIMPULAN DAN SARAN	121
6.1	Kesimpulan	121
6.2	Saran	121
	DAFTAR REFERENSI	123
	A KODE PROGRAM	125
	B FILE GRAMMAR UNTUK BAHASA PEMROGRAMAN JAVA	143

DAFTAR GAMBAR

2.1	Keterkaitan Antarmodul	11
2.2	Jenis <i>Coupling</i>	12
2.3	Contoh Graf	17
2.4	Contoh Graf Berarah dan Graf Tak Berarah	18
2.5	Contoh Graf Terhubung dan Graf Tak Berhubung	18
2.6	Contoh Pohon dan Bukan Pohon	19
2.7	Contoh Pohon Berakar	19
2.8	Contoh Subpohon	20
2.9	Contoh Penelusuran Pohon	21
2.10	Cara Kerja <i>Compiler</i>	22
2.11	Bagian dari <i>Compiler</i>	22
2.12	Proses yang Terjadi di dalam <i>Compiler</i>	23
2.13	AST Hasil <i>Parsing</i> Menggunakan ANTLR	30
3.1	Rancangan Diagram Kelas	36
3.2	Diagram Aktivitas	37
4.1	Rancangan Antarmuka Utama	39
4.2	Rancangan Antarmuka Dialog <i>File Chooser</i>	40
4.3	Diagram Kelas Rinci	50
4.4	Kelas ListenerDIT	51
4.5	Kelas ListenerNOC	52
4.6	Kelas ListenerCBO	53
4.7	Kelas ListenerRFC	55
4.8	Kelas ListenerLCOM	58
4.9	Kelas Controller	59
5.1	Hasil <i>Unit Testing</i> DIT untuk Skenario 1	70
5.2	Hasil <i>Unit Testing</i> DIT untuk Skenario 2	71
5.3	Hasil <i>Unit Testing</i> DIT untuk Skenario 3	72
5.4	Hasil <i>Unit Testing</i> DIT untuk Skenario 4	73
5.5	Hasil <i>Unit Testing</i> NOC untuk Skenario 1	74
5.6	Hasil <i>Unit Testing</i> NOC untuk Skenario 2	75
5.7	Hasil <i>Unit Testing</i> NOC untuk Skenario 3	77
5.8	Hasil <i>Unit Testing</i> NOC untuk Skenario 4	78
5.9	Hasil <i>Unit Testing</i> NOC untuk Skenario 5	80
5.10	Hasil <i>Unit Testing</i> CBO untuk Skenario 1	81
5.11	Hasil <i>Unit Testing</i> CBO untuk Skenario 2	82
5.12	Hasil <i>Unit Testing</i> CBO untuk Skenario 3	83
5.13	Hasil <i>Unit Testing</i> CBO untuk Skenario 4	84
5.14	Hasil <i>Unit Testing</i> CBO untuk Skenario 5	85
5.15	Hasil <i>Unit Testing</i> CBO untuk Skenario 6	86
5.16	Hasil <i>Unit Testing</i> CBO untuk Skenario 7	87

5.17 Hasil <i>Unit Testing</i> CBO untuk Skenario 8	88
5.18 Hasil <i>Unit Testing</i> CBO untuk Skenario 9	89
5.19 Hasil <i>Unit Testing</i> CBO untuk Skenario 10	90
5.20 Hasil <i>Unit Testing</i> CBO untuk Skenario 11	91
5.21 Hasil <i>Unit Testing</i> CBO untuk Skenario 12	92
5.22 Hasil <i>Unit Testing</i> CBO untuk Skenario 13	93
5.23 Hasil <i>Unit Testing</i> CBO untuk Skenario 14	94
5.24 Hasil <i>Unit Testing</i> CBO untuk Skenario 15	95
5.25 Hasil <i>Unit Testing</i> RFC untuk Skenario 1	96
5.26 Hasil <i>Unit Testing</i> RFC untuk Skenario 2	97
5.27 Hasil <i>Unit Testing</i> RFC untuk Skenario 3	98
5.28 Hasil <i>Unit Testing</i> RFC untuk Skenario 4	99
5.29 Hasil <i>Unit Testing</i> RFC untuk Skenario 5	100
5.30 Hasil <i>Unit Testing</i> RFC untuk Skenario 6	101
5.31 Hasil <i>Unit Testing</i> RFC untuk Skenario 7	102
5.32 Hasil <i>Unit Testing</i> RFC untuk Skenario 8	103
5.33 Hasil <i>Unit Testing</i> RFC untuk Skenario 9	104
5.34 Hasil <i>Unit Testing</i> RFC untuk Skenario 10	105
5.35 Hasil <i>Unit Testing</i> RFC untuk Skenario 11	106
5.36 Hasil <i>Unit Testing</i> RFC untuk Skenario 12	107
5.37 Hasil <i>Unit Testing</i> RFC untuk Skenario 13	108
5.38 Hasil <i>Unit Testing</i> RFC untuk Skenario 14	109
5.39 Hasil <i>Unit Testing</i> LCOM untuk Skenario 1	110
5.40 Hasil <i>Unit Testing</i> LCOM untuk Skenario 2	112
5.41 Hasil <i>Unit Testing</i> LCOM untuk Skenario 3	113
5.42 Hasil <i>Unit Testing</i> LCOM untuk Skenario 4	114
5.43 Hasil <i>Unit Testing</i> LCOM untuk Skenario 5	115
5.44 Hasil <i>Unit Testing</i> LCOM untuk Skenario 6	116

DAFTAR TABEL

4.1	Fungsi dari Elemen Milik <i>Frame</i> Utama	40
4.2	Fungsi dari Elemen Milik Dialog <i>File Chooser</i>	40
4.3	Irisan <i>Instance Variable</i> yang Digunakan oleh <i>Method</i> di KelasLCOM	48
5.1	Spesifikasi Perangkat Keras	65
5.2	Spesifikasi Perangkat Lunak	65
5.3	Tabel Kemungkinan Skenario <i>Unit Testing</i>	67
5.4	Irisan <i>Instance Variable</i> yang Digunakan oleh <i>Method</i> untuk Skenario 1	110
5.5	Irisan <i>Instance Variable</i> yang Digunakan oleh <i>Method</i> untuk Skenario 2	111
5.6	Irisan <i>Instance Variable</i> yang Digunakan oleh <i>Method</i> untuk Skenario 3	113
5.7	Tabel Perbandingan Hasil untuk Kelas ListenerCBO	117
5.8	Tabel Perbandingan Hasil untuk Kelas ListenerDIT	118
5.9	Tabel Perbandingan Hasil untuk Kelas ListenerLCOM	118
5.10	Tabel Perbandingan Hasil untuk Kelas ListenerNOC	118
5.11	Tabel Perbandingan Hasil untuk Kelas ListenerRFC	118
5.12	Tabel Perbandingan Hasil untuk Kelas Controller	119
5.13	Tabel Perbandingan Hasil untuk Kelas CheckingAccount	119
5.14	Tabel Perbandingan Hasil untuk Kelas Account	119
5.15	Tabel Perbandingan Hasil untuk Kelas SavingsAccount	120
5.16	Tabel Perbandingan Hasil untuk Kelas AccountDemo	120

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi, khususnya perangkat lunak, saat ini telah menjadi kebutuhan tak tergantikan dalam hidup manusia. Perangkat lunak dikembangkan agar pekerjaan manusia yang bisa diotomatiskan dapat dilakukan oleh komputer. Hal tersebut mendorong kemunculan perusahaan-perusahaan yang bergerak di bidang pengembangan perangkat lunak. Tingkat persaingan yang tinggi ini mengharuskan pengembang perangkat lunak berlomba untuk membuat perangkat lunak yang berkualitas.

Kualitas suatu perangkat lunak tidak hanya dilihat dari seberapa efektif dan efisien perangkat lunak tersebut dalam menyelesaikan masalah. Kualitas perangkat lunak juga dilihat dari bagaimana perangkat lunak tersebut dapat dikembangkan dan dipelihara. Sebuah perangkat lunak dikatakan berkualitas jika pengembangan fitur dan pemeliharannya mudah dilakukan. Hal ini berkaitan dengan tingkat *coupling* yang rendah.

Coupling adalah sebuah istilah yang digunakan saat terjadi kebergantungan antara satu modul dengan modul lainnya pada sebuah perangkat lunak.[1] Jika suatu perangkat lunak memiliki tingkat *coupling* yang tinggi, maka kode program semakin sulit untuk dimengerti dan diubah. Kebalikannya, jika suatu perangkat lunak memiliki tingkat *coupling* yang rendah, maka kode program semakin mudah untuk dimengerti dan diubah.

Kesulitan pengubahan kode program pada perangkat lunak dapat membuat perusahaan merugi. Jika terjadi pengubahan fitur pada perangkat lunak, perusahaan harus terlebih dahulu mempelajari ulang kode program sebelum bisa mengimplementasikan pengubahannya. Perusahaan menjadi kehilangan waktu dan tenaga yang sebenarnya bisa dialokasikan untuk kegiatan lainnya. Hal ini berpengaruh pada tingkat persaingan sebuah perusahaan, karena perusahaan harus cepat tanggap dalam melakukan penambahan fitur.

Atas dasar tersebut, diperlukan pengurangan tingkat *coupling* pada perangkat lunak. Hal ini dapat dilakukan dengan mengukur tingkat *coupling* yang dimiliki sebuah perangkat lunak. Hasil dari pengukuran tersebut dapat memperlihatkan seberapa eratny kebergantungan antarmodul. Tidak hanya itu saja, hasil pengukuran juga dapat menjadi patokan seberapa besar usaha yang harus dikerahkan untuk mengurangi tingkat *coupling*.

Skripsi ini diharapkan menghasilkan perangkat lunak yang dapat melakukan pengukuran *coupling* yang dimiliki oleh perangkat lunak/kode program lainnya. Skripsi ini berpusat pada pengukuran *coupling* di perangkat lunak berbahasa Java. Selain itu, skripsi ini juga diharapkan dapat menjadi batu loncatan bagi skripsi lainnya yang ingin melakukan hal serupa, namun dengan jenis *coupling* yang berbeda.

1.2 Rumusan Masalah

Dari latar belakang di atas, timbul beberapa masalah yang dibahas pada skripsi ini. Beberapa masalah tersebut di antaranya:

1. Apa jenis *coupling* yang dapat diukur?

2. Bagaimana cara mengukur *coupling*?
3. Bagaimana implementasi pengukuran *coupling* tersebut dalam perangkat lunak?
4. Bagaimana mengukur kualitas perangkat lunak yang dihasilkan?

1.3 Tujuan

Tujuan dari penulisan skripsi ini adalah:

1. Menentukan jenis *coupling* mana saja yang dapat diukur
2. Mengetahui bagaimana cara mengukur *coupling*
3. Mengimplementasikan cara pengukuran *coupling* ke perangkat lunak yang dibuat
4. Mendefinisikan pengukuran kuantitas perangkat lunak yang dihasilkan

1.4 Batasan Masalah

Rumusan masalah yang telah disebutkan di atas masih memiliki ruang lingkup yang cukup luas. Karena keterbatasan waktu dan kemampuan yang dimiliki, maka skripsi ini hanya memfokuskan pada batasan masalah sebagai berikut:

1. Kode program yang diukur tingkat *coupling*-nya menggunakan bahasa Java.
2. Pengukuran *coupling* dilakukan secara sintaksis.

1.5 Metodologi Penelitian

Berikut ini adalah langkah-langkah yang dilakukan dalam skripsi ini.

1. Studi Pustaka,
Tahap ini dilakukan untuk memperdalam pengetahuan mengenai dasar-dasar teori yang berhubungan dengan *coupling*. Studi pustaka lain yang perlu dipelajari adalah hubungan *coupling* dengan bahasa pemrograman Java.
2. Analisis Jenis *Coupling*,
Tahap ini dilakukan untuk memutuskan jenis *coupling* manakah yang dapat diukur dan apa algoritma yang sesuai untuk mengukur jenis *coupling* tersebut.
3. Menganalisis Kebutuhan Perangkat Lunak,
Tahap ini dilakukan untuk menganalisis kebutuhan apa saja yang dibutuhkan oleh perangkat lunak, seperti apa struktur data yang cocok untuk diimplementasikan dan apa masukan dan keluaran yang dibutuhkan oleh perangkat lunak.
4. Melakukan Implementasi Perangkat Lunak,
Tahap ini dilakukan untuk melakukan pengimplementasian algoritma dan struktur data yang telah dirancang pada analisis kebutuhan perangkat lunak. Pengimplementasian ini menggunakan bahasa pemrograman Java.
5. Melakukan Pengujian dan Eksperimen,
Tahap ini dilakukan untuk mengetahui apakah perangkat lunak yang telah dibangun telah memenuhi standar-standar yang telah ditentukan sebelumnya.

6. Melakukan Dokumentasi.

Tahap ini dilakukan untuk mendokumentasikan proses pengerjaan skripsi yang dilakukan.

1.6 Sistematika Pembahasan

Sistematika pembahasan skripsi ini dibagi menjadi enam buah bab, berikut adalah sistematika pembahasan pada skripsi ini.

1. Bab 1 Pendahuluan,

Bab ini membahas mengenai latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

2. Bab 2 Landasan Teori,

Bab ini membahas mengenai pengetahuan dan teori dasar mengenai *coupling* dan jenis-jenisnya.

3. Bab 3 Analisis,

Bab ini membahas mengenai hasil analisis terhadap masalah dan kebutuhan perangkat lunak yang dibangun, sebagai jawaban permasalahan yang dihadapi saat ini. Pada bab ini juga disertakan diagram aktivitas dan diagram kelas usulan.

4. Bab 4 Perancangan,

Bab ini membahas perancangan perangkat lunak, seperti apa masukan dan keluaran, rancangan antarmuka, rancangan struktur data, dan rancangan kelas (atribut, metode, dan hubungan antarkelas).

5. Bab 5 Implementasi dan Pengujian,

Bab ini membahas bagaimana mengimplementasikan perangkat lunak. Selain itu, bab ini membahas proses pengujian dan hasil pengujian untuk membuktikan bahwa perangkat lunak yang dibuat telah memenuhi standar-standar yang telah ditetapkan sebelumnya.

6. Bab 6 Kesimpulan dan Saran.

Bab ini berisi kesimpulan yang dapat diambil dari pengujian perangkat lunak. Bab ini juga berisi saran untuk pengembangan perangkat lunak sejenis.