

ABSTRAK

Salah satu bidang kajian dalam bidang teknologi informasi adalah rekayasa perangkat lunak. Dalam rekayasa perangkat lunak, terdapat konsep yang mendasari berbagai jenis metodologi pengembangan perangkat lunak yang dikenal dengan *SDLC* (*Software Development Life Cycle*). Dalam tahapan *SDLC*, pengembang sistem memodelkan kelas-kelas dan relasi-relasi yang menyusun sistem yang akan dibuatnya ke dalam bentuk diagram kelas. Diantara beberapa aplikasi yang biasa dipakai untuk membuat diagram kelas, hanya sedikit aplikasi yang dapat melakukan sinkronisasi secara dua arah antara kode sumber dan diagram kelas.

Pada tugas akhir ini, dibuat sebuah aplikasi untuk membantu proses pembuatan diagram kelas yang dapat melakukan sinkronisasi dua arah antara kode sumber dan diagram kelas. Aplikasi yang telah dibuat dapat membuat diagram kelas berdasarkan kode sumber Java ataupun berkas *XML*. Aplikasi juga dapat membuat *template* kode sumber berdasarkan diagram kelas yang baru dibuat. Hasil akhir dari penelitian ini adalah aplikasi berhasil melakukan sinkronisasi secara dua arah antara kode sumber dan diagram kelas.

Kata kunci: diagram kelas, *xml*, java, kode sumber, sinkronisasi

ABSTRACT

One area of study in the field of information technology is software engineering. In software engineering, there are concepts that underlie different types of software development methodology known as SDLC (Software Development Life Cycle). In SDLC stages, system developers model classes and relationships that make up a system into the form of a class diagram. Among several applications commonly used to create a class diagram, only a few applications can perform a round trip engineering between the source code and its class diagram.

An application has been made to assist the process of making a class diagram which can perform a round trip engineering between the source code and class diagram. The application can create a class diagram based on the Java source code or an XML file. The application can also create a template based on the newly created class diagram. The end result of this research is the application successfully perform round trip engineering between the source code and its class diagram.

Keyword: class diagram, xml, java, source code, round trip engineering

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR SIMBOL.....	xiii
DAFTAR ALGORITMA.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Pembahasan.....	2
1.4 Batasan Masalah	2
1.5 Sistematika Penyajian.....	3
BAB II LANDASAN TEORI	4
2.1 Object Oriented Programming.....	4
2.1.1 Kelas.....	4
2.1.2 <i>Method</i>	4
2.1.3 <i>Inheritance</i>	5
2.2 Java	5
2.2.1 Java Grammar	5

2.3	<i>Unified Modeling Language</i>	8
2.4	<i>Markup Language</i>	8
2.4.1	<i>Extensible Markup Language</i>	8
2.5	Aplikasi Visualisasi Diagram Kelas	9
2.5.1	Visual Paradigm.....	9
2.5.2	Astah	9
2.6	Penelitian Sejenis.....	10
BAB III ANALISIS DAN DISAIN		11
3.1	Analisis	11
3.1.1	DiagramObject.....	11
3.1.2	Parser.....	12
3.1.3	DiagramProcessor	12
3.1.4	Synchronizer	12
3.1.5	Drawer.....	13
3.1.6	Format <i>XML</i>	13
3.2	Gambaran Keseluruhan	16
3.2.1	Persyaratan Antarmuka Eksternal	16
3.2.2	Antarmuka dengan Pengguna	16
3.2.3	Antarmuka Perangkat Keras	17
3.2.4	Antarmuka Perangkat Lunak.....	17
3.2.5	Fitur-fitur Produk Perangkat Lunak	17
3.3	Disain Perangkat Lunak.....	21
3.3.1	Pemodelan Perangkat Lunak.....	21
3.3.2	Disain Antarmuka	33
BAB IV PENGEMBANGAN PERANGKAT LUNAK.....		37
4.1	Implementasi Modul Sinkronisasi	37

4.2	Implementasi Antarmuka Jendela Utama.....	38
4.3	Implementasi Fitur <i>Generate Diagram</i>	40
4.4	Implementasi Fitur Generate Source Code.....	41
4.5	Implementasi Fitur Sinkronisasi Diagram ke Kode Sumber	42
4.6	Implementasi Fitur Sinkronisasi Kode Sumber Ke Diagram	44
BAB V TESTING DAN EVALUASI SISTEM		46
5.1	Pengujian Modul Penulisan Kode Sumber ke Berkas <i>XML</i>	46
5.2	Pengujian Modul Penulisan Berkas <i>XML</i> ke Kode Sumber	49
5.3	Pengujian Dengan <i>Unit Testing</i>	53
5.4	Pengujian Antarmuka	54
5.5	Perbandingan Dengan Aplikasi Sejenis.....	57
5.6	<i>Stress Test</i>	58
5.7	Pengisian Kuisisioner oleh Pengguna	59
BAB VI KESIMPULAN DAN SARAN		61
6.1	Kesimpulan.....	61
6.2	Saran	63
DAFTAR PUSTAKA		65
RIWAYAT HIDUP PENULIS		66
LAMPIRAN.....		67
A.	Lampiran Perbandingan Hasil Diagram Kelas	68

DAFTAR GAMBAR



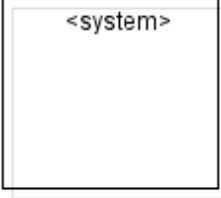
Gambar 2.1 Sintaks untuk mendeklarasi kelas dalam java	6
Gambar 2.2 Sintaks untuk mendeklarasi atribut dalam sebuah kelas	7
Gambar 2.3 Sintaks untuk mendeklarasi sebuah method dalam sebuah kelas	7
Gambar 3.1 Format penulisan tag project	13
Gambar 3.2 Format penulisan tag class	14
Gambar 3.3 Format penulisan tag pos.....	14
Gambar 3.4 Format penulisan tag attr.....	14
Gambar 3.5 Format penulisan tag method	15
Gambar 3.6 Format penulisan tag param	15
Gambar 3.7 Format penulisan tag agregasi, asosiasi, dan komposisi	15
Gambar 3.8 Format penulisan tag interface	16
Gambar 3.9 Rancangan use case diagram.....	22
Gambar 3.10 <i>Activity diagram</i> untuk fitur <i>generate diagram</i>	25
Gambar 3.11 <i>Activity diagram</i> untuk fitur <i>save image</i>	26
Gambar 3.12 <i>Activity diagram</i> untuk fitur <i>XML Editor</i>	27
Gambar 3.13 Rancangan <i>sequence diagram</i> untuk fitur <i>generate diagram</i>	28
Gambar 3.14 Rancangan <i>sequence diagram</i> untuk fitur <i>save image</i>	29
Gambar 3.15 Rancangan <i>sequence diagram</i> untuk fitur <i>xml editor</i>	29
Gambar 3.16 Rancangan <i>package diagram</i> aplikasi	30
Gambar 3.17 Rancangan <i>diagram kelas</i> aplikasi	31
Gambar 3.18 Disain antarmuka untuk jendela utama	34
Gambar 3.19 Disain antarmuka untuk jendela fitur <i>generate diagram</i>	35
Gambar 3.20 Jendela penambahan elemen <i>diagram kelas</i>	35
Gambar 4.1 Implementasi antarmuka jendela utama	39
Gambar 4.2 Implementasi tampilan untuk fitur <i>Generate Diagram</i>	40
Gambar 4.3 Berkas <i>XML</i> yang dibuat oleh aplikasi berdasarkan kode sumber....	41
Gambar 4.4 <i>Template</i> kode sumber yang dibuat oleh aplikasi	41
Gambar 4.5 Penambahan elemen pada salah satu kelas	42
Gambar 4.6 Atribut baru pada kelas <i>Drawer</i>	43
Gambar 4.7 Hasil <i>update</i> pada kode sumber	43

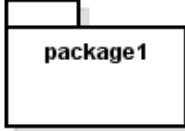
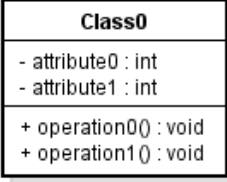
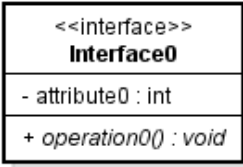
Gambar 4.8 Penambahan atribut baru pada kode sumber.....	44
Gambar 4.9 Pemberitahuan perubahan kode sumber pada aplikasi.....	44
Gambar 4.10 Atribut baru berhasil ditambahkan pada diagram kelas	45
Gambar 5.1 Potongan Kode Program Algoritma Penulisan Berkas XML	47
Gambar 5.2 Potongan Kode Program Algoritma Penulisan <i>Template</i> Kode Sumber	51
Gambar 5.3 Hasil <i>Whitebox Testing</i> Menggunakan JUnit.....	53


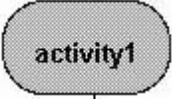

DAFTAR TABEL

Tabel 3.1 Deskripsi use case diagram untuk fitur generate diagram	22
Tabel 3.2 Deskripsi use case diagram untuk fitur save image	22
Tabel 3.3 Deskripsi use case diagram untuk fitur synchronize.....	23
Tabel 3.4 Deskripsi use case diagram untuk fitur save XML	23
Tabel 3.5 Deskripsi use case diagram untuk fitur save source code	23
Tabel 5.1 Hasil Pengujian method writeObjectXML	47
Tabel 5.2 Hasil Pengujian method writeObjectXML	51
Tabel 5.3 <i>Test case</i> untuk fitur <i>generate diagram</i>	54
Tabel 5.4 Test case untuk fitur sinkonisasi	55
Tabel 5.5 Hasil pengujian dari <i>test case</i> untuk fitur <i>generate diagram</i>	55
Tabel 5.6 Hasil pengujian dari <i>test case</i> untuk fitur sinkronisasi	56
Tabel 5.7 Perbandingan Hasil Diagram Kelas dengan Hasil Dari Aplikasi Lain .	57
Tabel 5.8 Perbandingan Hasil Stress Test dengan Hasil dari Aplikasi Lain.....	59
Tabel 5.9 Perhitungan Poin dari Hasil Kuisisioner	60

DAFTAR SIMBOL

Use Case Diagram (UML 2.0)		
No	Gambar	Keterangan
1.		Menggambarkan aktor atau pengguna aplikasi.
2.		Menggambarkan proses atau aksi yang dapat dilakukan oleh aktor pada aplikasi.
3.		Menggambarkan sistem tempat proses dijalankan

Class Diagram (UML 2.0)		
No	Gambar	Keterangan
1.		Menggambarkan paket tempat menyimpan sekumpulan kelas
2.		Menggambarkan sebuah kelas beserta atribut dan <i>method</i> -nya
3.		Menggambarkan sebuah <i>interface</i> beserta atribut dan <i>method</i> -nya

Activity Diagram (UML 1.3)		
No	Gambar	Keterangan
1.		Menandakan dimulainya aktivitas pada sebuah sistem.
2.		Menandakan aktivitas apa yang akan dilakukan oleh pengguna aplikasi.
3.		Menandakan akhir aliran proses sistem

DAFTAR ALGORITMA

Algoritma 4.1 Proses penghitungan waktu perubahan terakhir pada direktori.....	38
Algoritma 4.2 Proses pengubahan kode sumber berdasarkan log.....	38