

ABSTRAK

Praktikum pemrograman dasar bahasa C merupakan praktikum yang wajib diikuti oleh seluruh mahasiswa. Dalam pengerjaan praktikum, mahasiswa mencari sebuah penyelesaian masalah yang diberikan oleh dosen. Dalam pencarian sebuah penyelesaian masalah, terdapat banyak solusi yang dapat digunakan untuk menyelesaikan masalah. Pengelompokan perlu dilakukan untuk mencari berapa kemungkinan solusi yang dapat digunakan dalam satu penyelesaian masalah. Menggunakan algoritma *running karp-rabin greedy string tiling* akan dicari tabel kemiripan dari kumpulan hasil praktikum mahasiswa. Dari tabel kemiripan yang terbentuk, maka akan dapat terlihat kelompok-kelompok mahasiswa dalam menyelesaikan sebuah permasalahan. Banyaknya kelompok yang akan terbentuk bergantung kepada banyaknya sampel *file* yang digunakan dan kompleksitas soal yang diberikan. Semakin banyak sampel *file* dan semakin kompleks soal yang diberikan, menghasilkan kelompok yang lebih beragam. Jumlah data yang diuji sebanyak delapan kasus, tigapuluh sampel *file*, dan satu sampel kelas.

Kata kunci : *running karp-rabin greedy string tiling, threshold clustering, problem solving's solutions, string similarity*

ABSTRACT

Practical C programming language is a practical basis which shall be followed by all students. In fulfillment of the practicum, student searching a solution to the problem solving which given by lecturer. In the search for a settlement of the problem, there are many solutions that can be used to solve the problem. Clustering should be done to look at how the potential solutions that can be used in a problem solving. Using running karp-rabin greedy string tiling algorithm to search the table of similarity from the set of student's practicum result. From the similarity's table that is formed, it will show the groups of students in solving a problem. The number of groups that will be formed depends on the amount of sample files used and the complexity of the given problem. The more sample files and increasingly complex given problem, generating a more diverse group. The amount of data that examined is eight cases, thirty sample file, and a sample class.

Keyword : running karp-rabin greedy string tiling algorithm, threshold clustering, problem solving's solutions, string similarity

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PRAKATA	ii
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN	iv
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN.....	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xvii
DAFTAR NOTASI/ LAMBANG	xviii
DAFTAR SINGKATAN	xx
DAFTAR ISTILAH.....	xxi
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Pembahasan.....	2
1.4 Batasan Masalah.....	3
1.5 Sistematika Pembahasan.....	4
BAB 2. KAJIAN TEORI.....	6
2.1 Algoritma <i>Running Karp-Rabin Greedy String Tiling</i>	6
2.2 <i>Hashing</i>	9
2.3 K-gram.....	10
2.4 Similaritas	10
2.5 <i>Lexical Analyzer</i>	11
2.6 Token.....	11
2.7 <i>Finite State Automata</i>	14
2.8 Bahasa C	15
2.9 Bahasa Java	16
2.10 UML	16
2.11 MinGW	17
BAB 3. ANALISIS DAN DISAIN	18

3.1	Analisis	18
3.2	Gambaran Keseluruhan	19
3.2.1	Persyaratan Antarmuka Eksternal	19
3.2.2	Antarmuka dengan Pengguna.....	19
3.2.3	Antarmuka Perangkat Keras.....	21
3.2.4	Antarmuka Perangkat Lunak.....	21
3.2.5	Fitur – fitur Produk Perangkat Lunak	22
3.3	Disain Perangkat Lunak.....	28
3.3.2	Disain Penyimpanan Data.....	69
3.3.3	Disain Antarmuka.....	70
BAB 4.	PENGEMBANGAN PERANGKAT LUNAK.....	75
4.1	Implementasi <i>Class</i> /Modul	75
4.1.1	<i>Class</i> Koneksi.....	75
4.1.2	<i>Class</i> TextFieldLimiter	75
4.1.3	<i>Class</i> FindFile.....	76
4.1.4	<i>Class</i> CompileFile	78
4.1.5	<i>Class</i> CheckingInputOutput	82
4.1.6	<i>Class</i> ReadingFile.....	86
4.1.7	<i>Class</i> CekToken.....	87
4.1.8	<i>Class</i> Filtering	92
4.1.9	<i>Class</i> FiniteStateAutomata.....	93
4.1.10	<i>Class</i> Conversion	97
4.1.11	<i>Class</i> HashTable	101
4.1.12	<i>Class</i> MatchTiles	102
4.1.13	<i>Class</i> Pattern.....	103
4.1.14	<i>Class</i> KarpRabin.....	103
4.1.15	<i>Class</i> Similaritas	109
4.1.16	<i>Class</i> SubCategorization.....	111
4.1.17	<i>Class</i> Categorization.....	111
4.2	Implementasi Penyimpanan Data.....	113
4.2.1	Tabel Hasil	113
4.3	Implementasi Antarmuka.....	114

4.3.1	Tampilan Halaman Utama	114
4.3.2	Tampilan <i>Show</i> Seluruh Analisis.....	115
4.3.3	Tampilan Analisis Satu Kelas	116
4.3.4	Tampilan Analisis Satu Mahasiswa.....	117
4.3.5	Tampilan Analisis Satu Kasus Satu Kelas	118
4.3.6	Tampilan Analisis Satu Topik.....	119
4.3.7	Tampilan Analisis Satu Topik Satu Kelas	120
4.3.8	Tampilan Halaman Petunjuk.....	121
BAB 5.	TESTING DAN EVALUASI SISTEM.....	122
5.1	Rencana Pengujian	122
5.2	Pelaksanaan Pengujian	122
5.2.1	White Box	122
5.2.2	Pengujian Dengan Kasus	129
5.2.3	Kesimpulan Pengujian Kasus.....	159
BAB 6.	KESIMPULAN DAN SARAN.....	161
6.1	Kesimpulan	161
6.2	Saran.....	163
DAFTAR PUSTAKA		164
RIWAYAT HIDUP PENULIS.....		166

DAFTAR GAMBAR

Gambar 2.1 Algoritma <i>scanpattern</i>	8
Gambar 2.2 Algoritma <i>Running Karp-Rabin Greedy String Tiling</i>	9
Gambar 2.3 Perhitungan similaritas.....	10
Gambar 2.4 Daftar kata kunci dalam bahasa C	12
Gambar 2.5 Daftar penanda escape dalam bahasa C	13
Gambar 2.6 Contoh penulisan <i>finite state automata</i>	15
Gambar 3.1 <i>Use case</i> aplikasi pengkategorisasian bahasa C	28
Gambar 3.2 <i>Activity diagram</i> masukan folder direktori	29
Gambar 3.3 <i>Activity diagram</i> mencari <i>file</i>	31
Gambar 3.4 <i>Activity diagram</i> kompilasi <i>file</i>	33
Gambar 3.5 <i>Activity diagram</i> membaca <i>file</i>	34
Gambar 3.6 <i>Activity diagram</i> cek <i>input</i> dan <i>output</i>	36
Gambar 3.7 <i>Activity diagram</i> proses.....	37
Gambar 3.8 <i>Activity diagram</i> konversi kode program menjadi token	38
Gambar 3.9 <i>Activity diagram</i> pengkonversian kode.....	39
Gambar 3.10 <i>Activity diagram</i> pencocokan	41
Gambar 3.11 <i>Activity diagram</i> pengukuran kemiripan.....	42
Gambar 3.12 <i>Activity diagram</i> pengelompokan	43
Gambar 3.13 <i>Activity diagram</i> menyimpan analisis	44
Gambar 3.14 <i>Activity diagram</i> menampilkan analisis	45
Gambar 3.15 <i>Activity diagram</i> menampilkan analisis keseluruhan.....	46
Gambar 3.16 <i>Activity diagram</i> menampilkan analisis satu kelas	47
Gambar 3.17 <i>Activity diagram</i> menampilkan analisis satu mahasiswa	48
Gambar 3.18 <i>Activity diagram</i> menampilkan analisis satu kelas satu kasus	49
Gambar 3.19 <i>Activity diagram</i> menampilkan analisis satu topik	50
Gambar 3.20 <i>Activity diagram</i> menampilkan analisis satu topik satu kelas	51
Gambar 3.21 <i>Activity diagram</i> <i>print</i>	52
Gambar 3.22 <i>Class diagram</i>	53
Gambar 3.23 <i>Class diagram</i> pada <i>package</i> <i>utility</i>	54
Gambar 3.24 <i>Class diagram</i> pada <i>package</i> <i>readingfile</i>	54
Gambar 3.25 <i>Class diagram</i> pada <i>package</i> <i>rabinkarp</i>	55

Gambar 3.26 <i>Class diagram</i> pada <i>package</i> fsa	56
Gambar 3.27 <i>Class diagram</i> pada <i>package</i> categorization.....	56
Gambar 3.28 FSA untuk konstanta <i>integer</i> pada bahasa C.....	57
Gambar 3.29 FSA untuk konstanta real pada bahasa C	57
Gambar 3.30 FSA untuk konstanta karakter pada bahasa C	58
Gambar 3.31 FSA untuk <i>string literals</i> pada bahasa C	58
Gambar 3.32 FSA untuk <i>keyword</i> dan <i>identifier</i> pada bahasa C.....	59
Gambar 3.33 FSA untuk operator pada bahasa C	59
Gambar 3.34 FSA untuk simbol pada bahasa C	60
Gambar 3.35 FSA untuk komentar pada bahasa C.....	60
Gambar 3.36 FSA Keseluruhan.....	61
Gambar 3.37 <i>String</i> kode program mahasiswa satu	62
Gambar 3.38 <i>String</i> kode program mahasiswa dua	62
Gambar 3.39 Hasil setelah proses <i>filtering string</i> kode program mahasiswa satu	63
Gambar 3.40 Hasil setelah proses <i>filtering string</i> kode program mahasiswa dua.	63
Gambar 3.41 <i>Dice Similarity Coeficiens</i>	68
Gambar 3.42 ERD.....	69
Gambar 3.43 <i>User interface</i> halaman utama.....	70
Gambar 3.44 <i>User interface</i> menu <i>show</i> seluruh analisis	71
Gambar 3.45 <i>User interface</i> menu analisis satu kelas	72
Gambar 3.46 <i>User interface</i> menu analisis satu mahasiswa	73
Gambar 3.47 <i>User interface</i> menu analisis <i>problem solving</i>	74
Gambar 4.1 Implementasi <i>class</i> Koneksi.....	75
Gambar 4.2 Implementasi <i>class</i> TextFieldLimiter	76
Gambar 4.3 Implementasi <i>class</i> FindFile.....	76
Gambar 4.4 Implementasi <i>method</i> getAllFileC.....	77
Gambar 4.5 Implementasi <i>class</i> CompileFile	78
Gambar 4.6 Implementasi <i>method</i> compilingFile	79
Gambar 4.7 Implementasi <i>method</i> compileCFile	79
Gambar 4.8 Implementasi <i>method</i> getAllFileHasCompile	80
Gambar 4.9 Implementasi <i>method</i> getFileHasFailedCompile	81
Gambar 4.10 Implementasi <i>class</i> CheckingInputOutput	82

Gambar 4.11 Implementasi <i>method</i> getFileHasCorrectAnswers	83
Gambar 4.12 Implementasi <i>method</i> getFileHasWrongAnswers.....	84
Gambar 4.13 Implementasi <i>method</i> getInputOutput.....	84
Gambar 4.14 Implementasi <i>method</i> getFileInputOutput.....	85
Gambar 4.15 Implementasi <i>class</i> ReadingFile	86
Gambar 4.16 Implementasi <i>method</i> readTheFile	86
Gambar 4.17 Implementasi <i>class</i> CekToken.....	88
Gambar 4.18 Implementasi <i>method</i> isAlphabet	88
Gambar 4.19 Implementasi <i>method</i> isNumber	89
Gambar 4.20 Implementasi <i>method</i> isAlphabetNumber	89
Gambar 4.21 Implementasi <i>method</i> isOperator.....	90
Gambar 4.22 Implementasi <i>method</i> isSymbol	90
Gambar 4.23 Implementasi <i>method</i> isSeparator.....	91
Gambar 4.24 Implementasi <i>method</i> isKeyword	91
Gambar 4.25 Implementasi <i>class</i> Filtering	92
Gambar 4.26 Implementasi <i>method</i> splitKodeProgram.....	93
Gambar 4.27 Implementasi <i>class</i> FiniteStateAutomata.....	94
Gambar 4.28 Implementasi <i>method</i> generateFSA (1)	95
Gambar 4.29 Implementasi <i>method</i> generateFSA (2)	96
Gambar 4.30 Implementasi <i>method</i> getListFSA	97
Gambar 4.31 Implementasi <i>class</i> Conversion	98
Gambar 4.32 Implementasi <i>method</i> convertForIntoWhile.....	99
Gambar 4.33 Implementasi <i>method</i> convertSwitchIntoIf	100
Gambar 4.34 Implementasi <i>class</i> HashTable	101
Gambar 4.35 Implementasi <i>method</i> add	101
Gambar 4.36 Implementasi <i>method</i> get	102
Gambar 4.37 Implementasi <i>class</i> MatchTiles	102
Gambar 4.38 Implementasi <i>class</i> Pattern.....	103
Gambar 4.39 Implementasi <i>class</i> KarpRabin.....	104
Gambar 4.40 Implementasi <i>method</i> scanPattern	105
Gambar 4.41 Implementasi <i>method</i> distToNextTile.....	106
Gambar 4.42 Implementasi <i>method</i> jumpNextUnmarkedTile	107

Gambar 4.43 Implementasi <i>method</i> generateHashValue	107
Gambar 4.44 Implementasi <i>method</i> markString	108
Gambar 4.45 Implementasi <i>method</i> isHavingTheSameMatch.....	108
Gambar 4.46 Implementasi <i>method</i> runningRabinKarpGreedyStringTiling	109
Gambar 4.47 Implementasi <i>class</i> Similaritas	110
Gambar 4.48 Implementasi <i>method</i> calculateSimilarity	110
Gambar 4.49 Implementasi <i>method</i> getLengMatches	110
Gambar 4.50 Implementasi <i>class</i> SubCategorization	111
Gambar 4.51 Implementasi <i>class</i> Categorization	112
Gambar 4.52 Implementasi <i>method</i> generateCategorization	112
Gambar 4.53 Implementasi <i>method</i> getSimilarityMatriks	113
Gambar 4.54 Tampilan Halaman Utama	115
Gambar 4.55 Tampilan <i>Show</i> Seluruh Analisis.....	116
Gambar 4.56 Tampilan Analisis Satu Kelas	117
Gambar 4.57 Tampilan Analisis Satu Mahasiswa	118
Gambar 4.58 Tampilan Analisis Satu Kasus Satu Kelas	119
Gambar 4.59 Tampilan Analisis Satu Topik	120
Gambar 4.60 Tampilan Analisis Satu Topik Satu Kelas	121
Gambar 4.61 Tampilan Halaman Petunjuk	121
Gambar 5.1 Kode program kasus tanpa <i>input user</i> untuk kategori satu	130
Gambar 5.2 Kode program kasus tanpa <i>input user</i> untuk kategori dua	130
Gambar 5.3 Kode program kasus tanpa <i>input user</i> untuk kategori tiga.....	130
Gambar 5.4 Kode program kasus dua <i>input</i> dan dua <i>output</i> untuk kategori satu	131
Gambar 5.5 Kode program kasus dua <i>input</i> dan dua <i>output</i> untuk kategori dua	132
Gambar 5.6 Kode program kasus dua <i>input</i> dan dua <i>output</i> untuk kategori tiga	132
Gambar 5.7 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori satu ...	134
Gambar 5.8 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori dua....	134
Gambar 5.9 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori tiga....	135
Gambar 5.10 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori empat.....	135
Gambar 5.11 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori lima.	136
Gambar 5.12 Kode program kasus satu <i>input</i> dan <i>banyak</i> output kategori enam	136

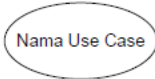
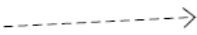
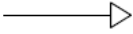
Gambar 5.13 Kode program kasus satu <i>input</i> dan banyak <i>output</i> kategori tujuh	137
Gambar 5.14 Kode program kasus satu <i>input</i> dan banyak <i>output</i> kategori delapan	137
Gambar 5.15 Kode program kasus satu <i>input</i> dan banyak <i>output</i> kategori sembilan	138
Gambar 5.16 Kode program kasus percabangan untuk kategori satu.....	139
Gambar 5.17 Kode program kasus percabangan untuk kategori dua	140
Gambar 5.18 Kode program kasus percabangan untuk kategori tiga	140
Gambar 5.19 Kode program kasus percabangan untuk kategori empat	141
Gambar 5.20 Kode program kasus percabangan untuk kategori lima	141
Gambar 5.21 Kode program kasus percabangan untuk kategori enam	142
Gambar 5.22 Kode program kasus percabangan untuk kategori tujuh.....	142
Gambar 5.23 Kode program kasus pengulangan (<i>while</i>) untuk kategori satu	144
Gambar 5.24 Kode program kasus pengulangan (<i>while</i>) untuk kategori dua	144
Gambar 5.25 Kode program kasus pengulangan (<i>while</i>) untuk kategori tiga	145
Gambar 5.26 Kode program kasus pengulangan (<i>while</i>) untuk kategori empat .	145
Gambar 5.27 Kode program kasus pengulangan (<i>while</i>) untuk kategori lima.....	146
Gambar 5.28 Kode program kasus pengulangan (<i>for</i>) untuk kategori satu	147
Gambar 5.29 Kode program kasus pengulangan (<i>for</i>) untuk kategori dua	148
Gambar 5.30 Kode program kasus pengulangan (<i>for</i>) untuk kategori tiga.....	148
Gambar 5.31 Kode program kasus pengulangan (<i>for</i>) untuk kategori empat	149
Gambar 5.32 Kode program kasus <i>array</i> satu dimensi untuk kategori satu	150
Gambar 5.33 Kode program kasus <i>array</i> satu dimensi untuk kategori dua.....	151
Gambar 5.34 Kode program kasus <i>array</i> satu dimensi untuk kategori tiga.....	151
Gambar 5.35 Kode program kasus <i>array</i> satu dimensi untuk kategori empat....	152
Gambar 5.36 Kode program kasus <i>array</i> satu dimensi untuk kategori lima.....	152
Gambar 5.37 Kode program kasus <i>array</i> satu dimensi untuk kategori enam.....	153
Gambar 5.38 Kode program kasus <i>array</i> satu dimensi untuk kategori tujuh	153
Gambar 5.39 Kode program kasus <i>array</i> dua dimensi untuk kategori satu.....	155
Gambar 5.40 Kode program kasus <i>array</i> dua dimensi untuk kategori dua	155
Gambar 5.41 Kode program kasus <i>array</i> dua dimensi untuk kategori tiga	156
Gambar 5.42 Kode program kasus <i>array</i> dua dimensi untuk kategori empat.....	156

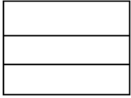
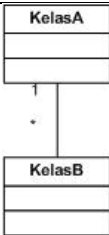
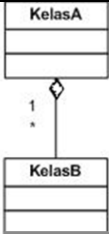
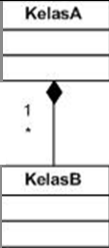
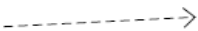
Gambar 5.43 Kode program kasus <i>array</i> dua dimensi untuk kategori lima	157
Gambar 5.44 Kode program kasus <i>array</i> dua dimensi untuk kategori enam.....	157
Gambar 5.45 Kode program kasus <i>array</i> dua dimensi untuk kategori tujuh	158
Gambar 5.46 Kode program kasus <i>array</i> dua dimensi untuk kategori delapan ..	158
Gambar 5.47 Kode program kasus <i>array</i> dua dimensi untuk kategori sembilan	159

DAFTAR TABEL

Tabel 3.1 Tabel kumpulan token dan nilai <i>hashing</i> pada mahasiswa satu.....	64
Tabel 3.2 Tabel kumpulan token pada mahasiswa dua.....	65
Tabel 3.3 Tabel kumpulan token yang memiliki kesamaan.....	67
Tabel 4.1 Implementasi tabel hasil	113
Tabel 5.1 Tabel pengujian <i>class</i> CekToken	122
Tabel 5.2 Tabel pengujian <i>class</i> Conversion.....	123
Tabel 5.3 Tabel pengujian <i>class</i> Filtering	124
Tabel 5.4 Tabel pengujian <i>class</i> FiniteStateAutomata	126
Tabel 5.5 Tabel pengujian <i>class</i> MatchTiles.....	126
Tabel 5.6 Tabel pengujian <i>class</i> Pattern	127
Tabel 5.7 Tabel pengujian <i>class</i> Similaritas.....	127
Tabel 5.8 Tabel pengujian <i>class</i> Categorization.....	128
Tabel 5.9 Tabel pengujian <i>class</i> SubCategorization.....	128
Tabel 5.10 Tabel hasil pengujian kasus tanpa <i>input user</i>	129
Tabel 5.11 Tabel hasil pengujian kasus dengan dua <i>input</i> dan dua <i>output</i>	131
Tabel 5.12 Tabel hasil pengujian kasus dengan satu <i>input</i> dan banyak <i>output</i> ...	133
Tabel 5.13 Tabel hasil pengujian kasus percabangan.....	138
Tabel 5.14 Tabel hasil pengujian kasus pengulangan dengan <i>while</i>	143
Tabel 5.15 Tabel hasil pengujian kasus pengulangan dengan <i>for</i>	147
Tabel 5.16 Tabel hasil pengujian kasus <i>array</i> satu dimensi	149
Tabel 5.17 Tabel hasil pengujian kasus <i>array</i> dua dimensi	154

DAFTAR NOTASI/ LAMBANG

Notasi/Lambang	Nama	Arti
	<i>Actor</i>	Pengguna yang melakukan interaksi langsung dengan sistem, bisa manusia, aplikasi atau sistem, ataupun objek lain.
	<i>Use Case</i>	Digambarkan dengan lingkaran elips dengan nama <i>use case</i> berada ditengah lingkaran elips.
	<i>Association</i>	Penghubung <i>actor</i> dengan <i>use case</i> .
	<i>Include</i>	Terjadi jika terdapat perilaku yang serupa di lebih dari satu <i>use case</i> .
	<i>Extend</i>	Menambahkan perilaku <i>use case</i> pada <i>use case</i> dasar.
	<i>Initial State</i>	Titik awal untuk memulai aktifitas.
	<i>Final State</i>	Titik akhir untuk mengakhiri aktifitas.
	<i>Activity Status</i>	Menandakan status dari aktifitas setelah melakukan aktifitas.
	<i>Activity</i>	Menandakan aktivitas dalam aplikasi.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.
	<i>Fork/Join</i>	Menunjukkan kegiatan menggabungkan dua panel aktifitas menjadi satu atau sebaliknya memisahkan satu panel menjadi dua panel aktifitas.
	<i>Control Flow</i>	Arus aktifitas.

Notasi/Lambang	Nama	Arti
	<i>Note</i>	Catatan khusus untuk sebuah aktifitas.
	<i>Class</i>	Blok-blok pembangun program. Bagian atas <i>class</i> menunjukkan nama dari <i>class</i> , bagian tengah mendefinisikan atribut dari <i>class</i> , dan bagian bawah mendefinisikan <i>method</i> dari sebuah <i>class</i> .
	<i>Association</i>	Menunjukkan hubungan antar kelas yang tidak saling bergantung.
	<i>Agregation</i>	Ketika sebuah <i>class</i> A mengagregasi <i>class</i> B, maka objek A tidak dapat ada tanpa objek B, namun objek B dapat ada tanpa kehadiran objek A.
	<i>Composition</i>	Ketika sebuah <i>class</i> A mengkompisiasi <i>class</i> B, maka objek A tidak dapat ada tanpa objek B, dan objek B tidak dapat ada tanpa kehadiran objek A.
	<i>Dependency</i>	Menunjukkan ketergantungan antara satu <i>class</i> dengan <i>class</i> yang lain.
	<i>Generalization</i>	Menunjukkan <i>inheritance</i> dari satu <i>class</i> ke <i>class</i> lain.

Sumber : [1]

DAFTAR SINGKATAN

ERD	=	<i>Entity Relational Database</i>
FSA	=	<i>Finite State Automata</i>
GNU	=	<i>Gnu's Not Unix</i>
UML	=	<i>Undefined Model Language</i>

DAFTAR ISTILAH

<i>Activity diagram</i>	= Menjelaskan urutan kegiatan yang mendukung kegiatan kondisional maupun paralel.
Algoritma	= Suatu langkah atau aturan yang tepat dalam mendefinisikan urutan operasi sehingga setiap aturan menjadi lebih efektif dan jelas.
<i>Class diagram</i>	= Menjelaskan jenis objek yang terdapat dalam sistem dan berbagai macam hubungan statis yang terbentuk antar objek tersebut.
<i>Clustering</i>	= Pengelompokan data ke dalam suatu kelompok berdasarkan parameter tertentu, dalam satu kelompok anggota kelompok tersebut memiliki tingkat kemiripan yang tinggi.
<i>Compiler</i>	= Sebuah program komputer yang berguna untuk menerjemahkan program komputer yang ditulis menggunakan bahasa pemrograman tertentu menjadi program yang ditulis dalam bahasa pemrograman lain.
<i>File</i>	= Urutan data yang berisi mengenai informasi tertentu, yang memiliki atribut seperti nama <i>file</i> , ukuran <i>file</i> , ekstensi <i>file</i> dan lain sebagainya.
<i>Finite State Automata</i>	= Merupakan model matematika yang dapat menerima masukan dan mengeluarkan keluaran. Model ini memiliki <i>state</i> dalam jumlah tertentu dan dapat berpindah dari satu <i>state</i> ke <i>state</i> yang lain sesuai dengan masukan dan fungsi transisi. Model ini biasa digunakan dalam bahasa mesin.
<i>Hashing</i>	= Suatu cara untuk mentransformasi sebuah <i>string</i> menjadi suatu nilai yang unik dengan panjang tertentu (<i>fixed-length</i>) yang berfungsi sebagai penanda <i>string</i> tersebut.

<i>Lexical analyzer</i>	=	Proses mengubah kata – kata menjadi token - token.
<i>Pattern</i>	=	Merupakan pola dari suatu teks atau kalimat dengan panjang tertentu.
<i>Threshold</i>	=	Merupakan nilai yang menjadi titik acuan untuk parameter.
Token	=	sebuah unit terkecil atau elemen mendasar bahasa komputer dimana unit tersebut tidak dapat dibagi lagi.
<i>User interface</i>	=	Gambaran mengenai tampilan aplikasi atau sistem yang akan dibuat.
<i>White Box</i>	=	pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol disain program.