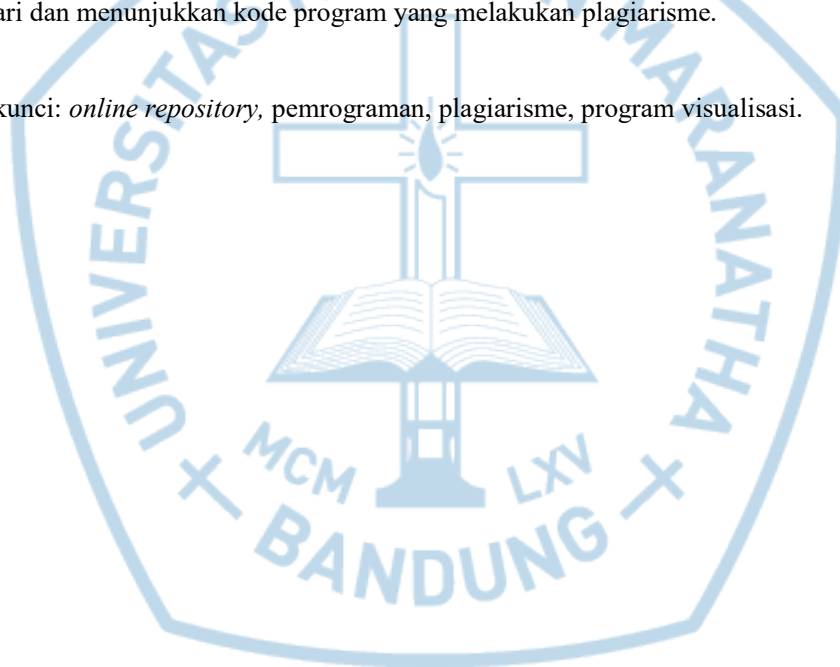


ABSTRAK

Membuat sebuah program bukan hal mudah, terutama untuk mereka yang baru pertama kali mempelajari pemrograman. Sebagian dari mereka akan mengatasi masalah itu dengan melakukan plagiarisme kode sumber program orang lain untuk keuntungan pribadi atau mendapat nilai yang baik dalam mata kuliah pemrograman jika orang yang melakukan adalah mahasiswa. Oleh karena itu, diperlukan sebuah metode alternatif untuk mengatasi masalah tersebut, sebagai contoh dengan mengembangkan sebuah aplikasi visualisasi kesamaan kode sumber program. Dengan menggunakan aplikasi visualisasi kesamaan kode sumber program dengan *online repository*, pengguna dapat lebih terbantu dalam mencari kode sumber program asli yang diplagiarisme dan melihat kesamaan diantara dua kode sumber program tersebut. Aplikasi *Plagiarism Observer* (Plago) dapat mencari kode sumber program yang persentase kemiripannya paling besar dari *repository* dan menyelesaikan plagiarisme antara 2 kode sumber program dengan cepat dan tepat selama pengguna memasukan tahap-tahap sesuai dengan kebutuhan aplikasi Plago. Untuk melakukan visualisasi, pengguna harus mengupload data kode sumber yang dideteksi melakukan plagiarisme. Berdasarkan hasil pengujian, aplikasi ini dianggap mampu mencari dan menunjukkan kode program yang melakukan plagiarisme.

Kata kunci: *online repository*, pemrograman, plagiarisme, program visualisasi.



ABSTRACT

Creating a program is not easy, especially for those who are learning programming for the first time. Some of them will overcome the problem by plagiarism the source code of other people's programs for personal gain or get good grades in programming courses if the person doing it is a student. Therefore, an alternative method is needed to overcome this problem, for example by developing an application to visualize the similarity of program source code. By using the application visualization of program source code similarity with the online repository, users can be more helped in finding the original program source code that is plagiarism and see the similarities between the two source code of the program. The Plagiarism Observer (Plago) application can search for the program source code which has the largest percentage of similarities from the repository and resolve plagiarism between 2 source code programs quickly and precisely as long as the user enters stages according to the needs of the Plago application. To visualize, users must upload source code data that is detected as plagiarism. Based on the test results, this application is considered capable of finding and showing program code that performs plagiarism.

Keywords: *online repository, programming, plagiarism, visualization program.*



DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN.....	iii
PRAKATA	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR <i>TABLE</i>	xv
DAFTAR SINGKATAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Pembahasan	3
1.4 Ruang Lingkup.....	3
1.5 Sumber Data	3
1.6 Sistematika Penyajian	4
BAB 2 LANDASAN TEORI.....	5
2.1 Plagiarisme	5
2.2 <i>Online Repository</i>	6
2.3 <i>Search Engine</i>	7
2.3.1 <i>Information Retrieval</i>	8
2.4 RKR GST (<i>Running-Karp-Rabin Greedy-String-Tiling</i>).....	9
2.5 <i>Abstract Syntax Tree</i>	10

2.6	Kakas Kesamaan Kode Sumber.....	14
2.7	<i>ACE</i>	16
2.8	Penelitian Terkait	18
BAB 3 ANALISIS DAN RANCANGAN SISTEM		23
3.1	Analisis	23
3.1.1	Penentuan <i>Abstract Syntax Tree Plagiarism</i>	24
3.2	Gambaran Aplikasi.....	31
3.3	Pemodelan Perangkat Lunak	32
3.3.1	<i>Use Case Diagram</i>	32
3.3.2	<i>Use Case Scenario</i>	32
3.3.3	<i>Activity diagram</i> Aplikasi	33
3.4	Rancangan Aplikasi	37
BAB 4 IMPLEMENTASI.....		41
4.1	Class Diagram Algoritma	41
4.2	Rancangan Skenario Pengujian	42
4.3	Implementasi Tampilan Antar Muka	46
4.3.1	Implementasi <i>Upload</i> Kode Program Ke dalam <i>Repository</i>	49
4.3.2	Implementasi Compare Kode Program	52
4.4	Implementasi Fitur Compare dengan Highlight	57
4.4.1	Implementasi RKRGS.....	57
4.4.2	Implementasi AST.....	62
4.4.3	Implementasi Highlight	67
4.5	Implementasi <i>Information Retrieval</i>	69
BAB 5 PENGUJIAN		70
5.1	Pengujian Black Box.....	70

5.2 Pengujian Efektivitas Aplikasi dalam Mendeteksi Kesamaan Kode Sumber	70
5.2.1 Dataset Pengujian	70
5.2.2 Skenario Satu	80
5.2.3 Skenario Dua	82
5.2.4 Skenario Tiga	84
5.2.5 Skenario Empat	84
5.2.6 Skenario Lima	88
BAB 6 KESIMPULAN DAN SARAN	91
6.1 Kesimpulan	91
6.2 Saran	92
Daftar pustaka	93
RIWAYAT HIDUP PENULIS	96



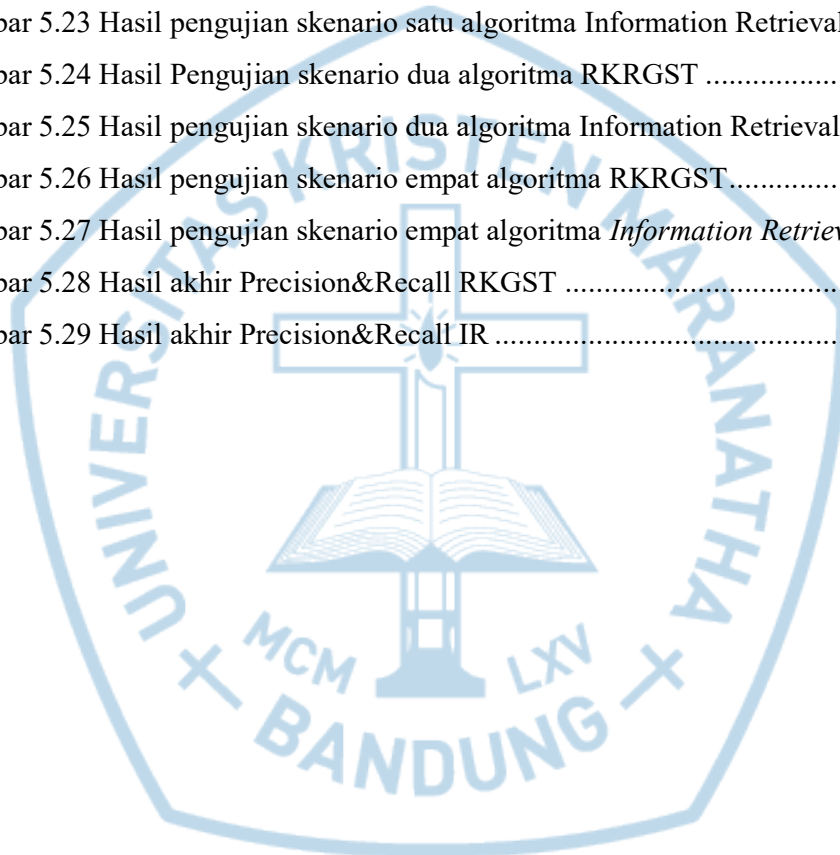
DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>GitHub</i>	6
Gambar 2.2 Tampilan <i>SourceTree</i>	7
Gambar 2.3 Arsitektur <i>Information Retrieval</i>	8
Gambar 2.4 Rumus pengukuran <i>Precision</i> dan <i>Recall</i>	9
Gambar 2.5 Algoritma RKR GST.....	10
Gambar 2.6 Kode Program <i>Abstract Syntax Tree</i>	11
Gambar 2.7 <i>Abstract Syntax Tree</i> untuk kondisi “if” dan “else”.....	11
Gambar 2.8 <i>Abstract Syntax Tree</i> untuk kondisi “for”.....	12
Gambar 2.9 <i>Abstract Syntax Tree</i> untuk kondisi “nested if”.....	12
Gambar 2.10 Bentuk linearisasi <i>Inorder Abstract Syntax Tree</i>	13
Gambar 2.11 Bentuk linearisasi <i>Preorder Abstract Syntax Tree</i>	13
Gambar 2.12 Bentuk linearisasi <i>Postorder Abstract Syntax Tree</i>	14
Gambar 2.13 Visualisasi Kakas <i>Jplag</i>	15
Gambar 2.14 Visualisasi <i>MOSS</i>	15
Gambar 2.15 Kode Program <i>Ace</i>	16
Gambar 2.16 Arsitektur <i>Ace</i>	16
Gambar 2.17 Implementasi <i>highlighting</i>	17
Gambar 2.18 Implementasi <i>class css</i>	18
Gambar 2.19 Bentuk penyusunan tahap dataset [23].....	19
Gambar 2.20 Contoh susunan kode original, plagiarisme, dan non-plagiarisme [23]	20
Gambar 2.21 Pembagian level plagiarisme [23].....	21
Gambar 3.1 Contoh kode sumber program original.....	24
Gambar 3.2 Linearisasi pohon kode sumber program original.....	26
Gambar 3.3 Kode sumber program plagiarisme.....	26
Gambar 3.4 Linearisasi pohon kode sumber program plagiarisme.....	28
Gambar 3.5 Contoh kesamaan kode sumber original dengan kode sumber plagiarisme.....	28
Gambar 3.6 Kode sumber program tidak plagiarisme.....	29
Gambar 3.7 Linearisasi pohon kode sumber program tidak plagiarisme.....	30

Gambar 3.8 Contoh kesamaan kode sumber original dengan kode sumber tidak plagiarisme	31
Gambar 3.9 Use Case Diagram	32
Gambar 3.10 <i>Activity diagram Login</i>	34
Gambar 3.11 <i>Activity Diagram Upload data ke repository</i>	34
Gambar 3.12 <i>Activity Diagram Upload data yang ingin dicari plagiarismenya</i> ...	35
Gambar 3.13 <i>Activity Diagram Pencarian kode sumber ke repository</i>	35
Gambar 3.14 <i>Activity Diagram Proses Visualisasi</i>	36
Gambar 3.15 <i>Activity Diagram Logout</i>	37
Gambar 3.16 Rancangan Tampilan Login	38
Gambar 3.17 Rancangan Tampilan Utama	38
Gambar 3.18 Rancangan Tampilan Pencarian Plagiarisme	39
Gambar 3.19 Rancangan Tampilan Input ke Repository.....	40
Gambar 4.1 Class Diagram RKRGSST	41
Gambar 4.2 Class Diagram AST	41
Gambar 4.3 Tampilan Login	46
Gambar 4.4 Tampilan Pengguna/ <i>User</i>	47
Gambar 4.5 Tampilan <i>Database</i>	48
Gambar 4.6 Tampilan Compare	48
Gambar 4.7 Tampilan Nama Tempat Berkas/ <i>Folder</i> teriisi.....	49
Gambar 4.8 Tampilan Tempat Berkas/ <i>Folder</i> terbuat.....	50
Gambar 4.9 Kode Program Membuat berkas/ <i>folder</i>	50
Gambar 4.10 Tampilan Pemilihan Data Kode Sumber yang Dipilih	51
Gambar 4.11 Tampilan Database Setelah <i>Upload</i>	51
Gambar 4.12 Kode Program Upload	52
Gambar 4.13 Tampilan Pemilihan Data Kode Sumber yang Dipilih	53
Gambar 4.14 Proses pemilihan bahasa pemrograman	53
Gambar 4.15 Tampilan Proses Search RKRGSST	54
Gambar 4.16 Tampilan proses Highlight RKRGSST	54
Gambar 4.17 Tampilan Proses Search Struktural.....	55
Gambar 4.18 Proses Highlight Struktural	55
Gambar 4.19 Kode sumber pencarian plagiarisme ke <i>repository</i>	56

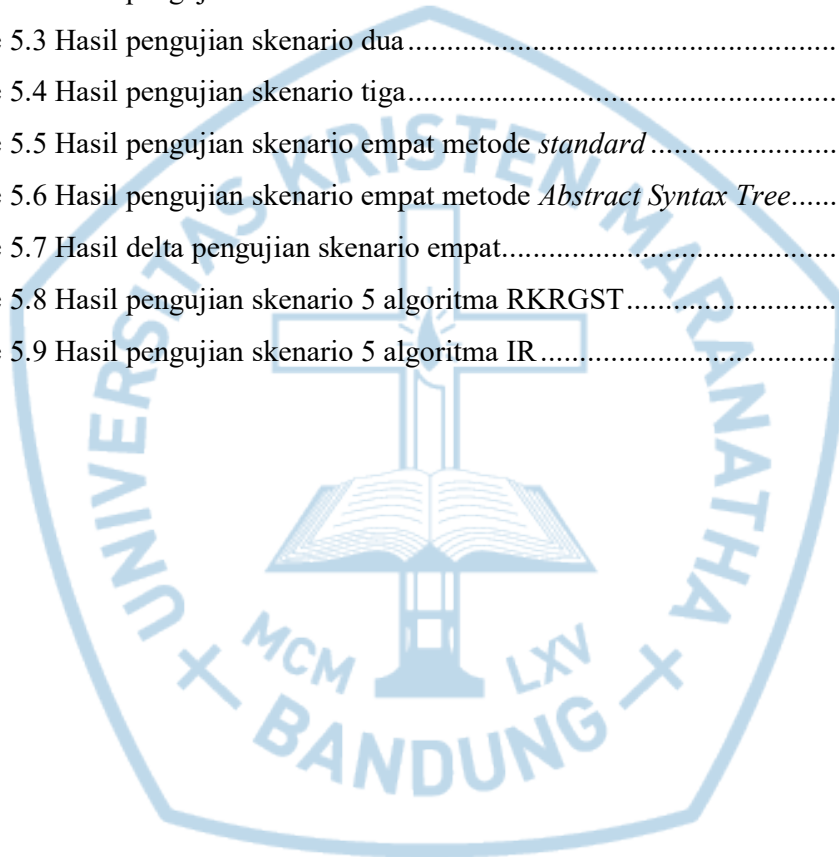
Gambar 4.20 Kode sumber <i>Highlight</i>	56
Gambar 4.21 Kode program TokenTuple.....	57
Gambar 4.22 Kode program TokenReader C++.....	58
Gambar 4.23 Kode program MatchTuple.....	58
Gambar 4.24 Implementasi RKRGS pada C++.....	59
Gambar 4.25 Implementasi RKRGS Pada Python.....	60
Gambar 4.26 Implementasi RKRGS pada Java.....	61
Gambar 4.27 Contoh fungsi konversi kode program Java ke dalam bentuk <i>tree</i> .	62
Gambar 4.28 Kode program MatchTuple.....	63
Gambar 4.29 Implementasi AST pada C++.....	64
Gambar 4.30 Kode konversi <i>tree</i> C++.....	64
Gambar 4.31 Implementasi AST pada Python.....	65
Gambar 4.32 Kode konversi <i>tree</i> Python.....	65
Gambar 4.33 Implementasi AST pada Java.....	66
Gambar 4.34 Kode konversi <i>tree</i> Java.....	66
Gambar 4.35 Implementasi <i>highlight</i> RKRGS.....	67
Gambar 4.36 Implementasi <i>highlight</i> AST.....	68
Gambar 4.37 Implementasi kode program <i>Information Retrieval</i>	69
Gambar 5.1 Soal Original Kasus 1.....	71
Gambar 5.2 Contoh soal plagiarisme Kasus 1.....	71
Gambar 5.3 Contoh soal non-plagiarisme Kasus 1.....	71
Gambar 5.4 Soal Original Kasus 2.....	72
Gambar 5.5 Contoh soal plagiarisme Kasus 2.....	72
Gambar 5.6 Contoh soal non-plagiarisme Kasus 2.....	72
Gambar 5.7 Soal Original Kasus 3.....	73
Gambar 5.8 Contoh soal plagiarisme Kasus 3.....	73
Gambar 5.9 Contoh soal non-plagiarisme Kasus 3.....	74
Gambar 5.10 Soal Original Kasus 4.....	74
Gambar 5.11 Contoh soal plagiarisme Kasus 4.....	75
Gambar 5.12 Contoh soal non-plagiarisme Kasus 4.....	75
Gambar 5.13 Soal Original Kasus 5.....	76
Gambar 5.14 Contoh soal plagiarisme Kasus 5.....	76

Gambar 5.15 Contoh soal non-plagiarisme Kasus 5	77
Gambar 5.16 Soal Original Kasus 6	77
Gambar 5.17 Contoh soal plagiarisme Kasus 6	78
Gambar 5.18 Contoh soal non-plagiarisme Kasus 6	78
Gambar 5.19 Soal Original Kasus 7	79
Gambar 5.20 Contoh soal plagiarisme Kasus 7	79
Gambar 5.21 Contoh soal non-plagiarisme Kasus 7	80
Gambar 5.22 Hasil Pengujian skenario satu algoritma RKGST.....	81
Gambar 5.23 Hasil pengujian skenario satu algoritma Information Retrieval.....	81
Gambar 5.24 Hasil Pengujian skenario dua algoritma RKGST	83
Gambar 5.25 Hasil pengujian skenario dua algoritma Information Retrieval	83
Gambar 5.26 Hasil pengujian skenario empat algoritma RKGST.....	86
Gambar 5.27 Hasil pengujian skenario empat algoritma <i>Information Retrieval</i> ..	86
Gambar 5.28 Hasil akhir Precision&Recall RKGST	89
Gambar 5.29 Hasil akhir Precision&Recall IR	89



DAFTAR TABLE

<i>Table 2.1</i> Jumlah kode plagiarisme setiap kasus [23]	22
<i>Table 4.1</i> Skenario 7 kasus.....	42
<i>Table 4.2</i> <i>Table</i> level plagiarisme perkasus	42
<i>Table 4.3</i> <i>Table</i> penggunaan dataset.....	43
<i>Table 5.1</i> Pengujian Black Box	70
<i>Table 5.2</i> Hasil pengujian skenario satu	81
<i>Table 5.3</i> Hasil pengujian skenario dua.....	82
<i>Table 5.4</i> Hasil pengujian skenario tiga.....	84
<i>Table 5.5</i> Hasil pengujian skenario empat metode <i>standard</i>	85
<i>Table 5.6</i> Hasil pengujian skenario empat metode <i>Abstract Syntax Tree</i>	85
<i>Table 5.7</i> Hasil delta pengujian skenario empat.....	85
<i>Table 5.8</i> Hasil pengujian skenario 5 algoritma RKRGS	88
<i>Table 5.9</i> Hasil pengujian skenario 5 algoritma IR.....	88



DAFTAR SINGKATAN

AST	<i>Abstract Syntax Tree</i>
RKR GST	<i>Running-Karp-Rabin Greedy-String-Tiling</i>
IR	<i>Information Retrieval</i>
JSON	<i>JavaScript Object Notation</i>

