

Deteksi Otomatis Plagiarisme *Source Code*

LAPORAN TUGAS AKHIR

Disusun sebagai syarat kelulusan tingkat sarjana

oleh :

CYNTHIA KUSTANTO / NIM. 13503066



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2007

Lembar Pengesahan Program Studi Sarjana Teknik Informatika

Deteksi Otomatis Plagiarisme *Source Code*

Tugas Akhir
Program Studi Sarjana Teknik Informatika ITB

Oleh
Cynthia Kustanto / 13503066

Telah disetujui dan disahkan sebagai laporan Tugas Akhir
di Bandung, pada tanggal Juni 2007

Pembimbing

Dr. Ir. M.M. Inggriani Liem
NIP 130796176

ABSTRAK

Detektor otomatis sangat membantu untuk menemukan praktek-praktek plagiarisme *source code*. Pada Tugas Akhir ini, telah dikembangkan prototipe perangkat lunak untuk mendeteksi plagiarisme *source code* secara otomatis. Detektor plagiarisme yang diberi nama **Deimos** tersebut dikembangkan dengan konteks untuk digunakan di kelas pemrograman dasar dengan jumlah mahasiswa mencapai ratusan.

Metode deteksi yang diimplementasikan pada **Deimos** adalah metode berbasis struktur karena *source code* adalah teks yang mempunyai struktur sesuai bahasa pemrogramannya. Terdapat dua proses sekuensial pada metode tersebut, yaitu mengubah *source code* menjadi representasi linier yang disebut *token string*, kemudian membandingkan setiap *token string* satu sama lain.

Untuk mengubah *source code* menjadi *token string*, diimplementasikan *scanner* dan *parser* yang sesuai dengan bahasa pemrograman target. Sedangkan pada perbandingan, algoritma yang dipakai adalah algoritma *Running Karp-Rabin Greedy String Tiling*. Algoritma tersebut mampu menemukan bagian-bagian yang identik pada dua buah *string* tanpa terpengaruh urutan maupun posisi *substring*, dan mempunyai parameter yang dapat merepresentasikan sensitivitas deteksi. Untuk setiap perbandingan dua buah program, nilai similaritas dihitung dan bagian-bagian *source code* yang diduga merupakan hasil praktek plagiarisme ditandai.

Setelah proses deteksi selesai, pengguna dapat melihat hasil deteksi dan menganalisis terjadinya kasus-kasus plagiarisme berdasarkan hasil deteksi tersebut. Disarankan agar hasil deteksi tidak menjadi keputusan final untuk penentuan kasus-kasus plagiarisme, melainkan hanya sebagai bahan pertimbangan. Keputusan final tetap ditentukan oleh pemeriksa, karena yang mengetahui kriteria otentisitas suatu *source code* adalah pemeriksa.

Deimos dibangun berdasarkan prinsip di atas dengan menggunakan bahasa pemrograman **PHP** dan **Java**. **PHP** digunakan untuk implementasi antarmuka *web* sedangkan **Java** digunakan pada implementasi komponen yang bertanggung jawab melakukan deteksi plagiarisme. **Deimos** mampu menemukan dugaan-dugaan praktek plagiarisme dan telah diuji menggunakan *source code* hasil pengumpulan kelas pemrograman dasar dengan peserta ratusan di program studi Teknik Informatika ITB.

Kata kunci : deteksi plagiarisme, plagiarisme *source code*

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa. Dengan rahmat dan bimbingan-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul "Deteksi Plagiarisme Otomatis pada *Source Code*" dengan baik dan tepat waktu.

Pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya pada:

1. Ibu Dr. Ir. M.M. Inggriani Liem selaku pembimbing Tugas Akhir, atas segala pendidikan dan bimbingan yang sangat berharga selama Tugas Akhir.
2. Ibu Sri Purwanti, S.T, M.T, atas segala kritik dan saran yang diberikan selaku penguji presentasi proposal dan seminar Tugas Akhir.
3. Ibu Yani Widayani, S.T, M.T, atas segala kritik dan saran yang diberikan selaku penguji prasidang dan sidang Tugas Akhir.
4. Bapak Riza Satria Perdana, S.T, M.T, atas segala kritik dan saran yang diberikan selaku penguji sidang Tugas Akhir.
5. Seluruh rekan-rekan asisten laboratorium *programming*, atas segala dukungannya.
6. Naryo, atas segala dukungannya, Ronny dan Arya, atas segala bantuan yang diberikan terhadap pembuatan Tugas Akhir ini, juga para sahabat.
7. Seluruh teman-teman seperjuangan Informatika angkatan 2003, atas segala kenangan yang diberikan.
8. Semua staf akademik Program Studi Teknik Informatika yang telah memberikan banyak ilmu selama masa perkuliahan, juga semua staf non akademik.

Tidak lupa, penulis mengucapkan banyak terima kasih pada Ayah dan Ibu tercinta atas segala kepercayaan, perhatian, dukungan, dan doa yang tak ada putusnya.

Semoga Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang membutuhkannya. Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu, saran dan kritik dari pembaca sangat penulis harapkan.

Bandung, Juni 2007

Penulis

DAFTAR ISI

Lembar Pengesahan Program Studi Sarjana Teknik Informatika.....	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR KODE.....	x
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang.....	I-1
1.2 Rumusan Masalah.....	I-2
1.3 Tujuan	I-3
1.4 Batasan Masalah	I-3
1.5 Metodologi.....	I-3
1.6 Sistematika Pembahasan	I-4
BAB II STUDI LITERATUR	II-1
2.1 Perkembangan Deteksi Plagiarisme Otomatis pada <i>Source Code</i>	II-1
2.1.1 Plagiarisme <i>Source Code</i> di Kelas Pemrograman	II-1
2.1.2 Karakteristik Plagiarisme di Kelas Pemrograman	II-3
2.1.3 Manfaat Sistem	II-4
2.1.4 Metode Deteksi Plagiarisme Otomatis	II-5
2.1.4.1 Metode <i>Attribute-counting</i>	II-5
2.1.4.2 Metode <i>Structure-based</i>	II-6
2.1.5 Penanganan Lebih Dari Satu Bahasa Pemrograman.....	II-7
2.2 Algoritma Pembandingan pada Sistem Deteksi Plagiarisme Otomatis	II-8
2.2.1 Algoritma <i>Greedy String Tiling</i>	II-8
2.2.2 <i>Tuning</i> Algoritma <i>Greedy String Tiling</i>	II-11
2.2.3 Algoritma <i>Running Karp-Rabin Greedy String Tiling</i>	II-13
2.3 Perangkat Lunak Pendeteksi Plagiarisme Otomatis pada <i>Source Code</i>	II-18
2.3.1 Plague	II-18
2.3.2 YAP	II-19
2.3.2.1 Persamaan dan Perbedaan Ketiga Versi YAP	II-19
2.3.2.2 YAP1	II-20
2.3.2.3 YAP2	II-22
2.3.2.4 YAP3	II-22
2.3.3 JPlag.....	II-23

BAB III ANALISIS.....	III-1
3.1 Deskripsi Sistem	III-1
3.2 Detektor Plagiarisme pada <i>Source Code</i>	III-2
3.2.1 Adopsi Aspek-aspek Perangkat Lunak Pendeteksi Plagiarisme	III-2
3.2.2 Karakteristik Kelas Pemrograman Dasar.....	III-4
3.2.3 Analisis Algoritma <i>Running Karp-Rabin Greedy String Tiling</i>	III-5
3.2.4 Spesifikasi Fungsional dan Non-fungsional Deimos.....	III-8
3.2.5 Use Case Deimos	III-9
3.2.6 Skenario Penggunaan Deimos	III-11
3.2.7 Rancangan Global Deimos	III-13
3.2.7.1 <i>Tokenizer</i>	III-15
3.2.7.2 <i>Comparator</i>	III-17
3.2.7.3 Aplikasi <i>web</i>	III-18
3.2.8 Penanganan Modifikasi-modifikasi <i>Source Code</i> yang Menyembunyikan Praktek Plagiarisme	III-20
BAB IV PERANCANGAN RINCI.....	IV-1
4.1 Aplikasi <i>Backend</i>	IV-1
4.2 Aplikasi <i>Web</i>	IV-6
4.3 Perancangan Representasi Persisten Kelas <i>Entity</i>	IV-7
4.4 Perancangan Antarmuka Aplikasi <i>Web</i>	IV-7
4.4.1 Halaman Utama	IV-8
4.4.2 <i>Form</i> Memulai Deteksi.....	IV-8
4.4.3 <i>Form</i> Menghapus Hasil Deteksi	IV-9
4.4.4 Halaman Utama Tampilan Hasil Deteksi	IV-9
4.4.5 Halaman Perbandingan Pasangan <i>Source Code</i>	IV-11
BAB V IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1 Implementasi.....	V-1
5.1.1 Lingkungan Implementasi dan Pengembangan	V-1
5.1.2 Batasan Implementasi	V-2
5.1.3 Implementasi Kelas	V-2
5.1.4 Implementasi Antarmuka.....	V-4
5.1.5 Implementasi Penyimpanan Data Persisten.....	V-4
5.1.6 Implementasi <i>Extendability</i> untuk Bahasa Pemrograman Lain	V-6
5.2 Pengujian	V-7
5.2.1 Lingkungan Pengujian	V-7
5.2.2 Persiapan Pengujian.....	V-8
5.2.2.1 Data Uji	V-8
5.2.2.2 Instalasi Prototipe	V-10

5.2.3	Kasus Uji	V-11
5.2.3.1	Pengujian unit.....	V-11
5.2.3.2	Pengujian Fungsional Berdasarkan <i>Use Case</i>	V-12
5.2.3.2.1	Butir Uji Mendeteksi Plagiarisme Otomatis.....	V-12
5.2.3.2.2	Butir Uji Melihat Hasil Deteksi	V-15
5.2.3.2.3	Butir Uji Menghapus Hasil Deteksi	V-16
5.2.3.3	Pengujian non-fungsional.....	V-16
5.2.4	Evaluasi Pengujian.....	V-17
BAB VI KESIMPULAN DAN SARAN.....		VI-1
6.1	Kesimpulan	VI-1
6.2	Saran	VI-2
DAFTAR REFERENSI.....		xi
DAFTAR PUSTAKA.....		xiv
Lampiran A CONTOH PENERAPAN ALGORITMA <i>GREEDY STRING TILING</i>		A-1
Lampiran B SKENARIO <i>USE CASE</i>		B-1
Lampiran C PERANCANGAN <i>SCANNER</i> DAN <i>PARSER</i>		C-1
Lampiran D <i>DIAGRAM SEQUENCE</i>		D-1

DAFTAR GAMBAR

Gambar II-1. Spektrum Plagiarisme [FAI87]	II-4
Gambar II-2. Struktur Data <i>Double Linked-list of Queues</i> yang Dipakai Untuk Menyimpan <i>Maximal-matches</i> pada Algoritma <i>Running Karp-Rabin Greedy String Tiling</i>	II-16
Gambar II-3. Bagian Atas Contoh Halaman yang Menampilkan Hasil Deteksi <i>JPlag</i> [LUT00]	II-24
Gambar II-4. Contoh Bagian dari Halaman yang Menampilkan Pasangan Program pada Tampilan Hasil Deteksi <i>JPlag</i> [LUT00]	II-25
Gambar II-5. Contoh <i>Source Code</i> Java dan Kumpulan <i>Token</i> yang Mewakili Setiap Barisnya pada <i>JPlag</i> (diambil dari [LUT00])	II-26
Gambar III-1. Deskripsi <i>Learning Management System Milestone</i>	III-1
Gambar III-2. Pengaruh nilai <i>minimum-match-length</i> pada deteksi plagiarisme	III-6
Gambar III-3. Diagram <i>Use Case Deimos</i>	III-10
Gambar III-4. Skenario Penggunaan <i>Deimos</i> Untuk Mendeteksi Plagiarisme Secara Otomatis	III-11
Gambar III-5. Skenario Penggunaan <i>Deimos</i> Untuk Melihat Hasil Deteksi Plagiarisme	III-12
Gambar III-6. Skenario Penggunaan <i>Deimos</i> Untuk Menghapus Hasil Deteksi Tertentu	III-13
Gambar III-7. Rancangan Global <i>Deimos</i>	III-14
Gambar III-8. Contoh Rancangan Tampilan Halaman Utama Hasil Deteksi pada <i>Deimos</i> .	III-19
Gambar III-9. Contoh Rancangan Tampilan Halaman Perbandingan Pasangan <i>Source Code</i> pada <i>Deimos</i>	III-20
Gambar IV-1. Diagram Kelas Aplikasi <i>Backend Deimos</i>	IV-3
Gambar IV-2. Diagram kelas yang mewakili <i>modeler</i> dan <i>controller</i> pada aplikasi web <i>Deimos</i>	IV-6
Gambar IV-3. Rancangan Antarmuka Generik Detektor Plagiarisme <i>Deimos</i>	IV-8
Gambar IV-4. Halaman Utama Aplikasi <i>Web Deimos</i>	IV-8
Gambar IV-5. Antarmuka Form Memulai Deteksi pada Aplikasi <i>Web Deimos</i>	IV-9
Gambar IV-6. Antarmuka Form Menghapus Hasil Deteksi	IV-9
Gambar IV-7. Halaman Tampilan Hasil Deteksi Utama pada Aplikasi <i>Web Deimos</i>	IV-10
Gambar IV-8. Antarmuka Halaman Perbandingan Pasangan <i>Source Code</i> pada Aplikasi <i>Web</i> <i>Deimos</i>	IV-11
Gambar V-1. Lingkungan Pengujian <i>Deimos</i>	V-7
Gambar D-1. Diagram <i>Sequence</i> untuk <i>Trigger</i> Deteksi	D-1
Gambar D-2. Diagram <i>Sequence</i> untuk <i>Tokenizing</i>	D-2
Gambar D-3. Diagram <i>Sequence</i> untuk Membandingkan Setiap Pasangan <i>Token String</i>	D-3
Gambar D-4. Diagram <i>Sequence</i> untuk Mengirimkan Notifikasi bahwa Deteksi Selesai	D-4
Gambar D-5. Diagram <i>Sequence</i> untuk Melihat Halaman Utama Hasil Deteksi	D-5
Gambar D-6. Diagram <i>Sequence</i> untuk Melihat Halaman Perbandingan Pasangan <i>Source</i> <i>Code</i>	D-6
Gambar D-7. Diagram <i>Sequence</i> untuk <i>Use Case</i> Menghapus Hasil Deteksi	D-7

DAFTAR TABEL

Tabel III-1. Ikhtisar Aspek-aspek Perangkat Lunak Pendeteksi Plagiarisme yang Akan Diimplementasikan pada Deimos	III-3
Tabel III-2. Spesifikasi Fungsional Deimos	III-8
Tabel III-3. Spesifikasi Non-fungsional Deimos.....	III-9
Tabel III-4. Definisi <i>Use Case</i> Deimos	III-10
Tabel III-5. Keterhubungan Spesifikasi Fungsional dengan <i>Use Case</i> Deimos.....	III-10
Tabel III-6. Tanggung Jawab Komponen Deimos beserta Input yang Dibutuhkan dan Hasilnya	III-14
Tabel III-7. Tanggung Jawab Setiap Subsistem pada Aplikasi <i>Backend</i> Deimos beserta Input yang Dibutuhkan dan Hasilnya	III-15
Tabel III-8. Penanganan Modifikasi-modifikasi yang Dilakukan Mahasiswa Kelas Pemrograman	III-21
Tabel IV-1. Rancangan Kelas Aplikasi <i>Backend</i> Deimos	IV-4
Tabel IV-2. Rancangan Kelas Aplikasi <i>Web</i> Deimos.....	IV-6
Tabel IV-3. Perancangan Penyimpanan Data Persisten dan Pemetaannya dengan Kelas <i>Entity</i>	IV-7
Tabel V-1. Detil Implementasi Penyimpanan Data Persisten Untuk Deimos	V-5
Tabel V-2. Hasil Pengujian Unit Pada Aplikasi <i>Backend</i> Deimos	V-11
Tabel V-3. Hasil Pengujian Spesifikasi Non-fungsional.....	V-16
Tabel C-1. Daftar <i>Token</i> yang Merepresentasikan Elemen <i>Source Code</i> Pascal	B-9
Tabel C-2. Daftar Simbol LISP serta Nilai <i>Token</i> yang Merepresentasikannya	B-11
Tabel C-3. Daftar <i>Keyword</i> LISP serta Nilai <i>Token</i> yang Merepresentasikannya	B-11

DAFTAR KODE

Kode II-1. Algoritma <i>Greedy String Tiling</i> ([WIS93a,WIS93b])	II-11
Kode II-2. Algoritma <i>Greedy String Tiling</i> yang Sudah Di- <i>tuning</i> [WIS93a,WIS93b].....	II-13
Kode II-3. <i>Pseudocode</i> Algoritma <i>scanpattern(s)</i> pada RKR-GST (diadopsi dari [WIS93b])	II-15
Kode II-4. Fungsi <i>markstrings(s)</i> pada Algoritma RKR-GST dengan <i>s</i> Merupakan <i>Search-length</i> [WIS93a,WIS93b]	II-17
Kode II-5. <i>Top-level Algorithm</i> untuk RKR-GST [WIS93a,WIS93b]	II-17
Kode III-1. Modifikasi Urutan <i>Statement</i> yang Merupakan Salah Satu Usaha Plagiator Untuk Menyembunyikan Praktek Plagiarisme	III-7