

RANCANG BANGUN APLIKASI PENDETEKSI PLAGIAT *SOURCE CODES* DENGAN *TOOLS JPLAG* (STUDI KASUS IT TELKOM PURWOKERTO)

Iwan Firmawan¹, Elisa Usada², Ipam Fuaddina Adam³

^{1,2,3}Informatika/S1 Informatika dan IT Telkom Purwokerto,

e-mail: ¹14102065@st3telkom.ac.id, ²elisa@ittelkom-pwt.ac.id, ³ipam@ittelkom-pwt.ac.id

ABSTRAK

Dalam upaya mencegah tindak plagiarisme di lingkungan kampus IT Telkom Purwokerto. Khususnya praktikum pemrograman komputer yang berdampak besar pada kompetensi dan keahlian masing-masing mahasiswa. Perancangan dan pembangunan aplikasi pendeteksi plagiat pada *source codes* dilakukan dengan mengintegrasikan *tools JPLAG*. *JPLAG* merupakan *tools* pendeteksi plagiat *source codes* otomatis yang menerapkan algoritma *Greedy String Tiling*. *Greedy String Tiling* bekerja dengan 2 tahap yaitu pencocokan berdasarkan *minimum match length* dan mengelompokkan kemiripan token. *JPLAG* dikembangkan oleh *Karlsruhe Institute of Technology* dari Jerman menggunakan program *Java* dan dipilih sebagai *tools* yang berintegrasi dengan aplikasi *Scaniplag* karena *tools* tersebut dapat dijalankan pada aplikasi berbasis *website* dan mendeteksi *source codes* secara masal. *JPLAG* mendukung bahasa pemrograman yang digunakan oleh praktikum di kampus IT Telkom Purwokerto seperti *Java*, *C#*, *C* dan *C++*. *JPLAG* tidak hanya membandingkan *byte* teks, melainkan dapat mengetahui sintaks dan struktur *source codes* yang berupaya menyamarkan plagiat. Penyebaran kuesioner *expert judgment* yang mengevaluasi tingkat plagiat *JPLAG* dengan 7 teknik modifikasi *source codes* kepada 5 dosen pemrograman. Menghasilkan 0% *expert* menilai tingkat plagiat *JPLAG* dinaikkan. 25,7 % *expert* menilai tingkat plagiat *JPLAG* selaras atau tetap. 74,3 % *expert* menilai tingkat plagiat *JPLAG* diturunkan. Dari hasil implementasi *dataset source codes* praktikum *S1 Sistem Informasi 2017*, jenis plagiarisme yang ditemukan tergolong dalam *basic level*.

Kata Kunci: *Greedy String Tiling*, *JPLAG*, plagiat *source codes*

1. PENDAHULUAN

Praktikum pemrograman komputer setiap tahunnya dilaksanakan oleh mahasiswa baru untuk mengasah logika dalam pemecahan masalah. Umumnya, banyak asisten praktikum kesulitan untuk mengidentifikasi plagiarisme pada *source codes*. Sangat sedikit asisten praktikum yang memiliki kesabaran untuk benar-benar mencari plagiarisme. Walaupun menemukan plagiarisme hal yang mungkin, kegiatan tersebut membutuhkan waktu yang cukup lama[1]. Adapun tindakan-tindakan yang tergolong sebagai plagiarisme pada *source codes* ialah mengambil sebagian *source codes* dari pihak lain yang sah tanpa melakukan pengakuan dari sumber lain, memproduksi ulang seluruh bagian *source codes* milik orang lain dalam bahasa pemrograman lain yang serupa, pelajar atau peneliti yang membayar seseorang untuk membuat program dan pelajar yang melanggar bekerja sama untuk menyelesaikan tugas pemrograman[2]. Sebuah survey yang ditujukan kepada asisten praktikum terkait penawaran sistem pendeteksi plagiat pada *source codes* telah dilakukan di area kampus IT Telkom Purwokerto melalui *Google* formulir. Hasil survey tersebut menunjukkan 70,8 % asisten praktikum mendukung pembuatan aplikasi praktikum yang berintegrasi dengan alat pendeteksi plagiat *source codes* otomatis. Aplikasi pendeteksi plagiat *source codes* pada penelitian ini diberi nama dengan *Scaniplag (Source Codes Anti Plagiarism)*.

Berdasarkan kriteria dan dukungan bahasa pemrograman yang telah dilakukan pada tahap studi literatur untuk mengeksplorasi alat pendeteksi plagiat *source codes* yang saat ini ada. *JPLAG* memiliki dominasi yang kuat untuk berintegrasi dengan aplikasi *Scaniplag*. Pasalnya, dukungan bahasa pemrograman yang disediakan *JPLAG* sesuai dengan bahasa pemrograman yang digunakan oleh kelas praktikum di kampus IT Telkom Purwokerto. Selain itu, pemasangan atau tahap instalasi yang diperlukan untuk mengintegrasikan *JPLAG* dengan aplikasi *Scaniplag* tergolong sederhana. Pemasangan *JPLAG* hanya memerlukan ruang penyimpanan yang memiliki hak akses untuk menjalankan program *JAVA* yang berekstensi *.jar. *JPLAG* bekerja dengan membandingkan semua *source codes* secara bergantian dan berpasangan.

Dengan adanya aplikasi yang dapat mendeteksi plagiat *source codes* otomatis. Diharapkan penilaian praktikum yang diberikan asisten praktikum kepada praktikan tidak subjektif dan tidak memakan banyak waktu. Karena setiap pengumpulan *source codes* ketepatan waktu dan prosentase plagiat dapat diukur secara otomatis melalui aplikasi. Sehingga asisten praktikum memiliki referensi dalam meninjau parameter penilaian setiap praktikan. Selain itu, *source codes* yang diidentifikasi plagiat dapat ditinjau oleh dosen praktikum sebagai *expert judgment* yang menilai tingkat plagiarisme yang dilakukan praktikan.

2. TINJAUAN PUSTAKA

Dalam mengeksplorasi *tools* pendeteksi plagiat *source codes*, peneliti meninjau beberapa jurnal yang telah disajikan pada Tabel 1 berikut ini.

Tabel 1. Review jurnal penelitian pendeteksi plagiat *source codes* otomatis.

No	Judul Jurnal	Keluaran	Hasil Penelitian
1	<i>Similarity Detection in Java Programming Assignments</i>	Sistem CAPlag (Computing Assignment Plagiarism)	Evaluasi sistem dilakukan dengan men-deteksi program java yang telah dimodifikasi dalam 6 level terdiri dari pengurangan atau penambahan komentar, mengganti variabel, mengacak <i>block sequences</i> , merubah struktur perintah, mengubah pernyataan program dan mengubah struktur kontrol. Hasilnya per-bandingan struktural lebih unggul dibandingkan dengan pendekatan atribut. Perbandingan struktural cocok untuk men-deteksi plagiat tugas pemrograman berskala besar.
2	<i>Efficient Source code Plagiarism Identification Based on Greedy String Tiling.</i>	Sebuah Sistem yang mengimplementasikan <i>Greedy String Tiling</i> yang dapat menyimpan pada file Excel.	Penambahan fungsi matrik untuk mengukur kemiripan pasangan <i>source codes</i> efektif mendeteksi <i>source codes</i> yang menyamakan plagiat dengan teknik modifikasi perubahan variabel, mengganti urutan dan nama fungsi kemudian mengacak-acak baris perintah. Evaluasi <i>tools</i> befokus pada kecepatan sistem mendeteksi 9 variasi token <i>source codes</i> c++.
3	<i>A Source code Similarity System for Plagiarism Detect-ion.</i>	Sistem SCSDS (<i>Source code Similarity Detector System</i>).	Berikut ini hasil perbandingan sistem dengan tools JPLAG : 1. Nilai kemiripan JPLAG ber-kurang pada saat mendeteksi modifikasi <i>source codes</i> dengan merubah <i>block sequences</i> . 2. Nilai kemiripan JPLAG berkurang pada saat mendeteksi modifikasi <i>source codes</i> dengan merubah struktur perintah. 3. Nilai kemiripan JPLAG berkurang pada saat mendeteksi <i>source codes</i> yang telah dirancang ulang (modifikasi guru).
4	<i>An Anti-Plagiarism Add-on for Web-CAT.</i>	Web-CAT integrasi <i>tools Plaggie2</i> .	Evaluasi Web-cat dilakukan dengan membandingkan <i>tools</i> JPLAG melalui teknik modifikasi berikut ini : 1. Plaggie2 dan JPLAG tidak berpengaruh pada teknik modifikasi mengubah nama <i>class</i> , <i>variabel</i> , <i>method</i> , <i>import library</i> JAVA dan mengganti property GUI. 2. Nilai kemiripan JPLAG lebih tinggi pada saat modifikasi <i>setters</i> dan <i>getters</i> . 3. Nilai kemiripan JPLAG dan Plaggie2 berkurang pada saat penggeseran <i>block sequence</i> .

Pada Tabel 1 *Review* jurnal penelitian sistem otomatis pendeteksi plagiat *source codes* diatas menunjukkan berbagai teknik-teknik modifikasi yang sering dilakukan untuk menyamakan tindakan plagiat dan mengevaluasi hasil kemiripan yang diperoleh sistem. Pada penelitian ini keluaran yang dihasilkan berbentuk aplikasi *website* yang berintegrasi dengan *tools* JPLAG sebagai pendeteksi plagiat *source codes*. Diakses melalui jaringan intranet kampus IT Telkom Purwokerto dan cara yang digunakan untuk mengevaluasi adalah dengan melakukan pengujian program C++ praktikum Prodi S1 Sistem Informasi 2017 dan validasi plagiat melalui kuesioner *expert judgment*.

3. METODE PENELITIAN

3.1 Objek dan Subjek Penelitian

Dari latar belakang yang telah diuraikan sebelumnya, penelitian ini dilakukan pada laboratorium komputer kampus IT Telkom Purwokerto. Subjek yang diteliti pada penelitian ini berkaitan dengan pendeteksi plagiat pada *source codes*. Pendeteksi plagiat *source codes* menggunakan *tools* JPLAG yang diintegrasikan dengan Aplikasi Scaniplag.

3.2 Studi Literatur

Studi literatur yang dilakukan pada penelitian ini adalah menggali informasi terkait teknologi yang telah dikembangkan untuk mendeteksi plagiat *source codes*. Penggalan informasi dilakukan dengan membaca jurnal-jurnal penelitian sebelumnya, mencari referensi aplikasi yang telah dipublikasikan pada situs Github dan *trial & error* aplikasi yang dapat dikembangkan.

3.2 Pengumpulan Data

3.2.1 Kuesioner

Terdapat dua kuesioner yang dilakukan pada penelitian ini. Kuesioner pertama dilakukan secara *online* melalui Google formulir untuk menggali permasalahan praktikum pemrograman di kampus IT Telkom Purwokerto yang ditujukan kepada asisten praktikum. Kuesioner kedua dilakukan secara *offline* melalui penyebaran langsung kepada 5 dosen *expert* pemrograman untuk menilai tingkat plagiarisme yang dihasilkan *tools* JPLAG dari 7 teknik modifikasi *source codes* yang bervariasi.

3.2.2 Dataset source codes praktikum

Dataset *source codes* sebagai bahan pengujian JPLAG yang diambil dari hasil praktikum Algoritma dan Pemrograman kelas S1 Sistem Informasi angkatan 2017. *Source codes* menggunakan bahasa pemrograman C++. Program yang dibuat oleh praktikan berupa program bilangan genap atau ganjil, mencari nilai maksimal dan minimal beserta cara pengurutannya.

3.3 Perancangan Sistem

Perancangan sistem dibuat dengan pemodelan dokumentasi UML yang terdiri *use case*, *class* diagram dan ERD (*Entity Relationship Diagram*). *Use case* dibuat untuk mendeskripsikan peran pengguna dalam suatu sistem. Pemodelan Class Diagram dibuat untuk menentukan struktur suatu obyek memiliki atribut dan fungsi untuk diimplementasikan menjadi sebuah program perangkat lunak komputer. Pemodelan *Entity relationship* diagram dibuat untuk menentukan format penyimpanan data berdasarkan atribut pada entitas yang dibutuhkan dan relasi antar entitas yang saling terbentuk.

3.4 Pembuatan Sistem

Pembuatan sistem dilakukan dengan model pengembangan *iterative*, model ini dipilih karena aplikasi yang akan dibangun berskala kecil dan spesifik untuk kepentingan praktikum dan membantu pendeteksi plagiat *source codes* yang tersubmit oleh praktikan. Model ini secara fundamental menguraikan proses pembangunan yang progresif melalui penambahan fitur lebih lanjut secara bertahap sampai dengan sistem pada kondisi stabil.

3.5 Pengujian Sistem

Pengujian sistem dilakukan dengan membuat kuesioner untuk *expert judgement* menilai hasil plagiat dari *tools* JPLAG. Pertanyaan-pertanyaan kuesioner mengacu pada simulasi 7 cara modifikasi *source code* jawaban soal nomer 1 modul 7 praktikum Algoritma dan pemrograman kelas Sistem Informasi B angkatan 2017.

3.6 Teknik Analisa Data

Analisa data dilakukan dengan membandingkan jumlah prosentase *expert* untuk menurunkan, menetapkan atau menaikkan tingkat plagiarisme *tools* JPLAG dari kuesioner *expert judgment* yang telah disebarakan kepada 5 dosen *expert* pemrograman. Hasil analisa tersebut kemudian diolah untuk mengetahui keselarasan tingkat plagiarisme *tools* JPLAG dengan penilaian *expert* untuk kebutuhan praktikum di kampus IT Telkom Purwokerto.

4. HASIL DAN PEMBAHASAN

4.1 Tahap Requirements

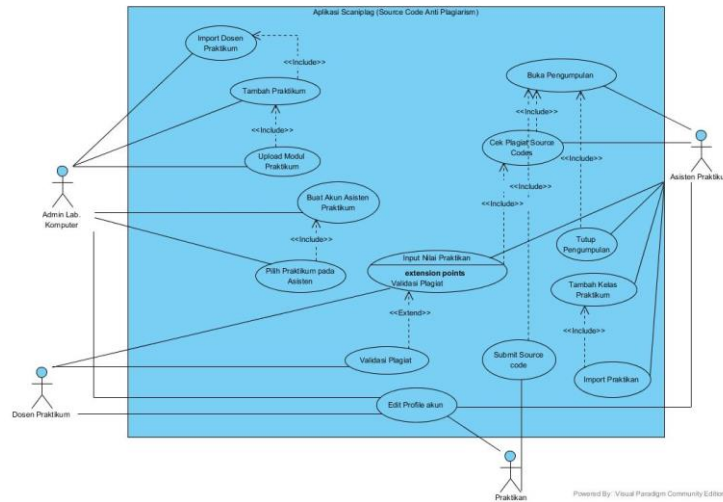
Tabel 2 dibawah ini merupakan kebutuhan fungsional sistem scaniplag untuk menangani praktikum.

Tabel 2. User requirement aplikasi Scaniplag

User	User Requirement	Kode UR
Admin	Import Dosen Praktikum	UR-1
Laboratorium Komputer	Import praktikan	UR-1
	Tambah Praktikum	UR-2
	Upload modul praktikum	UR-3
	Tambah Asisten praktikum	UR-4
	Pilih Asisten praktikum sesuai keputusan dosen pengampu praktikum.	UR-5
Asisten Praktikum	Buka <i>submit source codes</i>	UR-6
	Tutup <i>submit source codes</i>	UR-7
	Cek Plagiat <i>source codes</i> praktikan	UR-8
	Input nilai praktikum	UR-9
Praktikan	Submit <i>source code</i>	UR-10
Dosen Pratikum	Input nilai praktikum	UR-11
	Validasi Plagiat	UR-12

4.2 Tahap Design

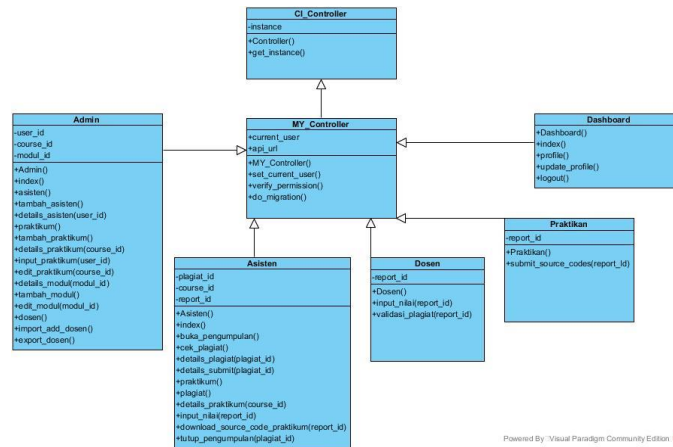
4.2.1 Use case diagram



Gambar 1. Use case Scaniplag

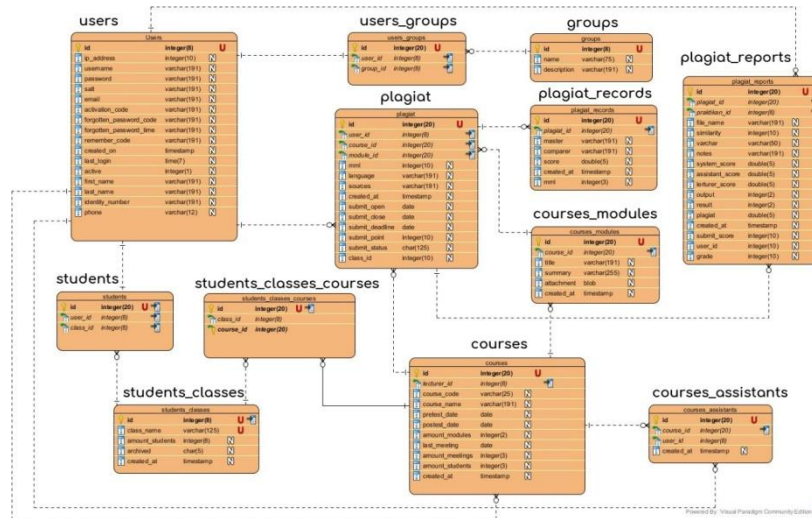
Pada Gambar 1 *Use case Scaniplag* memberikan penjelasan admin laboratorium komputer dapat melakukan import dosen praktikum untuk menambahkan data dosen praktikum pada sistem. *Case import* dosen praktikum wajib dilakukan Admin laboratorium komputer sebelum melakukan tambah praktikum. Admin laboratorium komputer dapat membuat akun asisten praktikum. *Case* membuat akun asisten praktikum wajib dilakukan admin laboratorium komputer sebelum melakukan pilih praktikum pada asisten. Asisten praktikum dapat melakukan *import* praktikan setelah membuat kelas praktikum. Asisten praktikum dapat melakukan buka pengumpulan agar praktikan dapat submit *source codes*. *Case* buka pengumpulan wajib dilakukan asisten praktikum sebelum tutup pengumpulan dan cek plagiat *source codes*. *Case* cek plagiat *source codes* wajib dilakukan asisten praktikum sebelum input nilai praktikan. Praktikan dapat *submit source codes* selama pengumpulan masih dibuka oleh asisten praktikum. Dosen praktikum dapat input nilai praktikan dan melakukan validasi plagiat yang dihasilkan dari *tools* JPLAG. Semua aktor dapat edit *profile* akun untuk mengubah informasi e-mail atau mengganti *password*.

4.2.2 Class diagram



Gambar 2. Class diagram aplikasi Scaniplag

Pada Gambar 2 *Class diagram aplikasi Scaniplag* diatas terlihat ada 3 hirarki generalisasi, ditingkat pertama *class CI_Controller* sebagai induk class yang memiliki semua fungsional pada *framework Codeigniter*. Tingkat kedua *class MY_Controller* merupakan anak class dari *CI_Controller* dan induk *class* dari 3 *class* dibawahnya. *Class MY_Controller* memiliki fungsional untuk membuat *session user* yang berhasil *login*, mengatur hak akses dan melakukan migrasi *database*. Tingkat ketiga merupakan *class-class* spesifik yang terdiri dari admin, asisten, praktikan dan dashboard yang memiliki fungsional sesuai dengan *use case*.



Gambar 3. ERD aplikasi Scaniplag

4.3 Tahap Development

Tabel 3. Versi pengembangan aplikasi Scaniplag.

No	Versi	Fitur Aplikasi
1	1.0.0	Pengguna terdiri dari admin dan asisten. Admin praktikum dapat menambah dosen, praktikum dan asisten. Asisten dapat <i>upload source codes</i> , deteksi plagiat dan input nilai.
2	1.1.1	Pengguna terdiri dari admin, asisten dan praktikan. Asisten dapat membuat kelas, import praktikan dan membuka atau menutup pengumpulan <i>source codes</i> praktikan. Praktikan dapat
3	1.2.1	Pengguna terdiri dari admin, asisten, praktikan dan dosen. Dosen dapat memberikan nilai akhir praktikan. Dosen dapat menentukan <i>source codes</i> praktikan plagiat atau tidak.

4.4 Tahap Testing

Tabel 4. Hasil kuesioner *expert judgment*.

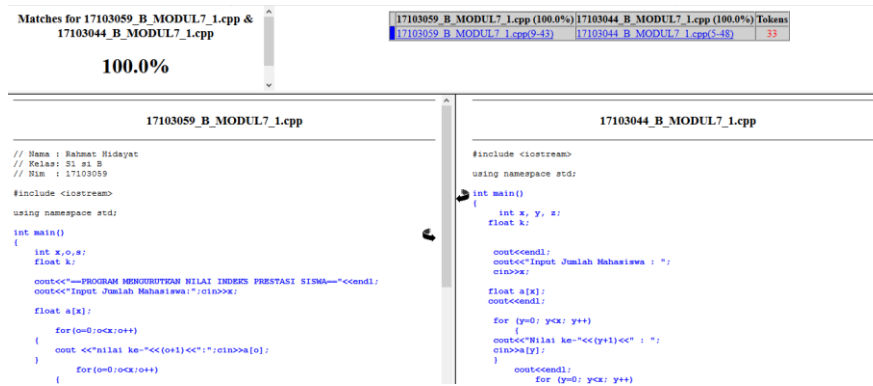
1	Modifikasi <i>source code</i> dengan menambah atau mengurangi komentar.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	100	0	0	2	3
2	Modifikasi <i>source code</i> dengan mengubah variabel atau tipe data.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	60	40	0	1	4
3	Modifikasi <i>source code</i> dengan menambah atau mengurangi <i>output</i> program.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	80	20	0	2	3
4	Modifikasi <i>source code</i> dengan mengubah urutan blok perintah.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	60	40	0	2	3
5	Modifikasi <i>source code</i> dengan mengubah struktur perintah (<i>for</i> menjadi <i>while</i>) yang memiliki eksekusi sama.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	60	40	0	2	3
6	Modifikasi Guru <i>source code</i> dengan menggabungkan modifikasi <i>block sequence</i> dan tipe data.				
	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	60	40	0	0	5
7	Modifikasi <i>Advanced source code</i> dengan menggabungkan semua teknik modifikasi.				

	Ya (%)	Tidak (%)	(%) JPLAG++	(%) JPLAG= <i>expert</i>	(%) JPLAG--
	60	40	0	0	5
Total			0	9	26
Total Expert (%)			0	25,7	74,3

Pada Tabel 4 diatas menjelaskan bahwa dari masing-masing penilaian *expert* terhadap prosentase tingkat plagiat yang dideteksi oleh JPLAG. Berapa jumlah *expert* yang menilai prosentase plagiat JPLAG perlu dinaikkan yang diwakili oleh kolom (%) JPLAG ++. Berapa jumlah *expert* yang menilai prosentase plagiat JPLAG sesuai dengan *expert* yang diwakili oleh kolom (%) JPLAG=*expert*. Dan berapa jumlah *expert* yang menilai prosentase plagiat JPLAG diturunkan. Kemudian masing-masing jumlah tersebut ditotal sampai dengan memperoleh prosentase *expert* menemukan kecondongan untuk menaikkan, menurunkan atau menetapkan prosentase plagiat sesuai JPLAG. Dapat dilihat dari perolehan kuesioner tersebut, sebanyak 74,3 % *expert* menilai prosentase tingkat plagiat JPLAG perlu diturunkan.

4.5 Tahap Implementasi

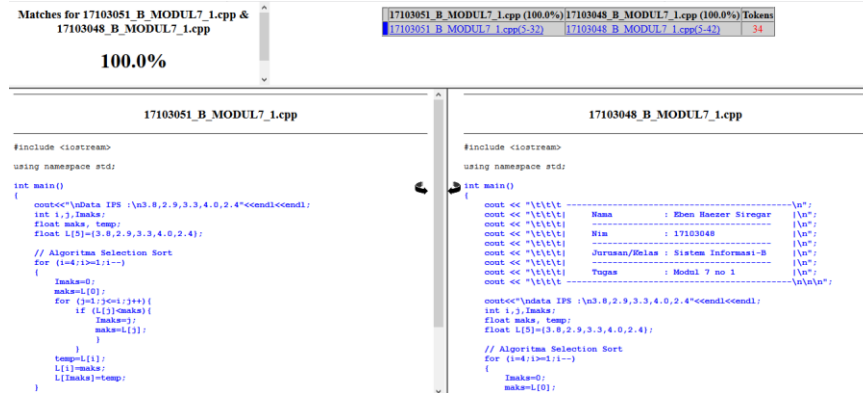
Untuk tahap implementasi pertama, aplikasi Scaniplag digunakan untuk mendeteksi dataset *source codes* soal nomor 1 modul 7 dari praktikum kelas S1 Sistem Informasi 2017. Implementasi ini dilakukan oleh asisten praktikum dengan mengunggah semua *source codes* praktikan dalam format arsip berekstensi *.zip. Setelah proses mengunggah *source codes* selesai, asisten praktikum kemudian menentukan tingkat kemiripan sebagai parameter JPLAG mendeteksi plagiat pada *source codes*. Hasil pendeteksi plagiat menemukan 3 jenis modifikasi *source codes* yang terdiri dari modifikasi komentar, modifikasi variabel atau tipe data dan modifikasi *output* seperti yang ditunjukkan pada Gambar 4 sampai dengan 5 berikut ini.



Gambar 4. Modifikasi komentar.



Gambar 5. Modifikasi variabel atau tipe data.



Gambar 6. Modifikasi output.

Baris perintah yang bertambah pada *source codes* NIM 17102048 yang ditunjukkan pada Gambar 6 Potongan modifikasi output diatas telah mengalami penurunan tingkat plagiarisme yang dideteksi oleh JPLAG.

5. KESIMPULAN

Berdasarkan penyebaran kuesioner *expert judgment*, diperoleh hasil prosentase *expert* menilai tingkat plagiat dari masing-masing modifikasi berikut ini:

- a. 100% *expert* menilai modifikasi komentar merupakan plagiat.
- b. 60% *expert* menilai modifikasi variabel atau tipe data merupakan plagiat.
- c. 80% *expert* menilai modifikasi *output* merupakan plagiat.
- d. 60% *expert* menilai modifikasi *block sequence* merupakan plagiat.
- e. 60% *expert* menilai modifikasi struktur perintah merupakan plagiat.
- f. 60% *expert* menilai modifikasi guru merupakan plagiat.
- g. 60% *expert* menilai modifikasi *advanced* merupakan plagiat.
- h. 74,3% *expert* menilai prosentase tingkat plagiat JPLAG perlu diturunkan.

Dari hasil implementasi aplikasi Scaniplag, ditemukan jenis modifikasi komentar, modifikasi variabel atau tipe data dan modifikasi *output* pada dataset *source codes* C++ yang diambil dari soal nomor 1 modul 7 praktikum kelas S1 Sistem Informasi 2017. Berdasarkan jurnal “*Source code plagiarism detection: The Unix way*” jenis modifikasi tersebut tergolong sebagai plagiator *basic level*.

6. SARAN

Dalam upaya pengembangan aplikasi Scaniplag untuk memberikan kontribusi penanganan tindak plagiarisme pada bidang pendidikan khususnya pemrograman komputer. Beberapa hal berikut ini perlu diperhatikan oleh penelitian selanjutnya guna meningkatkan kontribusi yang lebih baik seperti :

- a. Aplikasi Scaniplag perlu terintegrasi dengan sistem informasi kampus.
- b. Aplikasi Scaniplag perlu menyediakan hasil *output* program otomatis untuk mengetahui apakah *output* sesuai dengan instruksi soal atau tidak.
- c. Diperlukan modifikasi program JPLAG untuk menurunkan sensitifitas prosentase tingkat plagiat.

DAFTAR PUSTAKA

[1] L. Prechelt, G. Malpohl, and M. Philippsen, “Finding Plagiarisms among a Set of Programs with JPlag,” *J. Univers. Comput. Sci.*, vol. 8, no. 11, pp. 1016–1038, 2002.

[2] K. Z. Haider, T. Nawaz, S. Din, and A. Javed, “Efficient Source Code Plagiarism Identification Based on Greedy String Tilling,” vol. 10, no. 12, pp. 204–210, 2010.