



**APLIKASI DETEKSI PLAGIARISME SOURCE CODE PADA JAVA
PROGRAMING DENGAN PENERAPAN LEVELING SENSITIVITAS
MENGUNAKAN METODE LEKSIKAL**

TUGAS AKHIR

Muhamad Arif Firmansyah

41516310025

UNIVERSITAS
MERCU BUANA

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

2020



**APLIKASI DETEKSI PLAGIARISME SOURCE CODE PADA JAVA
PROGRAMING DENGAN PENERAPAN LEVELING SENSITIVITAS
MENGUNAKAN METODE LEKSIKAL**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

Muhamad Arif Firmansyah

41516310025

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

LEMBAR PERNYATAAN ORISINALITAS

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41516310025
Nama : Muhamad Arif Firmansyah
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java Programing
Dengan Penerapan Leveling Sensitivitas Menggunakan
Metode Leksikal

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.



Jakarta, ~~18~~ September 2020



Muhamad Arif Firmansyah

UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Muhamad Arif Firmansyah
NIM : 41516310025
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java
Programing Dengan Penerapan Leveling Sensitivitas
Menggunakan Metode Leksikal

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 2 September 2020



Muhamad Arif Firmansyah

UNIVERSI
MERCU BUANA

SURAT PERNYATAAN LUARAN TUGAS AKHIR

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Muhamad Arif Firmansyah
 NIM : 41516310025
 Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java Programming Dengan Penerapan Leveling Sensitivitas Menggunakan Metode Leksikal

Menyatakan bahwa Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis	Status
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi	Diajukan <input checked="" type="checkbox"/>
		Jurnal Nasional Terakreditasi	
		Jurnal International Tidak Bereputasi	Diterima <input type="checkbox"/>
		Jurnal International Bereputasi	
Disubmit/dipublikasikan di :	Nama Jurnal	: JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer) – Nusa Mandiri	
	ISSN	: P-ISSN: 2685-8223 E-ISSN: 2527-4864	
2	Kertas Kerja, Merupakan material hasil penelitian sebagai kelengkapan Artikel Jurnal. Terdiri dari (minimal 4)	Literatur Review	<input checked="" type="checkbox"/>
		Hasil analisa & perancangan aplikasi	<input checked="" type="checkbox"/>
		Source code	<input checked="" type="checkbox"/>
		Data set	<input type="checkbox"/>
		Tahapan eksperimen	<input checked="" type="checkbox"/>
		Hasil eksperimen seluruhnya	<input checked="" type="checkbox"/>
3	HAKI Disubmit / Terdaftar	HKI	Diajukan <input type="checkbox"/>
		Paten	Tercatat <input type="checkbox"/>
		No & Tanggal Permohonan	:
		No & Tanggal Pencatatan	:

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 28 September 2020




 Muhamad Arif Firmansyah

LEMBAR PERSETUJUAN PEMBIMBING



LEMBAR PERSETUJUAN PEMBIMBING

Nama Mahasiswa : Muhamad Arif Firmansyah
NIM : 41516310025
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java
Programming Dengan Penerapan Leveling
Sensitivitas Menggunakan Metode Leksikal

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 2 September 2020

Menyetujui,

(Sri Dianing Asri, ST, M Kom)
Dosen Pembimbing

MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI



LEMBAR PERSETUJUAN PENGUJI

NIM : 41516310025
Nama : Muhamad Arif Firmansyah
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java
Programing Dengan Penerapan Leveling Sensitivitas
Menggunakan Metode Leksikal

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 07 September 2020



(Desi Ramayanti, S.Kom., MT)

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI



LEMBAR PERSETUJUAN PENGUJI

NIM : 41516310025
Nama : Muhamad Arif Firmansyah
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java Programming
Dengan Penerapan Leveling Sensitivitas Menggunakan
Metode Leksikal

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 07 September 2020



(Dr. Ida Nurhaida, M.T.)

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI



LEMBAR PERSETUJUAN PENGUJI

NIM : 41516310025
Nama : Muhamad Anf Firmansyah
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java Programming Dengan Penerapan Leveling Sensitivitas Menggunakan Metode Leksikal

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 07 September 2020

(Diky Firdaus, S.Kom, MM)

UNIVERSITAS
MERCU BUANA

LEMBAR PENGESAHAN



LEMBAR PENGESAHAN

NIM : 41516310025
Nama : Muhamad Arif Firmansyah
Judul Tugas Akhir : Aplikasi Deteksi Plagiarisme Source Code Java
Programing Dengan Penerapan Leveling Sensitivitas
Menggunakan Metode Leksikal

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 07 September 2020

Menyetujui,

(Sri Dianing Asri, ST, M.Kom)
Dosen Pembimbing

Mengetahui,

UNIVERSITAS

(Diky Firdaus, S.Kom, M.M)
Koord. Tugas Akhir Teknik Informatika

(Desi Ramayanti, S.Kom, MT)
Ka. Prodi Teknik Informatika

MERCU BUANA

ABSTRAK

Nama : Muhamad Arif Firmansyah
NIM : 41516310025
Pembimbing TA : Sri Dianing Asri, ST, M.Kom
Judul : Aplikasi Deteksi Plagiarisme Source Code Java
Programing Dengan Penerapan Leveling Sensitivitas
Menggunakan Metode Leksikal

Plagiarisme merupakan tindakan mengambil karya orang lain dan mengakuinya sebagai karyanya sendiri. Aplikasi ini dikembangkan dengan menggunakan metode analisis leksikal dan Jaccard similarity coefficient. Metode analisis leksikal digunakan untuk merubah kode menjadi token unik. Token ini yang akan digunakan untuk mengidentifikasi struktur dari program yang diperiksa. Terdapat dua pilihan level sensitifitas yang bisa di gunakan, pada level 0 jika token pada source kesatu sama dengan token pada source kedua program akan langsung mencatat adanya kemiripan tanpa memastikan isi dari token tersebut, sedangkan pada Level 1 jika token pada source kesatu sama dengan token pada source kedua, program tidak langsung mencatat sebagai satu kemiripan, melainkan program akan memastikan kembali isi dari token tersebut apakah terdapat juga kemiripan atau tidak, jika ternyata isi dari token berbeda program akan mencatat tidak adanya kemiripan. Nilai presentase kemiripan kode program yang dihasilkan dari proses pencocokan token ini yang nantinya digunakan sebagai acuan penentuan plagiarism dengan ketentuan jika nilai presentase lebih besar dari 30 % maka source dianggap plagiat, namun jika lebih kecil dari 30% tidak dianggap sebagai plagiat.

Kata kunci:

Analisis Leksikal, Source Code Java, Deteksi Plagiarismem, Jaccard similarity coefficient.

MERCU BUANA

ABSTRACT

Name : Muhamad Arif Firmansyah
Student Number : 41516310025
Counsellor : Sri Dianing Asri, ST, M.Kom
Title : Application of Plagiarism Detection Source Code
Java Programing with the Application of Sensitivity
Leveling Using the Lexical Method and the Jaccard
simi-larity coefficient

Plagiarism is the act of taking someone else's work and recognizing it as his own work. This application was developed using the lexical analysis method and the Jaccard similarity coefficient. The lexical analysis method is used to convert the code into a unique token. This token will be used to identify the structure of the program being examined. There are two choices of sensitivity levels that can be used, at level 0 if the token in the first source is the same as the token in the source, the two programs will immediately record the similarity without ensur-ing the contents of the token, while at Level 1 if the token in the first source is the same as the token in the source. The second source, the program does not immediately record as similarity, but the program will reassure the contents of the token whether there is similarity or not, if it turns out that the contents of the tokens are different, the program will record no similarities. The percentage val-ue is similar to the program code generated from the token matching process which will later be used as a reference for determining plagiarism provided that if the percentage value is greater than 30% then the source is considered plagia-rism, but if it is less than 30% it is not considered plagiarism.

Keywords:

Lectic Analysis, Java Source Code, Plagiarismem Detection, Jaccard similarity coefficient

MERCU BUANA

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah yang telah melimpahkan rahmat serta berkat-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini. Penulisan tugas akhir ini ditujukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik Jurusan Teknik Informatika.

Terselesaikannya penulisan tugas akhir ini tidak lepas dari peran serta beberapa pihak, baik secara langsung maupun secara tidak langsung. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada pihak-pihak yang telah ikut membantu dalam penulisan tugas akhir ini, baik dalam memberikan bimbingan, petunjuk, kerjasama, kritikan, maupun saran antara lain kepada:

1. Ibu Desi Ramayanti, S.Kom, MT selaku Ketua Program Studi Teknik Informatika.
2. Bapak Diky Firdaus, S.Kom, MM selaku koordinator Tugas Akhir Program Studi Teknik Informatika.
3. Ibu Sri Dianing Asri, ST, M.Kom selaku Dosen Pembimbing, yang telah banyak membantu terutama dalam memberikan bimbingan, dukungan dan perhatian, sehingga penulis dapat menyelesaikan laporan tugas akhir ini.
4. Kepada Orang Tua tersayang, dan keluarga yang selalu memberikan motivasi serta do'anya dalam pembuatan tugas akhir ini.
5. Seluruh civitas akademika Teknik Informatika di Universitas Mercu Buana, yang telah memberikan bekal ilmu selama penulis menimba ilmu.
6. Kepada teman-teman seperjuangan Universitas Mercu Buana yang tidak bisa semua penulis sebutkan satu persatu.

Akhir kata, penulis berharap semoga tugas akhir ini dapat bermanfaat bagi kita semua dan menjadi bahan masukan dalam dunia pendidikan terutama dalam perkembangan teknologi.

Jakarta, 21 Mei 2020

Muhamad Arif Firmansyah

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS.....	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR ...	iii
SURAT PERNYATAAN LUARAN TUGAS AKHIR.....	iv
LEMBAR PERSETUJUAN PEMBIMBING	vi
LEMBAR PERSETUJUAN PENGUJI.....	vi
LEMBAR PERSETUJUAN PENGUJI.....	vii
LEMBAR PERSETUJUAN PENGUJI.....	viii
LEMBAR PENGESAHAN.....	ix
ABSTRAK.....	x
ABSTRACT	xi
KATA PENGANTAR.....	xii
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
NASKAH JURNAL	1
KERTAS KERJA	A
BAGIAN 1. LITERATUR REVIEW	B
BAGIAN 2 ANALISIS DAN PERANCANGAN	E
BAGIAN 3 SOURCE CODE.....	J
BAGIAN 4 DATASET	LL
BAGIAN 5 TAHAPAN EKSPERIMEN	NN
BAGIAN 6 HASIL SEMUA EKSPERIMEN	TT

DAFTAR GAMBAR

Gambar 1 Flowchart program	4
Gambar 2 Source code yang sudah di-input.....	4
Gambar 3 Hasil scanning LV 0	5
Gambar 4 Hasil scanning LV 1	5
Gambar 5 Metodologi Waterfall.....	F
Gambar 6 Flow Chart.....	H
Gambar 7 Activity Diagram.....	I
Gambar 8 Halaman Awal.....	OO
Gambar 9 Upload File 1	PP
Gambar 10 Upload File 2.....	QQ
Gambar 11 Hasil Scan Menggunakan Level 0	RR
Gambar 12 Hasil Scan Menggunakan Level 1	SS



DAFTAR TABEL

Tabel 1 Daftar nama Class Source 1.....	5
Tabel 2 Daftar nama Class Source 2.....	5
Tabel 3 Hasil perhitungan kemiripan level 0	5
Tabel 4 Hasil perhitungan kemiripan level 1	6
Table 5 Data Token.....	LL
Table 6 Tabel Hasil Skenario Uji Coba Penelitian.....	TT



NASKAH JURNAL

APLIKASI DETEKSI PLAGIARISME SOURCE CODE PADA JAVA PROGRAMING DENGAN PENERAPAN LEVELING SENSITIVITAS MENGGUNAKAN METODE LEKSIKAL

Muhamad Arif Firmansyah¹, Ade Fathudin²;
Jurusan Teknik Informatika¹
Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta
Email : afirmansyahx@gmail.com

Abstrak

Plagiarisme merupakan tindakan mengambil karya orang lain dan mengakuinya sebagai karyanya sendiri. Aplikasi ini dikembangkan dengan menggunakan metode *analisis leksikal* dan Jaccard similarity coefficient. Metode *analisis leksikal* digunakan untuk merubah kode menjadi token unik. Token ini yang akan digunakan untuk mengidentifikasi struktur dari program yang diperiksa. Terdapat dua pilihan level sensitifitas yang bisa di gunakan, pada level 0 jika token pada *source* kesatu sama dengan token pada *source* kedua program akan langsung mencatat adanya kemiripan tanpa memastikan isi dari *token* tersebut, sedangkan pada Level 1 jika token pada *source* kesatu sama dengan token pada *source* kedua, program tidak langsung mencatat sebagai satu kemiripan, melainkan program akan memastikan kembali isi dari token tersebut apakah terdapat juga kemiripan atau tidak, jika ternyata isi dari token berbeda program akan mencatat tidak adanya kemiripan. Nilai presentase kemiripan kode program yang dihasilkan dari proses pencocokan token ini yang nantinya digunakan sebagai acuan penentuan *plagiarism* dengan ketentuan jika nilai presentase lebih besar dari 30 % maka *source* dianggap *plagiat*, namun jika lebih kecil dari 30% tidak dianggap sebagai plagiat.

Kata Kunci: Analisis Leksial, Source Code Java, Deteksi Plagiarismem, Jaccard similarity coefficient

Abstract

Plagiarism is the act of taking someone else's work and recognizing it as his own work. This application was developed using the lexical analysis method and the Jaccard similarity coefficient. The lexical analysis method is used to convert the code into a unique token. This token will be used to identify the structure of the program being examined. There are two choices of sensitivity levels that can be used, at level 0 if the token in the first source is the same as the token in the source, the two programs will immediately record the similarity without ensuring the contents of the token, while at Level 1 if the token in the first source is the same as the token in the source. The second source, the program does not immediately record as similarity, but the program will reassure the contents of the token whether there is similarity or not, if it turns out that the contents of the tokens are different, the program will record no similarities. The percentage value is similar to the program code generated from the token matching process which will later be used as a reference for determining plagiarism provided that if the percentage value is greater than 30% then the source is considered plagiarism, but if it is less than 30% it is not considered plagiarism.

Keywords: Lectic Analysis, Java Source Code, Plagiarismem Detection, Jaccard similarity coefficient

1. PENDAHULUAN

Plagiat didefinisikan sebagai tindakan mengambil “karangan [i.e., karya] ... orang lain dan menjadikannya seolah-olah karangan [i.e., karya] ... sendiri” (“Kamus Besar Bahasa Indonesia (KBBI) Online”, 2019). Plagiarisme juga bisa terjadi pada kode program komputer.

Plagiarisme memiliki dampak kognitif bagi mahasiswa, yakni, plagiarisme membuat mahasiswa menjadi malas berpikir dan mengembangkan kemampuan intelektualnya (Arista and Listyani, 2015). Selain itu plagiator akhirnya cenderung tidak mengandalkan kemampuan dirinya lagi (Nursalam, Bani, and Munirah, 2016). Oleh karena itu tindakan plagiat sangat berbahaya bagi mahasiswa tingkat awal bahkan sampai tingkat akhir.

Mendeteksi plagiarisme tidak mudah, begitu pula ketika mendeteksi plagiarisme pada kode program. Mendeteksi secara manual, yaitu pemeriksaan yang dilakukan secara visual dengan membandingkan satu kode program dengan lainnya, memakan waktu yang cukup banyak, rentan terhadap banyak variasi subjektivitas, dan rentan terhadap inkonsistensi.

Aplikasi yang sudah ada mempunyai tingkat sensitivitas yang tinggi, sehingga tidak cocok dengan karakteristik plagiarisme. Selain itu, aplikasi dibutuhkan untuk mencegah adanya sifat subjektif, inkonsistensi dalam pemberian vonis plagiarisme, dan inkonsistensi terhadap memberi definisi plagiarisme.

2. METODE PENELITIAN

2.1 Analisis Leksikal

Analisis leksikal merupakan fungsi analisis dalam *compiler* yang bertugas mendekomposisi program sumber menjadi bagian-bagian kecil (*token*). *Analisis leksikal* atau *scanner* bertugas mengidentifikasi semua besaran pembangun bahasa (leksikal) yang ada pada *source code*. *Scanner* menerima masukan *source code* berupa serangkaian karakter kemudian memilah-milahnya menjadi bagian - bagian kecil yang disebut token. *Token-token* ini akan menjadi masukan bagi analisis selanjutnya yaitu analisis *sintaksis*. Dari fungsi *scanner* secara umum seperti telah disebutkan di atas, maka tugas *scanner* secara rinci adalah:

1. Membaca serangkaian karakter dari *source code*.
2. Mengenalinya ke dalam satuan leksikal.
3. Mengubahnya menjadi *token* dan menentukan jenis *tokennya*.
4. Mengirimkan *token* ke proses analisis selanjutnya, yaitu analisis *sintaksis*.
5. Mengabaikan karakter *white space* (spasi, *enter*, ganti baris, penanda akhir *file*) dan komentar (*remark*) apabila ada di dalam *source code*.
6. Menangani *error*.
7. Menangani tabel simbol

Scanner bekerja berdasarkan prinsip mesin *Finite State Automata* (FSA). Diagram keadaan dapat digunakan untuk membantu mengkonstruksi *scanner*. Besaran pembangun bahasa (leksikal) meliputi:

1. *Identifier*

Identifier atau pengenal dapat berupa:

- a. Kata tercadang atau kata kunci (*reserve word*) yang telah didefinisikan oleh bahasa pemrograman.
- b. Kata yang dideklarasikan sendiri oleh pemakai (si pembuat program) disebut dengan variabel.

2. Nilai *konstanta*

Nilai *konstanta* disini dapat berupa: *integer*, *real*, *boolean*, *character*, *String*, dan sebagainya.

3. *Operator* dan *Delimiteter*
 - a. *Operator* aritmatika (+, -, *, /

- b. *Operator* logika (<, =, >, <=, >=, !=, <>)
- c. *Delimiter* berguna sebagai pemisah atau pembatas, contoh:
- d. Karakter sebagai berikut () { } ; , :
- e. Karakter *white space*.

2.2 Jaccard similarity coefficient

Metode ini digunakan untuk menghitung tingkat kemiripan dua himpunan. Perhitungan *Jaccard similarity coefficient* menghasilkan nilai similarity antara nol sampai satu, dimana semakin besar nilainya. Hasil dari perhitungan akan dihasilkan nilai similaritas dokumen.maka semakin besar pula kemungkinan plagiarisme antara kedua kode tersebut. Formula dari *jaccard similarity coefficient* adalah.

$$\text{Similarity}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \times 100$$

2.3 Perhitungan nilai kemiripan

Saat kedua file sudah dibentuk kedalam *token token* maka proses berlanjut kedalam tahap perhitungan nilai kemiripan dari kedua *source*.

Metode perhitungan kemiripan berdasarkan posisi baris *token* yang sama pada kedua buah file yang di cocokan, pada Level 1 jika token pada *source* kesatu sama dengan token pada *source* kedua, program tidak langsung mencatat sebagai satu kemiripan melainkan program akan memastikan kembali isi dari token tersebut apakah terdapat juga kemiripan atau tidak, jika ternyata isi dari token berbeda program akan mencatat tidak adanya kemiripan, Sedangkan pada Level 0 jika token pada *source* kesatu sama dengan token pada *source* kedua program akan langsung mencatat adanya kemiripan tanpa memastikan isi dari *token* tersebut.

Jika baris token pada kedua *source* berbeda karena jumlah token yang terbentuk pada kedua *source* berbeda, pencocokan akan dilakukan pada baris token terpendek dan sisanya akan dianggap sebagai ketidakmiripan.

Menentukan nilai kemiripan menggunakan Jaccard similarity coefficient.

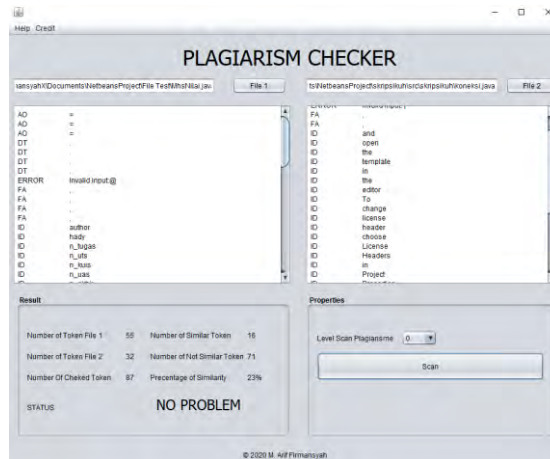
$$\text{Similarity}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \times 100$$

2.4 Analisa kebutuhan sistem

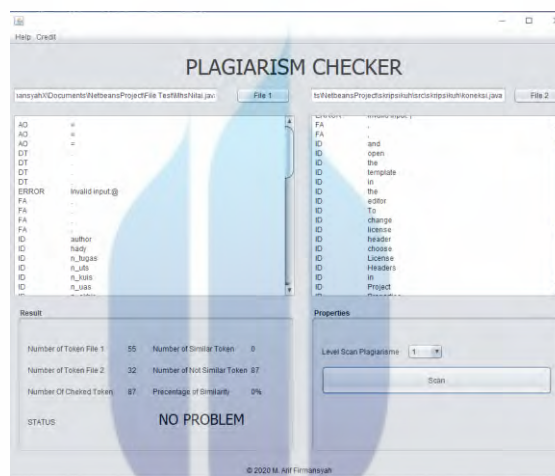
Data yang dibutuhkan pada system ini adalah adanya file class java, data diperoleh dari dosen kuliah Pemrograman Berorientasi Objek (PBO), data ini yang nantinya akan dihitung nilai presentasinya menggunakan metode *analisis leksikal* dan pencocokan string.

2.5 Arsitektur

Flowchart dari proses system secara keseluruhan bisa dilihat pada gambar berikut.



Gambar 3 Hasil scanning LV 0



Gambar 4 Hasil scanning LV 1

3.2 Hasil Analisis

Pengujian sistem dilakukan dengan memeriksa dua source code yang telah di input ke program. Pengujian berlanjut kepada 6 file class yang sudah disiapkan, daftar file class yang di uji ditunjukkan pada Tabel 1 dan Tabel 2

Tabel 1 Daftar nama Class Source 1

No	Nama Class	TOTAL TOKEN
1	Mahdi_nilai.java	39
2	Fery_nilai.java	24
3	Nasa_nilai.java	55

Tabel 2 Daftar nama Class Source 2

No	Nama Class	TOTAL TOKEN
1	Nindi_nilai.java	19
2	Fattah_nilai.java	55
3	Eca_koneksi.java	32

Tabel 3 Hasil perhitungan kemiripan level 0

No	TOKEN SAMA	TOKEN TIDAK SAMA	KEMIRIPAN
1	2	56	4%

2	1	78	1%
3	16	71	23%

Tabel 4 Hasil perhitungan kemiripan level 1

No	TOKEN SAMA	TOKEN TIDAK SAMA	KEMIRIPAN
1	2	56	4%
2	1	78	1%
3	0	87	0%

Tabel 3 menunjukkan terdapat 3 hasil pengujian Level 0 sedangkan pada Tabel 4 menunjukkan terdapat 3 hasil pengujian Level 1.

4. KESIMPULAN

Berdasarkan pembahasan dan pengujian terhadap Aplikasi Deteksi Plagiarisme Source Code Pada Java sebelumnya, penulisan ini mengimplementasikan proses dan menghasilkan beberapa kesimpulan yaitu :

1. Dengan aplikasi ini pemeriksaan *source code* secara visual menjadi otomatis.
2. Hasil dari aplikasi Mendapatkan nilai persentase kemiripan yang bisa digunakan sebagai acuan telah terjadinya tindakan plagiarisme.

5. SARAN

Dalam pembuatan Aplikasi Deteksi Plagiarisme Source Code pada Java Programming Dengan Penerapan Leveling Sensitivitas Menggunakan Metode Leksikal masih terdapat banyak kekurangan dan jauh dari kata sempurna. Maka dari itu masih perlu banyak penyempurnaan. Berikut beberapa saran untuk pengembangan dari aplikasi ini :

1. *File* yang dapat diperiksa aplikasi ini hanya dua buah, agar lebih menghemat waktu kedepannya pemeriksaan bisa dilakukan lebih dari dua buah *file* yang di input.
2. *File* yang bisa diperiksa pada aplikasi ini hanya *file class.java* saja, pengembangan selanjutnya dapat ditingkatkan untuk memeriksa satu aplikasi full dari *project* tersebut dan menambahkan Level lain dengan metode pengecekan yang berbeda.

DAFTAR PUSTAKA

- [1] Liliana, Gregorius Satia Budhi, Anthony Wibisono, and Ricky Tanojo. 2012. "Pengecekan Plagiarisme Pada Code Dalam Bahasa C++." *Jurnal Informatika* 11 (1): 70–78. <https://doi.org/10.9744/informatika.11.1.70-78>.
- [2] Mirza, Olfat M., Mike Joy, and Georgina Cosma. 2017. "Suitability of BlackBox Dataset for Style Analysis in Detection of Source Code Plagiarism." In *7th International Conference on Innovative Computing Technology, INTECH 2017*, 90–94. IEEE. <https://doi.org/10.1109/INTECH.2017.8102424>.
- [3] Kasmala, Raras, Arif Budimansyah, and U. Tresna Lenggana. 2017. "Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) Dan Algoritma Huffman." *Jurnal Online Informatika* 2 (1): 1. <https://doi.org/10.15575/join.v2i1.79>.

- [4] Nursalam, Nursalam, Suddin Bani, and Munirah Munirah. 2016. "Bentuk Kecurangan Akademik (Academic Cheating) Mahasiswa PGMI Fakultas Tarbiyah Dan Keguruan UIN Alauddin Makassar." *Jurnal Ilmu Tarbiyah Dan Keguruan* 16 (2): 127-38. <https://doi.org/10.24252/lp.2013v16n2a1>.
- [5] Salmuasih, Sunyoto,A., 2013, Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity, Program Studi teknikInformatika, STMIK AMIKOM Yogyakarta, Yogyakarta, pp. 23-28.
- [6] Putra,D.A., Sijaini,H., dan Pratiwi,H.S., 2015, Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah, Program Studi Teknik Informatika, Fakultas Teknik, Universitas Tanjungpura, Vol.1, No.1.
- [7] Firdaus,A., Ernawati dan Vatesia,A., 2014, Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani Dan Metode Cosine Similarity. Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu, Bengkulu, Vol.10, No.1, pp. 97-110.
- [8] Roy, C K, and J R Cordy. 2007. "A Survey on Software Clone Detection Research." *Computing* 115 (2007-541): 115. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.7869>.
- [9] Ling, Juen, I Putu Eka N. Kencana, and Tjokorda Bagus Oka. 2014. "Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square." *E-Jurnal Matematika* 3 (3): 92. <https://doi.org/10.24843/MTK.2014.v03.i03.p070>.
- [10] "Kamus Besar Bahasa Indonesia (KBBI) Online." 2019. Kamus Besar Bahasa Indonesia (KBBI) Online. 2019. <https://kbbi.web.id/plagiat>.



KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan:

1. Literatur review.
2. Analisis dan perancangan.
3. Source code.
4. Hasil pengambilan data.
5. Tahapan eksperimen.
6. Hasil semua eksperiment

