

**PENGARUH ABSTRACT SYNTAX TREE PADA SISTEM  
DETEKSI KEMIRIPAN KODE SUMBER MENGGUNAKAN  
ALGORITMA RATCLIFF/OBERSHELP**

Diajukan Sebagai Syarat Untuk Menyelesaikan  
Pendidikan Program Strata-1 pada  
Jurusan Teknik Informatika



Oleh:

Muhammad Sholeh  
NIM: 09021281823172

Jurusan Teknik Informatika  
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA  
2022

## LEMBAR PENGESAHAN SKRIPSI

### PENGARUH ABSTRACT SYNTAX TREE PADA SISTEM DETEKSI KEMIRIPAN KODE SUMBER MENGGUNAKAN ALGORITMA RATCLIFF/OBERSHELP

Oleh:

Muhammad Sholeh  
NIM: 09021281823172

Indralaya, 22 Maret 2022

Pembimbing I,



Dr. Abdiansah, S.Kom., M.Cs.  
NIP. 198410012009121005

Pembimbing II,



Alvi Syahrini Utami, M.Kom.  
NIP. 19781222200642003

Mengetahui,



## TANDA LULUS UJIAN SIDANG SKRIPSI

Pada hari Selasa tanggal 22 Maret 2022 telah dilaksanakan ujian sidang ujian sidang skripsi oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya

Nama : Muhammad Sholeh

NIM : 09021281823172

Judul : Pengaruh Abstract Syntax Tree Pada Sistem Deteksi Kemiripan Kode Sumber Menggunakan Algoritma Ratcliff/Obershelp

1. Ketua Penguji

Dian Palupi Rini, M.Kom., Ph.D.  
NIP. 197802232006042002

2. Pembimbing I

Dr. Abdiansah, S.Kom., M.Cs  
NIP. 198410012009121005

3. Pembimbing II

Alvi Syahrini Utami, M.Kom.  
NIP. 197812222006042003

4. Penguji I

Novi Yusliani, S.Kom., M.T.  
NIP. 198211082012122001

5. Penguji II

M Naufal Rachmatullah S.Kom., M.T.  
NIP. 167106011292006

Mengetahui,



Alvi Syahrini Utami, M.Kom.

NIP. 197812222006042003

## HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Sholeh

NIM : 09021281823172

Jurusan : Teknik Informatika

Judul Skripsi : Pengaruh Abstract Syntax Tree Pada Sistem Deteksi Kemiripan Kode Sumber Menggunakan Algoritma Ratcliff/Obershelp

Hasil pengecekan *Software iThenticate/Turnitin* : 10%

Menyatakan bahwa Laporan Proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapa pun.



Indralaya, 22 Maret 2022



Muhammad Sholeh  
NIM. 09021281823172

## **MOTTO DAN PERSEMBAHAN**

*“It does not matter how slowly you go as long as you don’t stop”*

*– Conficius*

Kupersembahkan karya tulis ini kepada:

- Allah Subhanahu Wa Ta ’ala dan Rasulullah SAW
- Ummiii dan Abuya
- Adikku Haniyah, Karimah dan Adibah
- Dosen pembimbing Akademik dan Skripsi Saya
- Universitas Sriwijaya
- Sahabat dan teman-teman seperjuangan

**THE ABSTRACT SYNTAX TREE EFFECTS IN  
SOURCE CODE SIMILARITY DETECTION SYSTEM USING  
RATCLIFF/OBERSHELP ALGORITHM**

By:

Muhammad Sholeh (09021281823172)

Department of Informatics, Faculty of Computer Science, Sriwijaya University

Email: [muhammadsholeh.dev@gmail.com](mailto:muhammadsholeh.dev@gmail.com)

**ABSTRACT**

*In today's technological era, source code has become something that is mostly written. Among the diversity of existing programming source code, many of them are the result of acts of plagiarism. Process of detecting plagiarism in the source code is not an easy to do. Based on the information above, this research uses the Abstract Syntax Tree method because of its ability to determine the structure of a source code which is then represented by tokens. The Ratcliff/Obershelp algorithm was chosen to perform similarity calculations based on the token structure of the source code. The data used in this study amounted to 25 source code documents with details of 5 main source code documents and 20 modified source code documents. The results of this study indicate that the Abstract Syntax Tree method affects the calculation of the similarity of the source code, and also the developed system can detect the similarity of the source code using the Ratcliff/Obershelp algorithm.*

*Keywords: Abstract Syntax Tree, Similarity Detection, Source Code, Plagiarism, Ratcliff/Obershelp*

**PENGARUH ABSTRACT SYNTAX TREE PADA SISTEM DETEKSI  
KEMIRIPAN KODE SUMBER MENGGUNAKAN  
ALGORITMA RATCLIFF/OBERSHELP**

Oleh:

Muhammad Sholeh (09021281823172)

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email: muhammadsholeh.dev@gmail.com

**ABSTRAK**

Dalam perkembangan era teknologi saat ini, kode sumber menjadi hal yang banyak dibuat atau dituliskan. Diantara diversitas kode sumber pemrograman yang ada, tidak sedikit diantaranya merupakan hasil tindak kecurangan plagiarisme. Proses deteksi adanya tindakan plagiarisme dalam kode sumber bukanlah merupakan sesuatu yang mudah. Berdasarkan keterangan diatas, penelitian ini menggunakan metode *Abstract Syntax Tree* karena kemampuannya untuk mengetahui struktur dari sebuah kode sumber yang kemudian direpresentasikan melalui token. Algoritma Ratcliff/Obershelp dipilih untuk melakukan perhitungan kemiripan berdasarkan token struktur kode sumber. Data yang digunakan dalam penelitian berjumlah 25 dokumen kode sumber dengan rincian 5 dokumen kode sumber utama dan 20 dokumen kode sumber modifikasi. Hasil penelitian ini menunjukkan bahwa metode *Abstract Syntax Tree* mempengaruhi perhitungan kemiripan kode sumber, dan juga sistem yang dikembangkan dapat melakukan deteksi kemiripan kode sumber menggunakan algoritma Ratcliff/Obershelp.

Kata Kunci: *Abstract Syntax Tree*, Deteksi Kemiripan, Kode Sumber, Plagiarisme Ratcliff/Obershelp

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Allah SWT, Tuhan Semesta Alam atas berkat, rahmat, rahim dan karunia-Nya yang telah diberikan kepada penulis sehingga Tugas Akhir berjudul “Pengaruh Abstract Syntax Tree Pada Sistem Deteksi Kemiripan Kode Sumber Menggunakan Algoritma Ratcliff/Obershelp” dapat disusun dengan baik sebagai syarat dalam menyelesaikan studi Strata-1 program studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Sriwijaya. Pada kesempatan ini penulis ingin mengucapkan ucapan terima kasih yang luar biasa besar kepada semua pihak yang memberikan dukungan, motivasi dan bimbingan selama penyusunan Tugas Akhir dan penelitian. Secara khusus ucapan terima kasih ini ditujukan kepada:

1. Allah Subhanallahu Wa Ta’ala atas segala berkah, rahmat, rahim dan karunia-Nya.
2. Keluarga tercinta, Ummi dan Abuya yang selalu memberikan semangat dan dukungan positif.
3. Adik-adikku Haniyah, Karimah dan Adibah yang selalu menemani dan membantu dalam banyak hal.
4. Bapak Jaidan Jauhari, M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Ibu Alvi Syahrini Utami, M.Kom. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

6. Bapak Dr. Abdiansah, S.Kom., M.Cs. selaku Dosen Pembimbing I yang senantiasa membagikan ilmu, membimbing, memberi arahan dan memotivasi saya untuk menyelesaikan Tugas Akhir ini dengan baik.
7. Ibu Alvi Syahrini Utami, M.Kom. yang dalam hal ini selaku Dosen Pembimbing II yang senantiasa membagikan ilmu, membimbing, memberi arahan dan memotivasi saya untuk menyelesaikan Tugas Akhir ini dengan baik.
8. Ibu Dian Palupi Rini, M.Kom., Ph.D. selaku ketua penguji Tugas Akhir yang telah memberikan nasihat dan saran yang membangun
9. Ibu Novi Yusliani S.Kom., M.T. selaku penguji I Tugas Akhir yang telah memberikan nasihat dan saran yang membangun.
10. Bapak Muhammad Naufal Rachmatullah S.Kom., M.T. selaku penguji II Tugas Akhir yang telah memberikan nasihat dan saran yang membangun.
11. Ibu Rizki Kurniati, M.T. selaku Dosen Pembimbing Akademik yang senantiasa memberikan saran, arahan dan bimbingan kepada saya selama masa perkuliahan.
12. Seluruh Dosen Jurusan Teknik Informatika, juga Dosen Fakultas Ilmu Komputer yang telah banyak memberikan bekal, arahan, dan saran selama masa perkuliahan.
13. Seluruh Staf Administrasi dan Pegawai yang telah membantu dalam urusan administrasi.

14. Sahabat – sahabatku: Febriansyah, Igarsa, Agung, Firmansyah, Renaldi, dan Nabila, serta seluruh teman seperjuangan Teknik Informatika Angkatan 2018 yang selalu sedia menjadi rekan diskusi baik dalam penggerjaan Tugas Akhir maupun selama masa perkuliahan.

Penulis secara penuh menyadari akan kekurangan dalam penyusunan Tugas Akhir ini. Kekurangan ini semata – mata adalah karena keterbatasan pengetahuan juga pengalaman penulis. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan Tugas akhir ini sehingga dapat membawa manfaat lebih banyak dan luas. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi kita semua.

Indralaya, 21 Februari 2022

Muhammad Sholeh

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL .....</b>	i
<b>LEMBAR PENGESAHAN SKRIPSI .....</b>	ii
<b>TANDA LULUS UJIAN SIDANG SKRIPSI.....</b>	iii
<b>HALAMAN PERNYATAAN .....</b>	iv
<b>MOTTO DAN PERSEMBAHAN.....</b>	v
<b>ABSTRACT .....</b>	vi
<b>ABSTRAK .....</b>	vii
<b>KATA PENGANTAR.....</b>	viii
<b>DAFTAR ISI.....</b>	xi
<b>DAFTAR TABEL.....</b>	xiv
<b>DAFTAR GAMBAR.....</b>	xv
<b>DAFTAR LAMPIRAN .....</b>	xvi
<b>BAB I PENDAHULUAN.....</b>	I-1
1.1. Pendahuluan.....	I-1
1.2. Latar Belakang .....	I-1
1.3. Rumusan Masalah.....	I-3
1.4. Tujuan Penelitian .....	I-3
1.5. Manfaat Penelitian .....	I-3
1.6. Batasan Masalah .....	I-4
1.7. Sistematika Penulisan .....	I-4
1.8. Kesimpulan .....	I-6
<b>BAB II KAJIAN LITERATUR .....</b>	II-1
2.1. Pendahuluan.....	II-1
2.2. Landasan Teori.....	II-1
2.2.1. Plagiarisme Kode Sumber.....	II-1
2.2.2. <i>Text Mining</i> .....	II-3
2.2.3. Pra-pengolahan Teks .....	II-3
2.2.4. <i>Abstract Syntax Tree</i> .....	II-4
2.2.5. ANTLR ( <i>Another Tool for Language Recognition</i> ) .....	II-5

2.2.6. <i>Mean Absolute Error</i> .....	II-6
2.2.7. Algoritma Ratcliff/Obershelp.....	II-7
2.2.8. <i>Rational Unified Process</i> .....	II-7
2.3. Penelitian Lain yang Relevan .....	II-9
2.4. Kesimpulan .....	II-12
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>III-1</b>
3.1. Pendahuluan.....	III-1
3.2. Pengumpulan Data.....	III-1
3.2.1. Jenis dan Sumber Data .....	III-1
3.2.2. Metode Pengumpulan Data .....	III-2
3.3. Tahapan Penelitian .....	III-3
3.3.1. Mengumpulkan Data .....	III-3
3.3.2. Menentukan Kerangka Kerja Penelitian .....	III-4
3.3.3. Menentukan Kriteria Pengujian .....	III-6
3.3.4. Menentukan Format Data Pengujian.....	III-6
3.3.5. Menentukan Alat Bantu Penelitian.....	III-8
3.3.6. Melakukan Pengujian Penelitian.....	III-9
3.3.7. Melakukan Analisis dan Kesimpulan Hasil Pengujian Penelitian.....	III-11
3.4. Metode Pengembangan Perangkat Lunak.....	III-12
3.4.1. Fase Insepsi .....	III-12
3.4.2. Fase Elaborasi .....	III-13
3.4.3. Fase Konstruksi .....	III-13
3.4.4. Fase Transisi.....	III-14
3.5. Kesimpulan .....	III-14
<b>BAB IV PENGEMBANGAN PERANGKAT LUNAK .....</b>	<b>IV-1</b>
4.1. Pendahuluan.....	IV-1
4.2. Fase Insepsi .....	IV-1
4.2.1. Pemodelan Sistem .....	IV-1
4.2.2. Kebutuhan .....	IV-2
4.2.3. Analisis dan Perancangan.....	IV-5
4.2.4. Implementasi .....	IV-13

4.3.	Fase Elaborasi .....	IV-20
4.3.1.	Pemodelan Sistem .....	IV-20
4.3.2.	Kebutuhan .....	IV-21
4.3.3.	Analisis dan Perancangan.....	IV-22
4.4.	Fase Konstruksi.....	IV-28
4.4.1.	Kebutuhan .....	IV-28
4.4.2.	Implementasi .....	IV-29
4.5.	Fase Transisi.....	IV-31
4.5.1.	Pemodelan Sistem .....	IV-31
4.5.2.	Kebutuhan .....	IV-31
4.5.3.	Analisis dan Perancangan.....	IV-32
4.5.4.	Implementasi .....	IV-34
4.6.	Kesimpulan .....	IV-37
<b>BAB V</b>	<b>HASIL DAN ANALISIS PENELITIAN .....</b>	<b>V-1</b>
5.1.	Pendahuluan.....	V-1
5.2.	Hasil Program .....	V-1
5.3.	Data Hasil Penelitian.....	V-3
5.3.1.	Konfigurasi Percobaan .....	V-3
5.3.2.	Data Hasil Konfigurasi I .....	V-4
5.3.3.	Data Hasil Konfigurasi II .....	V-5
5.4.	Analisis Hasil Penelitian .....	V-7
5.4.1.	Analisis Hasil Perhitungan Algoritma Ratcliff/Obershelp .....	V-8
5.4.2.	Analisis Hasil Perhitungan Menggunakan Abstract Syntax Tree.....	V-8
5.5.	Kesimpulan .....	V-10
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>VI-1</b>
6.1.	Pendahuluan.....	VI-1
6.2.	Kesimpulan .....	VI-1
6.3.	Saran .....	VI-2
<b>DAFTAR PUSTAKA .....</b>	<b>xvii</b>	

## DAFTAR TABEL

Tabel III-1. Pengukuran Tingkat Kemiripan Kode Sumber.....	III-7
Tabel III-2. Tabel Bantuan Pengukuran Kesalahan Absolut .....	III-7
Tabel III-3. Hasil Pengukuran Tingkat Kemiripan Kode Sumber .....	III-11
Tabel III-4. Hasil Tabel Bantuan Pengukuran Kesalahan Absolut.....	III-12
Tabel IV-1. Kebutuhan Fungsional Sistem .....	IV-2
Tabel IV-2. Kebutuhan Non-Fungsional Sistem .....	IV-2
Tabel IV-3. Cuplikan Hasil Tahap Pra-pengolahan.....	IV-7
Tabel IV-4. Cuplikan ID Tipe Token.....	IV-8
Tabel IV-5. Cuplikan Hasil Tahap Tokenisasi .....	IV-8
Tabel IV-6. Cuplikan Hasil Perhitungan Tingkat Kemiripan.....	IV-13
Tabel IV-7. Tabel Definisi Aktor.....	IV-14
Tabel IV-8. Tabel Definisi <i>Use-Case</i> .....	IV-14
Tabel IV-9. Skenario Memasukkan Dokumen Kode Sumber.....	IV-15
Tabel IV-10. Skenario Memproses Dokumen Kode Sumber.....	IV-17
Tabel IV-11. Skenario Melakukan Perhitungan Kemiripan Kode Sumber.....	IV-19
Tabel IV-12. Tabel Implementasi Kelas .....	IV-30
Tabel IV-13. Rencana Pengujian <i>Use-Case</i> Memasukkan Kode Sumber.....	IV-32
Tabel IV-14. Rencana Pengujian <i>Use-Case</i> Memproses Data Menggunakan metode <i>Abstract Syntax Tree</i> .....	IV-33
Tabel IV-15. Rencana Pengujian <i>Use-Case</i> Melakukan deteksi kemiripan kode sumber menggunakan metode AST dan Algoritma Ratcliff/Obershelp.....	IV-33
Tabel IV-16. Pengujian <i>Use-Case</i> Memasukkan Kode Sumber .....	IV-34
Tabel IV-17. Pengujian <i>Use-Case</i> Memproses Data Menggunakan Metode Abstact Syntax Tree .....	IV-35
Tabel IV-18. Pengujian <i>Use-Case</i> Perhitungan Kemiripan Kode Sumber Menggunakan Algoritma Ratcliff/Obershelp .....	IV-36
Tabel V-1. Hasil Pengujian Menggunakan Konfigurasi I .....	V-4
Tabel V-2. Hasil Pengujian Menggunakan Konfigurasi II.....	V-6
Tabel V-3. Hasil Pengukuran Kesalahan Absolut Dengan Konfigurasi II .....	V-7

## DAFTAR GAMBAR

Gambar II-1. Contoh <i>Abstract Syntax Tree</i> dari Potongan Kode Sumber .....	II-4
Gambar II-2. Arsitektur <i>Rational Unified Process</i> .....	II-6
Gambar III-1. Alur Tahapan Penelitian.....	III-3
Gambar III-2. Diagram Alir Sistem Deteksi Kemiripan Kode Sumber.....	III-4
Gambar III-3. Contoh Pohon Sintaks Hasil Proses <i>Parsing</i> dengan ANTLR..	III-10
Gambar IV-1. Alur Kerja Fitur Masukan Kode Sumber.....	IV-3
Gambar IV-2. Alur Kerja Fitur Pemrosesan Kode Sumber.....	IV-4
Gambar IV-3. Alur Kerja Fitur Perhitungan Kemiripan Antar Kode Sumber ...	IV-5
Gambar IV-4. Diagram <i>Use-Case</i> .....	IV-13
Gambar IV-5. Rancangan Tampilan Antarmuka Sistem.....	IV-21
Gambar IV-6. Diagram Aktivitas Memasukkan Dokumen Kode Sumber.....	IV-23
Gambar IV-7. Diagram Aktivitas Melakukan Pemrosesan Kode Sumber Menggunakan Metode <i>Abstract Syntax Tree</i> .....	IV-24
Gambar IV-8. Diagram Aktivitas Melakukan Perhitungan Kemiripan Dengan algoritma Ratcliff/Obershelp.....	IV-25
Gambar IV-9. Diagram Alur Memasukkan Kode Sumber.....	IV-26
Gambar IV-10. Diagram Alur Melakukan Pemrosesan Kode Sumber Menggunakan Metode <i>Abstract Syntax Tree</i> .....	IV-27
Gambar IV-11. Diagram Alur Melakukan Perhitungan Kemiripan Dengan algoritma Ratcliff/Obershelp.....	IV-28
Gambar IV-12. Diagram Kelas Sistem Yang Dikembangkan.....	IV-29
Gambar IV-13. Tampilan Antarmuka Sistem.....	IV-31
Gambar V-1. Tampilan Fitur Masukan Kode Sumber .....	V-2
Gambar V-2. Tampilan Fitur Pemrosesan Kode Sumber .....	V-2
Gambar V-3. Tampilan Fitur Perhitungan Kemiripan.....	V-3
Gambar V-4. Grafik Hasil Pengujian Konfigurasi I.....	V-5
Gambar V-5. Grafik Hasil Pengujian Konfigurasi II .....	V-6

## **DAFTAR LAMPIRAN**

1. Hasil Tahapan Pra-pengolahan
2. Hasil Tahapan Tokenisasi
3. Kode Program

# **BAB I**

## **PENDAHULUAN**

### **1.1. Pendahuluan**

Pada bab ini akan dibahas berkenaan dengan garis besar pokok-pokok pikiran dalam penelitian ini. Pokok pikiran yang akan dibahas antara lain latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian. Pokok-pokok pikiran yang diuraikan akan dijadikan acuan dalam kajian penelitian ini.

### **1.2. Latar Belakang**

Perkembangan teknologi dan informasi saat ini maju dengan sangat pesat sehingga mempermudah manusia dalam bertukar informasi. Seiring dengan kemajuannya, banyak terjadi pula pelanggaran di dalamnya, salah satunya adalah pelanggaran hak cipta berupa plagiarisme. Plagiarisme dapat menyasar ke dalam berbagai bidang, salah satunya pada bidang pendidikan yang berupa plagiarisme terhadap kode sumber. Kode sumber merupakan sekumpulan baris kode berupa pernyataan atau deklarasi dalam bahasa pemrograman komputer yang ditulis dan dapat dibaca oleh manusia (Prasetya dan Jalal, 2014). Plagiarisme terhadap kode sumber merupakan kegiatan membuat sebuah program yang berasal dari program lain dengan transformasi di dalamnya (Kaučič et al., 2010). Transformasi yang dilakukan dapat beragam dari menyalin sebagian kecil kode sumber asli hingga menyalin seluruh bagian dari kode sumber asli.

Dalam penelitian yang dilakukan oleh Sraka dan Kaučič (2009) sebanyak 100 dari 138 (72,5%) mahasiswa yang ikut dalam survei anonim menyatakan bahwa mereka setidaknya pernah melakukan plagiarisme baik terhadap kode sumber ataupun tugas lain setidaknya sekali selama masa studi mereka. Lewat penelitian ini, dapat dikatakan bahwa plagiarisme di tingkat mahasiswa marak terjadi, khususnya plagiarisme terhadap kode sumber yang kemudian akan sangat mempengaruhi kualitas pendidikan ilmu komputer. Untuk mengurangi perkembangan akademik yang buruk, penelitian mengenai deteksi plagiarisme terhadap kode sumber sangat dibutuhkan.

Terdapat beberapa penelitian sebelumnya mengenai deteksi kemiripan kode sumber dengan menggunakan berbagai algoritma. Algoritma yang banyak digunakan diantaranya seperti algoritma Levenshtein, algoritma Jaro-Winkler, algoritma Ratcliff/Obershelp dan *Cosine Similarity*. Selain itu, terdapat pula metode yang dapat membantu atau mempengaruhi proses deteksi adanya kemiripan kode sumber, salah satunya adalah metode *Abstract Syntax Tree* (AST). Metode AST merupakan metode *tree-based* yang dapat melakukan *parsing* kode sumber menjadi sebuah struktur dari suatu kode sumber (Rusdianto dan Chaniago, 2019). Metode ini banyak digunakan dalam pelbagai penelitian seperti yang dilakukan Prasetya dan Jalal (2014), dan Zhang dan Liu (2013).

Algoritma Ratcliff/Obershelp dapat digunakan dalam mendeteksi plagiarisme pada kode sumber. Algoritma ini banyak digunakan dalam mendeteksi kemiripan dokumen khususnya dokumen kode sumber, diantaranya yang dilakukan oleh Joane et al. (2017), Ilyankou (2014), dan (Hidayat et al., 2020). Lewat penelitiannya Ilyankou (2014) menyatakan algoritma

Ratcliff/Obershelp memiliki akurasi deteksi kemiripan yang lebih baik dengan angka 4%-18.6% dibandingkan algoritma Jaro-Winkler dalam menemukan kata yang benar dalam daftar 53 kata yang salah eja.

Dengan adanya referensi penelitian terkait sebelumnya, metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp akan menjadi metode yang digunakan dalam penelitian deteksi kemiripan kode sumber. Diharapkan dengan adanya kombinasi dua metode ini, plagiarisme terhadap kode sumber dapat dideteksi dengan akurasi sistem yang lebih baik.

### **1.3. Rumusan Masalah**

Rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan deteksi kemiripan antar kode sumber menggunakan metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp?
2. Bagaimana kinerja metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp dalam mendeteksi kemiripan antar kode sumber?

### **1.4. Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun perangkat lunak deteksi kemiripan kode sumber menggunakan metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp.
2. Mengukur kinerja metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp dalam mendeteksi kemiripan antar kode sumber.

### **1.5. Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Sistem yang dibuat dapat membantu pengguna untuk mendeteksi adanya plagiarisme pada dokumen kode sumber.
2. Hasil penelitian dapat dijadikan sebagai rujukan untuk penelitian terkait di masa mendatang.

### **1.6. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Kode sumber yang akan diteliti persentase kemiripannya menggunakan bahasa pemrograman Java.
2. Data yang digunakan merupakan kode sumber yang dipergunakan dalam penelitian milik Karnalim et al. (2019). Data berupa tugas mahasiswa dalam mata kuliah pengantar pemrograman.
3. Perhitungan deteksi kemiripan kode sumber dilakukan antara satu dokumen kode sumber dengan satu dokumen kode sumber modifikasi lainnya.
4. Kode sumber masukan merupakan kode sumber yang sudah berjalan dan berfungsi dengan baik.

### **1.7. Sistematika Penulisan**

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

## **BAB I. PENDAHULUAN**

Pada bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian dan batasan masalah. Pokok-pokok pikiran ini akan menjadi dasar dan acuan pengembangan penelitian pada bab selanjutnya.

## **BAB II. KAJIAN LITERATUR**

Pada bab ini dibahas landasan teori yang digunakan di dalam penelitian, termasuk di dalamnya mengenai plagiarisme, *Abstract Syntax Tree*, algoritma Ratcliff/Obershelp, dan penelitian terkait yang relevan.

## **BAB III. METODOLOGI PENELITIAN**

Pada bab ini dibahas proses pengumpulan data dan tahapan-tahapan di dalam penelitian. Tahapan penelitian dibahas lebih rinci berdasarkan kerangka kerja tertentu. Di bagian akhir bab ini akan dimuat rancangan manajemen proyek penelitian.

## **BAB IV. PENGEMBANGAN PERANGKAT LUNAK**

Pada bab ini dibahas mengenai perancangan dan lingkungan implementasi sistem deteksi kemiripan antar kode sumber dengan metode *Abstract Syntax Tree* dan algoritma Ratcliff/Obershelp, hasil eksekusi sistem, dan hasil pengujian.

## **BAB V. HASIL DAN ANALISIS PENELITIAN**

Pada bab ini dibahas mengenai hasil dari implementasi dan pengujian metode yang telah dirancang dan kemudian disajikan. Analisis yang diberikan akan digunakan sebagai dasar dari kesimpulan yang akan diambil di dalam penelitian ini.

## **BAB VI. KESIMPULAN DAN SARAN**

Pada bab ini dibahas mengenai kesimpulan dari keseluruhan uraian yang didapatkan dalam penelitian. Selain itu, disajikan pula saran-saran yang

diharapkan berguna untuk pengembangan sistem deteksi kemiripan antar kode sumber.

### **1.8. Kesimpulan**

Pada Bab ini telah dibahas mengenai latar belakang penelitian serta acuan penting dalam penelitian seperti latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan.

## DAFTAR PUSTAKA

- Agrawal, M., & Sharma, D. K. (2017). A state of art on source code plagiarism detection. *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016, October, 236–241.* <https://doi.org/10.1109/NGCT.2016.7877421>
- Anwar, A. (2014). A Review of RUP (Rational Unified Process). In *Ashraf Anwar International Journal of Software Engineering (IJSE)* (Issue 5).
- Budhy Prasetya, E., & Fadly Dzil Jalal, A. (2014). *Seminar Nasional Sains dan Teknologi 2014 I Fakultas Teknik Universitas Muhammadiyah Jakarta* (Vol. 12).
- Dang, S., & Ahmad, P. H. (2014). Text Mining : Techniques and its Application Text Mining View project Text Mining: Techniques and its Application. *IJETI International Journal of Engineering & Technology Innovations*, 1. [www.ijeti.com](http://www.ijeti.com)
- Firdaus, A., Ernawati, E., & Vatresia, A. (2014). APLIKASI PENDETEKSI KEMIRIPAN PADA DOKUMEN TEKS MENGGUNAKAN ALGORITMA NAZIEF & ADRIANI DAN METODE COSINE SIMILARITY. In *Jurnal Teknologi Informasi* (Vol. 10, Issue 1). <http://research.pps.dinus.ac.id>
- Fischer, G., Lusiardi, J., & Wolff Von Gudenberg, J. (2007). *Abstract Syntax Trees-and their Role in Model Driven Software Development.*
- Gipp, B., & Meuschke, N. (2011). Citation pattern matching algorithms for citation-based plagiarism detection: Greedy citation tiling, citation chunking and longest common citation sequence. *DocEng 2011 - Proceedings of the 2011 ACM Symposium on Document Engineering*, 249–258. <https://doi.org/10.1145/2034691.2034741>
- Ilyankou, I. (2014). *Comparison of Jaro-Winkler and Ratcliff/Obershelp algorithms in spell check.*
- Joane, Y. L., Sinsuw, A., & Jacobus, A. (2017). Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks. *Journal Teknik Informatika*, 11(1).
- Karnalim, O., Budi, S., Toba, H., & Joy, M. (2019). Source code plagiarism detection in academia with information Retrieval: Dataset and the observation. *Informatics in Education*, 18(2), 321–344. <https://doi.org/10.15388/infedu.2019.15>

- Kaučič, B., Sraka, D., Ramsák, M., & Krašna, M. (2010). Observations on plagiarism in programming courses. *CSEDU 2010 - 2nd International Conference on Computer Supported Education, Proceedings*, 2, 181–184.  
<https://doi.org/10.5220/0002800701810184>
- Mulik, S., Shinde, S., & Kapase, S. (2011). *Comparison of Parsing Techniques For Formal Languages*. 3(4), 1611–1615.
- Smith, C. J. (2012). Rules of conduct: plagiarism. *Ethical Behaviour in the E-Classroom*, 45–57. <https://doi.org/10.1016/b978-1-84334-689-0.50005-7>
- Sraka, D., & Kaučič, B. (2009). Source code plagiarism. *Proceedings of the International Conference on Information Technology Interfaces*, ITI, 461–466.  
<https://doi.org/10.1109/ITI.2009.5196127>