

**RANCANG BANGUN APLIKASI DETEKSI PLAGIARISME
MENGUNAKAN ALGORITMA WINNOWER
DENGAN PENDEKATAN TRIWORD**

SKRIPSI

**Oleh :
MAULANA MUHAMMAD EL-SULTHAN
NIM. 16650106**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**RANCANG BANGUN APLIKASI DETEKSI PLAGIARISME
MENGUNAKAN ALGORITMA WINNOWER DENGAN
PENDEKATAN TRIWORD**

SKRIPSI

HALAMAN PENGANTAR

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
MAULANA MUHAMMAD EL-SULTHAN
NIM. 16650106**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN
RANCANG BANGUN APLIKASI DETEKSI PLAGIARISME
MENGGUNAKAN ALGORITMA WINNOWER
DENGAN PENDEKATAN TRIWORD

SKRIPSI

Oleh :
MAULANA MUHAMMAD EL-SULTHAN
NIM. 16650106

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 2020

Dosen Pembimbing I

Dosen Pembimbing II

Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN**RANCANG BANGUN APLIKASI DETEKSI PLAGIARISME
MENGUNAKAN ALGORITMA WINNOWER
DENGAN PENDEKATAN TRIWORD****SKRIPSI****Oleh :****MAULANA MUHAMMAD EL-SULTHAN
NIM. 16650106****Telah Dipertahankan Didepan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal 26 Juni 2020**

Susunan Dewan Penguji		Tanda Tangan
Penguji Utama	: <u>Dr. Muhammad Faisal, M.T</u> NIP. 19740510 200501 1 007	()
Ketua Penguji	: <u>Hani Nurhayati, M.T</u> NIP. 19780625200801 2 006	()
Sekretaris Penguji	: <u>Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	()
Anggota Penguji	: <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Maulana Muhammad El-Sulthan
NIM : 16650106
Fakultas/Jurusan : Sains dan Teknologi/Teknik Infomatika
Judul Skripsi : Rancang Bangun Aplikasi Deteksi Plagiarisme
Menggunakan Algoritma *Winnowing* Dengan Pendekatan
Triword

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 15 Juni 2020

Yang membuat pernyataan,



Maulana Muhammad El-Sulthan
NIM. 16650106

HALAMAN MOTTO

*“Selama ini telah kususuri setiap jalanan, dan harus kuakui tidak ada jalan
terbaik kecuali jalan bersama Al-Qur'an”*

HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Puji syukur kehadiran Allah, shalawat dan salam bagi Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Kedua orang tua penulis tercinta, Bapak Abdul Halim, S.Ag M.Ag dan Ibu Hullatin, S.Ag M.Pd yang tak pernah lelah mendidik, menyayangi, dan menjadi panutan sosok yang luar biasa bagi penulis sejak kecil. Ribuan kalimat bahagia dan syukur yang tak akan cukup penulis tulis disini teruntuk mereka berdua, semoga rahmat dan kasih sayang Allah selalu tercurahkan pada mereka

Seluruh guru-guru penulis khususnya Almarhum KH. Muhammad Maftuh Sa'id *Al-Hafizh At-Tijani* dan KH. Agus Fahim Maftuh *Al-Hafizh* yang telah menjadi sosok suri tauladan dan selalu membimbing penulis tanpa kenal lelah

Dosen pembimbing penulis Bapak Fresy Nugroho, M.T dan Bapak Dr. Cahyo Crysdiyan yang membimbing penulisan skripsi ini dengan penuh kesabaran dan selalu memotivasi penulis untuk semangat dan pantang menyerah. Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, yang telah membimbing dan memberikan ilmunya yang sangat bermanfaat

Teman seperjuangan sekaligus teman hidup mulai pertama kali penulis mengucapkan *ijab qabul*, beliau adalah Fauziyah Amini. Sosok teman hidup yang selalu mendukung dan selalu semangat untuk belajar bersama dan saling melengkapi satu sama lain.

Teman-teman organisasi HTQ selama di UIN Maulana Malik Ibrahim Malang yang telah mengajarkan penulis berbagai hal untuk terus belajar semakin baik. Keluarga Teknik Informatika Andromeda (Teknik Informatika angkatan 2016) yang telah memberikan semangat dan doanya

Orang-orang yang penulis sayangi, yang tak bisa penulis sebutkan satu per satu yang selalu memberikan semangat dan motivasinya kepada penulis untuk menyelesaikan skripsi ini. Penulis ucapkan terimakasih yang luar biasa. Semoga ukhwah kita tetap terjaga dan selalu diridhoi Allah SWT. Allahumma Aamiin.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur penulis panjatkan ke hadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayahNya kepada kita, sehingga penulis bisa menyelesaikan skripsi dengan tepat waktu, yang berjudul “Rancang Bangun Aplikasi Deteksi Plagiarisme Menggunakan Algoritma *Winnowing* dengan Pendekatan *Triword*”. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Didalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, Selaku Dosen Pembimbing I sekaligus Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Fresy Nugroho, M.T, selaku Dosen Pembimbing I sekaligus Dosen Wali yang telah membimbing penulis dari awal kuliah hingga penyusunan skripsi ini selesai.
5. Para staff laboran Fakultas Sains dan Teknologi yang telah bersedia memberikan data.
6. Orang tua tercinta yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat terselesaikan.
7. Istri tercinta juga anggota keluarga dan kerabat yang senantiasa memberikan doa dan dukungan semangat kepada penulis.
8. Sahabat-sahabat seperjuangan yang tiada henti memberi dukungan dan motivasi kepada penulis serta target bersama untuk lulus skripsi dan wisuda bersama

9. Rekan-rekan Andromeda yang selalu memberikan semangat dan doa kepada penulis.
10. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, 26 Juni 2020

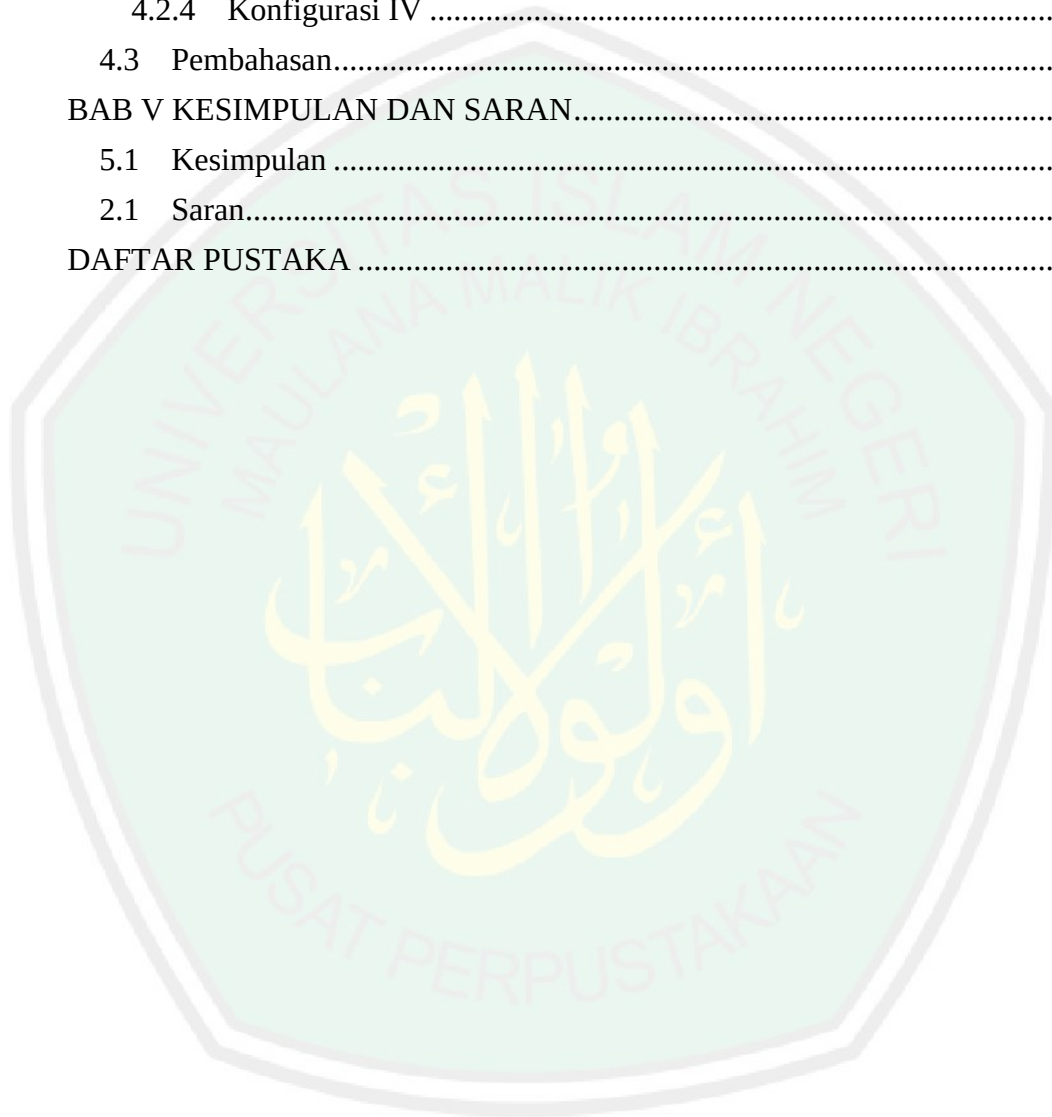
Penulis



DAFTAR ISI

HALAMAN PENGAJUAN.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN.....	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
ABSTRAK	xiii
ABSTRACT.....	xiv
الملخص.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan.....	6
BAB II STUDI PUSTAKA.....	7
2.1 Deteksi Plagiarisme.....	7
2.2 Kerangka Teori.....	9
BAB III DESAIN PENELITIAN.....	10
3.1 Sumber Data.....	10
3.2 Desain Sistem.....	10
3.2.1 Database Repository	11
3.2.2 Pre-processing	11
3.2.3 Algoritma Winnowing	18
3.2.4 Jaccard Coefficient.....	22
3.3 Pengujian Hasil Aplikasi.....	24
3.3.1 Pengujian Tingkat Error	24
3.3.2 Pengujian Kecepatan.....	25

BAB IV UJI COBA DAN PEMBAHASAN	26
4.1 Skenario Uji Coba	26
4.2 Hasil Uji Coba	28
4.2.1 Konfigurasi I	29
4.2.2 Konfigurasi II	31
4.2.3 Konfigurasi III	33
4.2.4 Konfigurasi IV	35
4.3 Pembahasan	37
BAB V KESIMPULAN DAN SARAN	42
5.1 Kesimpulan	42
2.1 Saran	43
DAFTAR PUSTAKA	44



DAFTAR GAMBAR

Gambar 2.1 Kerangka Teori Teknik <i>Document Fingerprinting</i>	10
Gambar 3.1 Desain Sistem	11
Gambar 3.2 Salah Satu Teks Abstrak Skripsi	12
Gambar 3.3 <i>Flowchart</i> Penghapusan <i>Whitespace Intensity</i>	13
Gambar 3.4 <i>Flowchart</i> Pengelompokan Teks Menjadi 3 Kata.....	15
Gambar 3.5 <i>Flowchart</i> Fungsi MD5	16
Gambar 3.6 <i>Flowchart</i> Fungsi <i>Rolling Hash</i>	18
Gambar 3.7 <i>Flowchart</i> Pengelompokan Nilai <i>Hash</i> Menjadi <i>Window</i>	20
Gambar 3.8 <i>Flowchart Fingerprint</i>	22
Gambar 3.9 <i>Flowchart Jaccard Coefficient</i>	24
Gambar 4.1 Halaman Hasil Presentase Plagiarisme	30
Gambar 4.2 <i>Output</i> Aplikasi Plagiarism Checker X.....	41
Gambar 4.3 <i>Output</i> Aplikasi Deteksi Plagiarisme menggunakan Algoritma <i>Winnowing</i> Dengan Pendekatan <i>Triword</i>	41

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Konfigurasi I.....	31
Tabel 4.2 Hasil Pengujian Konfigurasi II	33
Tabel 4.3 Hasil Pengujian Konfigurasi III	35
Tabel 4.4 Hasil Pengujian Konfigurasi IV	37
Tabel 4.5 Rata-Rata Presentase Nilai <i>Error</i> Dan Kecepatan Tiap Konfigurasi.....	39



ABSTRAK

El-Sulthan, Maulana Muhammad. 2020. ***Rancang Bangun Aplikasi Deteksi Plagiarisme Menggunakan Algoritma Winnowing dengan Pendekatan Triword***. Skripsi. Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Fresy Nugroho, M.T. (II) Dr. Cahyo Crysdiyan.

Kata Kunci : *Plagiarisme, Winnowing, Triword, Koefisien Jaccard*.

Praktik plagiarisme secara global tidak terlalu kasat mata dan dapat diamati dengan mudahnya. Salah satu sasaran plagiarisme adalah tugas akhir mahasiswa yaitu skripsi. Penguji skripsi adalah seorang dosen yang umumnya memiliki keterbatasan kemampuan untuk mengoreksi ratusan skripsi mahasiswa. Oleh karena itu peran aplikasi deteksi plagiarisme sangat dibutuhkan untuk membantu kinerja dosen untuk mengoreksi hasil skripsi mahasiswa. Dalam penelitian ini aplikasi deteksi plagiarisme dibangun menggunakan *Algoritma Winnowing* dengan pendekatan *Triword*. Algoritma ini mengimplementasikan teknik *Document Fingerprinting* untuk menentukan nilai *fingerprint* dari teks skripsi kemudian dihitung presentase kemiripannya menggunakan koefisien jaccard. Besar kecilnya tingkat *error* algoritma ini ditentukan oleh nilai konfigurasi nilai prima dan *window* yang digunakan sebagai parameter awal. Pada tahap pengujiannya penulis menghitung rata-rata nilai *error* dari 4 konfigurasi yang berbeda. Hasilnya adalah Rata-rata nilai *error* konfigurasi I = 15%, konfigurasi II = 22%, konfigurasi III = 17%, konfigurasi IV = 18%. Berdasarkan hasil pengujian tersebut disimpulkan bahwa konfigurasi I menghasilkan nilai *error* tekecil yaitu 15%, dengan konfigurasi nilai prima 4 dan *window* 2. Hal tersebut membuktikan bahwa konfigurasi tersebut merupakan konfigurasi terbaik dibandingkan 3 konfigurasi lainnya.

ABSTRACT

El-Sulthan, Maulana Muhammad. 2020. ***Design and Build Plagiarism Detection Application Using Winnowing Algorithm with Triword Approach.*** Essay. Department of Informatics Engineering, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim of Malang. Counselor: (I) Fresy Nugroho, M.T. (II) Dr. Cahyo Crysdian.

Keywords: *Plagiarism, Winnowing, Triword, Jaccard Coefficient.*

The plagiarism globally is not very visible and can be easily observed. One of the targets of plagiarism is the student's final project, Essay. Essay examiners are lecturers who generally have limited ability to correct hundreds of student essays. Therefore the role of plagiarism detection application is needed to help lecturers' performance to correct the results of student essays. In this essay a plagiarism detection application is built using the Winnowing Algorithm with the Triword approach. This algorithm implements the Document Fingerprinting technique to determine the fingerprint value of the essay text and then calculates the percentage of similarity using the jaccardic coefficient. The size of the error rate of this algorithm is determined by the configuration value of the prime and window values used as initial parameters. In the testing phase the authors calculate the average error value of 4 different configurations. The result is the average value of configuration error I = 15%, configuration II = 22%, configuration III = 17%, configuration IV = 18%. Based on the test results it was concluded that configuration I produced a small error value of 15%, with a configuration of prime value 4 and window 2. This proves that the configuration is the best configuration compared to the other 3 configurations.

الملخص

السلطان، مولانا محمد. 2020. تصميم تطبيق الكشف عن الانتحال باستخدام الخوارزمية التذرية بطريقة تريوورد. البحث الجامعي. قسم الهندسة المعلوماتية كلية العلوم والتكنولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (1) فرسي نوغرو هو الماجستير (2) الدكتور جهيو كريسيديان.

الكلمات المفتاحية: الانتحال ، التذرية، تريوورد ، معامل جاكرد. الانتحال أمر خفي، ولا تمكن ملاحظته بسهولة. والبحث الجامعي يكون فيه أكثر المشكلات المتعلقة بالانتحال و بجانب آخر يكون المشرف أو المصحح للبحث الجامعي لهم القدرة المحددة في تصحيحه لأن لا يزال عدده يزداد أكثر في كل حين. لذلك، يصبح تطبيق الكشف عن الانتحال من اللوازم لا بد من توافرها لمساعدة المشرف في التصحيح والتحقيق. في هذا البحث صمم تطبيق الانتحال باستخدام الخوارزمية التذرية بطريقة تريوورد. هذه الخوارزمية تنفذ بصمة المستند لتقييم البصمة من نص البحث الجامعي، ثم يحسب التشبيه بمعامل جاكرد. تم تعيين عدد الأخطاء في الخوارزمية بتكوين القيمة الأولية وويندو المستخدمة كالمعاملة الأولى. الباحث يجد الاختلافات في متوسط قيمة الخطأ من أربعة تكوينات وهي $1 = 22\%$ ، $2 = 22\%$ ، $3 = 17\%$ ، $4 = 18\%$. فالخلاصة أن التكوين الأول ينتج أقل الأخطاء وهي 15% ، مع تكوين القيمة الأولية 4 وويندو 2. فهذا يشير إلى إثبات أن ذلك التكوين هو أفضل من ثلاثة تكوينات أخرى.

BAB I PENDAHULUAN

1.1 Latar Belakang

Praktik plagiarisme secara global tidak terlalu kasat mata dan dapat diamati dengan mudahnya. Faktanya, plagiarisme tidak hanya terjadi pada kalangan pelajar namun bahkan terjadi pada tingkat dunia. Di China, hampir seluruh barang di dunia mereka tiru. Sudah menjadi rahasia umum bahwa produk China merupakan produk imitasi yang tidak wajar dan tidak jarang menjadi bahan candaan bagi masyarakat Indonesia. Pada dunia akademis tingkat nasional, Indonesia juga mengalami hal yang serupa, sebagaimana dikutip dari tempo.co (2016:1) Guru Besar Ilmu Politik dari *Northwestern University, Chicago, Amerika Serikat* yang bernama *Winters*, mengaku masih menemukan plagiarisme dalam penulisan esai akademik mahasiswa Indonesia. Dalam temuannya, *Winters* menegaskan, jumlah konten plagiarisme dalam esai mahasiswa Indonesia berbeda-beda. Ada yang plagiarisme hanya 4 persen, ada pula yang 50 persen. "Sayangnya ada pula yang plagiat esai mereka sampai 99 persen," katanya.

Hal tersebut telah membuktikan bahwa pada era global saat ini praktik plagiarisme juga telah memasuki dunia akademis. Karya tulis yang seharusnya menjadi media untuk pembelajaran, banyak disalah gunakan untuk kepentingan pribadi. Salah satu bentuk penyalahgunaan karya tulis adalah menjiplak karya tulis orang lain yang mana perilaku ini disebut plagiat. Pada tahun 2013 lalu, telah ditemukan karya tulis mahasiswa di salah satu universitas Indonesia yang mengandung unsur plagiat (Herqutanto, 2013). Tidak tanggung-tanggung, hampir 40 mahasiswa dicurigai telah melakukan praktik plagiarisme. Masalah tersebut

segera terselesaikan karena pihak kampus dengan sigap telah membentuk tim untuk melakukan penyelidikan dan menindak lanjuti semua masalah terkait dengan hal tersebut. Kejadian itu sangatlah mencoreng nama baik kampus, sebagai pelajar kita seharusnya perlu memahami dengan baik terkait isu plagiarisme, tidak hanya pada tingkat mahasiswa saja, namun seluruh civitas akademika. Bila pemahaman tersebut sudah baik, maka kejadian seperti itu tidak akan terulang dan seluruh energi yang kita miliki dapat dicurahkan untuk kemajuan pendidikan. Sangatlah tidak pantas apabila seorang pelajar masih melakukan praktik plagiarisme, karena kegiatan tersebut meruntuhkan tembok kejujuran yang sangat kita junjung.

Plagiarisme berdasarkan sudut pandang agama islam juga merupakan tindakan yang tidak terpuji dan termasuk pada kategori kebohongan. Seorang plagiator adalah pelaku kebohongan yang mengakui karya orang lain sebagai karyanya sendiri. Orang yang suka berbohong dalam islam dikecam akan mendapat hukuman yang sangat pedih, meskipun kebohongannya tidak diketahui oleh manusia, Allah *Subhanahu Wata'ala* pasti mengetahui kebohongannya. Sebagaimana dijelaskan dalam firman Allah *Subhanahu Wata'ala*:

قُلْ إِنْ تَخْفُوا مَا فِي صُدُورِكُمْ أَوْ تَبْدُوهُ يَعْلَمُهُ اللَّهُ وَيَعْلَمُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ وَاللَّهُ عَلَى كُلِّ شَيْءٍ قَدِيرٌ

“katakanlah: jika kalian menyembunyikan apa yang ada di dalam dada kalian, atau kalian menampakkannya, maka Allah mengetahui semua itu. Allah mengetahui apa yang ada di langit dan apa yang ada di bumi. Dan Allah maha kuasa atas segala sesuatu” (QS. Al Imran: 29).

Ironisnya meskipun negara maupun agama telah melarang praktik plagiarisme, perilaku tersebut kian marak hampir di semua kalangan. Maraknya praktik plagiarisme ini dipicu dengan berkembangnya teknologi internet yang memungkinkan semua orang dapat mengakses informasi dari belahan dunia manapun dan kapanpun waktunya. Adanya internet banyak membawa dampak positif, namun juga ada dampak negatif yang diterima bagi umat manusia. Orang-orang dapat dengan mudah mencari karya tulis orang lain kemudian menjiplaknya. Plagiarisme ini seringkali hanya dianggap sebagai pelanggaran etika dan bukan pelanggaran hukum, padahal seorang plagiat dapat menerima hukuman atas apa yang telah diperbuatnya sebagaimana diatur dalam Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

Salah satu sasaran plagiarisme adalah skripsi. Menurut Kamus Besar Bahasa Indonesia, skripsi adalah karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademisnya. Skripsi bertujuan untuk menguji kemampuan mahasiswa untuk membuat karya tulis ilmiah sesuai dengan bidangnya. Mahasiswa yang mampu membuat skripsi dianggap mampu menggabungkan pengetahuan, kreativitas, analisis, dan penggambaran suatu masalah yang sesuai dengan bidangnya. Penguji skripsi adalah seorang dosen, dan dosen memiliki keterbatasan kemampuan untuk mengoreksi ratusan skripsi mahasiswa. Oleh karena itu peran aplikasi deteksi plagiarisme sangat dibutuhkan untuk membantu kinerja dosen untuk mengoreksi hasil skripsi mahasiswa.

Saat ini, banyak aplikasi deteksi plagiarisme yang telah dibangun baik oleh individu maupun institusi atau kelompok dan dapat ditemukan di Internet. Beberapa

software untuk deteksi plagiarisme seperti *Turnitin*, *WCopyFind* yang digunakan oleh *University of Virginia*, *Crosscheck* yang ditemukan oleh *Elsevier*, *Springer* yang diciptakan oleh *Massachusetts Institute of Technology pers*, dan penerbit ternama lainnya, *Docoloc* yang terintegrasi dengan sistem manajemen konferensi EDAS (Alzahrani *et. al* , 2019).

Berdasarkan pemaparan singkat tentang beberapa hal di atas yang menjadi landasan untuk dilakukannya penelitian ini, Penulis berinisiatif untuk membangun suatu aplikasi deteksi plagiarisme untuk memproses file skala besar yaitu file Skripsi Mahasiswa Jurusan Teknik Informatika Universitas Maulana Malik Ibrahim Malang. Dalam hal ini yang akan menjadi objek penelitian adalah algoritma dan data tugas akhir mahasiswa Mahasiswa Jurusan Teknik Informatika Universitas Maulana Malik Ibrahim Malang. Aplikasi yang diharapkan dapat menampilkan hasil dari algoritma *Winnowing* dengan pendekatan *Triword* dan menghasilkan presentase tingkat plagiarisme dengan rata-rata presentase nilai *error* terkecil.

1.2 Pernyataan Masalah

1. Seberapa tinggi rata-rata presentase nilai *error* dari aplikasi pendeteksi plagiarisme menggunakan algoritma *Winnowing* dengan pendekatan *Triword*?
2. Seberapa cepat waktu yang diperlukan oleh algoritma *Winnowing* dengan pendekatan *Triword* untuk mendeteksi plagiarisme?

1.3 Tujuan Penelitian

1. Mengukur presentase *error* dari aplikasi deteksi plagiasi menggunakan algoritma *Winnowing* dengan pendekatan *Triword*.
2. Mengukur kecepatan dari algoritma *Winnowing* dengan pendekatan *Triword* untuk mendeteksi plagiarisme.

1.4 Manfaat Penelitian

Adapun manfaat penelitian ini bagi Universitas adalah:

1. Sebagai alat bantu untuk mengoreksi skripsi mahasiswa.
2. Mempersingkat waktu koreksi skripsi mahasiswa.
3. Sebagai sarana untuk melatih kemampuan menulis mahasiswa.
4. Untuk mengetahui kemiripan skripsi diri sendiri dengan skripsi orang lain.
5. Untuk mengetahui kemiripan skripsi diri sendiri dengan skripsi orang lain.

1.5 Batasan Masalah

1. Penelitian ini berfokus pada abstrak skripsi Mahasiswa Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dokumen yang diuji merupakan file masukan yang berekstensi *.doc* (*Word 97-2005 Document*).
3. Dokumen yang diuji merupakan *text* digital dan bukan gambar.
4. Aplikasi ini hanya mampu mendeteksi kemiripan kata dan tidak bisa mendeteksi kemiripan makna (sinonim).
5. Aplikasi ini hanya mampu mendeteksi *text* dalam bahasa yang sama dan menggunakan huruf *alphabet*.

1.6 Sistematika Penulisan

Penelitian ini tersusun dalam laporan yang terdiri dari beberapa bab pembahasan sebagai berikut:

1. **Bab I Pendahuluan** : Pada bab pertama ini berisi tentang latar belakang penelitian, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian,
2. **Bab II Studi Pustaka** : Bab ini berisi tentang *literatur review* dan penelitian serupa yang telah dilakukan.
3. **Bab III Desain Penelitian** : bab ini berisi perancangan dan implementasi algoritma *Winnowing* dengan pendekatan *Triword* pada aplikasi deteksi plagiarisme,
4. **Bab IV Hasil dan Uji Coba** : bab ini berisi hasil data pengolahan yang dilakukan penelitian ini oleh aplikasi deteksi plagiarisme yang menggunakan algoritma *Winnowing* dengan pendekatan *Triword*,
5. **Bab V Kesimpulan dan Saran** : bab ini berisi kesimpulan dari penelitian ini, serta saran yang sangat bermanfaat untuk penelitian ini di masa yang akan datang,
6. **Daftar Pustaka** : bab ini berisi referensi penelitian terkait dengan penelitian ini,

BAB II STUDI PUSTAKA

2.1 Deteksi Plagiarisme

Penelitian Deteksi plagiarisme sebelumnya telah dilakukan oleh Schleimer *et. al* (2003). Mereka melakukan penelitian terhadap sebuah teknik *Document Fingerprint* yang berkaitan dengan deteksi plagiarisme secara akurat. Penelitian tersebut mengembangkan sebuah algoritma yang bernama *Winnowing*, algoritma *Fingerprint* lokal yang efisien, dan mereka telah menunjukkan bahwa kinerja *Winnowing* sangat baik. Mereka juga memberikan hasil percobaan pada data Web, dan melaporkan pengalaman dengan MOSS, layanan pendeteksi plagiarisme yang banyak digunakan.

Sepuluh tahun kemudian Peneliti asal Indonesia Ridho (2013) telah melakukan riset pengembangan dari algoritma *Winnowing* yang menggunakan pendekatan *k-grams* dalam membentuk *fingerprint* dokumen. *Fingerprint* yang dibentuk memiliki teknik yang berbasis *Character-based*. Penelitian ini mencoba menggunakan teknik *Fingerprint* yang berbeda, yaitu teknik *Phrase-based*. Teknik yang berbasis *phrase* ini akan memecah dokumen teks menjadi token-token *Biword*. Token-token ini akan dienkripsi menjadi nilai MD5, agar token tersebut memiliki nilai *hash* yang sama panjang dan dapat dijadikan sebagai *fingerprint* dokumen teks. Dengan menerapkan pendekatan *Biword* pada algoritma *Winnowing*, algoritma ini dapat melakukan pengecekan frasa setiap dokumen dan kemudian disimpan dalam sebuah *array*. Sehingga dalam menampilkan teks yang

memiliki nilai yang sama, algoritma dapat menampilkan kembali nilai *array* yang berbentuk token *Biword* yang dianggap sebagai *fingerprint* sebuah dokumen.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Jody *et. al* (2015). Mereka melakukan riset untuk menggabungkan Algoritma *Winnowing* dengan *Synonym Recognition*. Berdasarkan percobaan yang mereka lakukan, dapat dilihat bahwa penggunaan *Synonym Recognition* pada *Winnowing* sangat berpengaruh dalam menentukan kalimat termirip dalam satu dokumen teks dengan nilai keberhasilan sebesar 88% dimana jauh berbeda dibandingkan dengan tanpa *Synonym Recognition* yang hanya menghasilkan nilai keberhasilan sebesar 19%

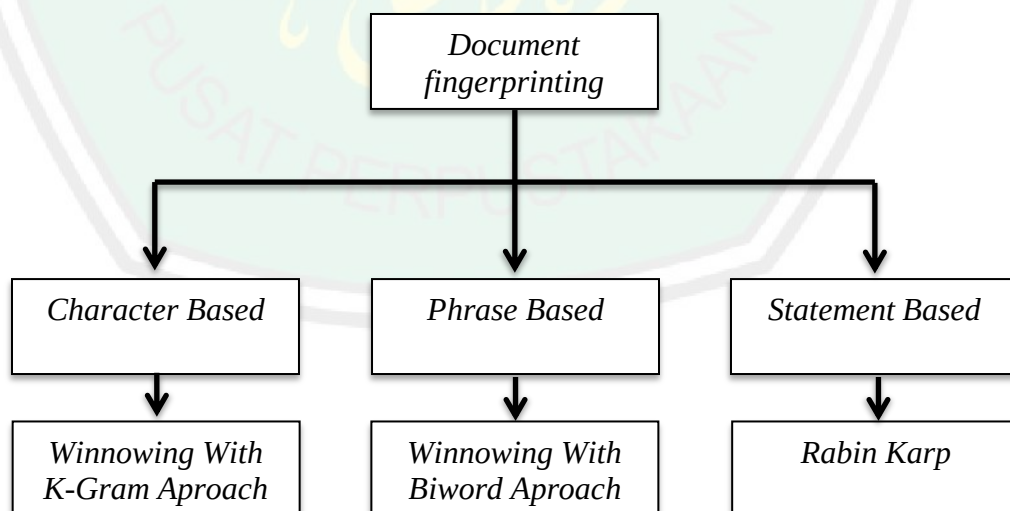
Penelitian tidak berhenti sampai disitu, 2 tahun kemudian Alamsyah (2017) melakukan penelitian untuk membandingkan akurasi Algoritma *Winnowing* dengan Algoritma *Rabin Karp* dalam menghitung tingkat plagiarisme judul skripsi mahasiswa. Dia berhasil memperoleh sebuah kesimpulan bahwa Algoritma *Winnowing* lebih baik daripada algoritma *Rabin Karp* karena menghasilkan tingkat presentase yang lebih kecil dan waktu proses yang lebih cepat, Berdasarkan hasil Pengujian terhadap perbandingan dua algoritma tersebut, dapat dilihat kemungkinan kemiripan teks judul skripsi yang terkecil adalah dengan menggunakan pendekatan algoritma *winnowing* yaitu pada ujicoba ke 8 dengan nilai $n\text{-gram} = 9$ dan $window = 3$, proses waktu 0.0257 dengan tingkat kemiripan terkecil yaitu 32.6 %.

Baru baru ini Hasan *et. al* (2018) telah mengembangkan sebuah *plug-in* pendeteksi plagiarisme yang digunakan pada *E-Learning*. *Plug-in* deteksi plagiarisme mampu mendeteksi tugas-tugas yang ada dalam format dokumen teks.

Algoritma yang digunakan adalah algoritma *Winnowing* , dan teknik *Hashing* yang dipilih untuk algoritma tersebut adalah *Rolling Hash*. Nilai kesamaan dihitung menggunakan *Coeffecien Jaccard*. Sebagai demonstrasi kelayakan, implementasi dilakukan di UMN E-learning berbasis Moodle. Hasil pengujian menunjukkan kombinasi parameter (*k-gram*, *window length*, dan *base prime number*) untuk *plug-in* dan *f-measure* berhasil mendeteksi plagiarisme pada tugas siswa di E-learning UMN dengan baik.

2.2 Kerangka Teori

Berdasarkan penjelasan pada sub bab sebelumnya, dapat disimpulkan bahwa berbagai alat pendeteksi plagiarisme telah dikembangkan berdasarkan berbagai teknik pendeteksian. Saat ini, teknik pencocokan *Document Fingerprinting* memainkan peran penting dalam alat deteksi tersebut. Kerangka teori dari Teknik *Document Fingerprinting* dijelaskan pada Gambar 2.1 (Kent and Salim, 2010) :



Gambar 2.1 Kerangka teori teknik *Document Fingerprinting*

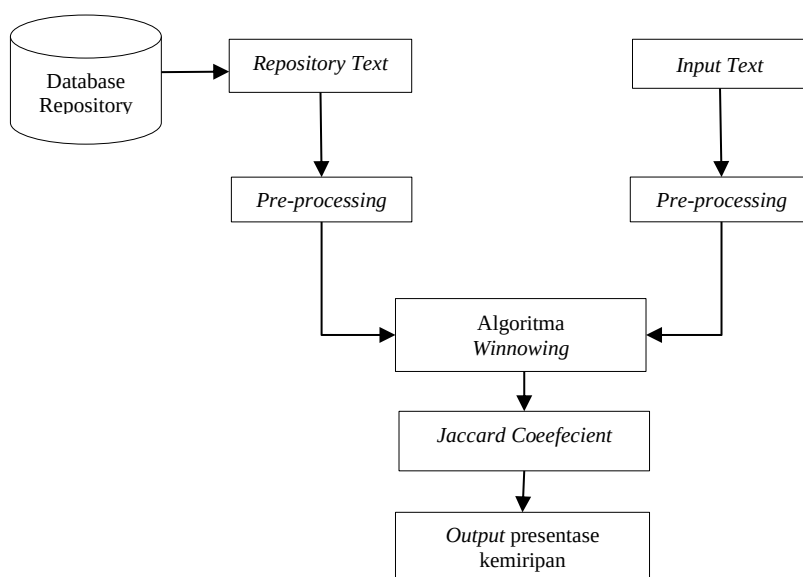
BAB III DESAIN PENELITIAN

3.1 Sumber Data

Sumber data pada penelitian ini adalah abstrak skripsi mahasiswa jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang yang berasal dari *E-Theses* UIN Maulana Malik Ibrahim Malang. File yang diunduh diambil bagian abstraknya saja kemudian diunggah ke database aplikasi pendeteksi plagiarisme. Dokumen skripsi yang di unduh awalnya berekstensi *PDF*, untuk memudahkan pengambilan abtrak, penulis mengubah ekstensi dokumen menjadi *DOC*.

3.2 Desain Sistem

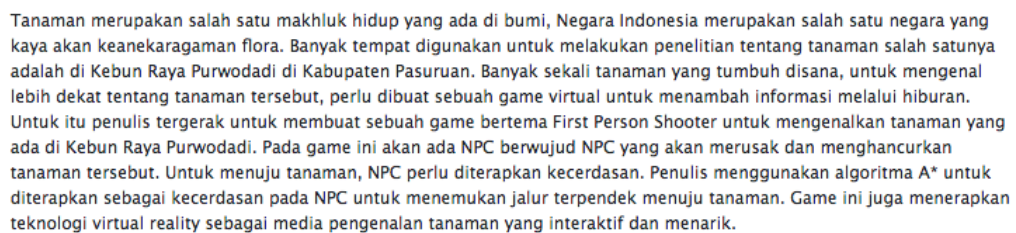
Desain sistem merupakan alur jalannya aplikasi, pada tahap ini penulis menggambarkan desain sistem dalam bentuk blok diagram. Gambar 3.1 merupakan desain sistem dari aplikasi pendeteksi plagiarisme.



Gambar 3.1 Desain Sistem

3.2.1 Database Repository

Database Repository merupakan tempat untuk menyimpan data berupa teks abstrak tugas akhir mahasiswa Teknik Informatika UIN Maulana Malik Ibrahim Malang. Jumlah Keseluruhan teks abstrak yang ada di *database* adalah 120. Rata-rata ukuran tiap *file* teks abstrak yang dimuat dalam *database* adalah 23 KB (*Kilo Bytes*) dengan ukuran teks terkecil berukuran 23 KB dan teks terbesar berukuran 90 KB. Gambar 3.2 merupakan contoh dari salah satu *file* teks yang akan diunggah ke *database repository*.



Tanaman merupakan salah satu makhluk hidup yang ada di bumi, Negara Indonesia merupakan salah satu negara yang kaya akan keanekaragaman flora. Banyak tempat digunakan untuk melakukan penelitian tentang tanaman salah satunya adalah di Kebun Raya Purwodadi di Kabupaten Pasuruan. Banyak sekali tanaman yang tumbuh disana, untuk mengenal lebih dekat tentang tanaman tersebut, perlu dibuat sebuah game virtual untuk menambah informasi melalui hiburan. Untuk itu penulis tergerak untuk membuat sebuah game bertema First Person Shooter untuk mengenalkan tanaman yang ada di Kebun Raya Purwodadi. Pada game ini akan ada NPC berwujud NPC yang akan merusak dan menghancurkan tanaman tersebut. Untuk menuju tanaman, NPC perlu diterapkan kecerdasan. Penulis menggunakan algoritma A* untuk diterapkan sebagai kecerdasan pada NPC untuk menemukan jalur terpendek menuju tanaman. Game ini juga menerapkan teknologi virtual reality sebagai media pengenalan tanaman yang interaktif dan menarik.

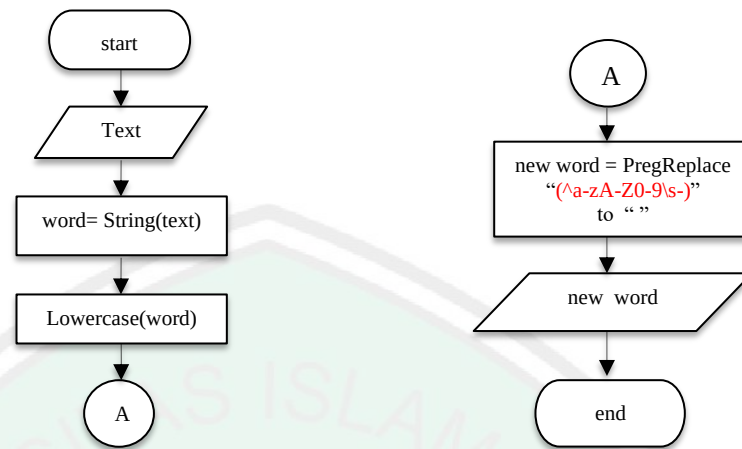
Gambar 3.2 Salah Satu Teks Abstrak Skripsi

3.2.2 Pre-processing

Pada tahap preprocessing dilakukan 4 proses, yaitu sebagai berikut:

3.2.2.1 Penghapusan karakter tidak relevan (*whitespace intensivity*)

Langkah pertama adalah menghapus karakter yang tidak relevan (*whitespace insensivity*) pada dokumen teks. Penghapusan dilakukan pada semua huruf yang bukan A-Z, a-z, 0-9. Kemudian teks diubah menjadi huruf kecil semua sehingga nantinya hanya karakter-karakter yang berupa huruf atau angka yang akan diproses lebih lanjut. *Flowchart* penghapusan *whitespace intensivity* dijelaskan pada Gambar 3.3 :



Gambar 3.3 Flowchart penghapusan *whitespace intensity*

Contohnya adalah sebagai berikut:

Teks masukan:

"RANCANG BANGUN APLIKASI DETEKSI PLAGIASI MENGGUNAKAN ALGORITHMMA
TRIWORD WINNOWING"

Setelah penghapusan *whitespace intensity* :

rancang bangun aplikasi deteksi plagiasi menggunakan algorithmma triword
winnowing

3.2.2.2 Pengelompokan teks menjadi 3 kata

Langkah kedua adalah mengelompokkan text menjadi berbentuk *Triword* (3 kata). *Triword* merupakan suatu teknik *matching* pada proses tokenisasi atau pemisahan kata (*fingerprint*) pada suatu teks. Konsep *Triword* itu sendiri menyerap teknik *phrase-based*. Pada tahap awal akan dilakukan konversi setiap dokumen untum satu set *trigram* atau tiga kata.

Berikut adalah contoh penerapan konsep *Triword*:

Teks masukan:

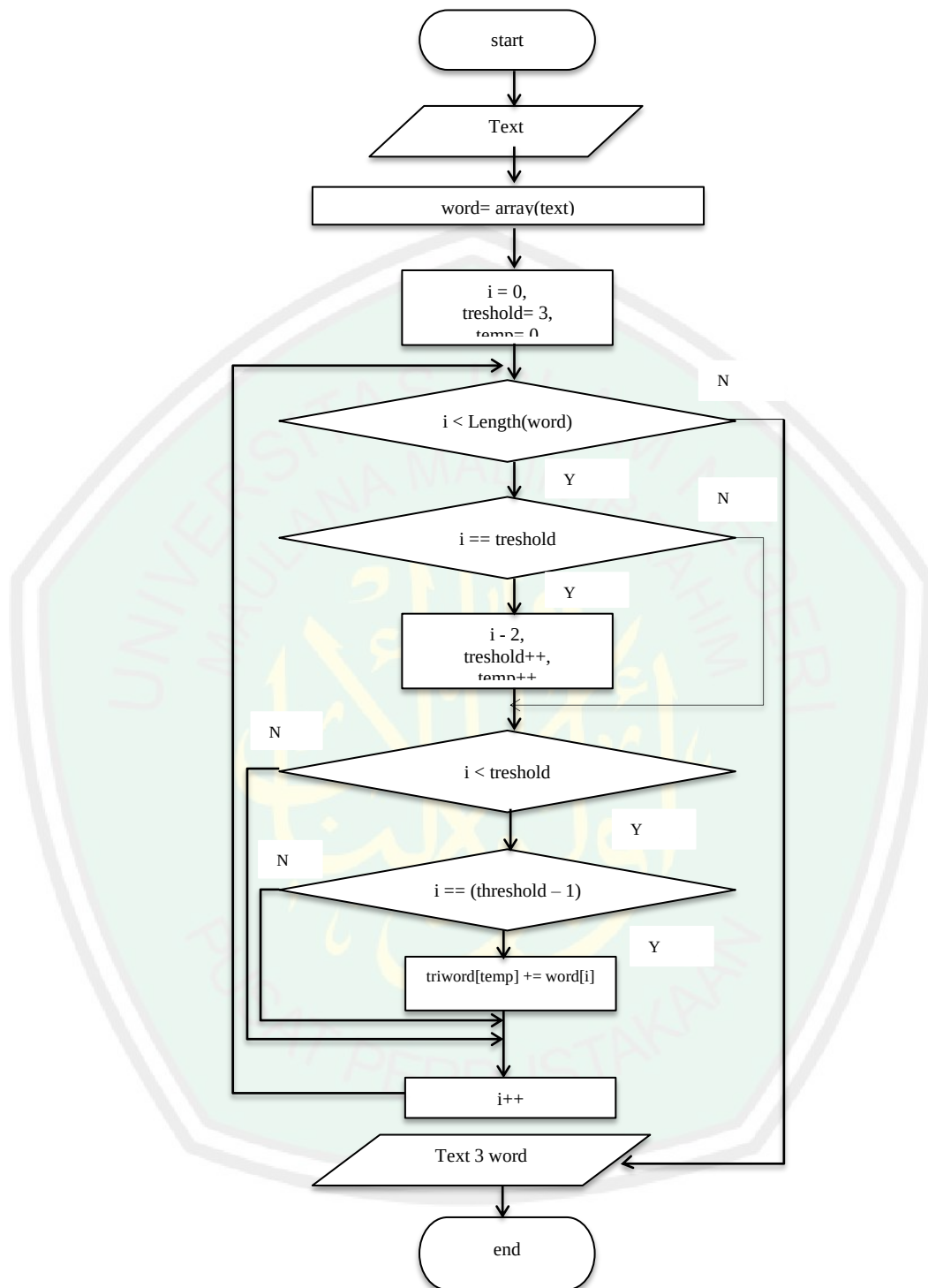
rancang bangun aplikasi deteksi plagiasi menggunakan algoritma triword winnowing

Setelah dilakukan tokenisasi *Triword* :

[1] = {rancang bangun aplikasi}
 [2] = {bangun aplikasi deteksi}
 [3] = {aplikasi deteksi plagiasi}
 [4] = {deteksi plagiasi menggunakan}
 [5] = {plagiasi menggunakan algoritma}
 [6] = {menggunakan algoritma triword}
 [7] = {algoritma triword winnowing}

Flowchart mengelompokan teks menjadi berbentuk *triword* adalah sebagaimana

Gambar 3.4 :



Gambar 3.4 Flowchart pengelompokkan text menjadi 3 kata

3.2.2.3 Menyamakan jumlah huruf teks 3 kata menggunakan fungsi MD5

Pada proses ini teks yang sudah terbentuk pada proses sebelumnya di enkripsi menggunakan MD5 agar setiap karakter mempunyai panjang yang sama.

Berikut adalah contoh penerapan enkripsi MD5:

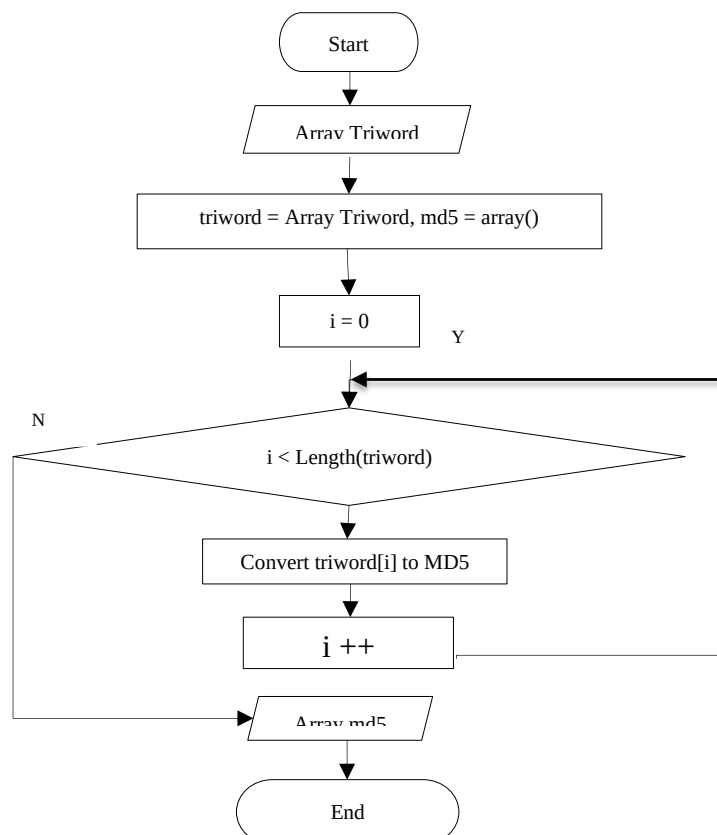
Teks masukan:

rancang bangun aplikasi

Setelah dilakukan enkripsi MD5 :

aab09c50a6b54c166e320c5adea38deb

Gambar 3.5 adalah *Flowchart* Menyamakan jumlah huruf teks 3 kata menggunakan fungsi MD5.



Gambar 3.5 *Flowchart* fungsi MD5.

3.2.2.4 Mengkonversi Teks menjadi kode ASCII dengan rumus *rolling hash*

pada tahap ini dilakukan perubahan teks dari proses sebelumnya menjadi kode ASCII menggunakan rumus persamaan *Rolling Hash*. *Rolling hash* biasa dikenal dengan fungsi *hash*. Fungsi *hash* menerima masukan berupa teks yang terdiri dari banyak kata yang panjangnya beragam kemudian mengkonversikannya menjadi teks yang terdiri dari kata yang panjangnya sama. Berikut Rumus 3.1 adalah persamaan dari fungsi *hash*:

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b + c_k \quad (3.1)$$

Keterangan:

c = nilai *ASCII* karakter (desimal)

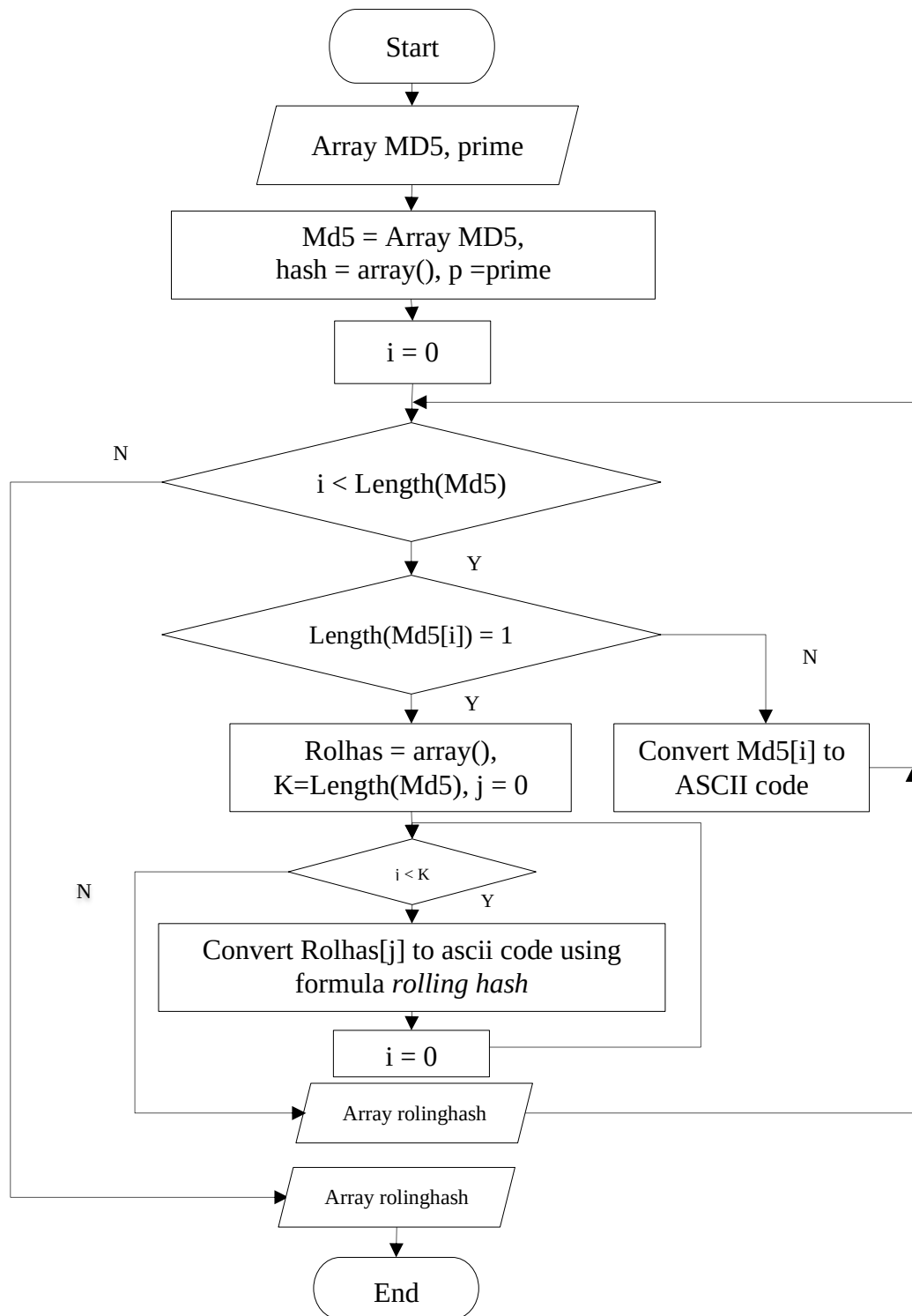
b = basis (bilangan prima)

k = banyak karakter

Berikut adalah Salah satu contoh perhitungan Fungsi *hash* pada kata “rancang” dengan nilai *prime* = 2:

$$\begin{aligned} H_{(\text{rancang})} &= \text{ascii}(\text{r}) * 2^{(6)} + \text{ascii}(\text{a}) * 2^{(5)} + \text{ascii}(\text{n}) * 2^{(4)} + \text{ascii}(\text{c}) * 2^{(3)} + \text{ascii}(\text{a}) * 2^{(2)} + \\ &\quad \text{ascii}(\text{n}) * 2^{(1)} + \text{ascii}(\text{g}) * 2^{(0)} \\ &= 114 * 64 + 197 * 99 + 110 * 16 + 99 * 8 + 114 * 197 + 110 * 2 + 103 * 1 \\ &= 7296 + 19503 + 1760 + 792 + 22458 + 220 + 103 \\ &= 27326 \end{aligned}$$

Flowchart Mengkonversi Teks menjadi kode ASCII dengan rumus *rolling hash* dijelaskan pada Gambar 3.6:



Gambar 3.6 Flowchart fungsi rolling hash

3.2.3 Algoritma *Winnowing*

Pada tahap ini Algoritma *Winnowing* akan dijelaskan secara rinci. Tahapan Algoritma *Winnowing* yang diterapkan pada desain sistem ini beberapa langkah, yaitu sebagai berikut:

3.2.3.1 Pengelompokan nilai *hash* menjadi *window* seukuran *w*

Nilai-nilai *hash* yang telah terbentuk, selanjutnya dibentuk dalam beberapa *window* dengan ukuran *w*. *Window* merupakan pembagian atau pengelompokan beberapa nilai *hash* dengan ukuran yang ditentukan. Dari *window* yang telah dibentuk dilakukan pemilihan nilai *hash* terkecil pada tiap *window* untuk dijadikan *fingerprint* tiap dokumen. Berikut adalah contoh hasil pembentukan *window* dari kumpulan nilai *hash*:

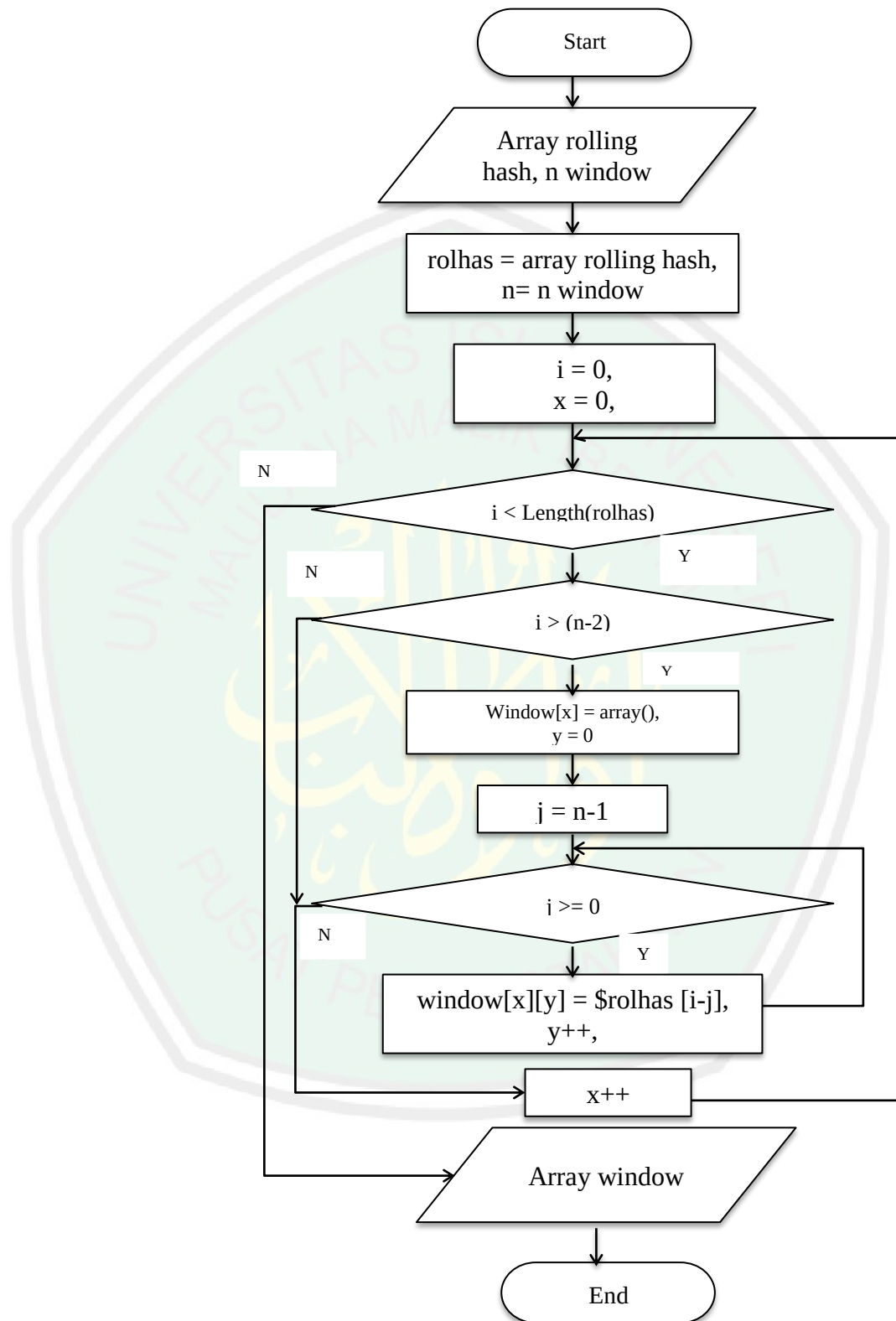
Teks masukan:

```
[1] = {792407567384}
[2] = {540146196934}
[3] = {803517371972}
[4] = {466664202804 }
[5] = {557337504340}
[6] = {714960651840}
[7] = {570907424356}
```

Hasil dari proses pembentukan *window* :

```
W-1 : {792407567384 540146196934 803517371972 466664202804}
W-2 : {540146196934 803517371972 466664202804 557337504340}
W-3 : {803517371972 466664202804 557337504340 714960651840}
W-4 : {466664202804 557337504340 714960651840 570907424356}
```

Flowchart algoritma *windowing* adalah sebagaimana Gambar 3.7:



Gambar 3.7 Flowchart Pengelompokan nilai hash menjadi window seukuran w

3.2.3.2 Memilih nilai *hash* terkecil pada setiap *window* untuk dijadikan nilai *fingerprint*

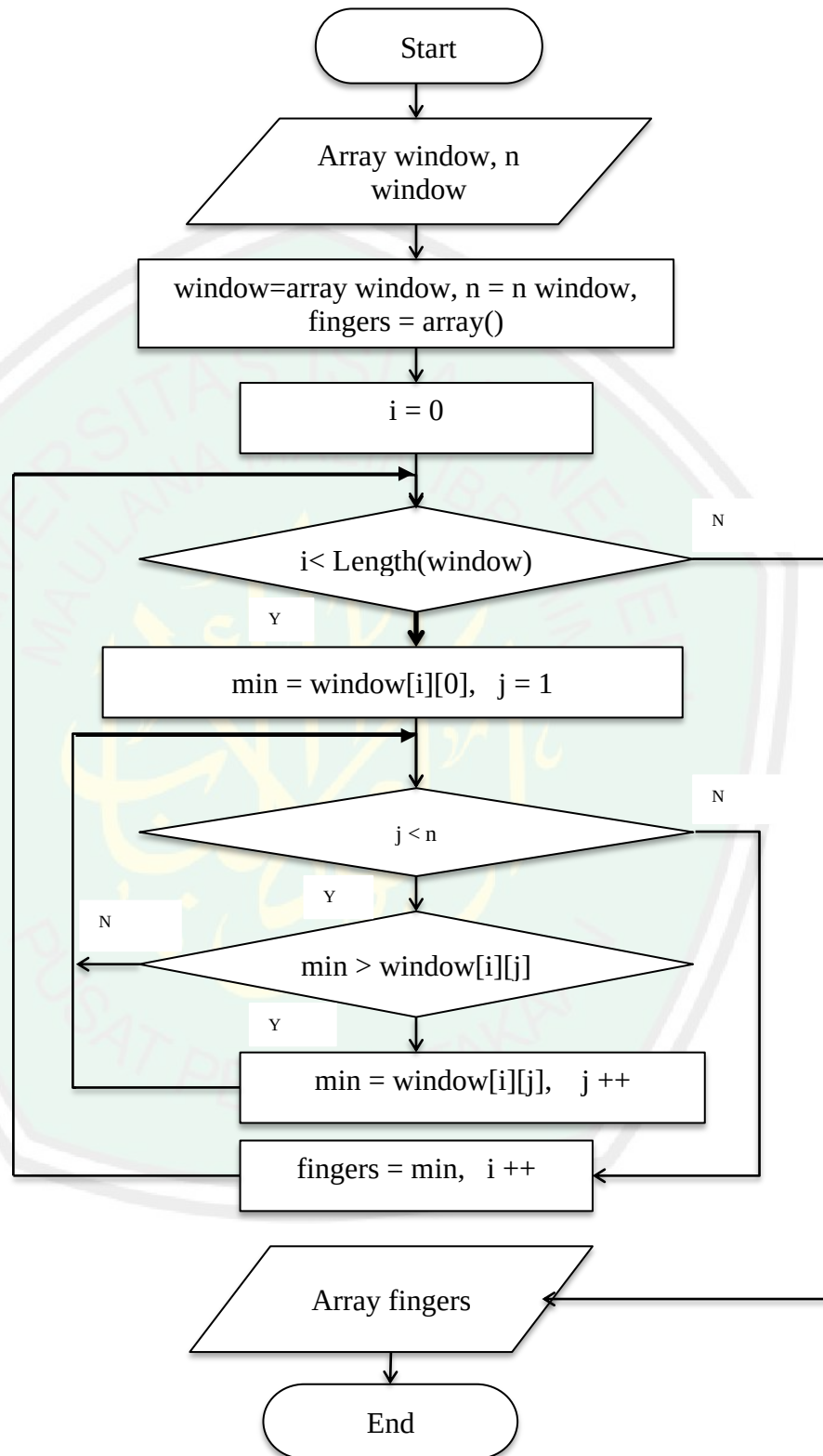
Setelah nilai *Rolling Hash* sudah dikelompokkan menjadi beberapa *window*, selanjutnya dari *window* yang telah dibentuk dilakukan pemilihan nilai *hash* terkecil pada tiap *window* untuk dijadikan *fingerprint* tiap dokumen. Dari proses sebelumnya didapatkan nilai *window* sebagai berikut:

W-1 : {792407567384 540146196934 803517371972 **466664202804**}
 W-2 : {540146196934 803517371972 **466664202804** 557337504340}
 W-3 : {803517371972 **466664202804** 557337504340 714960651840}
 W-4 : {**466664202804** 557337504340 714960651840 570907424356}

Nilai *hash* yang bercetak tebal merupakan nilai *hash* terkecil pada setiap *window*, nilai-nilai tersebut dikumpulkan menjadi satu seperti berikut:

F-1 : {466664202804}
 F-2 : {466664202804}
 F-3 : {466664202804}
 F-4 : {466664202804}

Gambar 3.8 merupakan *Flowchart* pemilihan *fingerprint*..:



Gambar 3.8 *Flowchart* fingerprint

3.2.4 Jaccard Coefficient

Langkah berikutnya adalah melakukan perhitungan persentase kemiripan menggunakan persamaan *Jaccard Coefficient*. *Jaccard coefficient* merupakan persamaan yang digunakan untuk menentukan tingkat kemiripan antara dua dokumen teks pada algoritma *winnowing*. Berikut persamaan *jaccard coefficient*:

$$\text{Coeffesien Jaccard} = \frac{|A \cap B|}{|A \cup B|} \times 100 \quad (3.2)$$

Keterangan:

$D(A,B)$ = Similarity

A = fingerprint teks 1

B = fingerprint teks 2

Berikut adalah contoh perhitungan *jaccard coefficient* terhadap 2 kumpulan nilai fingerprint :

Fingerprint 1:

F-1 : {466664202804} F-2 : {466664202804} F-3 : {466664202804} F-4 : {466664202804}
--

Fingerprint 2:

F-1 : {470337647994} F-2 : {474166965900}
--

Jumlah Fingerprints 1 = 4

Jumlah Fingerprints 2 = 2

Union fingerprints 1 dan 2 = 6

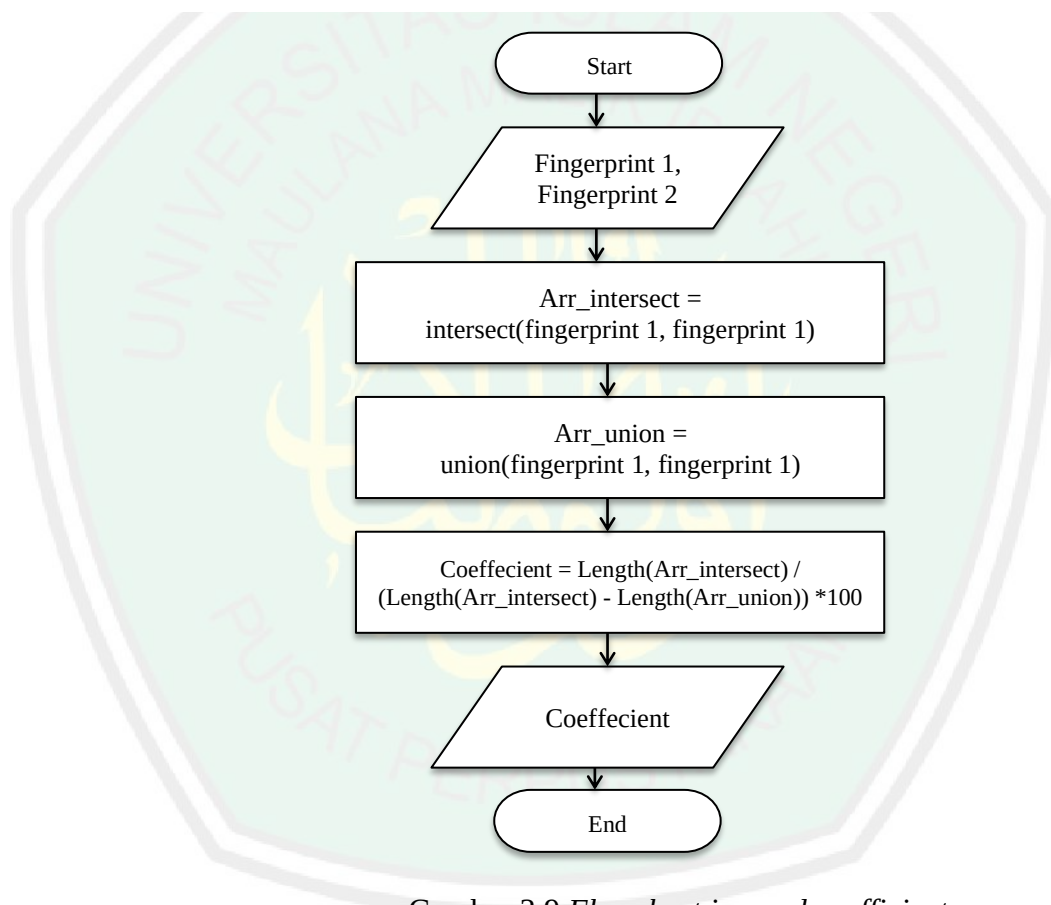
Intersection = 0

$$(Union - Intersection) = 6$$

$$Coeffisien Jaccard = (Intersection / (Union - Intersection)) * 100$$

$$= (0/6) * 100 = 0 \%$$

Flowchart Menghitung presentase kemiripan berdasarkan nilai fingerprint menggunakan persamaan jaccard coefficient dijelaskan pada Gambar 3.9:



Gambar 3.9 Flowchart jaccard coefficient

3.3 Pengujian Hasil Aplikasi

Tahap akhir pada penelitian ini adalah melakukan uji coba terhadap aplikasi yang telah dibangun berdasarkan tahap-tahap sebelumnya. Dengan demikian dapat diketahui tingkat keberhasilan aplikasi ini. Pengujian yang dilakukan adalah sebagai berikut:

3.3.1 Pengujian Tingkat Error

Untuk menghitung ketepatan *output* aplikasi maka dilakukan pengujian terhadap *output* algoritma *Winnowing* dengan Pendekatan *Triword* menggunakan rumus Rata-rata presentase nilai *error*. Menurut Li *et. al* (2019) Secara umum Rata-rata presentase nilai *error* dalah sebagaimana Rumus 3.3 berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \hat{y}_{xi}|}{y_{xi}} \right) \times 100\% \quad (3.3)$$

Keterangan:

n = Banyaknya pengujian.

y_{xi} = Nilai sebenarnya (*Ground truth*) .

$\hat{y}_{xi}$ = *Output* Aplikasi.

Data terkait “nilai sebenarnya” didapatkan dari *output* aplikasi *Plagiarism Checker X* yang merupakan salah satu software *similarity checker* yang direkomendasikan untuk dimiliki oleh pengelola jurnal. hasil dari pengecekan dengan menggunakan software ini tidak jauh berbeda dengan pengecekan yang dilakukan dengan menggunakan software sejenis yang seperti *Turnitin*. Software ini menurut situs resminya memiliki pelanggan dari OHIO University, King Saud University, IEEE, dan sebagian pengelola jurnal di Indonesia.

3.3.2 Pengujian Kecepatan

Pengujian ini bertujuan untuk mengetahui seberapa cepat waktu yang diperlukan algoritma *Winnowing* dengan Pendekatan *Triword* untuk mendeteksi plagiarisme. Adapun rumus yang digunakan adalah rumus *Mean* sebagaimana rumus 3.4 (Jody *et. al*, 2015).

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.4)$$

Keterangan:

$\bar{X}$ = Rata-rata Kecepatan proses pengujian.

n = Banyaknya pengujian.

X_i = Kecepatan proses pengujian ke- i .

BAB IV UJI COBA DAN PEMBAHASAN

4.1 Skenario Uji Coba

Tahap uji coba dilakukan untuk mengetahui apakah aplikasi yang dibuat sudah sesuai dengan perancangan dan layak digunakan oleh *user*. Langkah-langkah uji coba terhadap aplikasi Deteksi Plagiarisme adalah sebagai berikut :

1. Menyiapkan data uji sejumlah 30 data. Semua data ini merupakan *file* abstrak skripsi yang unduh secara acak dari situs etheses.uin-malang.ac.id. Penulis mengunduh 30 *file* skripsi dari situs tersebut berupa file ekstensi pdf, kemudian dari *file* tersebut diambil teks abstraknya saja kemudian diubah ekstensinya menjadi doc.
2. Melakukan cek plagiarisme terhadap 30 data *input* kepada Aplikasi Deteksi Plagiarisme Menggunakan Algoritma Winnowing dengan Pendekatan Triword. Tiap konfigurasi akan di uji dengan 30 data yang secara acak. Adapun konfigurasi nilai prima dan *window* yang digunakan adalah :
 - a. Konfigurasi I : Prima 2 dan *Window* 4

Konfigurasi ini memiliki nilai prima terkecil yaitu 2, dan menggunakan nilai *window* 4, dengan begitu nilai prima dan *window* di konfigurasi ini merupakan nilai prima dan *window* yang paling kecil dari konfigurasi yang lain. Penulis ingin menguji apakah semakin kecil nilai prima dan *window* mempengaruhi presentase *error* atau tidak.

b. Konfigurasi II : Prima 3 dan *Window* 10

Konfigurasi ini memiliki nilai prima dan *window* yang lebih besar dari konfigurasi I yaitu 3, dan 10. Dengan nilai *window* yang besar dapat diprediksi bahwa kecepatan proses aplikasi akan meningkat.

c. Konfigurasi III : Prima 5 dan *Window* 4

Konfigurasi ini memiliki nilai prima yang lebih besar dari konfigurasi I dan II yaitu 5, dan menggunakan nilai *window* yang sama dengan konfigurasi I. Dengan nilai prima yang lebih besar dari nilai *window* penulis ingin menguji apakah presentase *error* aplikasi akan berkurang atau tidak.

d. Konfigurasi IV : Prima 7 dan *Window* 10

Konfigurasi ini memiliki nilai yang lebih besar dari konfigurasi sebelumnya yaitu 7 dan menggunakan nilai *window* 10. Penulis ingin menguji apakah semakin tinggi nilai prima dan *window* mempengaruhi presentase *error* atau tidak.

Uji coba untuk tiap konfigurasi dilakukan 30 kali yang mana *output* dari cek plagiarisme data tersebut akan dibandingkan dengan *output* dari *Aplikasi Plagiarism Checker X*. Kecepatan proses aplikasi setiap uji coba juga akan dicatat oleh penulis.

3. Menghitung rata-rata nilai *error* dari hasil uji coba pada tiap konfigurasi. Pengujian ini bertujuan untuk mengetahui konfigurasi mana yang mampu menghasilkan *error* terkecil dalam mendeteksi plagiarisme dokumen teks. Semakin dekat nilai yang diukur dengan sampel nilai sebenarnya maka

semakin kecil nilai *error* tersebut. Adapun data yang digunakan sebagai sampel nilai sebenarnya adalah *output* dari aplikasi *Plagiarism Checker X*. Menurut Li *et. al* (2019) Secara umum Rata-rata presentase nilai *error* dalah sebagaimana Rumus 4.1 berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \hat{y}_{xi}|}{y_{xi}} \right) \times 100\% \quad (4.1)$$

Keterangan:

n = Banyaknya pengujian.

y_{xi} = Nilai sebenarnya (*Ground truth*) .

$\hat{y}_{xi}$ = *Output* Aplikasi.

4. Menghitung rata-rata kecepatan proses aplikasi. Pengujian ini bertujuan untuk mengetahui seberapa cepat waktu yang diperlukan algoritma *Winnowing* dengan Pendekatan *Triword* untuk mendeteksi plagiarisme. Adapun rumus yang digunakan adalah rumus *Mean* (Jody *et. al*, 2015).

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$$

Keterangan:

$\bar{X}$ = Rata-rata Kecepatan proses pengujian.

n = Banyaknya pengujian.

X_i = Kecepatan proses pengujian ke- i .

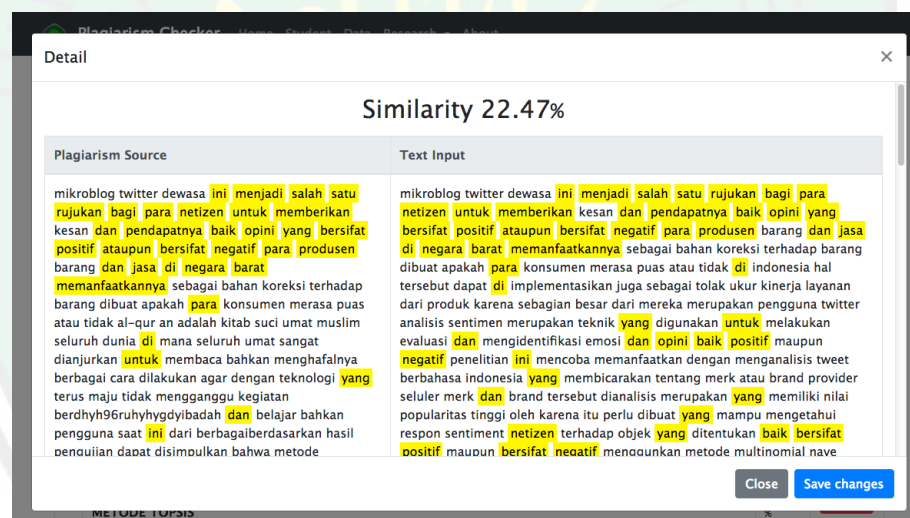
4.2 Hasil Uji Coba

Hasil uji coba yang akan dipaparkan oleh penulis merupakan hasil yang diperoleh dari pengujian terhadap 4 macam konfigurasi yang mana masing-masing

konfigurasi akan diuji coba dengan 30 data. Data uji yang berjumlah 30 tersebut akan dibandingkan oleh sistem dengan 120 data yang ada di dalam *database*.

4.2.1 Konfigurasi I

Konfigurasi nilai prima dan *window* pada pengujian ini adalah prima = 2 dan *window* = 4. Pengujian ini menggunakan 30 data uji yang mana semua data tersebut telah terdeteksi plagiarisme dengan persentase yang berbeda-beda. *Output* dari sistem akan dibandingkan dengan nilai sebenarnya. Tampilan hasil presentase plagiarisme ditunjukkan pada Gambar 4.1. Hasil uji coba terhadap 30 data yang dipaparkan pada Tabel 4.1.



Gambar 4.1 Halaman Hasil Presentase Plagiarisme

Tabel 4.1 Hasil Pengujian Konfigurasi I

pengujian ke-	Nama File	Waktu Proses (detik)	Presentase Kemiripan (persen)	Plagiarism Checker X	Persentase Nilai Error
1	11650093	4.6	100%	100%	0%
2	11650069	4.3	100%	100%	0%
3	10650002	9.9	40%	32%	25%
4	11650033	4.1	100%	100%	0%
5	11650042	4.4	100%	100%	0%
6	10650005	11.0	52%	83%	37%
7	12650022	4.3	100%	87%	15%
8	11650050	4.0	100%	100%	0%
9	10650006	10.2	41%	78%	47%
10	11650105	4.0	100%	100%	0%
11	11650065	4.9	100%	100%	0%
12	10650004	10.5	13%	79%	84%
13	12650018	4.1	100%	98%	2%
14	10650009	10.1	45%	100%	55%
15	11650081	3.8	100%	100%	0%
16	10650010	10.2	53%	100%	47%
17	11650087	4.4	100%	98%	2%
18	11650090	4.1	100%	100%	0%
19	10650008	10.0	54%	60%	10%
20	11650103	4.6	100%	100%	0%
21	12650033	4.2	100%	92%	9%
22	11650108	4.0	100%	100%	0%
23	12650013	3.7	100%	100%	0%
24	10650003	10.6	24%	49%	51%
25	12650028	3.9	100%	100%	0%
26	10650001	10.7	34%	61%	44%
27	12650029	3.6	100%	100%	0%
28	12650038	4.0	100%	100%	0%
29	12650036	4.3	100%	100%	0%
30	10650007	10.2	35%	54%	35%
	Total	186.7			464%

Berdasarkan Hasil Pengujian konfigurasi I pada Tabel 4.1 diperoleh perhitungan rata-rata *error* dan rata-rata kecepatan sebagai berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \hat{y}_{xi}|}{y_{xi}} \right) \times 100\%$$

$$\text{Error} = \frac{\left(\frac{|100 - 100|}{100} \right) + \left(\frac{|100 - 100|}{100} \right) + \left(\frac{|32 - 40|}{32} \right) + \dots + \left(\frac{|54 - 35|}{54} \right)}{30} \times 100\%$$

$$\text{Error} = \frac{0 + 0 + 25 + 0 + 0 + 37 + 15 + 0 + 47 + 0 + 0 + 84 + 2 + 55 + \dots + 35}{30} \times 100\%$$

$$\text{Error} = \frac{464}{30} \times 100\%$$

$$\text{Error} = 15\%$$

$$\text{Kecepatan} = \frac{(\text{jumlah semua kecepatan})}{(\text{banyaknya pengujian})}$$

$$\text{Kecepatan} = \frac{186.7}{30}$$

$$\text{Kecepatan} = 6.2 \text{ Detik}$$

Dari hasil perhitungan rata-rata *error* dan rata-rata kecepatan tersebut diperoleh kesimpulan bahwa uji coba Konfigurasi I dengan nilai *prima* = 2 dan *window* = 4 memiliki rata-rata persentase *error* sebesar 15% dan rata-rata kecepatan 6,2 detik.

4.2.2 Konfigurasi II

Konfigurasi nilai *prima* dan *window* pada pengujian ini adalah *prima* = 3 dan *window* = 10. Pengujian ini menggunakan 30 data uji yang mana semua data tersebut telah terdeteksi plagiarisme dengan persentase yang berbeda-beda. *Output* dari sistem akan dibandingkan dengan nilai sebenarnya. Berikut adalah uji coba terhadap 30 data yang dipaparkan pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Konfigurasi II

pengujian ke-	Nama File	Waktu Proses (detik)	Presentase Kemiripan (persen)	Plagiarism Checker X	Persentase Nilai <i>Error</i>
1	12650060	4.0	100%	100%	0%
2	12650067	4.0	100%	100%	0%
3	10650001	10.5	33%	61%	46%
4	12650088	4.0	100%	54%	85%
5	12650098	4.6	100%	88%	14%
6	10650002	10.1	40%	32%	25%
7	12650115	4.4	100%	100%	0%
8	12650120	4.2	100%	100%	0%
9	10650003	10.2	25%	49%	3%
10	13650002	4.6	100%	97%	46%
11	12650083	4.3	100%	100%	25%
12	10650004	10.0	27%	79%	49%
13	13650065	4.9	100%	94%	66%
14	13650013	4.0	100%	90%	11%
15	13650026	4.5	100%	98%	2%
16	10650005	10.7	54%	83%	35%
17	12650113	4.3	100%	100%	0%
18	13650036	4.9	100%	73%	37%
19	10650006	10.5	42%	78%	46%
20	13650041	4.3	100%	97%	3%
21	13650047	4.4	100%	100%	0%
22	10650007	10.4	35%	54%	16%
23	13650039	4.7	100%	100%	35%
24	13650052	4.3	100%	86%	8%
25	10650008	10.6	55%	60%	55%
26	13650067	4.5	100%	100%	43%
27	13650072	3.9	100%	95%	6%
28	10650009	10.3	45%	100%	0%
29	10650010	10.2	57%	100%	5%
30	13650073	4.1	100%	100%	0%
Total		190.4			662%

Berdasarkan Hasil Pengujian konfigurasi II pada Tabel 4.2 diperoleh perhitungan rata-rata *error* dan rata-rata kecepatan sebagai berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \hat{y}_{xi}|}{y_{xi}} \right) \times 100\%$$

$$\text{Error} = \frac{\left(\frac{|100 - 100|}{100} \right) + \left(\frac{|100 - 100|}{100} \right) + \dots + \left(\frac{|100 - 100|}{100} \right)}{30} \times 100\%$$

$$\text{Error} = \frac{0 + 0 + 46 + 85 + 14 + 25 + \dots + 0}{30} \times 100\%$$

$$\text{Error} = \frac{662}{30} \times 100\%$$

$$\text{Error} = 22\%$$

$$\text{Kecepatan} = \frac{(\text{jumlah semua kecepatan})}{(\text{banyaknya pengujian})}$$

$$\text{Kecepatan} = \frac{190.4}{30}$$

$$\text{Kecepatan} = 6.3 \text{ Detik}$$

Dari hasil perhitungan rata-rata *error* dan rata-rata kecepatan tersebut diperoleh kesimpulan bahwa uji coba Konfigurasi II dengan nilai *prima* = 3 dan *window* = 10 memiliki rata-rata persentase *error* sebesar 22% dan rata-rata kecepatan 6,3 detik.

4.2.3 Konfigurasi III

Konfigurasi nilai *prima* dan *window* yang digunakan pada pengujian ini adalah *prima* = 5 dan *window* = 4. Pengujian ini menggunakan 30 data uji yang mana semua data tersebut telah terdeteksi plagiarisme dengan persentase yang berbeda-beda. *Output* dari sistem akan dibandingkan dengan nilai sebenarnya. Berikut adalah uji coba terhadap 30 data yang dipaparkan pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Konfigurasi III

pengujian ke-	Nama File	Waktu Proses (detik)	Presentase Kemiripan (persen)	Plagiarism Checker X	Persentase Nilai <i>Error</i>
1	13650074	3.8	100%	100%	0%
2	13650079	4.0	100%	94%	6%
3	10650001	10.4	34%	61%	44%
4	13650085	4.2	100%	100%	0%
5	13650090	4.3	100%	100%	0%
6	10650002	10.2	40%	32%	25%
7	10650003	10.0	24%	49%	51%
8	13650083	4.1	100%	100%	0%
9	10650004	9.8	13%	79%	84%
10	13650122	4.0	100%	100%	0%
11	13650094	4.2	100%	100%	0%
12	10650005	10.5	52%	83%	37%
13	13650130	4.1	100%	98%	2%
14	13650133	4.2	100%	100%	0%
15	10650006	10.3	42%	78%	46%
16	13650143	5.8	100%	85%	18%
17	14650008	4.3	100%	100%	0%
18	14650013	4.6	100%	92%	9%
19	14650001	4.0	100%	89%	12%
20	10650007	10.3	35%	54%	35%
21	14650065	4.1	100%	100%	0%
22	14650046	4.6	100%	73%	37%
23	10650008	10.1	56%	60%	7%
24	14650044	4.2	100%	96%	4%
25	10650010	10.1	55%	100%	45%
26	14650048	4.6	100%	100%	0%
27	14650058	4.4	100%	100%	0%
28	10650009	10.1	46%	100%	54%
29	14650070	4.6	100%	97%	3%
30	14650071	4.0	100%	100%	0%
	Total	187.9			520%

Berdasarkan Hasil Pengujian konfigurasi III pada Tabel 4.3 diperoleh perhitungan rata-rata *error* dan rata-rata kecepatan sebagai berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \tilde{y}_{xi}|}{y_{xi}} \right) \times 100\%$$

$$\text{Error} = \frac{\left(\frac{|100 - 100|}{100} \right) + \left(\frac{|94 - 100|}{94} \right) + \dots + \left(\frac{|100 - 100|}{100} \right)}{30} \times 100\%$$

$$\text{Error} = \frac{0 + 6 + 44 + 0 + 0 + 25 + 51 + 0 + 84 + \dots + 0}{30} \times 100\%$$

$$\text{Error} = \frac{520}{30} \times 100\%$$

$$\text{Error} = 17\%$$

$$\text{Kecepatan} = \frac{(\text{jumlah semua kecepatan})}{(\text{banyaknya pengujian})}$$

$$\text{Kecepatan} = \frac{187.9}{30}$$

$$\text{Kecepatan} = 6.3 \text{ Detik}$$

Dari hasil perhitungan rata-rata *error* dan rata-rata kecepatan tersebut diperoleh kesimpulan bahwa uji coba Konfigurasi III dengan nilai *prima* = 5 dan *window* = 4 memiliki rata-rata persentase *error* sebesar 17% dan rata-rata kecepatan 6,3 detik.

4.2.4 Konfigurasi IV

Konfigurasi nilai *prima* dan *window* yang digunakan adalah *prima* = 7 dan *window* = 10. Pengujian ini menggunakan 30 data uji yang mana semua data tersebut telah terdeteksi plagiarisme dengan persentase yang berbeda-beda. *Output* dari sistem akan dibandingkan dengan nilai sebenarnya. Berikut adalah uji coba terhadap 30 data yang dipaparkan pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Konfigurasi IV

pengujian ke-	Nama File	Waktu Proses (detik)	Presentase Kemiripan (persen)	Plagiarism Checker X	Persentase Nilai <i>Error</i>
1	10650001	10.2	53%	61%	13%
2	12650084	4.0	100%	100%	0%
3	13650025	4.3	100%	100%	0%
4	10650010	10.2	53%	100%	47%
5	13650133	4.2	100%	100%	0%
6	14640014	5.1	100%	82%	22%
7	10650009	10.4	45%	100%	55%
8	14650094	5.1	100%	87%	15%
9	14650096	4.2	100%	100%	0%
10	10650008	10.1	57%	60%	5%
11	10650102	10.0	100%	94%	6%
12	12650017	10.0	100%	100%	0%
13	10650007	10.4	34%	54%	37%
14	14650113	3.8	100%	86%	16%
15	14650072	5.2	100%	82%	22%
16	10650006	10.5	46%	78%	41%
17	12650051	11.4	100%	100%	0%
18	12650065	9.0	100%	94%	6%
19	10650005	10.6	66%	83%	20%
20	13650008	9.7	100%	100%	0%
21	13650015	9.3	100%	96%	4%
22	10650004	10.0	27%	79%	66%
23	13650022	9.8	100%	94%	6%
24	13650061	8.9	100%	100%	0%
25	10650003	10.1	27%	49%	45%
26	13650019	10.4	100%	96%	4%
27	12650072	9.2	100%	58%	72%
28	10650002	10.1	41%	32%	28%
29	13650093	4.0	100%	88%	14%
30	13650132	11.5	100%	100%	0%
	Total	251.7			546%

Berdasarkan Hasil Pengujian konfigurasi IV pada Tabel 4.4 diperoleh perhitungan rata-rata *error* dan rata-rata kecepatan sebagai berikut:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_{xi} - \hat{y}_{xi}|}{y_{xi}} \right) \times 100\%$$

$$\text{Error} = \frac{\left(\frac{|61 - 100|}{61} \right) + \left(\frac{|100 - 100|}{100} \right) + \dots + \left(\frac{|100 - 100|}{100} \right)}{30} \times 100\%$$

$$\text{Error} = \frac{13 + 0 + 0 + 47 + 0 + 22 + 55 + 15 + 0 + \dots + 0}{30} \times 100\%$$

$$\text{Error} = \frac{546}{30} \times 100\%$$

$$\text{Error} = 18\%$$

$$\text{Kecepatan} = \frac{(\text{jumlah semua kecepatan})}{(\text{banyaknya pengujian})}$$

$$\text{Kecepatan} = \frac{251.7}{30}$$

$$\text{Kecepatan} = 8.4 \text{ Detik}$$

Dari hasil perhitungan rata-rata *error* dan rata-rata kecepatan tersebut diperoleh kesimpulan bahwa uji coba Konfigurasi IV dengan nilai *prima* = 5 dan *window* = 4 memiliki rata-rata persentase *error* sebesar 18% dan rata-rata kecepatan 8,4 detik.

4.3 Pembahasan

Setelah melakukan rangkaian uji coba terhadap Aplikasi Deteksi Plagiarisme Menggunakan Algoritma Winnowing Dengan Pendekatan Triword maka dapat diperoleh akurasi dan kecepatan pada tiap konfigurasi sebagaimana pada Tabel 4.5.

Tabel 4.5 Rata-Rata Presentase Nilai *Error* Dan Kecepatan Tiap Konfigurasi

No	Uji Coba	Prima	Window	Nilai Error	Kecepatan
1	Konfigurasi I	2	4	15%	6,2 Detik
2	Konfigurasi II	3	10	22%	6,3 Detik
3	Konfigurasi III	5	4	17%	6,3Detik
4	Konfigurasi IV	7	10	18%	8,4 Detik

Setelah melakukan rangkaian uji coba terhadap Aplikasi Deteksi Plagiarisme Menggunakan Algoritma Winnowing Dengan Pendekatan Triword penulis akan membahas hasil percobaan pada tiap konfigurasi. Berikut adalah hasil pembahasan dari tiap konfigurasi:

1. Konfigurasi I: Prima 2 dan Window 4

Konfigurasi I yang memiliki nilai prima dan *window* paling kecil dibanding konfigurasi lain yaitu 2 dan 4, mampu menghasilkan *output* dengan persentase *error* sebesar 15% dengan rata-rata kecepatan 6,2 detik.

2. Konfigurasi II: Prima 3 dan Window 10

Konfigurasi II dengan nilai prima dan *window* yg lebih besar dari konfigurasi I yaitu 3 dan 10, mampu menghasilkan *output* dengan persentase *error* sebesar 22% dengan rata-rata kecepatan 6,3 detik.

3. Konfigurasi III: Prima 5 dan Window 4

Konfigurasi III dengan nilai prima lebih besar dari konfigurasi II yaitu 5 dan nilai *window* 4, mampu menghasilkan *output* dengan persentase *error* sebesar 17% dengan rata-rata kecepatan 6,3 detik.

4. Konfigurasi IV: Prima 7 dan Window 10

Konfigurasi IV dengan nilai prima dan *window* yg lebih besar dari konfigurasi III yaitu 7 dan 10, mampu menghasilkan *output* dengan persentase *error* sebesar 18% dengan rata-rata kecepatan 8,4 detik.

Berdasarkan pembahasan tersebut diperoleh kesimpulan bahwa Konfigurasi I dengan nilai prima = 2 dan *window* = 4 dapat menghasilkan *output* dengan rata-rata persentase *error* terkecil yaitu 15%. Konfigurasi I juga mampu mendeteksi plagiarisme dengan waktu paling cepat yaitu 6,2 detik.

Penulis menyimpulkan bahwa aplikasi ini belum bisa menghasilkan *output* dengan tingkat *error* 0% karena beberapa hal, diantaranya:

1. Kurangnya data di dalam *database repository*. Aplikasi ini hanya mencari kesamaan dari 120 data di *database* yang diambil dari situs etheses.uin-malang.ac.id. Sedangkan Aplikasi Plagiarism Checker X mampu mencari kesamaan dari teks input ke berbagai situs di internet.
2. Sistem kurang maksimal dalam proses *marking* kata yang Sama. Aplikasi ini hanya memilah teks menjadi 3 kata kemudian mencari sumber plagiarisme pada database. Aplikasi plagiarism Checker X mampu memilah teks menjadi beberapa kalimat kemudian mencari sumber plagiarisme dari internet. Gambar 4.2 dan Gambar 4.3 merupakan perbandingan dari *output* kedua aplikasi tersebut.

Pengujian adalah suatu proses eksekusi program yang bertujuan untuk menemukan kesalahan pada aplikasi. Penelitian ini membangkitkan test case dengan UML activity diagram. Uji coba contoh diambil pada ATM, library management dan uji coba aplikasi sistem penilaian pembelajaran.). Perkembangan komputer serta jaringan internet di tahun 1966 juga telah membuka peluang bagi para pengusaha finansial untuk mengembangkan bisnis mereka secara global Term dengan kelas khusus kadangkadang muncul dalam kelas lain, hal ini sangat sering terjadi dalam satu bahkan beberapa kelas. Term dalam kelas khusus ini dapat digunakan untuk membedakan teks antar kelas yang berbeda dan diberikan bobot yang lebih besar.

Secara umum, term yang lebih terkonsentrasi distribusi kelas cenderung memiliki kekuatan membedakan kelas. Penggunaan metode rule base digunakan pada soal agar bisa menjadi pengoreksi pada saat pencocokan suku kata nanti. Setelah dilakukan pengujian didapati akurasi sebesar 80 % untuk pengenalan gesture dan kesesuaian yang

Gambar 4.2 Output Aplikasi Plagiarism Checker X

pengujian adalah suatu proses eksekusi program yang bertujuan untuk menemukan kesalahan pada aplikasi penelitian ini membangkitkan test case dengan uml activity diagram uji coba contoh diambil pada atm library management dan uji coba aplikasi sistem penilaian pembelajaran perkembangan komputer serta jaringan internet di tahun 1966 juga telah membuka peluang bagi para pengusaha finansial untuk mengembangkan bisnis mereka secara global term dengan kelas khusus kadangkadang muncul dalam kelas lain hal ini sangat sering terjadi dalam satu bahkan beberapa kelas term dalam kelas khusus ini dapat digunakan untuk membedakan teks antar kelas yang berbeda dan diberikan bobot yang lebih besar secara umum term yang lebih terkonsentrasi distribusi kelas cenderung memiliki kekuatan membedakan kelas penggunaan metph1nhmuh1nh1ncjojpqajmh sh gd1node rule base digunakan pada soal agar bisa menjadi pengoreksi pada saat pencocokan suku kata nanti setelah dilakukan pengujian didapati akurasi sebesar 80 untuk pengenalan gesture dan kesesuaian yang

Gambar 4.3 Output Aplikasi Deteksi Plagiarisme menggunakan Algoritma Winnowing Dengan Pendekatan Triword

Dengan tingkat *error* sebesar 15% dapat disimpulkan bahwa aplikasi ini belum bisa mencari teks yang mengandung plagiarisme dengan sempurna. Aplikasi *Plagiarism Checker* yang dijadikan rujukan untuk menghitung akurasi pun belum mencapai akurasi yang sempurna, jadi Penulis menyimpulkan bahwa tidak ada aplikasi yang sempurna didunia ini, kesempurnaan adalah milik Allah *Subhanahu Wata'ala*. Allah mampu mengetahui segala kebenaran atau kecurangan di muka

bumi ini, termasuk kecurangan dalam praktik plagiarisme. Sebagaimana firman Allah *Subhanahu Wata'ala* :

قُلْ إِن تَخْفُوا مَا فِي صُدُورِكُمْ أَوْ تُبْدُوهُ يُعْلَمَهُ اللَّهُ وَيَعْلَمَ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ وَاللَّهُ عَلَى كُلِّ شَيْءٍ قَدِيرٌ

“Katakanlah: jika kalian menyembunyikan apa yang ada di dalam dada kalian, atau kalian menampakkannya, maka Allah mengetahui semua itu. Allah mengetahui apa yang ada di langit dan apa yang ada di bumi. Dan Allah maha kuasa atas segala sesuatu” (QS. Al Imran: 29).

Al-Mahalli dan As-Suyuti (1990) menjelaskan bahwa dalam ayat tersebut, Allah *Subhanahu Wata'ala* memerintahkan pada nabi Muhammad untuk menjelaskan pada kaumnya bahwa Allah *Subhanahu Wata'ala* akan mengetahui segala sesuatu di langit maupun di bumi meskipun kaumnya berusaha menyembunyikan hal tersebut ataupun menampakkannya, *Wallahu A'lamu Bis Shawaab*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan penelitian tentang deteksi plagiarisme menggunakan Algoritma *Winnowing* dengan pendekatan *Triword*, diperoleh kesimpulan sebagaimana berikut:

1. Rata-rata nilai *error* konfigurasi I = 15%, konfigurasi II = 22%, konfigurasi III = 17%, konfigurasi IV = 18%. Berdasarkan hasil pengujian tersebut disimpulkan bahwa konfigurasi I menghasilkan nilai *error* tekecil yaitu 15%, dengan konfigurasi nilai prima 4 dan *window* 2. Hal tersebut membuktikan bahwa konfigurasi tersebut merupakan konfigurasi terbaik dibandingkan 3 konfigurasi lainnya.
2. Rata-rata kecepatan konfigurasi I = 6.2 detik, konfigurasi II = 6.3 detik, konfigurasi III = 6.3 detik, konfigurasi IV = 8.4 detik. Berdasarkan hasil pengujian tersebut disimpulkan bahwa konfigurasi I menghasilkan rata-rata waktu proses tercepat dalam mendeteksi plagiarisme yaitu 6.2 detik, dengan konfigurasi nilai prima 4 dan *window* 2. Hal tersebut membuktikan bahwa semakin kecil nilai prima dan *window* maka akan semakin cepat sistem dalam mendeteksi plagiarisme.

Dari hasil kesimpulan tersebut penulis menyatakan bahwa aplikasi ini belum bisa menghasilkan *output* dengan tingkat *error* 0% karena Kurangnya data di dalam *database repository* dan Sistem kurang maksimal dalam proses *marking* kata yang

Sama. Aplikasi ini juga belum mampu untuk mencari sumber plagiarisme dari internet sebagaimana telah dibahas pada bab sebelumnya.

2.1 Saran

Penulis menyadari bahwa penelitian ini masih dapat ditingkatkan kembali, diperlukan beberapa percobaan lain untuk menghasilkan tingkat *error* terkecil dengan waktu yang tercepat. Adapun saran untuk penelitian selanjutnya sebagai berikut:

1. Diperlukan uji coba dengan jumlah teks yang lebih banyak. Dikarenakan untuk memperoleh hasil penelitian yang lebih maksimal, aplikasi ini memerlukan uji coba terhadap teks dengan jumlah lebih banyak, misalnya teks skripsi mulai Bab I hingga Bab V.
2. Diperlukan uji coba dengan database *repository* yang lebih banyak, Dikarenakan semakin lengkap database *repository* maka akan mempengaruhi tingkat akurasi dan persentase *error* sebuah sistem dalam mendeteksi plagiarisme.
3. Diperlukan uji coba untuk mengetahui tingkat akurasi, presisi, *recall* dari aplikasi ini. Dikarenakan dengan mengukur nilai-nilai tersebut akan diketahui lebih detail bagaimana kualitas *output* dari kinerja sistem.

DAFTAR PUSTAKA

- Alamsyah, N. 2017. “Perbandingan Algoritma *Winnowing* Dengan Algoritma *Rabin Karp* Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi”, eJournal Kedokteran Indonesia, Vol 8, No.3, 2017: 2.
- Alzahrani, SM. Salim, N. and Abraham, A. 2013. “*Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods*”, IEEE Transactions On Systems, Man, And Cybernetics, Vol. 1, No. 1, 2012: 133-149.
- Al-Mahally, Imam Jalaluddin. As-suyutti, Imam Jalaluddin. 1990 “Tafsir Jalalain Berikut Asbab An-nujulnya”, Jilid I Bandung,: Sinar Baru,
- Hasan, EG. Wicaksana, A. Hansun, Seng. 2018. “*The Implementation of Winnowing Algorithm for Plagiarism Detection in Moodle-based E-learning*”, IEEE Computer Society, DOI: 10.1109/ICIS.2018.8466429, 2018: 1
- Herqutanto. 2013. “Plagiarisme, Runtuhnya Tembok Kejujuran Akademik”, eJournal Kedokteran Indonesia, Vol. 1, No. 1, 2013: 1-2.
- Jody. Wibowo , AT. Arifianto, A. 2015. “Analsis dan Implementasi Algoritma *Winnowing* dengan *Synonym Recognition* pada Deteksi Plagiarisme untuk Dokumen Teks Berbahasa Indonesia”, e-Proceeding of Engineering : Vol.2, No.3 Desember 2015: 7674.
- Kent, CK. Salim, N. 2010. “*Features Based Text Similarity Detection*”, Skripsi Tidak Diterbitkan. Malaysia: Faculty of Computer Science and Information System.
- Ridho, M. 2013. “Rancang Bangun Aplikasi Pendeteksi Penjiplakan Dokumen menggunakan Algoritma *Biword Winnowing*”. Skripsi Tidak Diterbitkan. Pekanbaru: UIN Sultan Syarif Kasim Jurusan Teknik Informatika.
- Schleimer, S. Wilkerson, DS. Aiken, A. 2003. “*Winnowing: Local Algorithms for Document Fingerprinting*”. San Diego: Proceedings of the ACM SIGMOD International Conference On Management Of Data.
- KBBI (Kamus Besar Bahasa Indonesia) Edisi Online. <https://kbbi.web.id/>. (diunduh pada tanggal 6 Oktober 2019).
- Li, G., Zhao, S., Wu, J., Li, C., & Liu, Y. (2019). DV-Hop localization algorithm based on minimum mean square error in internet of things. *Procedia computer science*, 147, 458-462.

Tempo, 2016. “Seperempat Esai Mahasiswa Indonesia Terdeteksi Plagiat”.
Malang:Tempo.

<https://nasional.tempo.co/read/743504/seperempat-esai-mahasiswa-indonesia-terindikasi-plagiat>. (diunduh pada tanggal 9 November 2019).

Undang-Undang No.28 Tahun 2014 tentang Hak Cipta Pasal 112.

