



# Perbandingan Algoritma *Winnowing* dan Algoritma *Rabin-Karp* pada Aplikasi Pendeteksi Kesamaan Dokumen Skripsi

Jumadil Nangi<sup>a\*</sup>, Ida Bagus Gede Pala Asmara<sup>a</sup>, Muh. Ihsan Sarita<sup>a</sup>, Laode Muh. Golok Jaya<sup>a</sup>,  
Hasmina Tari Mokui<sup>b</sup>, LM Tajidun<sup>a</sup>

<sup>a</sup>Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo

<sup>b</sup>Teknik Elektro, Fakultas Teknik, Universitas Halu Oleo

Naskah masuk: 10 Januari 2024; Diterima untuk publikasi: 2 Februari 2024

DOI: 10.21456/vol14iss2pp131-142

---

## Abstract

Plagiarism is often found in the academic world and is considered a serious violation because it involves the appropriation of ideas, opinions, or writings of others. A thesis is the final work of a student that must meet scientific standards and be tailored to the field of study. It is important to check the similarity of the thesis using a web-based sistem to prevent plagiarism and ensure academic integrity. This sistem will be developed as a web-based platform with the aim of comparing the *Winnowing* algorithm and the *Rabin-Karp* algorithm in checking the similarity of thesis/final project texts with existing thesis data. In principle, both methods involve searching for strings using hashing functions to compare the sought string ( $m$ ) with the compared string ( $n$ ) by comparing the results of the hashing function used. However, the *Winnowing* algorithm differs in that it does not use all hash values from each formed set of grams. The hash values formed in the previous stage will be divided into a window of size ( $w$ ). In this research, the sistem testing uses data from Computer Engineering students at Halu Oleo University to facilitate checking the plagiarism level of theses using the *Rabin-Karp* and *Winnowing* algorithms. In this study, the *Rabin-Karp* and *Winnowing* algorithms have been implemented successfully in the plagiarism checking sistem for students' theses. The test results for the comparison of the *Winnowing* and *Rabin-Karp* algorithms in terms of processing time show that the *Rabin-Karp* algorithm takes 1.509 seconds, while the *Winnowing* algorithm takes 1.508 seconds. Subsequent testing using Normalized Mean Absolute Error (NMAE) reveals that the *Rabin-Karp* algorithm has an absolute error value of 0.1829, while the *Winnowing* algorithm has a value of 0.0194. Therefore, based on the NMAE test, the *Winnowing* algorithm performs better than the *Rabin-Karp* algorithm.

**Keywords:** Detection Sistem, Thesis, Web, *Winnowing*, *Rabin-Karp*

## Abstrak

Plagiarisme sering ditemukan didunia akademik dan dianggap sebagai pelanggaran serius karena melibatkan pengambilan ide, pendapat, atau tulisan orang lain. Skripsi adalah karya akhir mahasiswa yang harus memenuhi standar ilmiah dan disesuaikan dengan bidang studi. Penting untuk melakukan pengecekan kesamaan skripsi menggunakan sebuah sistem berbasis *website* untuk mencegah plagiat dan memastikan integritas akademik. Sistem ini akan dibuat berbasis *website* dengan tujuan membandingkan algoritma *Winnowing* dan algoritma *Rabin-karp* dalam pengecekan kesamaan atau kemiripan naskah skripsi/tugas akhir dengan data-data skripsi yang sudah ada sebelumnya. Pada prinsipnya kedua metode ini sama yaitu melakukan pencarian string yang menggunakan fungsi *hashing* untuk membandingkan string yang dicari ( $m$ ) dengan string yang dibandingkan ( $n$ ) dengan membandingkan hasil dari fungsi *hashing* yang digunakan. Namun pada algoritma *Winnowing* perbedaannya tidak menggunakan semua nilai *hash* dari setiap rangkaian gram yang dibentuk. Nilai *hash* yang dibentuk pada tahap sebelumnya akan dibagi ke dalam *window* berukuran ( $w$ ). Pada penelitian ini pengujian sistem menggunakan data skripsi mahasiswa Teknik Informatika Universitas Halu Oleo untuk memudahkan pengecekan tingkat plagiarisme skripsi terhadap algoritma *Rabin-Karp* dan algoritma *Winnowing*. Pada penelitian ini algoritma *Rabin-Karp* dan algoritma *Winnowing* telah diimplementasikan dengan baik pada sistem pengecekan kemiripan skripsi mahasiswa. Hasil pengujian perbandingan algoritma *Winnowing* dan algoritma *Rabin-Karp* dari sisi waktu proses algoritma *Rabin-Karp* waktu proses 1.509 detik sedangkan algoritma *Winnowing* 1.508 detik. Pengujian selanjutnya dengan *Normalized Mean Absolute Error* (NMAE) algoritma *Rabin-Karp* memiliki nilai absolute error di nilai 0.1829 sedangkan algoritma *Winnowing* memiliki nilai 0.0194 maka dari pengujian NMAE algoritma *Winnowing* memiliki kinerja yang lebih baik dari pada algoritma *Rabin-Karp*.

**Keywords:** Sistem Pendeteksi, Skripsi, *Web*, *Winnowing*, *Rabin-Karp*

\*) *Corresponding author*: jumadilnangi87@gmail.com

## 1. Pendahuluan

Perkembangan teknologi informasi pada saat ini berkembang begitu pesat, dengan adanya teknologi dapat membantu atau meringankan pekerjaan manusia. Selain itu berkembangnya suatu teknologi tentu ada dampak positif dan negatifnya, salah satu contoh dampak negatifnya adalah orang-orang cenderung ingin sesuatu yang instan, bentuknya yaitu seperti plagiat. Plagiat atau plagiarisme adalah pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seakan-akan karangan atau pendapat sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri.

Plagiarisme selalu menjadi sorotan, tak terkecuali pada sektor akademis baik dari tingkat sekolah hingga jenjang perguruan tinggi. Tindakan ini sangat tidak mencerminkan seorang yang terpelajar, selain mematikan kreativitas juga kebebasan berinovasi menjadi sempit karena kurang menghargai karya orang lain. Plagiarisme atau penjiplakan karya tulis orang lain dalam dunia akademik dipicu oleh banyak faktor. Salah satu faktor pemicunya adalah penulis (mahasiswa) ingin segera menyelesaikan skripsinya agar bisa meraih gelar akademik secepatnya tanpa harus bekerja keras sesuai proses riset dan penulisan ilmiah yang benar. Salah satu plagiarisme yang sering terjadi di kalangan mahasiswa adalah plagiarisme di dalam tugas akhir (skripsi) (Alamsyah and Rasyidan, 2019).

Tugas akhir (skripsi) merupakan salah satu bentuk karya tulis ilmiah yang dibuat oleh mahasiswa pada tahap akhir dari masa studinya. Tugas akhir (skripsi) dibuat berdasarkan hasil penelitian, kajian terhadap permasalahan yang diperoleh dari pelaksanaan PKL (Praktik Kerja Lapangan), atau permasalahan rill lainnya. Jadi dengan adanya sistem ini dapat digunakan oleh dosen dan pihak jurusan untuk mendeteksi tingkat plagiarisme tugas akhir (skripsi) mahasiswa.

Tindakan plagiarisme secara perlahan harus dicegah dan dihilangkan dengan melakukan pendeteksian plagiat secara manual maupun dengan memanfaatkan metode pencocokan *string*. Dengan demikian melakukan pendeteksian plagiarisme secara manual sangat tidak efektif sehingga algoritma *Winnowing* atau algoritma *Rabin-Karp* dapat menjadi solusi untuk menyelesaikan hal tersebut (Aldian and Mubarak, 2019).

Algoritma *Winnowing* adalah sebuah metode yang digunakan untuk mendeteksi tindakan plagiarisme dengan menggunakan teknik *hashing*. Fungsinya sebagai sebuah *fingerprnt* dokumen, dengan *input* berupa dokumen teks. Algoritma ini akan menghasilkan sekumpulan nilai *hash* yang dihasilkan dari perhitungan ASCII pada setiap karakter dalam dokumen. Nilai-nilai *hash* ini kemudian digunakan sebagai *fingerprnt* untuk mendeteksi adanya plagiarisme (Sunardi *et al.*, 2018). Keunggulan

algoritma *winnowing* yaitu menyimpan nilai *hash* terkecil sehingga efisiensi dalam penggunaan penyimpanan dan memberikan kontrol terhadap sejauh mana kemiripan/kesamaan dianggap signifikan atau memiliki pengaturan nilai ambang batas. Adapun kelemahan algoritma *winnowing* kurang efisien untuk pola-pola kecil serta parameter sensitif bergantung pada pengaturan nilai ambang batas (Schleimer *et al.*, 2018).

Algoritma *Rabin-Karp* adalah metode pencarian string yang dikembangkan oleh *Michael Rabin* dan *Richard Karp*. Algoritma ini menggunakan teknik *hashing* untuk membandingkan *string* yang dicari dengan *string* yang dibandingkan. Fungsi *hashing* digunakan untuk mengubah data menjadi bilangan bulat yang relatif kecil yang dapat digunakan sebagai indeks pada *array*. Proses *hashing* merupakan bagian penting dari algoritma *Rabin-Karp*, dimana karakter atau tanda baca ditransformasikan menjadi nilai atau angka dengan menggunakan representasi ASCII (Suryati *et al.*, 2018). Keunggulan algoritma *rabin-karp* yaitu fleksibilitas pencarian Dimana pencarian pola bersifat dinamis dan adaptif, efisiensi pada pola pendek. Adapun kelemahan *rabin-karp* risiko tumpang tindih *hash* yang dapat mempengaruhi keakuratan pencarian dan kinerja bisa menurun ketika menghadapi pola yang sangat Panjang (Cormen *et al.*, 2019).

Algoritma *Winnowing* dan algoritma *Rabin-Karp* pada aplikasi pendeteksi kesamaan dokumen skripsi dapat memberikan hasil yang memadai dalam menemukan hasil kemiripan antara dokumen skripsi atau tugas akhir seperti ketepatan dalam mendeteksi kesamaan, waktu proses yang bersaing serta kemampuan menangani dataset besar dari kedua algoritma yaitu *Winnowing* dan algoritma *Rabin-Karp* (Faisal *et al.*, 2020). Penelitian ini menggunakan algoritma *Winnowing* dan algoritma *Rabin-Karp* karena kedua algoritma ini memiliki proses sama dalam pengambilan *fingerprnt* yaitu menggunakan nilai-nilai *hash* dari representasi ASCII sebagai data dalam mendeteksi tingkat kesamaan suatu *string*, tetapi dari sisi *Respons time*, *Normalized Mean Absolute Error* (NMAE) dan kompleksitas algoritma *Winnowing* dan algoritma *Rabin-Karp* terhadap dokumen skripsi/tugas akhir mahasiswa belum dapat diketahui rata-rata kompleksitas dari algoritma tersebut.

Penelitian sebelumnya yang dilakukan oleh (Alamsyah and Rasyidan, 2019) dihasilkan sebuah sistem deteksi plagiarisme judul skripsi menggunakan algoritma *winnowing* dengan menggunakan nilai *n-gram*, *window*, dan bilangan prima. Kombinasi nilai *n-gram* = 3, *window* = 3, dan bilangan prima = 2 diperoleh tingkat plagiarisme yang tinggi sebesar 73.86%. Penelitian lainnya (Aldian and Mubarak, 2019) dihasilkan pendeteksi plagiarisme pada dokumen berupa *text* berbasis *website* menggunakan algoritma *rabin-karp* dengan menggunakan 10

dokumen yang diuji dengan nilai k-gram = 1 diperoleh presentasi kemiripan sebesar 57,14%. Penelitian lainnya oleh Parewe *et. al.*, (2021), dihasilkan perbandingan algoritma *winnowing* dan algoritma *manber* dalam mendeteksi berita *hoax* dengan parameter perbandingannya berupa *response time* dan nilai presentasi kesamaan teks berita *hoax*, menjelaskan hasil persentase 65% dari segi *response time* algoritma *manber* lebih baik dari algoritma *winnowing*, tetapi dari segi ketepatan kesamaan diperoleh *winnowing* lebih baik dari pada *manber*.

Berdasarkan uraian di atas peneliti merancang dan mengembangkan sebuah sistem berbasis *website* membandingkan algoritma *Winnowing* dan algoritma *Rabin-Karp* berdasarkan *response time*, NMAE, serta *space complexity* dalam mencari kemiripan/kesamaan dokumen skripsi/tugas akhir.

Penelitian ini bertujuan untuk membandingkan algoritma *rabin-karp* dan algoritma *winnowing* dalam mendeteksi kesamaan dokumen skripsi berbasis *website* dengan parameter perbandingannya berupa *response time*, *Normalized Mean Absolute Error* (NMAE), dan *space complexity*. Metode pengembangan sistem *Rational Unified Process* (RUP), keunggulan dengan menggunakan RUP fleksibilitas yang dapat disesuaikan dengan kebutuhan penelitian, dokumentasi yang baik, dan menejemen proyek yang efektif (Kruchten, 2003). Pengujiannya menggunakan *blackbox testing* dengan pengujian fungsionalitas eksternal sistem dengan memperhatikan bagaimana fungsi-fungsi tersebut diimplementasikan (Yani *et al.*, 2020).

Manfaat dari penelitian ini yaitu membandingkan algoritma *winnowing* dan *rabin-karp* dengan parameter perbandingannya berupa *response time*, NMAE, serta *space complexity*. Selain itu, sistem ini juga dapat berfungsi untuk membantu mengatasi masalah kesamaan dokumen skripsi dengan melihat hasil *similarity* yang dihasilkan dari kedua algoritma tersebut serta dapat memberikan kontribusi dalam mencegah tindakan kesamaan dokumen skripsi di dunia akademik dengan memberikan alat yang efektif bagi dosen dan pihak jurusan untuk mendeteksi kesamaan dalam skripsi/tugas akhir mahasiswa.

## 2. Kerangka Teori

### 2.1. Information Retrieval

*Information retrieval* adalah bidang yang mempelajari tentang sistem pengindeksan, pencarian, dan penyimpanan data yang tidak terstruktur khususnya teks. Dalam proses pengindeksan, terdapat tahap *stemming* yang merupakan proses mengubah bentuk kata menjadi kata dasar. Proses ini sangat dipengaruhi oleh bahasa yang digunakan, karena harus mengaplikasikan aturan morfologi dari bahasa tersebut (Zulfikar, 2017).

### 2.2. ASCII (American Standard Code for Information Interchange)

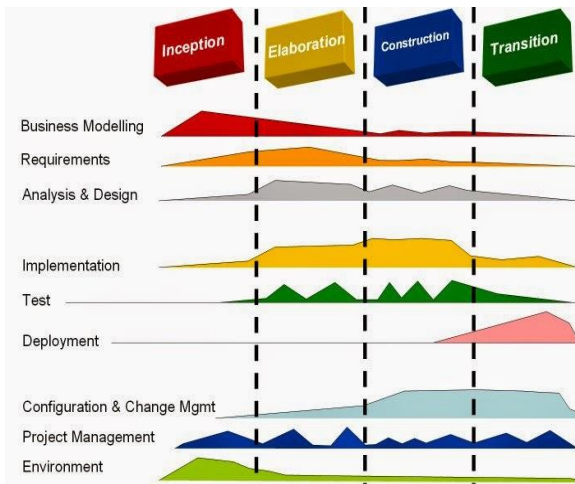
ASCII adalah standar internasional untuk kode huruf dan simbol seperti *Hex* dan *Unicode*. Namun, ASCII lebih bersifat universal dan selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. ASCII memiliki komposisi bilangan biner sebanyak 7 bit, namun disimpan sebagai kode 8 bit dengan menambahkan satu bit tambahan yang sering digunakan untuk uji prioritas (Tantoni and Zaen, 2018).

Tabel 1. Kode ASCII

Code	ASCII	Binnary	Code	ASCII	Binnary
65	A	01000001	97	A	01100001
66	B	01000010	98	B	01100010
67	C	01000011	99	C	01100011
68	D	01000100	100	D	01100100
69	E	01000101	101	E	01100101
70	F	01000110	102	F	01100110
71	G	01000111	103	G	01100111
72	H	01001000	104	H	01101000
73	I	01001001	105	I	01101001
74	J	01001010	106	J	01101010
75	K	01001011	107	K	01101011
76	L	01001100	108	L	01101100
77	M	01001101	109	M	01101101
78	N	01001110	110	N	01101110
79	O	01001111	111	O	01101111
80	P	01010000	112	P	01110000
81	Q	01010001	113	Q	01110001
82	R	01010010	114	R	01110010
83	S	01010011	115	S	01110011
84	T	01010100	116	T	01110100
85	U	01010101	117	U	01110101
86	V	01010110	118	V	01110110
87	W	01010111	119	W	01110111
88	X	01011000	120	X	01111000
89	Y	01011001	121	Y	01111001
90	Z	01011010	122	Z	01111010

### 2.3. Metode Rational Unified Process (RUP)

RUP merupakan kerangka kerja pengembangan perangkat lunak yang sistematis dan terstruktur M. (Zaminkar and Reshadinezhad, 2019). RUP menawarkan pendekatan iteratif dan incremental terhadap pengembangan perangkat lunak, sehingga memungkinkan pengembangan sistem secara progresif membangun dan meningkatkan produk perangkat lunak (Young, 2013). Fase yang digunakan dalam metode ini dapat dilihat pada Gambar 1.



Gambar 1. Metode RUP

RUP memiliki 4 tahap yaitu *inception*, *elaboration*, *construction*, dan *transition* (Kruchten, 2003).

#### 1. *Inception*

Tahapan ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modelling*), mendefinisikan kebutuhan akan sistem yang dibuat serta analisis dan desain.

#### 2. *Elaboration*

Tahap ini lebih pada analisis dan desain sistem (*prototype*).

#### 3. *Construction*

Tahapan ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak.

#### 4. *Transition*

Tahapan ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*.

### 2.4. Algoritma Rabin-Karp

Algoritma *Rabin-Karp* adalah metode pencarian *string* yang menggunakan fungsi *hashing* untuk membandingkan *string* yang dicari (*m*) dengan *string* yang dibandingkan (*n*) dengan membandingkan hasil dari fungsi *hashing* yang digunakan. Algoritma ini menggunakan fungsi *hashing* untuk menemukan pola dalam *string* teks. Algoritma ini menggunakan *k-gram* dan *hashing* sebagai karakteristiknya dan dilakukan setelah proses *preprocessing* (Purba and Situmorang, 2017). Fungsi *hashing* adalah proses penting dalam algoritma *Rabin-Karp*. Ini adalah fungsi matematis yang mengubah data menjadi bilangan bulat yang relatif kecil yang dapat digunakan sebagai indeks pada *array* (Suryati *et al.*, 2018).

### 2.5. Algoritma *Winnowing*

Algoritma *Winnowing* adalah metode yang digunakan untuk mengecek kemiripan kata (*fingerprint* dokumen) untuk mendeteksi plagiarisme. Secara teknis, *Winnowing* adalah perluasan dari implementasi algoritma *Rabin-Karp fingerprint*

dengan penambahan metode *window* (Apridiansyah *et al.*, 2022).

Tahapan *text preprocessing* sebagaimana yang telah dilakukan pada algoritma *Rabin-Karp* juga dilakukan pada algoritma *Winnowing*. Lanjutan proses memotong *string* sepanjang *k* dalam proses perhitungan *k-gram* dan lanjutan proses *hashing* terhadap seluruh pecahan *string* juga sama antara kedua algoritma. Perbedaannya dimana algoritma *Winnowing* tidak menggunakan semua nilai *hash* dari setiap rangkaian *gram* yang dibentuk. Nilai *hash* yang dibentuk pada tahap sebelumnya akan dibagi ke dalam *window* berukuran *w*. *Window* adalah proses pembentukan *substring* dari nilai *hash* sepanjang *w-gram*. Dari proses *Winnowing* akan menghasilkan *fingerprint* yang nanti akan digunakan untuk pencocokan plagiasi. *Window* pertama berisi nilai *hash* pertama sampai nilai *hash* ke-*w*. *Window* kedua dibentuk dari nilai *hash* kedua sampai nilai *hash* ke-*w+1* dan seterusnya sampai terbentuk *window* dari seluruh nilai *hash*. Dari *window* yang telah dibentuk dilakukan pemilihan nilai *hash* terkecil pada tiap *window* untuk dijadikan *fingerprint* tiap dokumen. Dari rangkaian nilai *hash* inilah akan dipilih nilai *hash* terkecil, jika terdapat 2 atau lebih nilai yang sama, nilai *hash* terkecil yang paling kanan yang akan di pilih (Sugiono *et al.*, 2018).

### 2.6. *Dice's Similarity Coficient* (Deteksi Kemiripan)

*Dice coefficient* atau *dice similarity* adalah metode untuk menghitung tingkat kemiripan antara 2 objek dengan cara mengalikan jumlah nilai irisan antara dokumen dan *query*, lalu membagi dengan jumlah nilai dari dokumen dan *query* (Parewe *et al.*, 2021). Terdapat pada persamaan (1) :

$$S = \frac{2C}{A+B} * 100\% \quad (1)$$

Keterangan:

S : *Similarity* (Kemiripan)

A : Jumlah *fingerprint* dengan nilai *hash* dari dokumen 1

B : Jumlah *fingerprint* dengan nilai *hash* dari dokumen 2

C : Jumlah *fingerprint* dengan nilai *hash* yang sama pada kedua dokumen

### 2.7. *Rolling Hash*

*Rolling hash* merupakan tahapan melakukan proses *hashing* pada *string* yang telah dibagi menjadi *k-gram*. Fungsi ini akan mengubah atau mentransformasikan data untuk menciptakan *fingerprint* yang disebut *hash value*. Fungsi *hash* yang baik adalah yang menghasilkan sedikit *hash collision*. Solusi yang digunakan untuk mengatasi *hash collision* adalah menggunakan *rolling hash* (Ramadhani, 2015). Dalam prosesnya, digunakan basis yang biasanya adalah bilangan prima yang cukup besar untuk meminimalkan terjadinya tabrakan. Salah satu *hash*

yang sering digunakan adalah *rolling hash* dengan persamaan (2) berikut:

$$H_{(c_1 \dots c_n)} = c_1 * b^{(n-1)} + c_2 * b^{(n-2)} + \dots + c_{(n-1)} * b^{(1)} + c_n \quad (2)$$

Keterangan:

- c : nilai ASCII pada karakter
- B : basis bilangan prima (tidak ditentukan)
- N : banyaknya karakter atau panjang rangkaian *n-gram*

### 2.3. Uji Validasi Normalized Mean Absolute Error (NMAE)

NMAE adalah metrik yang digunakan untuk mengukur tingkat kesalahan atau *error* antara prediksi dan nilai sebenarnya dalam konteks yang mempertimbangkan rentang atau skala nilai sebenarnya (Ati *et al.*, 2012). Kesalahan absolut mengacu pada perbedaan besar tanpa memperhatikan arah kesalahan. Perhitungan nilai NMAE dapat dilihat pada persamaan 3 dan persamaan 4 dibawah ini.

$$MAE = \frac{1}{n} \sum_{i=1}^n | \text{nilai similarity } A_i - \text{nilai similarity } B_i | \quad (3)$$

$$NMAE = \frac{MAE}{(A,B)_{max} - (A,B)_{min}} \quad (4)$$

keterangan:

- NMAE : *Normalized Mean Absolute Error*
- MAE : *Mean Absolute Error*, yang dihitung sebagai rata-rata nilai absolut dari selisih antara prediksi dan nilai sebenarnya
- (A,B)*max* : nilai maksimum antara nilai *similarity* A dan B
- (A,B)*min* : nilai minimum antara nilai *similarity* A dan B
- N : jumlah data.

Interpretasi hasil Uji NMAE:

- 1) NMAE memiliki rentang nilai dari 0 hingga 1.
- 2) Nilai NMAE yang lebih rendah menunjukkan kinerja yang lebih baik, dengan 0 yang menunjukkan bahwa prediksi adalah sempurna (tanpa kesalahan) dan 1 yang menunjukkan kesalahan yang setara dengan seluruh rentang nilai sebenarnya.

## 3. Metode

Pada bagian ini menjelaskan mengenai langkah-langkah yang dilakukan untuk mencapai tujuan penelitian.

### 3.1. Metode Pengumpulan Data

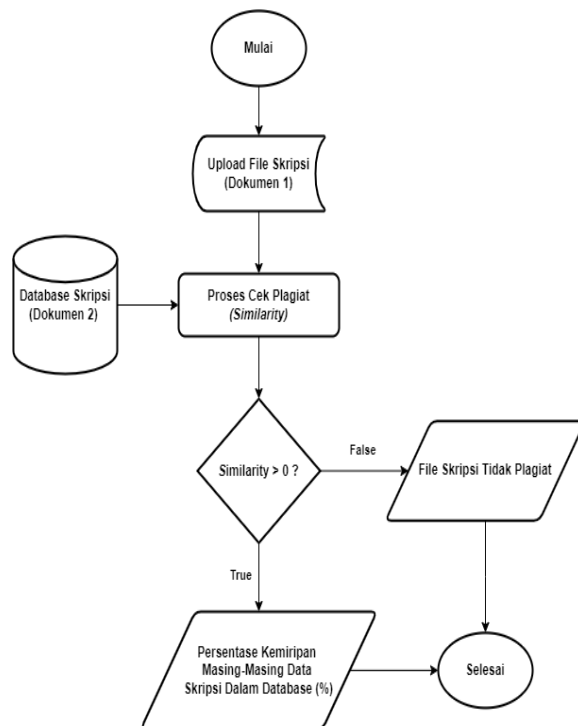
Dalam penelitian ini metode pengumpulan data yang digunakan dalam perancangan sistem dengan melakukan studi literatur. Studi literatur dilakukan dengan cara mengumpulkan referensi yang berkaitan dengan objek penelitian yang dilakukan dengan mencari dan mempelajari buku-buku, jurnal maupun *e-book*, serta penelusuan *internet* yang berkaitan dengan algoritma *Winnowing* dan algoritma *Rabin-Karp* sebagai penunjang yang dapat dijadikan sebagai landasan penelitian. Sehingga algoritma *Winnowing* dan algoritma *Rabin-Karp* dapat diimplementasikan dalam aplikasi pendeteksi plagiarisme tugas akhir (skripsi) Mahasiswa Teknik Informatika di Universitas Halu Oleo.

### 3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *Rational Unified Process* (RUP). Dalam metode pengembangan sistem ini, terdiri dari 4 tahapan pengembangan perangkat lunak, yaitu (Sutedi and Agarina, 2017) :

#### 3.2.1. Permulaan (Inception)

Tahap *inception* dilakukan dengan perancangan *business modeling canvas*, pendefinisian kebutuhan sistem, perancangan *use case diagram* dan pengujian perancangan.

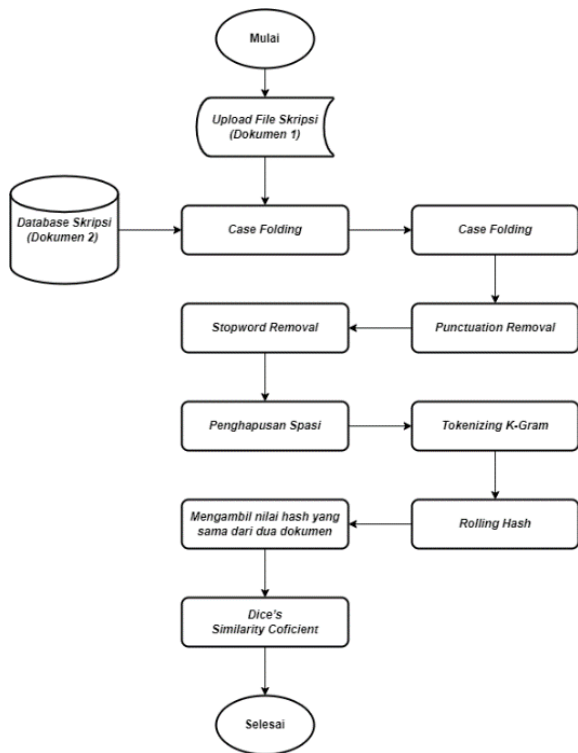


Gambar 2. Flowchart Alur Sistem

Gambar 2. merupakan *flowchart* alur sistem dimana user akan diminta melakukan login terlebih dahulu untuk mengakses sistem. Setelah berhasil maka user akan dialihkan ke halaman dashboard. Pada tampilan dashboard, user dapat melihat data.

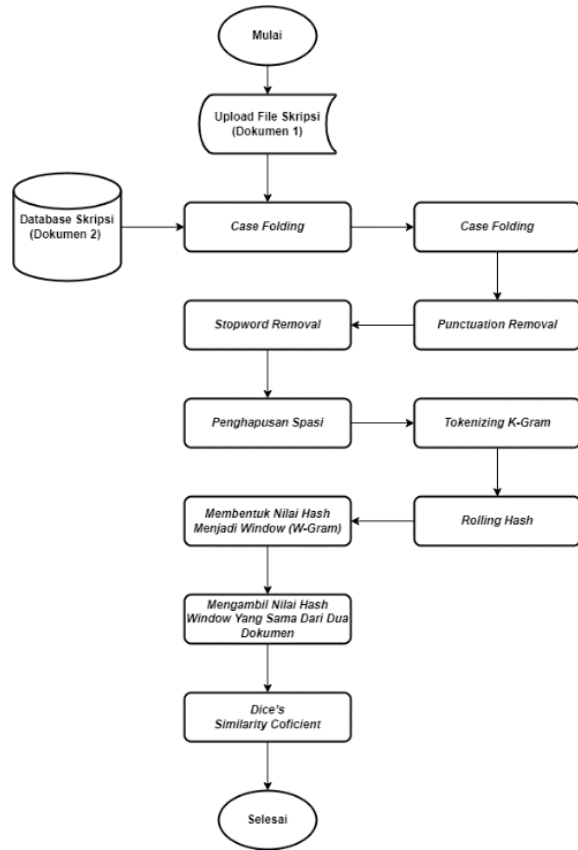
Kemudian *user* dapat mengupload file skripsi untuk melakukan pengecekan kesamaan serta perbandingan kemiripan dari algoritma *winnowing* dan algoritma *rabin-karp* yang berada pada *database*. Kemudian akan menampilkan persentase perbandingan nilai algoritma *winnowing* dan algoritma *rabin-karp*.

Gambar 3. merupakan *flowchart* algoritma *rabin-karp* *user* mulai mengupload file skripsi dan sistem akan mengambil file skripsi yang terdapat pada *database* kemudian akan masuk ke proses *text preprocessing*, setelah melakukan proses *text preprocessing*, selanjutnya ke tahap *tokenizing* kemudian dari proses tersebut akan diambil nilai *hash* nya berdasarkan algoritma *rabin-karp*, kemudian dihitung nilai *dice's similarity* menggunakan *dice's similarity coefficient*, *NMAE* dan *space complexity*.



Gambar 3. *Flowchart* Algoritma Rabin-Karp

Gambar 4. merupakan *flowchart* Algoritma *winnowing* *user* mulai mengupload file skripsi dan sistem akan mengambil file skripsi yang terdapat pada *database* kemudian akan masuk ke proses *text preprocessing*, setelah melakukan proses *text preprocessing*, selanjutnya ke tahap *tokenizing* kemudian dari proses tersebut akan membentuk nilai *hash* menjadi *window (W-Gram)* kemudian akan diambil nilai *hash* nya berdasarkan algoritma *winnowing*, kemudian dihitung nilai *dice's similarity* menggunakan *dice's similarity coefficient*, *NMAE* dan *space complexity*.



Gambar 4. *Flowchart* Algoritma Winnowing

*Text preprocessing* merupakan langkah awal yang akan dilakukan untuk memproses teks agar dapat digunakan dalam proses *tokenizing* teks. Tujuan dari *text preprocessing* adalah untuk mengembalikan teks menjadi bahasa yang alami. Secara umum dalam tahapan *text preprocessing*, yaitu (Mangundap *et al.*, 2022):

- 1) *Data Input*
- 2) *Case Folding*
- 3) *Punctuation Removal*
- 4) *Stopword Removal*
- 5) *Stemming*
- 6) *Penghapusan Spasi*

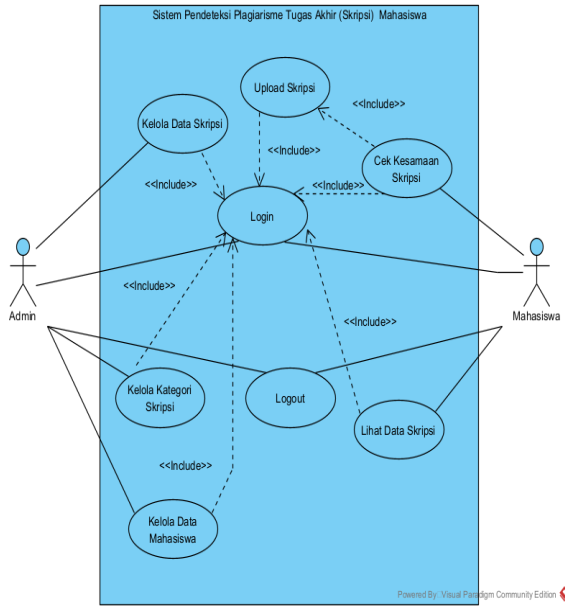
*Tokenizing* proses membagi karakter-karakter teks yang telah melalui proses *text preprocessing* sepanjang nilai *k* yang dibaca berurutan dari awal hingga akhir. Pada contoh kasus ini peneliti menggunakan nilai *k-gram* yaitu *4-gram*.

### 3.2.2. Perluasan/Perencanaan (*Elaboration*)

Tahap *elaboration* penambahan kebutuhan sistem dan *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram* serta tahapan pengujian.

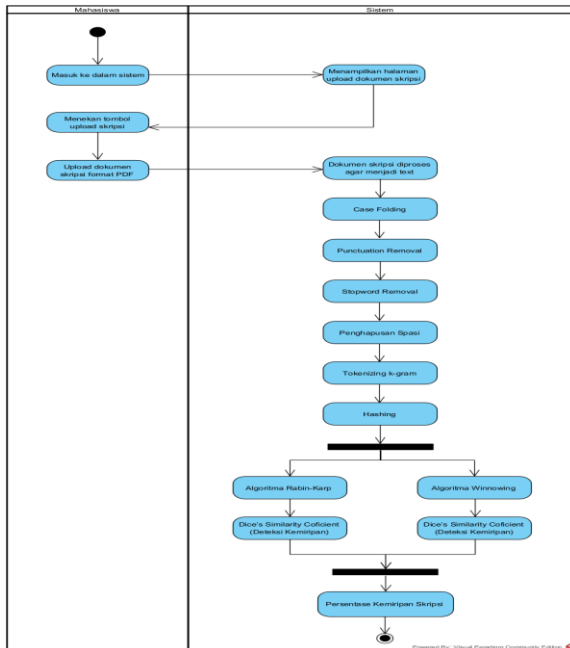
Gambar 5. merupakan *use diagram* dari sistem dimana admin dan mahasiswa dapat melakukan *login* terlebih dahulu. Admin dapat melakukan kelola data skripsi, kelola kategori skripsi, kelola data mahasiswa dan *logout* sistem. Sedangkan mahasiswa dapat

melakukan *upload* skripsi, cek kesamaan skripsi, lihat data skripsi, dan *logout* pada sistem.



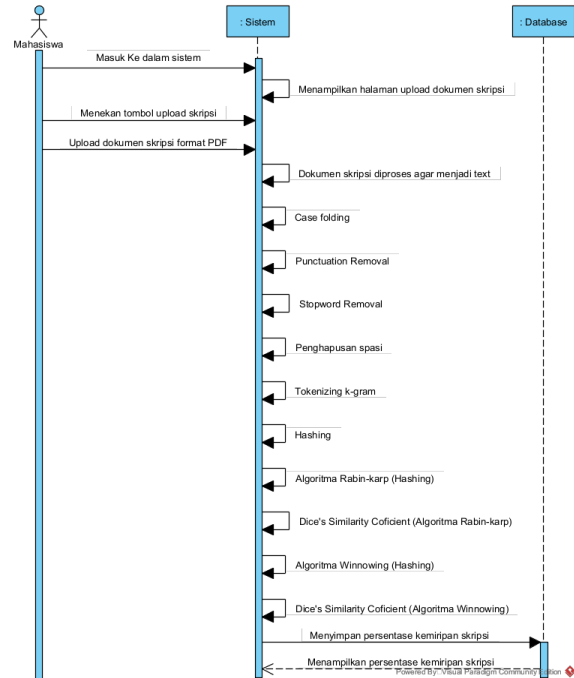
Gambar 5. Use Case Diagram

Gambar 6. menunjukkan aktivitas sistem *user* memasukkan dokumen skripsi kemudian sistem akan mengecek dengan dokumen skripsi yang berada di *database* dalam melakukan perbandingan algoritma *winning* dan Algoritma *Rabin-Karp* dengan 3 parameter yaitu *response time*, *NMAE*, dan *space complexity*.



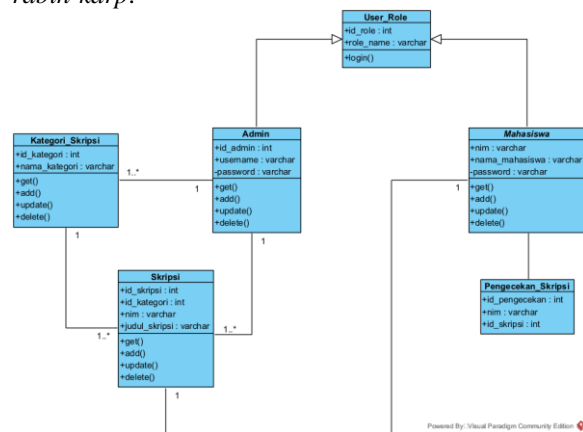
Gambar 6. Activity Diagram Perbandingan Algoritma *Wining* dan Algoritma *Rabin-Karp*

Gambar 7. menunjukkan interaksi pengguna dengan sistem dalam melihat hasil perbandingan algoritma *Wining* dan algoritma *Rabin-Karp*. Pengguna akan mengupload dokumen skripsi yang kemudian sistem akan mengecek dengan dokumen skripsi yang terdapat dalam *database*, kemudian sistem akan menampilkan hasil perbandingan algoritma *wining* dan algoritma *rabin-karp*.



Gambar 7. Sequence Diagram Perbandingan Algoritma *Wining* dan *Rabin-Karp*

Gambar 8. merupakan rancangan *class diagram* perbandingan algoritma *wining* dan algoritma *rabin-karp*.



Gambar 8. Class Diagram

### 3.2.3. Konstruksi (Construction)

Tahap *construction* tahapan dalam perancangan sistem yang disesuaikan dengan *usecase diagram*.

### 3.2.4. Transisi (Transition)

Tahap *transition* dilakukan pada tahapan instalasi sistem.

## 4. Hasil dan Pembahasan

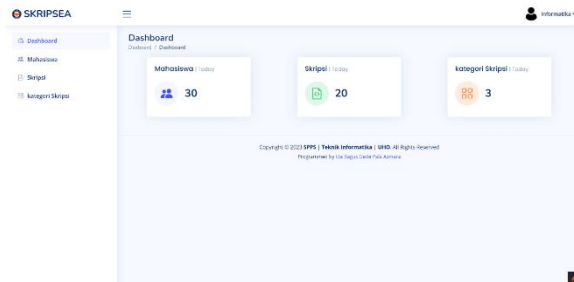
### 4.1. Implementasi Interface

Pada Gambar 9. merupakan tampilan halaman *login user* dimana admin dan mahasiswa memasukkan *username* dan *password* terlebih dahulu untuk masuk ke dalam aplikasi.



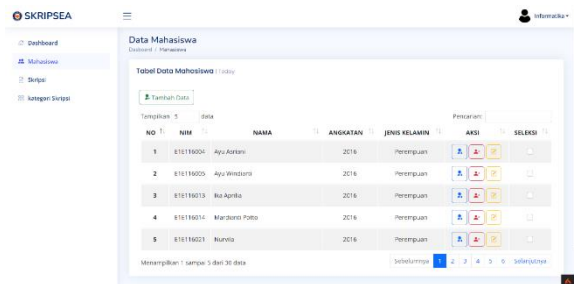
Gambar 9. Tampilan Halaman *Login User*

Gambar 10. merupakan halaman *Dashboard* merupakan halaman awal ketika *user login* sebagai mahasiswa dan admin.



Gambar 10. Tampilan Halaman *Dashboard*

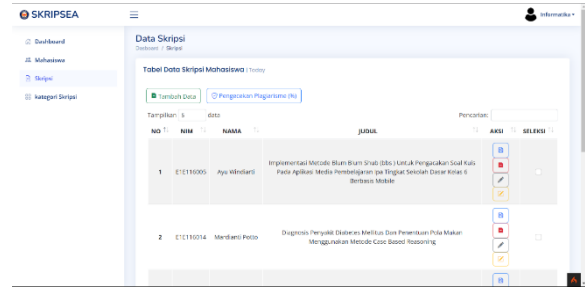
Gambar 11. merupakan halaman data mahasiswa merupakan halaman dimana terdapat data-data mahasiswa. Pada halaman ini terdapat fitur tambah, edit, detail, dan hapus data mahasiswa.



Gambar 11. Tampilan Halaman Data Mahasiswa

Gambar 12. halaman ini berisi tentang data skripsi mahasiswa. Pada halaman ini terdapat fitur

tambah, detail, edit, dan hapus data skripsi serta terdapat fitur pengecekan plagiarisme.



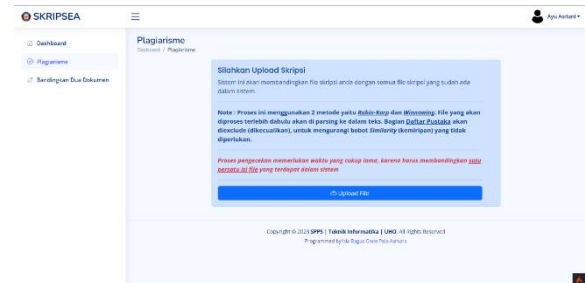
Gambar 12. Tampilan Halaman Data Skripsi

Gambar 13. halaman ini terdapat kategori dari skripsi yang terbagi menjadi 3 yaitu *Komputasi Berbasis Jaringan*, *Komputasi Cerdas* dan *Visualisasi dan Rekayasa Perangkat Lunak*. Dimana terdapat fitur tambah data kategori, hapus, edit dan detail kategori.



Gambar 13. Tampilan Halaman Kategori Skripsi

Gambar 14. merupakan halaman mahasiswa dapat mengunggah *File* skripsi yang ingin dicek kesamaan dokumen skripsi yang kemudian sistem akan membandingkan *File* skripsi anda dengan semua *File* skripsi yang sudah ada dalam sistem.

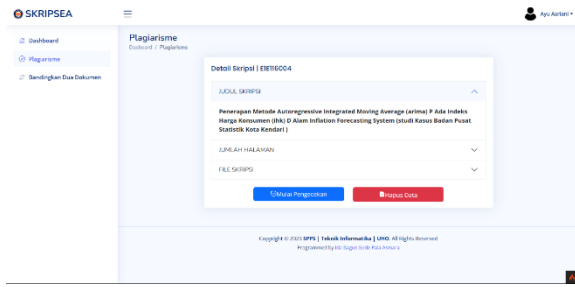


Gambar 14. Tampilan Halaman Perbandingan *Winnowing* dan *Rabin-Karp*

Gambar 15. merupakan halaman saat mahasiswa telah melakukan upload file di halaman plagiarisme kemudian akan lanjut ke halaman pengecekan plagiarisme dimana pada halaman ini akan menampilkan detail dari skripsi seperti judul, jumlah halaman dan *File* skripsi. Kemudian lakukan

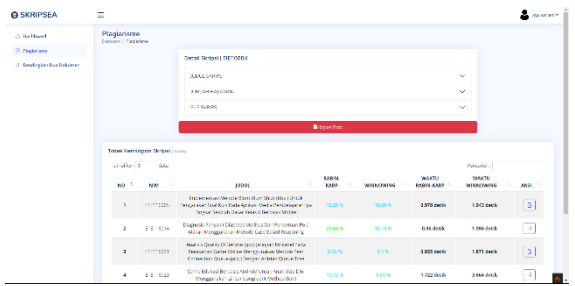


pengecekan plagiarisme dengan menekan tombol mulai pengecekan.



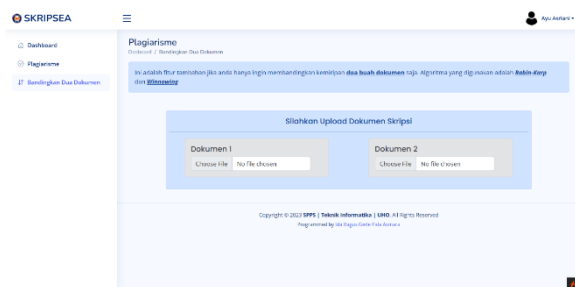
Gambar 15. Tampilan Halaman Pengecekan

Gambar 16. merupakan halaman hasil dari proses cek plagiarisme. Kemudian akan tampil tabel kemiripan skripsi yang sudah diunggah dengan skripsi yang lainnya.



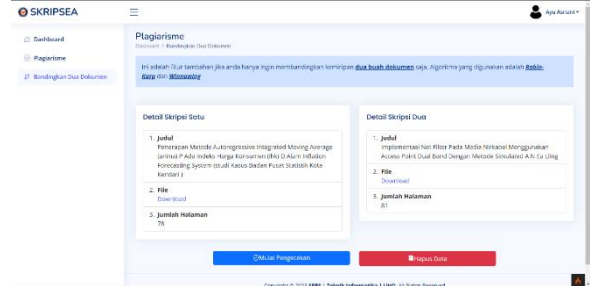
Gambar 16. Tampilan Halaman Hasil Pengecekan

Gambar 17. merupakan halaman apabila mahasiswa hanya ingin membandingkan antara dua dokumen saja untuk dicek kesamaan dokumen.



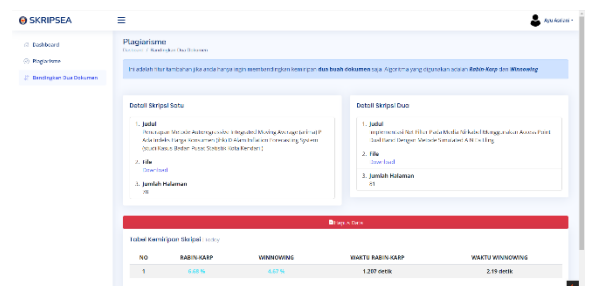
Gambar 17. Tampilan Halaman Bandingkan Dua Dokumen

Gambar 18. merupakan halaman setelah user memasukan atau mengunggah dua dokumen skripsi yang ingin dibandingkan. Kemudian akan tampil halaman detail dari dua dokumen tersebut. Untuk melakukan pengecekan mahasiswa dapat menekan tombol mulai pengecekan.



Gambar 18. Tampilan Halaman Detail Perbandingan Dua Dokumen

Gambar 19. merupakan halaman yang akan menampilkan tabel hasil dari pengecekan plagiarisme dari perbandingan dua dokumen dengan menggunakan dua algoritma yaitu Rabin-karp dan WInnowing dengan menampilkan similarity dan waktu proses dokumen dari kedua algoritma tersebut.



Gambar 19. Tampilan Halaman Hasil Perbandingan Dua Dokumen

#### 4.2. Pengujian Black Box

Berdasarkan pengujian menggunakan metode blackbox testing, ditemukan bahwa semua fitur yang telah direncanakan berfungsi dengan baik sesuai dengan perancangan yang telah dibuat. Adapun hasil pengujian black box dapat dilihat pada Tabel 2.

Tabel 2. Pengujian Black Box

No	Nama Form	Detail Uji	Output	Hasil Uji
1.	Pengujian Login	Menu Login	Menampilkan halaman login user	Sesuai
2.	Pengujian Menu Mahasiswa	Menu tambah mahasiswa	Menampilkan halaman tambah data mahasiswa	Sesuai
		Menu edit mahasiswa	Menampilkan halaman edit data mahasiswa	Sesuai
		Menu hapus mahasiswa	Menampilkan halaman hapus data mahasiswa	Sesuai
3.	Pengujian Menu Skripsi	Menu tambah skripsi	Menampilkan halaman tambah data skripsi	Sesuai

No	Nama Form	Detail Uji	Output	Hasil Uji
4.	Pengujian Menu Kategori Skripsi	Menu edit skripsi	Menampilkan halaman edit data skripsi	Sesuai
		Menu hapus skripsi	Menampilkan halaman hapus data skripsi	Sesuai
		Menu tambah kategori skripsi	Menampilkan halaman tambah kategori skripsi	Sesuai
		Menu edit kategori skripsi	Menampilkan halaman edit kategori data skripsi	Sesuai
5.	Pengujian Menu Cek Kesamaan Skripsi	Menu Upload dokumen skripsi	Menampilkan halaman dokumen skripsi yang telah diupload	Sesuai
		Menu Perbandingan Dua Dokumen Skripsi	Menampilkan halaman dokumen hasil kemiripan dokumen skripsi	Sesuai

#### 4.4. Pengujian Validasi NMAE (Normalized Mean Absolute Error)

Dalam pengujian validasi NMAE ini akan mengukur tingkat kesalahan atau *error* antara prediksi dalam hal ini algoritma *Rabin-Karp* dan algoritma *Winnowing* terhadap nilai sebenarnya dalam hal *platform* pengecekan kesamaan yaitu *Plagiarism Checker X* sebagai aplikasi atau *platform* yang dipercaya dalam konteks yang mempertimbangkan rentang atau skala nilai sebenarnya. Hasil pengujian Validasi NMAE dapat dilihat pada Tabel 3.

Tabel 3. Konversi Nilai *Similarity* ke Bilangan Desimal

No	Dokumen Skripsi 1	Dokumen Skripsi 2	Rabin-Karp	Winnowing	Plagiarism Checker X
1	Skripsi_1	Skripsi_2	0.1336	0.1068	0.11
2	Skripsi_1	Skripsi_3	0.2366	0.1818	0.18
3	Skripsi_1	Skripsi_4	0.0622	0.051	0.06
4	Skripsi_1	Skripsi_5	0.1172	0.0881	0.1
5	Skripsi_1	Skripsi_6	0.0682	0.0506	0.07
6	Skripsi_1	Skripsi_7	0.1415	0.0978	0.1415
7	Skripsi_1	Skripsi_8	0.2511	0.2223	0.2
8	Skripsi_1	Skripsi_9	0.1705	0.1267	0.13
9	Skripsi_1	Skripsi_10	0.1219	0.0769	0.08
10	Skripsi_1	Skripsi_11	0.1673	0.1333	0.14
11	Skripsi_1	Skripsi_12	0.1015	0.0647	0.07
12	Skripsi_1	Skripsi_13	0.1739	0.1026	0.1
13	Skripsi_1	Skripsi_14	0.1629	0.1278	0.12
14	Skripsi_1	Skripsi_15	0.1381	0.0833	0.09

No	Dokumen Skripsi 1	Dokumen Skripsi 2	Rabin-Karp	Winnowing	Plagiarism Checker X
15	Skripsi_1	Skripsi_16	0.0912	0.0686	0.08
16	Skripsi_1	Skripsi_17	0.1232	0.0837	0.09
17	Skripsi_1	Skripsi_18	0.115	0.0736	0.08
18	Skripsi_1	Skripsi_19	0.1204	0.0913	0.08
19	Skripsi_1	Skripsi_20	0.1313	0.0834	0.08
20	Skripsi_1	Skripsi_21	0.1417	0.0888	0.11

Berikut perhitungan NMAE *Rabin-Karp* terhadap *Plagiarism Checker X* :

$$\begin{aligned}
 MAE &= \frac{1}{20} (|0.1336 - 0.11| + |0.2366 - 0.18| \\
 &\quad + |0.0622 - 0.06| + |0.1172 - 0.1| \\
 &\quad + |0.0682 - 0.07| \\
 &\quad + |0.1415 - 0.1415| \\
 &\quad + |0.2511 - 0.2| + |0.1705 - 0.13| \\
 &\quad + |0.1219 - 0.08| + |0.1673 - 0.14| \\
 &\quad + |0.1015 - 0.07| + |0.1739 - 0.1| \\
 &\quad + |0.1629 - 0.12| + |0.1381 - 0.09| \\
 &\quad + |0.0912 - 0.08| + |0.1232 - 0.09| \\
 &\quad + |0.115 - 0.08| + |0.1204 - 0.08| \\
 &\quad + |0.1313 - 0.08| + |0.1417 - 0.11|) \\
 &= \frac{1}{20} (0.6993) \\
 &= 0.034965
 \end{aligned}$$

$$MAX_{(Rabin-Karp, Plagiarism Checker X)} = 0.2511$$

$$MIN_{(Rabin-Karp, Plagiarism Checker X)} = 0.06$$

$$NMAE = \frac{0.034965}{0.2511 - 0.06} = 0.182967033$$

Berikut perhitungan NMAE *Winnowing* terhadap *Plagiarism Checker X* :

$$\begin{aligned}
 MAE &= \frac{1}{20} (|0.1068 - 0.11| + |0.1818 - 0.18| \\
 &\quad + |0.051 - 0.06| + |0.0881 - 0.1| \\
 &\quad + |0.0506 - 0.07| \\
 &\quad + |0.0978 - 0.1415| \\
 &\quad + |0.2223 - 0.2| + |0.1267 - 0.13| \\
 &\quad + |0.0769 - 0.08| + |0.1333 - 0.14| \\
 &\quad + |0.0647 - 0.07| + |0.1026 - 0.1| \\
 &\quad + |0.1278 - 0.12| + |0.0833 - 0.09| \\
 &\quad + |0.0686 - 0.08| + |0.0837 - 0.09| \\
 &\quad + |0.0736 - 0.08| + |0.0913 - 0.08| \\
 &\quad + |0.0834 - 0.08| + |0.0888 - 0.11|) \\
 &= \frac{1}{20} (1 - 0.0669) \\
 &= 0.003345
 \end{aligned}$$

$$MAX_{(Winnowing, Plagiarism Checker X)} = 0.2223$$

$$MIN_{(Winnowing, Plagiarism Checker X)} = 0.0506$$

$$NMAE = \frac{0.003345}{0.2223 - 0.0506} = 0.019481654$$

Jadi berdasarkan perhitungan NMAE di atas diperoleh nilai NMAE (*Rabin-Karp, Plagiarism Checker X*) yaitu 0.182967033 dan nilai NMAE (*Winnowing, Plagiarism Checker X*) yaitu 0.019481654, maka algoritma *Winnowing* memiliki kinerja yang lebih baik daripada algoritma *Rabin-Karp*.

#### 4.4. Pengujian Space Complexity

*Space Complexity* adalah salah satu aspek penting dalam analisis kompleksitas algoritma yang mengukur berapa banyak memori (ruang) yang dibutuhkan oleh suatu algoritma selama proses. Pemahaman *Space Complexity* penting karena penggunaan memori yang berlebihan dapat mengakibatkan kinerja yang buruk dan bahkan kegagalan eksekusi program pada sistem dengan keterbatasan sumber daya. Hasil pengujian *space complexity* dapat dilihat pada Tabel 4.

Tabel 4. Pengujian *Space Complexity*

No	Dokumen Skripsi 1	Dokumen Skripsi 2	<i>Rabin-Karp</i>	<i>Winnowing</i>
1	Skripsi_1	Skripsi_2	26.88 Mb	94.16 Mb
2	Skripsi_1	Skripsi_3	26.85 Mb	94.91 Mb
3	Skripsi_1	Skripsi_4	26.81 Mb	93.46 Mb
4	Skripsi_1	Skripsi_5	33 Mb	123.26 Mb
5	Skripsi_1	Skripsi_6	28.22 Mb	106.01 Mb
6	Skripsi_1	Skripsi_7	26.64 Mb	91.25 Mb
7	Skripsi_1	Skripsi_8	27.12 Mb	90.28 Mb
8	Skripsi_1	Skripsi_9	27.95 Mb	102 Mb
9	Skripsi_1	Skripsi_10	27.68 Mb	98.07 Mb
10	Skripsi_1	Skripsi_11	28 Mb	102.88 Mb
11	Skripsi_1	Skripsi_12	27.29 Mb	99.61 Mb
12	Skripsi_1	Skripsi_13	28.36 Mb	108.61 Mb
13	Skripsi_1	Skripsi_14	26.76 Mb	93.61 Mb
14	Skripsi_1	Skripsi_15	32.92 Mb	121.72 Mb
15	Skripsi_1	Skripsi_16	28.4 Mb	108.51 Mb
16	Skripsi_1	Skripsi_17	28.14 Mb	103.66 Mb
17	Skripsi_1	Skripsi_18	32.65 Mb	118.13 Mb
18	Skripsi_1	Skripsi_19	26.61 Mb	91.08 Mb
19	Skripsi_1	Skripsi_20	27.06 Mb	95.95 Mb
20	Skripsi_1	Skripsi_21	33.39 Mb	125.86 Mb

Berdasarkan analisis Tabel 4. *Space Complexity* di atas, maka algoritma *Rabin-Karp* memiliki rata-rata *Space Complexity* sebesar 28.05 Mb dan algoritma *Winnowing* memiliki rata-rata *Space Complexity* sebesar 97.85 Mb. Jika dilihat besar rata-rata nilai *Space Complexity* dari masing-masing algoritma maka, algoritma *Winnowing* memiliki rata-rata *Space Complexity* yang besar dibandingkan algoritma *Rabin-Karp*.

#### 4.5. Pengujian Response Time

Pengujian *response time* dilakukan untuk mengukur waktu eksekusi algoritma *rabin-karp* dan *winnowing* dalam pengecekan kesamaan dokumen skripsi. Pengujian ini melibatkan 1 dokumen skripsi dengan 20 dokumen skripsi yang tersimpan pada *database* sistem. Adapun hasil pengujian dapat dilihat pada Tabel 5.

Tabel 5. Pengujian *Response Time*

No	Dokumen Skripsi 1	Dokumen Skripsi 2	Similarity (%)		Waktu Proses (Detik)	
			Rabin-Karb	Winnowing	Rabin-Karb	Winnowing
1	Skripsi_1	Skripsi_2	13.36	10,68	0,951	1,902
2	Skripsi_1	Skripsi_3	23.66	18.18	0,943	1,937
3	Skripsi_1	Skripsi_4	6.22	5.10	0,912	1,83
4	Skripsi_1	Skripsi_5	11.72	8.81	1,711	3,632
5	Skripsi_1	Skripsi_6	6.82	5.06	1,521	2,904
6	Skripsi_1	Skripsi_7	14.15	9.78	1,05	1,953
7	Skripsi_1	Skripsi_8	25.11	22.23	1,104	1,936
8	Skripsi_1	Skripsi_9	17.05	12.67	1,183	2,41
9	Skripsi_1	Skripsi_10	12.19	7.69	1,171	2,24
10	Skripsi_1	Skripsi_11	16.73	13.33	1,543	2,818
11	Skripsi_1	Skripsi_12	10.15	6.47	1,398	2,613
12	Skripsi_1	Skripsi_13	17.39	10.26	1,538	3,166
13	Skripsi_1	Skripsi_14	16.29	12.78	0,971	1,92
14	Skripsi_1	Skripsi_15	13.81	8.33	1,672	3,498
15	Skripsi_1	Skripsi_16	9.12	6.86	1,249	2,737
16	Skripsi_1	Skripsi_17	12.32	8.37	1,114	2,4
17	Skripsi_1	Skripsi_18	11.50	7.36	1,717	3,39
18	Skripsi_1	Skripsi_19	12.04	9.13	1,263	2,145
19	Skripsi_1	Skripsi_20	13.13	8.34	1,141	2,188
20	Skripsi_1	Skripsi_21	14.17	8.88	2,104	4,178

Berdasarkan Tabel 5. Pengujian *Response Time* atau waktu eksekusi dengan 20 dokumen algoritma *rabin-karp* rata-rata nya 1.072 detik, sedangkan algoritma *winnowing* 1.881 detik waktu eksekusinya. Atrinya dalam perbandingan *response time* algoritma *rabin-karp* lebih unggul dibandingkan dengan *winnowing*.

### 5. Kesimpulan

Hasil penelitian ini menyimpulkan bahwa perbandingan algoritma *Rabin-Karp* dan algoritma *Winnowing*, hasil evaluasi uji coba sistem yang telah dibuat dapat diketahui bahwa performa hasil persentase dari implementasi algoritma *Rabin-Karp* dan *Winnowing* dalam pengecekan kesamaan skripsi mahasiswa dapat berjalan dengan baik. Algoritma *Rabin-Karp* memiliki tingkat *similarity* yang tinggi daripada algoritma *Winnowing*. Namun, algoritma *Winnowing* memiliki tingkat konsistensi atau rata-rata *similarity* yang sama dengan aplikasi *Plagiarism*

Checker X. Berdasarkan *response time* algoritma Rabin-Karp memiliki waktu proses yang cenderung lebih cepat dengan rata-rata 1,072 detik dibandingkan dengan algoritma *Winnowing* dengan rata-rata 1.881 detik. Berdasarkan nilai rata-rata *Space Complexity*, algoritma *Winnowing* memiliki nilai rata-rata *Space Complexity* yang lebih besar dengan kapasitas 97,85 Mb dibandingkan algoritma Rabin-Karp dengan rata-rata 28.05 Mb.

#### Daftar Pustaka

- Alamsyah, N., Rasyidan, M., 2019. Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi pada Fakultas Teknologi Informasi Menggunakan Algoritma *Winnowing*. *Technologia*, 10(4), 197-201. <http://dx.doi.org/10.31602/tji.v10i4.2361>
- Aldian, Mubarak, 2019. Implementasi Algoritma Rabin-Karp untuk Pendeteksian Plagiarisme pada File Dokumen Berupa Text Berbasis Web. *Journal of Information Sistem Research*, 3(3), 150-154. <https://doi.org/10.47065/josh.v3i3.1404>
- Apridiansyah, Y., Wijaya, A., Purjiawan, A., 2022. Penerapan Fungsi Metode Rolling Hash pada Algoritma *Winnowing* untuk Mendeteksi Kemiripan Teks Abstrak Berbasis Web. *Jurnal Media Infotama*. 18(1), 128-133. <https://doi.org/10.37676/jmi.v18i1.1695>
- Ati, S.H., Saptono, R., Salamah, U., 2012. Peningkatan Efektivitas Metode User-item based Collaborative. *Jurnal ITSMART*. 1(1), 1-4. <https://doi.org/10.20961/itsmart.v1i1.574>
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2019. Introduction to Algorithms. *MIT Press*.
- Faisal, M., Nugroho, F., Sulthan, M.M.E., Amini, F., Hariyadi, M.A., Sedayu, A., 2020. Plagiarism Detection Using Manber and *Winnowing* Algorithm. *International Journal of Advanced Science And Technology*, 29(6s), 2130-2136.
- Kruchten, P., 2003. The Rational Unified Process: An Introduction. *Addison-Wesley*.
- Mangundap, G.H.O., Sujaini, H., Pratiwi, H.S., 2022. Implementasi Algoritma *Winnowing* pada Aplikasi Pendeteksi Kemiripan Dokumen. *Jurnal Edukasi dan Penelitian Informatika (Jepin)*, 8(1), 147-153. <http://dx.doi.org/10.26418/jp.v8i1.47822>
- Parewe, A.M.A.K., Aman, A., Dewang, D.P.M., 2021. Perbandingan Algoritma *Winnowing* Dan Algoritma Manber Dalam Mendeteksi Berita Hoax Di Media Sosia. *Semantik*, 3(1), 41-46.
- Purba, A.H., Situmorang, Z., 2017. Analisis Perbandingan Algoritma Rabin-Karp dan Levenshtein Distance dalam Menghitung Kemiripan Teks. *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, 2(2), 24-32. <https://doi.org/10.17605/jti.v2i2.187>
- Ramadhani, S., 2015. Sistem Pencegahan Plagiarism Tugas Akhir Menggunakan Algoritma Rabin-Karp (Studi Kasus: Sekolah Tinggi Teknik Payakumbuh). *Jurnal Teknologi Informasi dan Komunikasi Digital Zone*, 6(1), 44-52. <https://doi.org/10.31849/digitalzone.v6i1.88>
- Schleimer, S., Wilkerson, D.S., Aiken, A., 2018. *Winnowing: Local Algorithms for Document Fingerprinting*. *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 76-85. <https://doi.org/10.1145/872757.872770>
- Sugiono, Herwin, Hamdani, Erlin, 2018. Aplikasi Pendeteksi Tingkat Kesamaan Dokumen Teks: Algoritma Rabin Karp Vs. *Winnowing*. *Jurnal Teknologi & Komunikasi Digital Zone*. 9(1), 82-93. <https://doi.org/10.31849/digitalzone.v9i1.1242>
- Sunardi, Yudhana, A., Mukaromah, I.A., 2018. Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram dan Jaccard Similarity Terhadap Algoritma *Winnowing*. *Transmisi*, 20(3), 105-110. <https://doi.org/10.14710/transmisi.20.3.105-110>
- Suryati, T., Wibisono, Y., Wihardi, Y., 2018. Aplikasi Deteksi Plagiarisme Dokumen Skripsi dengan Algoritma Rabin-Karp. *JATIKOM*, 1 (2), 91-95. <https://doi.org/10.17509/jatikom.v1i2.25190>
- Sutedi, Agarina, M., 2017. Implementasi Rational Unified Process dalam Rancang Bangun Sistem Informasi Penjualan Hasil Bumi Berbasis Web pada CV. Aneka Mandiri Lestari Bandar Lampung. *Explore: Jurnal Sistem Informasi dan Telematika*, 8(2), 181-187. <https://doi.org/10.36448/jsit.v8i2.958>
- Tantoni, A., Zaen, M.T.A., 2018. Implementasi Double Caesar Cipher Menggunakan ASCII. *Jurnal Informatika & Rekayasa Elektronika (Jire)*, 1(2), 24-32.
- Yani, A., Setiawan, D., Sofian, N.E., Subagja, R., Desyani, T., 2020. Pengujian Aplikasi Reservasi Hotel di Legreen Hotel & Suite dengan Metode Black Box Testing Boundary Value Analysis. *Jurnal Teknologi Sistem Informasi dan Aplikasi*. 3(2), 114-118. [10.32493/jtsi.v3i2.4686](https://doi.org/10.32493/jtsi.v3i2.4686)
- Young, S., Nuryasin, I., Suharso, W., 2018. Penerapan Metode SDLC RUP (Rational Unified Proses) dalam Pembuatan Sistem Informasi E-kos Berbasis Web. *Jurnal Repositor*, 2(3). <https://doi.org/10.22219/repositor.v2i3.30496>
- Zaminkar M., Reshadinezhad, M.R., 2013. A Comparison Between Two Software Engineering Processes, RUP And Waterfall Models. *IJERT*, 2(7), 1348-1352. [10.17577/IJERTV2IS70559](https://doi.org/10.17577/IJERTV2IS70559)
- Zulfikar, A.F., 2017. Pengembangan Algoritma Stemming Bahasa Indonesia dengan Pendekatan Dictionary Base Stemming untuk Menentukan Kata Dasar dari Kata yang Berimbuhan. *Jurnal Informatika Universitas Pamulang*, 2(3), 143-146. <https://doi.org/10.32493/informatika.v2i3.1443>