

Penerapan Algoritma *Winnowing* untuk Mendeteksi Kemiripan Judul Skripsi

Dimas Bagus Prasetyo¹, Dina Maulina^{2 *}

¹ Universitas Amikom Yogyakarta; dimas.prasetyo@student.amikom.ac.id

^{*2} Universitas Amikom Yogyakarta; dina.m@amikom.ac.id

Abstrak: Syarat lulus dari perguruan tinggi salah satunya adalah harus menyelesaikan skripsi, dimana skripsi menjadi sarana untuk menuangkan karya tulis ilmiah dari hasil penelitian dengan pembahasan mengenai ilmu tertentu berdasarkan kajian pustaka oleh para ahli, hasil penelitian lapangan atau hasil pengembangan eksperimen. Setiap tahunnya Universitas AMIKOM Yogyakarta meluluskan mahasiswa yang membuat banyak skripsi yang telah dibuat yang memungkinkan judul yang diajukan dengan judul yang terdahulu memiliki kemiripan. Untuk mengatasi hal tersebut dibutuhkan suatu metode untuk mendeteksi kemiripan judul skripsi yaitu dengan algoritma *winnowing*. Algoritma *winnowing* digunakan sebagai pencarian dokumen *fingerprinting* yaitu metode untuk mendeteksi keakuratan salinan dari kalimat atau dokumen. Secara garis besar penerapan algoritma *winnowing* untuk mendeteksi kemiripan teks adalah membentuk *n-gram*, pembentukan *hashing*, pembentukan *window*, pencarian *fingerprinting*, dan mencari persentase dengan *jaccard coefficient*. Tujuan penelitian ini adalah untuk mengetahui proses algoritma *winnowing* dalam mendeteksi kemiripan judul skripsi dan untuk menghasilkan sebuah *prototype* sistem untuk mengecek kemiripan judul skripsi. Hasil akhir dari penerapan algoritma *winnowing* ini adalah suatu *prototype* sistem sederhana yang memiliki fitur untuk mendeteksi kemiripan judul skripsi yang dapat digunakan oleh dosen pembimbing untuk dapat mengecek judul yang telah diajukan mahasiswa. Jika judul yang diajukan banyak kemiripan dengan judul sebelumnya maka dosen pembimbing dapat merekomendasikan judul lain namun jika judul yang diajukan tidak banyak kemiripan maka dapat disetujui lalu dapat dimasukkan ke *database* judul.

Keywords: Algoritma *Winnowing*, Skripsi, *Fingerprint*, *Hashing*, *Window*

DOI: <https://doi.org/10.47134/jacis>

*Correspondensi: Dina Maulina

Email: dina.m@amikom.ac.id

Receive: 25 November 2024

Accepted: 27 November 20204

Published: 30 November 2024



Copyright: © 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Abstrak: In universities, there is the term thesis, which is a place to pour scientific papers from research results with a discussion of certain knowledge based on literature review by experts, field research results or experimental development results. Every year AMIKOM Yogyakarta University graduates students who make many theses that have been made which allows the title submitted with the previous title to have similarities. To overcome this, a method is needed to detect the similarity of thesis titles, namely the *winnowing* algorithm. The *winnowing* algorithm is used as a *fingerprinting* document search, which is a method to detect the accuracy of copies of sentences or documents. Broadly speaking, the application of *winnowing* algorithm to detect text similarity is forming *n-grams*, *hashing* formation, *window* formation, *fingerprinting* search, and finding percentage with *jaccard coefficient*. The purpose of this research is to know the process of *winnowing* algorithm in detecting the similarity of thesis title and to produce a *prototype* system to check the similarity of thesis title. The final result of the application of the *winnowing* algorithm is a simple *prototype* system that has a feature to detect the similarity of thesis titles that can be used by supervisors to be able to check the titles that have been submitted by students. If the proposed title has

many similarities with the previous title, the supervisor can recommend another title, but if the proposed title does not have many similarities, it can be approved and then can be entered into the title database.

Keywords: *Winnowing Algorithm, Thesis, Fingerprint, Hashing, Window*

PENDAHULUAN

Dalam dunia pendidikan tinggi tidak asing dengan istilah skripsi. Skripsi merupakan suatu istilah yang digunakan di Indonesia dalam rangka menuangkan suatu karya tulis secara ilmiah yang berupa paparan dari hasil penelitian dengan pembahasan suatu permasalahan dalam bidang ilmu tertentu berdasarkan kajian pustaka yang diteliti oleh para ahli, hasil dari penelitian di lapangan, dan atau hasil pengembangan suatu eksperimen[1]. Dengan berjalannya waktu semakin banyak judul skripsi yang diajukan oleh mahasiswa dan tidak menutup kemungkinan bahwa judul yang diajukan tersebut memiliki kemiripan atau bahkan sama dengan judul skripsi yang telah diajukan sebelumnya yang membuat persentase kesamaannya tinggi. Judul yang sama atau mirip tidak menutup kemungkinan isi atau topik dari skripsi tersebut hampir sama dan membuat tidak beragamnya judul yang dibuat. Untuk itu dibutuhkan suatu sistem yang dapat digunakan untuk mendeteksi kemiripan judul skripsi dengan menerapkan suatu algoritma yang dapat menentukan kemiripan teks, dan salah satu algoritma tersebut adalah algoritma *Winnowing*[2].

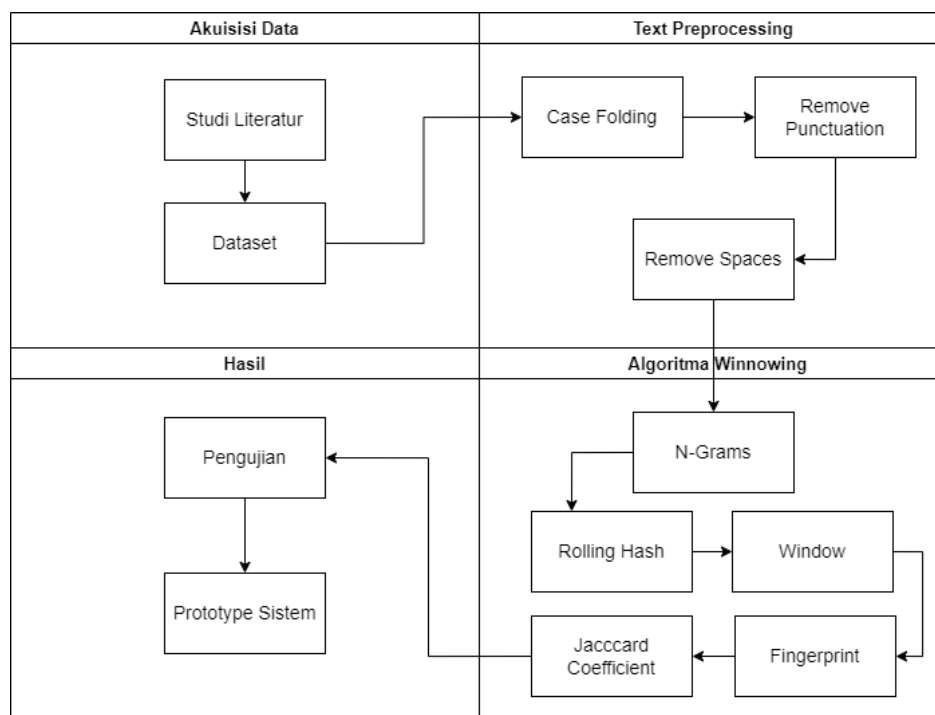
Algoritma *Winnowing* adalah salah satu dari banyak algoritma yang dapat digunakan untuk mendeteksi kemiripan kata atau kalimat dari kedua kalimat atau lebih yang dibandingkan. Algoritma *winnowing* digunakan sebagai pencarian dokumen *fingerprinting*. Dokumen *fingerprinting* adalah metode untuk mendeteksi keakuratan salinan dari sebuah kalimat ataupun dokumen. Teknik dalam *fingerprinting* dikenal dengan teknik *hashing* yang berfungsi untuk mengubah setiap *string* menjadi bilangan. *Fingerprint* inilah yang akan dijadikan dasar pembandingan antara teks[3]. Algoritma *winnowing* dapat digunakan untuk mendeteksi kemiripan teks karena sudah memenuhi syarat *whitespace intensitivity*[4] yakni dengan membuang karakter tidak relevan pada teks, seperti spasi, tanda baca, dll sehingga nantinya hanya tersisa huruf atau angka yang akan diproses lebih lanjut. Algoritma *winnowing* memiliki tingkat kemiripan yang efisien[5]. Jika dibandingkan dengan algoritma *fingerprint* lainnya, algoritma *Winnowing* mempunyai keunggulan dalam membandingkan suatu teks atau dokumen yaitu adanya penambahan konsep *window* yang dari *window* tersebut nilai *hash* yang dipilih menjadi nilai *fingerprint minimum*[6]. Oleh karena itu, penulis menjadikan algoritma *winnowing* tersebut untuk mengecek kemiripan judul skripsi yang nantinya digunakan untuk mengetahui judul skripsi yang telah sering digunakan dan diharapkan dengan adanya fitur tersebut dapat membantu dosen pembimbing dalam menentukan atau mempertimbangkan judul skripsi yang diajukan mahasiswa.

Pada tahun 2019, algoritma *Winnowing* digunakan untuk mendeteksi kemiripan teks skripsi dengan cara membandingkan kedua teks melalui teknik *fingerprinting*. Proses ini melibatkan pengubahan *n-gram* huruf menjadi angka yang disebut *hash*[2]. Dengan algoritma dan tujuan yang sama, *Winnowing* mampu menghasilkan tingkat kemiripan sebesar 34,78% menggunakan parameter *n-gram* = 5 dan *window* = 7[3]. Namun, ketika parameter diubah

menjadi $n\text{-gram} = 2$ dan $window = 2$, tingkat kemiripan meningkat menjadi 51,50%. Sebaliknya, dengan $n\text{-gram} = 5$ dan $window = 5$, tingkat kemiripan menurun drastis menjadi 5,26%[7]. Dalam aplikasi lain, algoritma Winnowing diterapkan untuk penilaian otomatis pada ujian daring, khususnya soal esai. Dengan $k\text{-gram} = 4$ dan $window = 4$, algoritma ini mampu menghasilkan tingkat kemiripan sebesar 69,23%[6]. Penelitian lebih lanjut menunjukkan bahwa nilai $k\text{-gram}$ yang lebih tinggi cenderung menurunkan tingkat kemiripan. Rata-rata tingkat kemiripan tercatat sebagai berikut: $k = 2$ sebesar 0,5299; $k = 3$ sebesar 0,1689; $k = 5$ sebesar 0,0283; $k = 7$ sebesar 0,0095. Selain itu, pemotongan string berdasarkan kata menggunakan $n\text{-gram}$ menunjukkan hasil sebagai berikut: penerapan *unigram* memiliki rata-rata tingkat kemiripan sebesar 0,0683; *bigram* sebesar 0,003; dan *trigram* serta *four-gram* tidak menunjukkan kemiripan (*similarity* = 0,000)[8].

METODE

Penelitian ini dilakukan dengan menerapkan beberapa langkah, secara garis besar alur langkah penelitian terdapat pada Gambar 1.



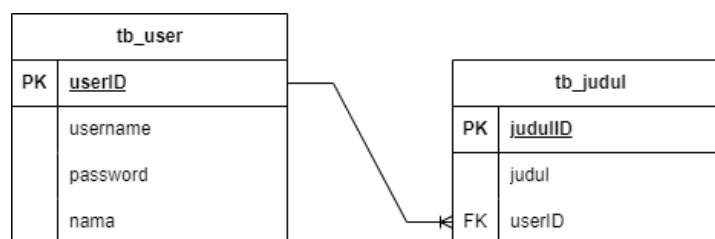
Gambar 1. Alur Penelitian

Langkah pertama adalah melakukan studi literatur mengenai penelitian yang akan dilakukan dengan sumber dari jurnal, skripsi, dan internet. Kemudian menyiapkan dataset berupa judul skripsi dari mahasiswa Fakultas Ilmu Komputer Universitas Amikom Yogyakarta dengan jumlah sekitar 900 data. Dataset tersebut kemudian diolah melalui *text preprocessing* dengan proses berupa *case folding*, *remove punctuation* dan *remove spaces*. Setelah melalui proses *preprocessing data*, maka masuk ke proses inti yaitu penerapan Algoritma Winnowing. Pertama menentukan nilai *N-Grams* terlebih dahulu kemudian dilanjutkan dengan membentuk nilai *hash* dari *n-gram* tersebut, yaitu suatu cara untuk mentransformasi sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu. Kemudian dilakukan pembentukan nilai *window* dari nilai *hash* tersebut sesuai dengan kebutuhan. Dilanjutkan dengan menentukan

fingerprint yaitu mencari nilai terkecil dari nilai *hash* setiap *window*. Langkah terakhir adalah proses penghitungan persentase menggunakan *Jaccard Coefissient* yakni menghitung persentase dari kemiripan judul skripsi. Data yang digunakan adalah data dari *resource center* Universitas AMIKOM Yogyakarta berupa data judul skripsi mahasiswa Fakultas Ilmu Komputer yang telah lulus dengan jumlah sekitar 900 judul yang tersimpan dalam *microsoft excel*. Data tersebut diinputkan ke dalam sistem pendeteksi kemiripan judul skripsi.

Desain Penelitian

Dalam membangun sistem ini dibuat *Class diagram* untuk mendeklarasikan struktur sistem dari pendefinisian kelas-kelas yang akan dibuat. Class adalah kumpulan dari objek yang memiliki struktur umum, *behavior* umum, relasi umum, dan kata yang umum[9]. Gambar 2 merupakan *class diagram* dari sistem yang akan dibuat.



Gambar 2. Class Diagram

Pada *class diagram* di atas, terdapat dua entitas yaitu user dan judul. Dalam entitas *user* terdapat *ID* sebagai *primary key*, *username*, *password* dan *nama*. Kemudian dalam entitas *judul* terdapat atribut *ID* sebagai *primary key* dan daftar judul. Kedua entitas tersebut memiliki relasi yaitu *one to many*. Tahap *preprocessing* adalah suatu hal yang penting dalam proses data mining maupun dalam penerapan algoritma. Konsep *preprocessing* adalah dengan mengurangi atribut yang tidak berguna atau tidak cukup berpengaruh dalam proses perhitungan algoritma.[10] Tahap *text preprocessing* pada penelitian ini meliputi *case folding* yaitu mengubah huruf kapital menjadi huruf kecil, *remove punctuation* yaitu menghilangkan tanda baca seperti (.), (:), ('), dll. Yang terakhir adalah *remove spaces* yaitu menghilangkan spasi antar kata.

HASIL DAN PEMBAHASAN

Penerapan Algoritma Winnowing

Algoritma *winnowing* adalah algoritma yang dapat digunakan untuk mengecek atau mendeteksi kemiripan atau kesamaan kata dengan fungsi *hashing* dan menghasilkan *fingerprinting* [11] Tahap dari perhitungan menggunakan algoritma *winnowing* adalah sebagai berikut:

1. Mengambil 2 judul untuk dibandingkan

Teks judul 1:

"Sistem Pakar Pendeteksi Kerusakan Pada Laptop Menggunakan Metode Naive Bayes

Teks judul 2:

"Sistem Pakar Pendeteksi Penyakit Tanaman dan Hama Buah Naga Menggunakan Naive Bayes"

2. Text Preprocessing

Dari 2 judul yang telah dipilih tersebut, maka akan dilakukan *text preprocessing* menggunakan *case folding*, *remove punctuation* dan *remove space* sehingga terbentuk seperti tabel 1.

Tabel 1. Text Preprocessing

Text preprocessing judul 1	Text preprocessing judul 2
sistempakarpendeteksikerusakanpadalaptopmen gunakanmetodenaiivebayes	sistempakarpendeteksipenyakitnamandanhamabua hngamenggunakannaivebayes

3. Pembentukan N-Gram

N-gram adalah sebuah metode yang yang dilakukan dengan mengambil suatu rangkaian *substring* dari *string* yang berjumlah *n* (rangkaian token dengan panjang *n*)[12] Pada penelitian ini, peneliti menggunakan *n-gram* = 3. Dari 2 judul tersebut terbentuk *n-gram* dengan nilai 3 seperti tabel 2.

Tabel 2. Pembentukan N-Gram

N-gram judul 1	N-gram judul 2
sis ist ste tem emp mpa pak aka kar arp rpe pen end nde det ete tek eks ksi sik ike ker eru rus usa sak aka kan anp npa pad ada dal ala lap apt pto top opm pme men eng ngg ggu gun una nak aka kan anm nme met eto tod ode den ena nai aiv ive veb eba bay aye yes	sis ist ste tem emp mpa pak aka kar arp rpe pen end nde det ete tek eks ksi sip ipe pen eny nya yak aki kit itt tta tan ana nam ama man and nda dan anh nha ham ama mab abu bua uah ahn hna nag aga gam ame men eng ngg ggu gun una nak aka kan ann nna nai aiv ive veb eba bay aye yes

4. Perhitungan Fungsi Hash

Hashing adalah metode untuk mengkonversi sebuah *string* menjadi suatu nilai unik dengan penentuan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut. Salah satu *hash* yang sering digunakan adalah *rolling hash*. Persamaan (1) adalah algoritma *rolling hash*[6]:

$$H(ck) = C1 * b^{(K-1)} + C1 * b^{(K-2)} + \dots + C1 * b^{(K-K)} \quad (1)$$

Keterangan:

- C1 = nilai ASCII dari setiap huruf
- b = basis bilangan prima
- k = jumlah N-Grams

Berikut merupakan penerapan rumus *rolling hash* untuk menghitung *n-gram* dari 2 judul tersebut

$$\begin{aligned}
 H(\text{apl}) &= \text{ASCII}(\text{s}) * 2^{(3-1)} + \text{ASCII}(\text{i}) * 2^{(2-1)} + \text{ASCII}(\text{s}) * 2^{(1-1)} \\
 &= 115 * 2^{(2)} + 105 * 2^{(1)} + 115 * 2^{(0)} \\
 &= 115 * 4 + 105 * 2 + 115 * 1 \\
 &= 460 + 210 + 115 \\
 &= 785 \text{ (dst.)}
 \end{aligned}$$

Jadi, berdasarkan perhitungan fungsi *hash* dari huruf “apl” menghasilkan nilai *hash* yaitu 785. Perhitungan tersebut dilakukan pada seluruh rangkaian *n-gram* sehingga menghasilkan nilai seperti tabel 3.

Tabel 3. Pembentukan N-Gram

N-gram judul 1	N-gram judul 2
785 766 793 775 734 757 749 699 736 728	785 766 793 775 734 757 749 699 736
781 760 724 741 718 737 773 733 763 777	728 781 760 724 741 718 737 773 733
735 744 749 805 795 761 699 732 720 761	763 782 745 760 745 779 785 707 754
742 685 702 701 738 728 791 798 777 767	768 793 768 705 743 703 740 708 737
748 727 749 735 756 785 741 699 732 717	704 712 745 719 703 728 701 723 766
759 754 747 786 745 712 721 739 716 757	706 733 737 691 715 707 748 727 749
772 697 707 731 801	737 756 785 741 699 732 718 757 739
	716 757 772 697 707 731 801

5. Pembentukan Window

Setelah dilakukan perhitungan fungsi *hash* pada setiap *n-gram*, maka selanjutnya adalah pembentukan *window*. *Window* merupakan pembagian atau pengelompokan hasil nilai *hash* ke dalam beberapa *window* dengan ukuran sebesar *w*. Setelah *window* terbentuk maka dapat dilakukan proses pemilihan *fingerprint* dengan ukuran terkecil dengan baik[5]. Proses pembentukan *window* sama dengan pembentukan *n-gram*, pada penelitian ini, peneliti menggunakan *window* = 3.

Tabel 4. Pembentukan Window

Window judul 1	Window judul 2
W1 = [785 766 793]	W1 = [785 766 793]
W2 = [766 793 775]	W2 = [766 793 775]
W3 = [793 775 734]	W3 = [793 775 734]
W4 = [775 734 757]	W4 = [775 734 757]
W5 = [734 757 749]	W5 = [734 757 749]
W6 = [757 749 699]	W6 = [757 749 699]
W7 = [749 699 736]	W7 = [749 699 736]
W8 = [699 736 728]	W8 = [699 736 728]
W9 = [736 728 781]	W9 = [736 728 781]
W10 = [728 781 760]	W10 = [728 781 760]
.....	
W51 = [759 754 747]	W51 = [707 748 727]
W52 = [754 747 786]	W52 = [748 727 749]
W53 = [747 786 745]	W53 = [727 749 737]
W54 = [786 745 712]	W54 = [749 737 756]
W55 = [745 712 721]	W55 = [737 756 785]
W56 = [712 721 739]	W56 = [756 785 741]
W57 = [721 739 716]	W57 = [785 741 699]
W58 = [739 716 757]	W58 = [741 699 732]
W59 = [716 757 772]	W59 = [699 732 718]
W60 = [757 772 697]	W60 = [732 718 757]
W61 = [772 697 707]	W61 = [718 757 739]
W62 = [697 707 731]	W62 = [757 739 716]
W63 = [707 731 801]	W63 = [739 716 757]
	W64 = [716 757 772]
	W65 = [757 772 697]
	W66 = [772 697 707]
	W67 = [697 707 731]
	W68 = [707 731 801]

6. Pencarian *Fingerprint*

Dokumen *fingerprint* merupakan metode yang digunakan untuk mendeteksi dari salinan antar teks atau antar dokumen baik dari seluruh teks dokumen ataupun sebagian dari teks dokumen[13]. Tabel 5 merupakan hasil pencarian *fingerprint* dari setiap *window*.

Jumlah <i>fingerprint</i> judul 1	= 63
Jumlah <i>fingerprint</i> judul 2	= 68
Gabungan <i>fingerprint</i> judul 1 dan 2	= 131
Jumlah <i>fingerprint</i> sama	= 49

Tabel 5. Hasil *Fingerprint*

Fingerprint judul 1	Fingerprint judul 2
766 766 734 734 734 699 699 699 728	766 766 734 734 734 699 699 699 728
728 724 724 718 718 718 733 733 733	728 724 724 718 718 718 733 733 733
735 735 735 744 749 761 699 699 699	745 745 745 745 745 707 707 707 754
720 720 685 685 685 701 701 728 728	768 768 768 703 703 703 708 704 704
777 767 748 727 727 727 735 735 741	704 712 703 703 701 701 701 706 706
699 699 699 717 717 747 747 745 712	706 691 691 691 707 707 727 727 737
712 712 716 716 716 697 697 697 707	737 741 699 699 699 718 718 716 716
	716 697 697 697 707

7. Perhitungan *Jaccard Coefficient*

Jaccard Coefficient adalah algoritma yang digunakan untuk membandingkan dua *sample* seperti dokumen berdasarkan kata-kata yang terkandung didalamnya. Algoritma ini berfungsi untuk mengukur tingkat kemiripan (*similarity*) antarai dua objek atau dokumen dengan membandingkan elemen-elemen yang dimiliki keduanya[12]. Persamaan 2 adalah rumus dari *jaccard coefficient*.

$$Similarity = \frac{Intersection}{Union - Intersection} \times 100\% \quad (2)$$

Keterangan:

Intersection = *fingerprint* yang sama
 Union = gabungan *fingerprint*

Setelah didapat hasil *fingerprint* dari setiap *window* maka dihitung persentase menggunakan *jaccard coefficient* dengan menerapkan rumus tersebut sehingga menghasilkan:

$$Similarity = \frac{49}{131 - 49} \times 100\% = 59,76\%$$

Jadi hasil persentase dari kemiripan 2 judul yang dibandingkan adalah 59,76%.

Pengujian

Pengujian merupakan langkah yang penting dilakukan dalam proses pengembangan sistem untuk mengevaluasi, menganalisa dan mengetahui tingkat akurasi serta kesesuaian hasil yang telah dicapai oleh sistem yang telah dirancang. Hasil dari pengujian yakni tingkat persentase kemiripan, untuk menentukan apakah persentase tersebut termasuk plagiasi maka dibutuhkan suatu ambang batas seperti Tabel 6 [2].

Tabel 6. Tingkat Plagiarisme

Tingkat Plagiarisme	
Plagiarisme Ringan	<= 35%
Plagiarisme Sedang	36% - 70%
Plagiarisme Tinggi	> 70%

Berikut merupakan pengujian dengan kombinasi *n-gram* dan *window* dengan menggunakan bilangan prima.

a. Pengujian dengan *n-gram* dan *window* yang sama

Tabel 7. Hasil Pengujian dengan N-Gram dan Window sama

No.	N-Gram	Window	Nilai basis	Persentase %
1.	2	2	2	70.89%
2.	3	3	2	59.76%
3.	5	5	2	24.24%
4.	7	7	2	21.05%

Tabel 7 menunjukkan hasil pengujian menggunakan nilai *N-Gram* dan *Window* yang sama. Ada empat pengujian dengan parameter berbeda, yaitu kombinasi nilai *N-Gram* dan *Window*. Kolom "Nilai Basis" tetap konsisten pada angka 2, sedangkan kolom "Persentase %" menunjukkan tingkat kemiripan yang dihasilkan dari setiap kombinasi. Berdasarkan tabel 7 diketahui bahwa saat nilai *N-Gram* dan *Window* rendah yaitu 2, memiliki tingkat kemiripan tertinggi tercatat sebesar 70,89%. Namun dengan meningkatnya nilai *N-Gram* dan *Window* secara bersamaan, maka tingkat kemiripanpun menurun dimana ketika *N-Gram* = 3, *Window* = 3 menghasilkan 59,76%; ketika *N-Gram* = 5, *Window* = 5 menghasilkan 24,24%; dan ketika *N-Gram* = 7, *Window* = 7 menghasilkan 21,05%

b. Pengujian dengan *n-gram* dan *window* yang berbeda

Tabel 8. Hasil Pengujian dengan n-gram dan window berbeda

No.	N-Gram	Window	Nilai basis	Persentase %
1.	2	5	2	72.00%
2.	2	7	2	71.23%
3.	3	2	2	58.33%
4.	3	5	2	47.67%
5.	3	7	2	41.38%
6.	5	2	2	35.79%
7.	5	3	2	29.59%
8.	5	7	2	17.82%

Pada tabel 8 menunjukkan hasil pengujian menggunakan nilai *N-Gram* dan *Window* yang berbeda. Seperti pada tabel sebelumnya, kolom "Nilai Basis" tetap konsisten di angka 2, sedangkan kolom "Persentase %" mencatat tingkat kemiripan berdasarkan kombinasi parameter *N-Gram* dan *Window*. Ketika nilai *N-Gram* lebih kecil yaitu 2 dengan *Window* lebih besar (5 dan 7) menghasilkan tingkat kemiripan tertinggi; sedangkan ketika *N-Gram* = 2, *Window* = 5 maka akan menghasilkan 72,00%; berbeda halnya dengan nilai *N-Gram* = 2, *Window* = 7 maka akan menghasilkan 71,23%. Namun ketika nilai *N-Gram* lebih besar (3 atau 5), maka persentase kemiripan cenderung menurun yaitu *N-Gram* = 3, *Window* = 2 akan menghasilkan 58,33%; sedangkan *N-Gram* = 5, *Window* = 7 akan menghasilkan 17,82%. Namun

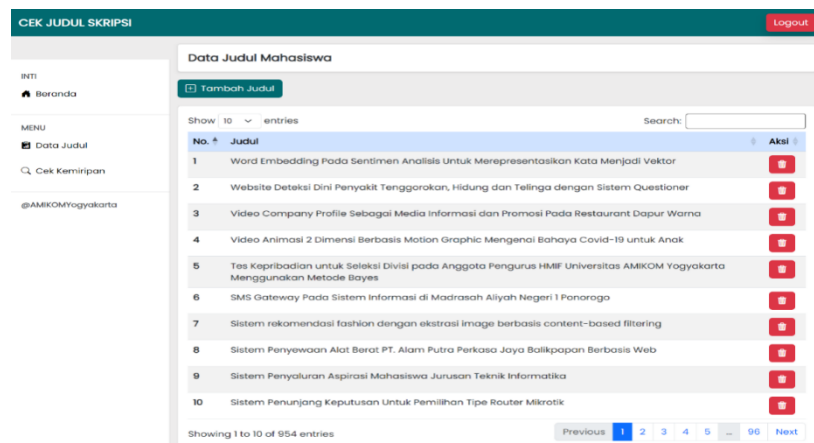
apabila dilihat dari tren tingkat kemiripan maka nilai *N-Gram* Rendah, *Window* Tinggi akan menghasilkan tingkat kemiripan tetap tinggi karena algoritma lebih fleksibel dalam mengidentifikasi pola. Namun ketika nilai *N-Gram* dan *Window* Tinggi: Tingkat kemiripan menurun drastis karena algoritma menjadi lebih selektif dan ketat dalam mencocokkan pola.

Kesimpulan Hasil Pengujian

Hasil yang didapat dari pengujian menunjukkan persentase yang berbeda dari setiap kombinasi dari nilai *n-gram* dan *window*. Dari hasil pengujian tersebut dapat ditarik Kesimpulan bahwa:

- Pengujian yang dilakukan dengan nilai *n-gram* dan *window* yang sama berpengaruh pada hasil *similarity*. Semakin besar *n-gram* dan *window* maka semakin kecil tingkat *similarity*nya dibuktikan pada tabel 6.
- Besar kecilnya nilai *n-gram* memiliki pengaruh terhadap pecahan *string* ke bentuk *n-gram*.
- Semakin kecil nilai *n-gram* maka semakin tinggi nilai *similarity* nya dengan syarat nilai *window* nya lebih besar
- Besar kecilnya nilai *window* memengaruhi jumlah *fingerprint* yang sama antar teks yang dibandingkan.
- Nilai *n-gram* yang besar akan menghasilkan *similarity* yang tinggi jika nilai *window* lebih kecil dari *n-gram*.

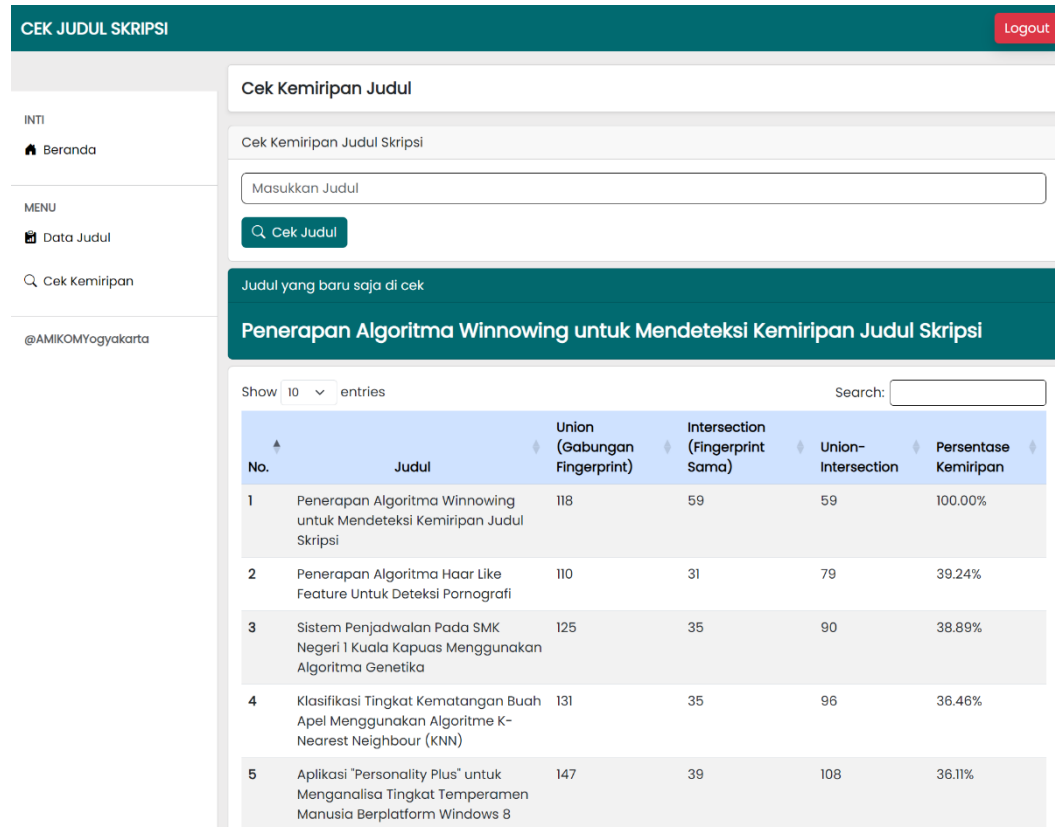
Dari penelitian yang dilakukan, maka menghasilkan suatu *prototype* sistem pendeteksi kemiripan judul skripsi berbasis website seperti pada Gambar 3 sampai dengan gambar Gambar 3 merupakan implementasi untuk menampilkan dan menambahkan judul yang telah ada sebelumnya ataupun menambahkan judul baru yang diajukan mahasiswa jika dosen pembimbing telah menyetujuinya.



Gambar 3. Tampilan Halaman Data Judul

Sedangkan hasil dari implementasi algoritma winnowing ini terdapat pada gambar 4 yang mampu menunjukkan persentase tingkat kemiripan judul yang diinputkan dengan perbandingan dengan judul terdahulu. Persentase diurutkan berdasarkan dari yang tertinggi ke yang paling rendah. Pada hal ini, dosen pembimbing dapat melihat tingkat kemiripan judul yang diinputkan sehingga dosen dapat menentukan jika judul yang diinputkan tersebut memiliki kemiripan yang tinggi dengan persentase yang tinggi maka dosen dapat merekomendasikan judul yang baru. Namun jika judul yang diinputkan tersebut memiliki

tingkat kemiripan dengan persentase yang rendah, maka dapat disimpulkan bahwa judul tersebut tidak memiliki banyak kesamaan dengan judul terdahulu, sehingga dosen dapat menyetujui judul tersebut dan dapat memasukkan judul baru tersebut ke dalam database melalui halaman data judul.



No.	Judul	Union (Gabungan Fingerprint)	Intersection (Fingerprint Sama)	Union- Intersection	Persentase Kemiripan
1	Penerapan Algoritma Winnowing untuk Mendeteksi Kemiripan Judul Skripsi	118	59	59	100.00%
2	Penerapan Algoritma Haar Like Feature Untuk Deteksi Pornografi	110	31	79	39.24%
3	Sistem Penjadwalan Pada SMK Negeri 1 Kuala Kapuas Menggunakan Algoritma Genetika	125	35	90	38.89%
4	Klasifikasi Tingkat Kematangan Buah Apel Menggunakan Algoritme K-Nearest Neighbour (KNN)	131	35	96	36.46%
5	Aplikasi "Personality Plus" untuk Menganalisa Tingkat Temperamen Manusia Berplatform Windows 8	147	39	108	36.11%

Gambar 4. Hasil Cek Kemiripan

Pengujian Aplikasi

Pengujian sistem penting untuk dilakukan di mana pengujian mengacu kepada kualitas sebuah sistem. Pengujian sistem adalah sesuatu yang harus diperhatikan serta amat dibutuhkan guna meyakinkan kualitas sistem yang dalam tahap pembuatan ataupun yang sudah ada agar dapat beroperasi dengan baik fungsinya. Pengujian sistem memiliki waktu tersendiri untuk mencoba sistem, berguna untuk mencari suatu kecacatan yang dapat diketahui dari awal yang dapat segera untuk diperbaiki[14].

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya berfokus pada memeriksa nilai keluaran berdasarkan nilai masukan masing-masing[15]. Tabel 9 merupakan hasil dari pengujian *prototype* sistem cek kemiripan judul skripsi menggunakan metode *black box testing*.

Tabel 9. Hasil Black Box Testing

No.	Fitur yang diuji	Skenario	Hasil yang diharapkan	Hasil pengujian
1.	Login	User memasukkan username dan password yang salah	Login gagal dan tetap berada pada halaman login	Sukses
2.	Login	User memasukkan "admin" dan password	Login berhasil dan akan diarahkan ke halaman	Sukses

		"admin"	dashboard	
3.	Tambah judul skripsi	User menambah judul dengan klik tambah judul lalu menginputkan judul dan klik simpan	Muncul pop up "data berhasil ditambahkan" menandakan data masuk ke database	Sukses
4.	Tambah Judul Skripsi (Import Excel)	User mengimportkan judul dengan data berupa excel melalui form import judul	Muncul keterangan banyak data yang berhasil diimport ke database.	Sukses
5.	Hapus judul	User melakukan penghapusan terhadap judul dengan meng klik icon hapus	Muncul pop up "data telah berhasil dihapus" data hilang dari tabel	Sukses
6.	Cek kemiripan	User menginputkan judul pada form inputan dan klik tombol cek kemiripan	Menunjukkan hasil pada tabel dengan memunculkan judul terdahulu, union, intersection, union-intersection, dan persentase kemiripan	Sukses
7.	Logout	User mengklik tombol logout	Berhasil logout dan kembali ke halaman admin	Sukses

SIMPULAN

Algoritma *Winnowing* terbukti dapat digunakan untuk mendeteksi kemiripan judul skripsi dengan pengujian menggunakan nilai n -gram dan *window* menghasilkan persentase yang beragam. Menurut hasil pengujian, nilai n -gram dan *window* berpengaruh pada persentase hasil kemiripan. Semakin besar nilai n -gram maka tingkat *similarity* akan rendah. Nilai n -gram yang besar akan menghasilkan tingkat *similarity* yang tinggi jika nilai *window*nya kecil. Pengujian dengan n -gram = 2 dan *window* = 5 menghasilkan persentase 72.00%. Dan pengujian dengan n -gram = 5 dan *window* = 7 menghasilkan persentase 17.82%. *Prototype* sistem yang dibuat diharapkan dapat membantu bagi dosen pembimbing untuk menentukan pengambilan keputusan atas judul yang telah diajukan oleh mahasiswa karena dapat mendeteksi kemiripan judul skripsi dengan judul skripsi yang terdahulu.

DAFTAR PUSTAKA

- [1] R. Permatasari, M. Arifin, and R. Padilah, "Studi Deskriptif Dampak Psikologis Mahasiswa Program Studi Bimbingan dan Konseling Universitas PGRI Banyuwangi Dalam Penyusunan Skripsi di Masa Pandemi Covid-19," *J. Bina Ilmu Cendekia*, vol. 2, no. 1, pp. 128–141, 2020, [Online]. Available: <http://jurnal.icjambi.id/index.php>
- [2] N. Alamsyah and M. Rasyidan, "Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi Pada Fakultas Teknologi Informasi Menggunakan Algoritma Winnowing," *Technol. J. Ilm.*, vol. 10, no. 4, p. 197, 2019, doi: 10.31602/tji.v10i4.2361.
- [3] M. P. Rihi, A. A. Pekuwali, and D. A. Sitaniapessy, "Winnowing Algorithm for Detecting the Similarity of Informatic Engineering Undergraduate Thesis Title Algoritma Winnowing untuk Mendeteksi Kesamaan Judul Skripsi Teknik Informatika," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 2, no. 2, pp. 42–52, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/malcom>

- [4] A. Pratomo, A. Irawan, and M. Risa, "Similarity detection design using Winnowing Algorithm as an effort to apply green computing," *J. Phys. Conf. Ser.*, vol. 1450, no. 1, 2020, doi: 10.1088/1742-6596/1450/1/012065.
- [5] M. E. Rijoly, W. Pramudita, B. P. Tomasouw, and Z. A. Leleury, "Perancangan Sistem Deteksi Plagiarisme Skripsi (Judul Dan Abstrak) Berbasis Matlab Menggunakan Algoritma Winnowing," *Tensor Pure Appl. Math. J.*, vol. 2, no. 2, pp. 67–76, 2021, doi: 10.30598/tensorvol2iss2pp67-76.
- [6] Normah, B. Rifai, S. Vambudi, and R. Maulana, "Analisa Sentimen Perkembangan Vtuber Dengan Metode Support Vector Machine Berbasis SMOTE," *J. Tek. Komput. AMIK BSI*, vol. 8, no. 2, pp. 174–180, 2022, doi: 10.31294/jtk.v4i2.
- [7] W. Desena and A. Solichin, "Pencarian Abstrak Tugas Akhir Mahasiswa Berdasarkan Tingkat Kemiripan Menggunakan Algoritma Winnowing dan Jaccard Similarity pada Universitas Budi Luhur," *Inform. J. Ilmu Komput.*, vol. 17, no. 2, p. 112, 2021, doi: 10.52958/iftk.v17i2.3628.
- [8] W. Hidayat, E. Utami, and A. D. Hartanto, "Pemilihan Parameter Terbaik pada Algoritma Winnowing dalam Mendeteksi Tingkat Kesamaan Dokumen Bahasa Indonesia," *Creat. Inf. Technol. J.*, vol. 7, no. 2, p. 119, 2021, doi: 10.24076/citec.2020v7i2.256.
- [9] H. Malius, Apriyanto, and A. A. H. Dani, "Sistem Informasi Sekolah Berbasis Web pada Sekolah Dasar Negeri (SDN) 109 Seriti," *Indones. J. Educ. Humanit.*, vol. 1, no. 3, pp. 156–168, 2021, [Online]. Available: <http://ijoehm.rcipublisher.org/index.php/ijoehm/article/view/28>
- [10] A. Bijaksana, P. Negara, H. Muhardi, and I. M. Putri, "Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes Dan Seleksi Fitur Information Gain Sentiment Analysis on Airlines Using Naïve Bayes Method and Feature Selection Information Gain," *Jtiik*, vol. 7, no. 3, pp. 599–606, 2020, doi: 10.25126/jtiik.202071947.
- [11] L. Sibarani, M. Magdalena, and A. Dharma, "Analisa Perbandingan Sistem Pendeteksian Kemiripan Judul Skripsi Menggunakan Algoritma Winnowing Dan Algoritma Rabin Karp," *REMIK (Riset dan E-Jurnal Manaj. Inform. Komputer)*, vol. 4, no. 1, p. 69, 2019, doi: 10.33395/remik.v4i1.10174.
- [12] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing," *Transmisi*, vol. 20, no. 3, p. 105, 2018, doi: 10.14710/transmisi.20.3.105-110.
- [13] R. M. Fauzi and J. C. Wibawa, "Implementasi Algoritma Winnowing untuk Mendeteksi Kemiripan Teks pada Artikel," 2018.
- [14] A. Ijudin and A. Saifudin, "Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 8, 2020, doi: 10.32493/informatika.v5i1.3717.
- [15] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.