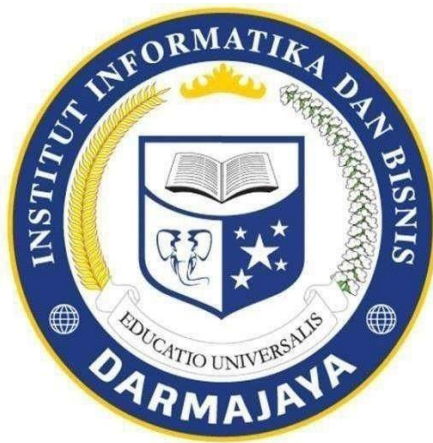


**IMPLEMENTASI ALGORITMA *TF-IDF* DAN *FUZZY MATCHING*
UNTUK DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS APLIKASI
ANDROID.**

**STUDI KASUS: JUDUL SKRIPSI PRODI TEKNIK INFORMATIKA IIB
DARMAJAYA**

SKRIPSI



Disusun oleh :

Arif Nur Listanto 2011010035

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA

BANDAR LAMPUNG

TAHUN 2024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Skripsi merupakan salah satu tahapan akhir dalam menyelesaikan studi di perguruan tinggi. Setiap mahasiswa pada jalur skripsi harus menentukan tema dan judul skripsi sesuai dengan bidang ilmu masing-masing mahasiswa. Skripsi harus ditulis dengan orisinal dalam arti tidak plagiat dan tidak memiliki kemiripan yang sangat tinggi antara satu penelitian dengan penelitian lainnya.

Setiap judul skripsi yang diajukan oleh mahasiswa akan di filter oleh ketua Kelompok Bidang Keilmuan (KBK). Di Darmajaya khususnya prodi Teknik Informatika terdapat 3 konsentrasi bidang keilmuan yaitu Artificial Intelligence, Mobile Technology, dan Multimedia. Masing-masing bidang keilmuan memiliki ketua KBK. KBK akan melakukan pengecekan dari setiap judul skripsi yang diajukan oleh mahasiswa dalam rangka menghindari judul identik, kemiripan yang sangat tinggi, dan plagiarisme dalam penelitian skripsi.

Masalah yang sering terjadi yaitu judul yang diajukan oleh mahasiswa seringkali identik, sangat mirip dengan judul yang sudah pernah diajukan sebelumnya, ataupun mahasiswa hanya mengganti judul dengan lokasi penelitian yang berbeda tanpa adanya nilai keterbaruan dari penelitian tersebut. Judul skripsi yang identik, sangat mirip, dan plagiat pasti ditolak oleh KBK. Saat ini terdapat ±2148 judul skripsi pada *spreadsheet* prodi teknik informatika. Sehingga menyulitkan untuk mendeteksi kemiripan pada judul skripsi yang diajukan. Untuk mendeteksi kemiripan pada judul skripsi.

Kombinasi algoritma *TF-IDF* dengan *Fuzzy Matching* merupakan salah satu algoritma yang cocok untuk mendeteksi tingkat kemiripan judul skripsi yang diajukan oleh mahasiswa.

Algoritma *TF-IDF* adalah metode yang digunakan untuk mengukur seberapa penting sebuah kata dalam sebuah dokumen[1], dimana *TF-IDF* secara umum didefinisikan sebagai 2 algoritma yang berbeda. *TF* (*Term Frequency*) merupakan algoritma yang menghitung seberapa sering sebuah kata muncul dalam sebuah dokumen, sedangkan *IDF* (*Inverse Document Frequency*) merupakan algoritma yang menghitung seberapa jarang sebuah kata muncul dalam sebuah dokumen. Algoritma kombinasi lainnya yaitu *Fuzzy Matching* merupakan metode pencocokan string yang menggunakan pendekatan terhadap pola dari string yang dicari, dimana algoritma *Fuzzy Matching* bekerja dengan cara yang berbeda dari pencocokan string biasa (*exact matching*)[2], algoritma ini berfokus pada menemukan kecocokan string yang “mirip”. Sehingga apabila kedua algoritma ini dikombinasikan maka akan menghasilkan tingkat akurasi yang cukup tinggi dalam mendeteksi kemiripan judul – judul skripsi mahasiswa.

Berdasarkan uraian diatas, maka penulis mengajukan judul skripsi **“IMPLEMENTASI ALGORITMA *TF-IDF* DAN *FUZZY MATCHING* UNTUK DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS APLIKASI ANDROID. STUDI KASUS: JUDUL SKRIPSI PRODI TEKNIK INFORMATIKA IIB DARMAJAYA”** untuk memudahkan kelompok bidang keilmuan (KBK) melakukan deteksi tingkat kemiripan pada judul – judul yang diajukan yang ada pada jurusan teknik informatika dengan algoritma *TF-IDF* dan *Fuzzy Matching*.

1.2 Identifikasi Masalah

Berdasarkan uraian pada latar belakang, ditemukan informasi mengenai sistem yang akan dibangun. Identifikasi masalah melalui latar belakang diantaranya:

1. Sulitnya bagi mahasiswa untuk mencari kemiripan ide judul mereka dengan judul yang sudah ada.
2. Pengecekan judul – judul mahasiswa yang diajukan masih dilakukan dengan cara manual yang tersimpan dalam bentuk *Microsoft Excel*.
3. Pertimbangan judul – judul yang telah diajukan ke kelompok bidang keilmuan (KBK) menjadi kurang efisien karena dilakukan satu persatu apabila judul yang diajukan sangat banyak, sehingga memungkinkan untuk terjadinya judul yang sama.

1.3 Rumusan Masalah

Perumusan masalah berdasarkan uraian latar belakang menjadi “Bagaimana kelompok bidang keilmuan (KBK) dapat mendeteksi kemiripan judul yang diajukan oleh mahasiswa teknik informatika dengan judul – judul yang sudah ada dan bagaimana mahasiswa dapat menentukan serta mendeteksi adanya kemiripan ide judulnya dengan judul yang sudah ada dengan menerapkan algoritma *TF-IDF* dan *Fuzzy Matching* untuk aplikasi berbasis android?”.

1.4 Batasan Masalah

Setiap penelitian memiliki batasan – batasan masalahnya. Batasan masalah untuk penelitian ini adalah:

1. Ruang lingkup penelitian ini hanya mencakup program studi teknik informatika IIB Darmajaya.

2. Pengguna aplikasi ini adalah kelompok bidang keilmuan (KBK) dan mahasiswa program studi teknik informatika.
3. Kombinasi algoritma *TF-IDF* dan *Fuzzy Matching* memiliki kekurangan dalam memahami konteks judul – judul, data yang belum pernah ada, hubungan antar judul yang kompleks, serta sumber daya yang tinggi untuk melakukan komputasi dengan algoritma ini.
4. Menggunakan Bahasa pemrograman *Python* sebagai *backend computing* dan database *JSON* untuk penyimpanan data judul.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini ialah:

1. Membantu menghindari adanya kemungkinan judul skripsi yang sama antara judul yang diajukan dengan judul yang sudah ada.
2. Meningkatkan efesiensi proses pertimbangan terhadap judul – judul yang diajukan oleh mahasiswa teknik informatika.
3. Membantu pendeteksian kemiripan judul yang diajukan secara efektif.

1.6 Manfaat Penelitian

Beberapa manfaat dari penelitian ini diantaranya ialah:

1. Membantu pengambilan keputusan pertimbangan kelompok bidang keilmuan (KBK) dalam menentukan judul – judul yang diajukan.
2. Membantu memberikan rekomendasi judul serupa yang cukup akurat.
3. Membantu dalam mendeteksi persentase kemiripan dari judul – judul yang diajukan.

1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Berisi penjelasan mengenai latar belakang identifikasi masalah, rumusan masalah, ruang lingkup penelitian, tujuan, manfaat dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi dasar teori yang digunakan sebagai acuan dalam pengambilan data, analisis, hasil penelitian dan pengujian.

BAB III METODE PENELITIAN

Dalam bab ini dijelaskan tentang metode pendekatan-pendekatan yang dilakukan untuk mendukung dalam proses penelitian.

DAFTAR PUSTAKA

BAB II

TINJAUAN PUSTAKA

2.1 Implementasi

Implementasi sebagai kegiatan mendistribusikan keluaran dari suatu kebijakan yang dijalankan oleh seorang pelaksana (untuk menyampaikan keluaran kebijakan) kepada suatu kelompok sasaran dalam upaya mencapai kebijakan tersebut[3].

2.2 Algoritma

Algoritma adalah sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk menyelesaikan suatu persoalan atau suatu permasalahan matematika dan logika dengan bantuan komputer[4].

Algoritma sebagai suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu[5].

Dari beberapa pengertian sebelumnya, dapat disimpulkan bahwa algoritma adalah serangkaian langkah-langkah yang terdefinisi dengan baik yang digunakan untuk menyelesaikan suatu masalah. Algoritma haruslah jelas, terstruktur, dan terdefinisi dengan baik agar dapat dimengerti dan diterapkan dengan benar.

3.3 Deteksi

Kata “deteksi” berasal dari bahasa latin "*Detectio*" yang berarti penemuan atau pengungkapan, dalam Bahasa Indonesia, deteksi merupakan penemuan, pengenalan, atau penentuan terhadap suatu objek[6].

3.4 Aplikasi Android

Aplikasi Android adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia[7], sedangkan menurut Nazaruddin, Aplikasi Android adalah perangkat lunak yang berjalan pada platform Android. Platform Android adalah sistem operasi mobile yang dikembangkan oleh Google dan bersifat open source.

Aplikasi Android adalah program komputer yang dirancang untuk dijalankan pada perangkat mobile yang menggunakan sistem operasi Android. Aplikasi Android dapat digunakan untuk berbagai macam keperluan, seperti komunikasi, hiburan, edukasi, dan bisnis[8]. Sedangkan menurut Sibero Aplikasi Android adalah program yang dibuat dengan menggunakan bahasa pemrograman Java dan SDK (Software Development Kit) Android. Aplikasi Android dapat diunduh dan dipasang pada perangkat Android dari Google Play Store atau sumber pihak ketiga lainnya.

Sehingga dapat disimpulkan bahwa aplikasi Android adalah program komputer yang dirancang untuk dijalankan pada perangkat mobile yang menggunakan sistem operasi Android. Aplikasi Android dapat digunakan untuk berbagai macam keperluan, dan dapat diunduh dan dipasang pada perangkat Android dari berbagai sumber.

Beberapa ciri – ciri aplikasi android diantaranya:

- Dibuat dengan menggunakan bahasa pemrograman Java dan SDK Android.
- Dapat diunduh dan dipasang pada perangkat Android dari Google Play Store atau sumber pihak ketiga lainnya.
- Berjalan pada sistem operasi Android.
- Digunakan untuk berbagai macam keperluan, seperti komunikasi, hiburan, edukasi, dan bisnis.

2.5 Algoritma *TF-IDF*

Algoritma *Term Frequency – Inverse Document Frequency* merupakan metode yang digunakan untuk menghitung bobot atau nilai setiap kata kunci dalam sebuah dokumen. Bobot atau nilai ini mencerminkan seberapa penting kata kunci dalam dokumen dan seberapa jarang kata kunci muncul di dalam dokumen.

Algoritma TF-IDF banyak digunakan dalam berbagai aplikasi pengolahan teks, seperti:

- Pencarian informasi: Algoritma ini membantu mesin pencari menemukan dokumen yang relevan dengan kueri pengguna.
- Klasifikasi teks: Algoritma ini membantu mengklasifikasikan dokumen ke dalam kategori tertentu.
- Ringkasan teks: Algoritma ini membantu mengidentifikasi kata kunci penting dalam dokumen dan membuat ringkasan yang informatif.
- Analisis sentimen: Algoritma ini membantu menentukan opini atau perasaan yang terkandung dalam sebuah teks.

2.5.1 *Term Frequency*

Term Frequency merupakan metode yang bekerja dengan cara mengukur seberapa sering kata kunci muncul dalam dokumen. Semakin sering kata kunci muncul, semakin tinggi nilai TF-nya.

$$tf(t, d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

Sumber: *Wikipedia tf-idf*^[4]

Keterangan:

$tf(t, d)$ = Nilai kata yang sering muncul dalam kumpulan dokumen.

t = term
 d = dokumen

2.5.2 Inverse Document Frequency

Sedangkan *Inverse Document Frequency*, Mengukur seberapa jarang kata kunci muncul di seluruh dokumen dalam kumpulan data. Semakin jarang kata kunci muncul, semakin tinggi nilai IDF-nya.

$$\text{idf}(t, D) = \log \frac{N}{|\{d : d \in D \text{ and } t \in d\}|}$$

Sumber: Wikipedia *tf-idf*^[4]

Keterangan:

$\text{idf}(t, D)$ = Nilai kata yang jarang muncul dalam kumpulan dokumen.

t = term

N = Total nilai dokumen dalam sebuah list.

d = Dokumen

D = Total dokumen

Nilai *TF-IDF* dihitung dengan mengalikan nilai *TF* dan nilai *IDF*, sehingga rumusnya adalah:

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Sumber: Wikipedia *tf-idf*^[4]

Keterangan:

$\text{tf}(t, d)$ = Total nilai kata yang sering muncul dalam kumpulan dokumen

$\text{idf}(t, D)$ = Nilai kata yang jarang muncul dalam kumpulan dokumen.

t	= <i>term</i>
d	= Dokumen
D	= Total dokumen

2.5.3 Langkah Langkah Algoritma *TF – IDF*

Misalkan kita memiliki dua dokumen:

- Dokumen 1: "Mobil berwarna merah itu melaju kencang di jalan tol."
- Dokumen 2: "Sepeda motor adalah kendaraan roda dua yang banyak digunakan di Indonesia."

Mari kita hitung nilai TF-IDF untuk kata kunci "mobil" pada kedua dokumen:

- Dokumen 1:

TF: Kata kunci "mobil" muncul 1 kali.

IDF: Asumsikan kata kunci "mobil" muncul dalam 10% dokumen dalam kumpulan data. Maka, $IDF = \log(N/n)$, di mana N adalah jumlah total dokumen dan n adalah jumlah dokumen yang mengandung kata kunci "mobil". Dalam kasus ini, $IDF = \log(10/1) = 2.3025$.

$$TF-IDF = 1 * 2.3025 = 2.3025$$

- Dokumen 2:

TF: Kata kunci "mobil" tidak muncul.

IDF: IDF sama seperti pada Dokumen 1.

$$TF-IDF = 0 * 2.3025 = 0$$

Dari contoh ini, kita dapat melihat bahwa nilai TF-IDF untuk kata kunci "mobil" pada Dokumen 1 lebih tinggi daripada Dokumen 2. Hal ini menunjukkan bahwa kata kunci "mobil" lebih penting bagi Dokumen 1 dibandingkan Dokumen 2.

2.6 Algoritma Fuzzy Matching

Algoritma Fuzzy Matching adalah teknik pencocokan string yang mempertimbangkan kemiripan antara dua string, bukan hanya persamaan kedua string. Hal ini memungkinkan algoritma untuk menemukan kecocokan bahkan ketika terdapat perbedaan kecil dalam ejaan, tata bahasa, atau struktur kalimat.

Algoritma Fuzzy Matching bekerja dengan menghitung jarak antara dua string. Jarak ini mewakili jumlah operasi yang diperlukan untuk mengubah satu string menjadi string lainnya. Operasi ini dapat berupa:

- Penyisipan: Menambahkan karakter ke string
- Penghapusan: Menghapus karakter dari string
- Penggantian: Mengubah karakter dalam string

Setelah menghitung jarak string, algoritma Fuzzy Matching akan menentukan apakah dua string cukup mirip untuk dianggap sebagai kecocokan. Nilai ambang batas kesamaan biasanya digunakan untuk menentukan hal ini.

2.6.1 Cara Kerja Algoritma Fuzzy Matching

Berikut adalah contoh bagaimana algoritma Fuzzy Matching dapat digunakan untuk menemukan kecocokan antara dua string:

String 1: "Sepatu lari"

String 2: "Sepatu lari merah"

Algoritma Levenshtein Distance akan menghitung jarak antara dua string ini sebagai satu. Hal ini karena hanya perlu satu operasi (penghapusan kata "merah") untuk mengubah String 1 menjadi String 2. Misalkan nilai ambang batas kesamaan untuk algoritma Fuzzy Matching adalah 0.8. Dalam hal ini, algoritma akan menganggap

String 1 dan String 2 sebagai kecocokan karena jaraknya (0.8) lebih besar dari nilai ambang batas.

2.7 UML (*Unified Modeling Language*)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[9].



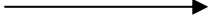
Perancangan Sistem Terdapat beberapa bagian dalam perancangan UML (*Unified Modelling Language*) yaitu sebagai berikut:


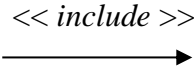
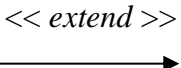
2.7.1 *Use Case Diagram*

Use case diagram adalah sebuah diagram UML yang digunakan untuk menggambarkan interaksi antara aktor dan sistem. Diagram ini menunjukkan apa yang dilakukan aktor dengan sistem, dan bagaimana sistem menanggapi[10].

Use Case Diagram memiliki beberapa simbol diantaranya:

Tabel 2.7.1 Simbol – Simbol *Use Case Diagram*



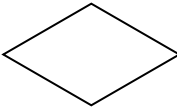

Simbol	Deskripsi
Simbol <i>Use Case</i> 	Fungsionalitas yang disediakan <i>Use Case Diagram</i> sebagai unit tukar pesan.
Actor 	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>Use Case</i> lain.
Association 	Abstraksi dari penghubung antara aktor dengan <i>use case</i>
Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>



	
<i>Include</i> 	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
<i>Extend</i> 	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2.7.2 Activity Diagram

Activity diagram merupakan alur kerja dari sebuah sistem yang digambarkan melalui diagram[11]. Berikut adalah simbol – simbol dari *activity diagram*:

Tabel 2.7.2 Simbol – Simbol *Activity Diagram*

Simbol	Deskripsi
Status Awal 	Status awal dari sebuah diagram aktivitas.
Aktivitas 	Merupakan simbol yang digunakan saat sistem sedang memproses suatu kegiatan, kalimatnya diawali dengan kata kerja.
Percabangan 	Merupakan simbol yang digunakan dimana ada pilihan aktivitas yang lebih dari satu.
Penggabungan 	Merupakan simbol yang digunakan oleh sistem dimana terdapat aktivitas lebih dari satu lalu digabungkan menjadi satu.

Status Akhir		Merupakan simbol yang menandakan aktivitas terakhir yang dilakukan oleh sistem.
Swimlane		Merupakan simbol yang digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.8 Database MongoDB

MongoDB adalah basis data *NoSQL* yang kuat dan fleksibel, banyak digunakan oleh para ahli dalam berbagai bidang industri. Keunggulannya dalam skalabilitas, kinerja, dan kemudahan penggunaan menjadikannya pilihan populer untuk aplikasi modern[12]. Tidak seperti basis data relasional tradisional yang menggunakan tabel dan skema tetap, *MongoDB* menyimpan data dalam dokumen fleksibel yang mirip dengan format *JSON*. Ini memungkinkan penyimpanan data yang tidak terstruktur atau semi-terstruktur dengan mudah. *MongoDB* dirancang untuk skalabilitas horizontal, artinya dapat didistribusikan di beberapa server untuk menangani beban kerja yang besar dan meningkatkan kinerja.

Beberapa keunggulan *MongoDB* diantaranya:

- Fleksibilitas skema
- Skalabilitas
- Kinerja tinggi
- Kemudahan penggunaan
- Dukungan komunitas yang kuat

2.9 Pengujian Black Box

Pengujian *Black Box* merupakan pengujian perangkat lunak dimana pengujian berasal dari persyaratan fungsional tanpa menguji

desain dan kode program. Pengujian dilakukan untuk mengetahui setiap fungsi, masukan, dan keluaran dari program sesuai dengan persyaratan yang dibutuhkan. Pengujian *black box* dilakukan dengan membuat kasus – kasus yang bersifat menguji semua fungsi dengan alat uji apakah sesuai dengan spesifikasi yang dibutuhkan.

2.10 Hasil Penelitian Sebelumnya

Berikut ini adalah beberapa literatur yang digunakan dalam penelitian, dapat dilihat pada tabel berikut :

Tabel 2.10.1 Tinjauan Pustaka Jurnal

No	Nama, Tahun	Judul	Tujuan Penelitian	Hasil Penelitian
1	Herlina Sari, 2021	Penerapan Algoritma Text Mining Dan TF-IDF Untuk Pengelompokan Topik Skripsi Pada Aplikasi Repository STMIK Budi Darma	Tujuan dari penelitian yang dibahas di halaman ini adalah untuk mengembangkan aplikasi repository yang dapat mengelompokkan topik skripsi secara otomatis menggunakan algoritma Text Mining dan TF-IDF.	Studi ini menggali bagaimana teknik text mining dan algoritma TF-IDF dapat digunakan untuk mengklasifikasikan dan mengelompokkan topik skripsi yang ada dalam repositori digital.
2	Ni Putu Anggi Yuliani, 2017	Sistem Temu Balik Dokumen Teks Menggunakan Metodeboolean dan <i>Term Wight Tf.Idf</i>	Tujuan penelitian dari jurnal tersebut adalah untuk mengembangkan Sistem Temu Balik Dokumen Teks yang efisien menggunakan metode Boolean dan Term Weight TF.IDF.	Metode Boolean: Penelitian ini menggunakan metode Boolean untuk memproses pencarian dokumen teks. Term Weight TF.IDF: Penelitian ini juga mengimplementasikan pembobotan term menggunakan TF.IDF untuk meningkatkan akurasi hasil temu balik.

				<p>Efisiensi Sistem: Hasil penelitian menunjukkan peningkatan efisiensi dalam sistem temu balik dokumen teks.</p> <p>Implementasi: Penelitian ini memberikan wawasan tentang implementasi sistem yang diusulkan dalam praktik nyata.</p>
3	Nilawati	Penggunaan Metode Cosine Similarity Dan Tf-Idf Untuk Klasifikasi Judul Seminar Proposal Pada Fakultas Teknik Universitas Jabal Ghafur	Tujuan penelitian ini untuk mengklasifikasikan judul seminar proposal pada Fakultas Teknik Universitas Jabal Ghafur dengan metode Cosine Similarity dan TF-IDF, berfokus pada pengembangan sistem secara otomatis mengkategorikan judul berdasarkan kesamaan konten, sehingga membantu pengelolaan pengetahuan dan organisasi informasi seminar.	<p>Penelitian ini menghasilkan sebuah sistem klasifikasi judul seminar proposal yang efektif. Sistem ini menggunakan metode Cosine Similarity dan TF-IDF (Term Frequency-Inverse Document Frequency) untuk mengukur kesamaan antara dokumen teks.</p>
4	Dhebys Suryani Hormansyah, 2018	Penerapan Metode Tf-Idf Dan N-Gram Pada Pengembangan Aplikasi Chatbot Berbasis Line Untuk Layanan Publik Kesehatan Di Kota Malang	Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi chatbot berbasis LINE yang menggunakan metode TF-IDF dan N-GRAM untuk	<p>Penelitian tersebut mengembangkan aplikasi chatbot berbasis LINE yang menggunakan metode TF-IDF dan N-gram untuk meningkatkan layanan publik kesehatan di Kota Malang.</p>

			meningkatkan layanan publik kesehatan di Kota Malang.	
5	Kairunnisa, 2019	Penerapan Fuzzy String Matching Pada Aplikasi Pencarian Judul Skripsi Teknik Informatika Berbasis Android	Tujuan dari penelitian ini adalah untuk mengembangkan sebuah aplikasi repository berbasis Android yang memanfaatkan algoritma Fuzzy String Matching	Aplikasi ini dirancang untuk membantu mahasiswa dalam mencari referensi skripsi dengan cepat dan efisien. Dengan aplikasi ini, pengguna dapat melakukan pencarian judul skripsi pada STMIK Budi Darma dan mendapatkan saran kata yang relevan,

BAB III METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Dalam metode pengumpulan data memiliki peranan yang penting dalam mendapatkan suatu informasi. Data yang relevan dengan pokok pembahasannya adalah indikator keberhasilan suatu penelitian. Dalam pengumpulan data, beberapa metode yang digunakan diantaranya:

3.1.1 Wawancara

Wawancara adalah proses pengumpulan data dengan tanya jawab secara langsung dengan orang – orang yang terkait. Berikut ini merupakan beberapa pertanyaan wawancara yang dilakukan yaitu:

- a. Kendala apa saja yang dialami oleh kelompok bidang keilmuan (KBK)?
- b. Berapa lama periode mahasiswa mengumpulkan judul skripsi?

3.1.2 Pengamatan

Penulis mengamati secara langsung lingkungan sistem yang akan dibangun. Ini bertujuan untuk mendapatkan gambaran yang jelas terhadap sistem yang akan dibangun.

3.1.3 Tinjauan Pustaka

Dalam pengumpulan data pada prosesnya dilakukan dengan mempelajari berbagai bentuk bahan tertulis yang berupa buku – buku, artikel, dokumen – dokumen, yang terkait secara langsung dengan sistem kemiripan judul skripsi.

3.2 Analisis Sistem

Dalam perancangan sebuah sistem, diperlukan langkah untuk mengidentifikasi dan mengevaluasi permasalahan atau hambatan dalam proses perancangan. Adapun kebutuhan dalam perancangan yang akan diimplementasikan diantaranya, perangkat keras dan perangkat lunak.

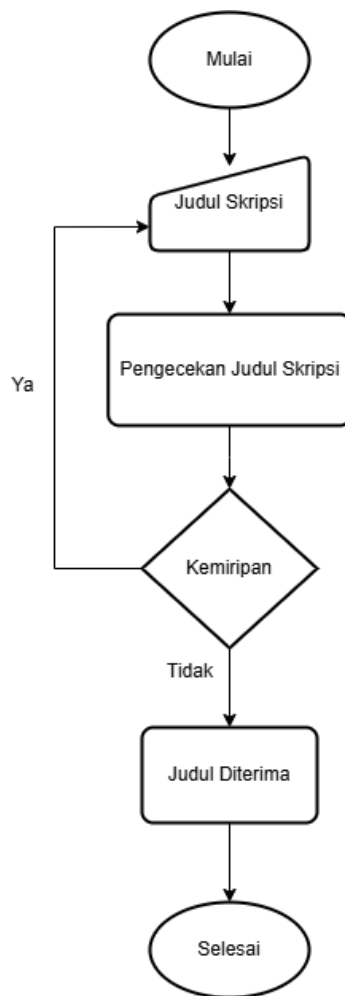
3.2.1 Analisis Kebutuhan Perangkat Lunak

Dalam implementasi aplikasi perangkat lunak, maka software yang dibutuhkan adalah sebagai berikut:

1. Sistem Operasi Windows
2. Bahasa Pemrograman Java
3. Android Studio
4. *Internet*

3.3 Perancangan Sistem

Desain aplikasi dibuat untuk memudahkan dalam pembuatan aplikasi yang berjalan berdasarkan kebutuhan sistem, berikut adalah alur sistem yang berjalan proses pengecekan kemiripan judul skripsi.



Gambar 3.3.1 *Flowchart* Proses Pengecekan

Berdasarkan alur diatas pada proses pengecekan kemiripan judul skripsi, dimulai dari memasukkan judul skripsi, selanjutnya sistem akan memproses tingkat kemiripan judul skripsi yang dimasukkan dengan beberapa judul skripsi yang sudah pernah ada. Apabila tingkat kemiripan judul skripsi tinggi, maka akan ditolak terhadap judul yang diajukan oleh mahasiswa, dan apabila tingkat kemiripan tidak terlalu tinggi maka judul memiliki kemungkinan untuk diterima.

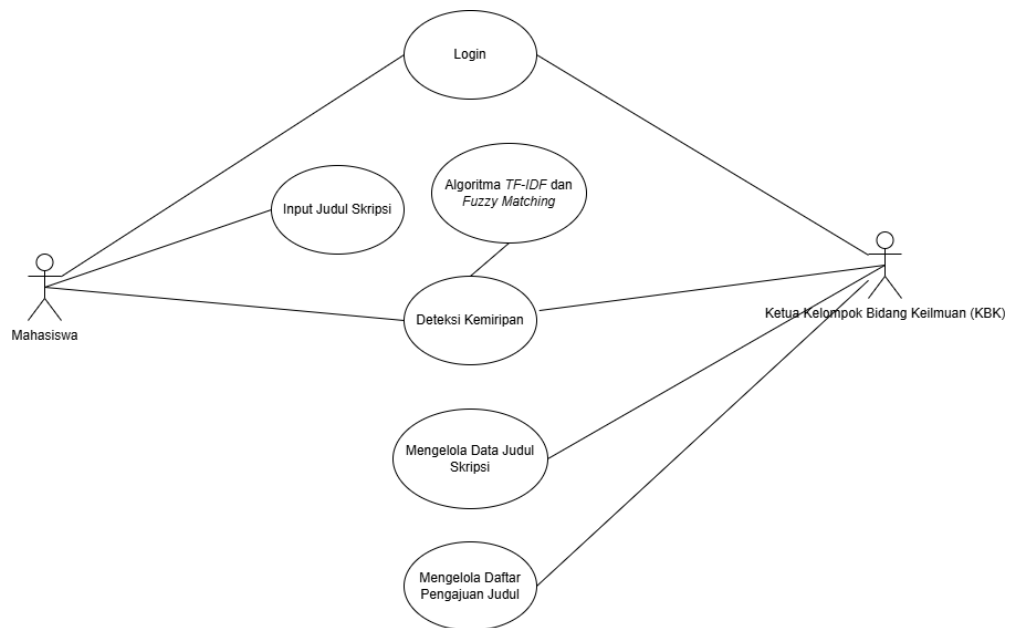
Berdasarkan masalah dan alur sistem yang ada, maka akan dibangun sistem yang dapat mengukur tingkat kemiripan judul skripsi menggunakan algoritma *TF-IDF* dengan kombinasi algoritma *Fuzzy Matching*.

3.4 Tahapan Membangun *Prototype*

Tools yang digunakan dalam perancangan ini adalah UML yaitu penggambaran *use case diagram*, *activity diagram*, dan *class diagram*.

3.4.1 Penggambaran *Usecase Diagram*

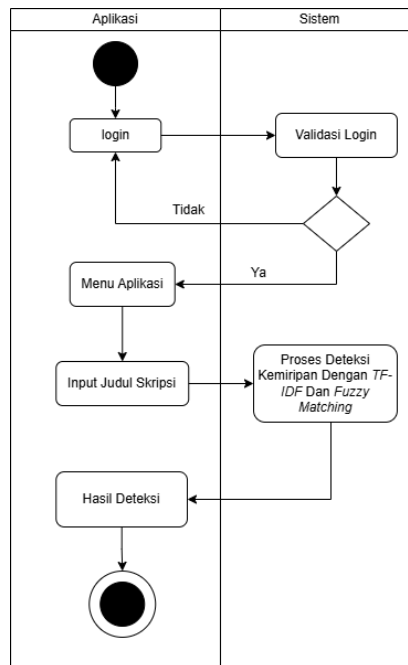
Usecase Diagram merupakan *tool* yang mendeskripsikan interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut merupakan *usecase diagram* dari sistem deteksi kemiripan judul skripsi.



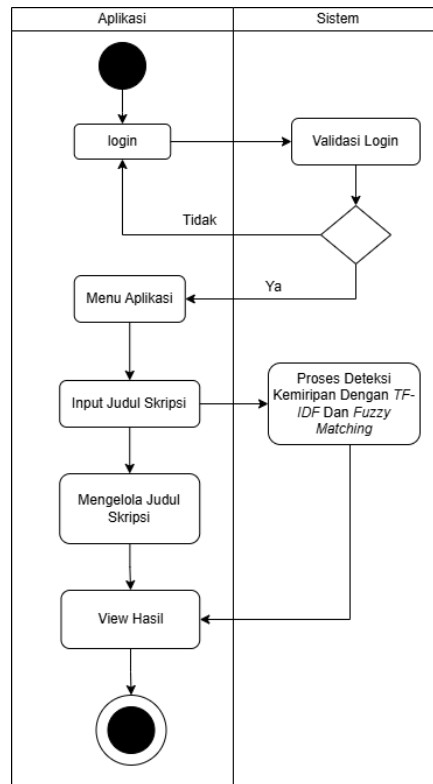
Gambar 3.4.1 Diagram *Usecase*

3.4.2 Penggambaran *Activity Diagram*

Activity Diagram merupakan aktivitas dari sebuah sistem atau proses bisnis. Berikut adalah *activity diagram* yang digunakan oleh sistem deteksi kemiripan judul skripsi.



Gambar 3.4.2 *Activity Diagram Mahasiswa*



Gambar 3.4.3 *Activity Diagram Admin*

3.5 Desain Aplikasi

3.5.1 Desain Tampilan Awal Aplikasi

Tampilan awal merupakan tampilan yang dihasilkan oleh program saat pertama kali program dijalankan. Dibawah ini merupakan gambar *prototype* dari tampilan awal aplikasi deteksi kemiripan judul skripsi.

Check Kemiripan Judul Skripsi

cari judul

hasil

hasil dari judul yang dicari,
serta deteksi tingkat
kemiripannya

Gambar 3.5.1 *Prototype* Tampilan Awal Aplikasi

3.5.2 Tampilan Halaman Login

Check Kemiripan Judul Skripsi

username

password

LOGIN

Gambar 3.5.2 *Prototype* Tampilan Login Aplikasi

3.5.3 Tampilan Awal Hak Akses Ketua KBK

Check Kemiripan Judul Skripsi

TAMBAH

EDIT

cari judul

hasil

hasil dari judul yang dicari,
serta deteksi tingkat kemiripannya

Gambar 3.5.3 *Prototype* Tampilan Awal Admin

3.5.4 Tampilan Tambah Judul Hak Akses Ketua KBK

Check Kemiripan Judul Skripsi

Tambah Judul

Nama Judul

Nama Mahasiswa

Tahun Pengajuan

TAMBAH

Gambar 3.5.4 *Prototype* Tampilan Tambah Judul

3.5.5 Tampilan Edit Judul Hak Akses Ketua KBK

Check Kemiripan Judul Skripsi

Edit Judul

Nama Judul

Nama Mahasiswa

Tahun Pengajuan

EDIT

Gambar 3.5.5 *Prototype* Tampilan Edit Judul

DAFTAR PUSTAKA

- [1] Y. Gu, Y. Wang, J. Huan, Y. Sun, and S. Xu, "An improved TFIDF algorithm based on dual parallel adaptive computing model," *Int. J. Embed. Syst.*, 2020, doi: 10.1504/IJES.2020.108278.
- [2] L. Mo, "Fuzzy matching algorithm of network information retrieval based on discrete mathematics," *Appl. Nanosci.*, 2023, doi: 10.1007/s13204-021-02190-y.
- [3] M. A. Dalma, "Implementasi : Pengertian, Tujuan dan Contoh," *Dosen Pintar*. 2021.
- [4] Sutiono, "Pengertian Algoritma, Flowchart dan Pseudocode," *DosenIT.com*. 2017.
- [5] R. Abidin, "Pengertian Algoritma Pemrograman," 25 April.
- [6] R. E. Izzaty, B. Astuti, and N. Cholimah, "Pengertian dan Tujuan Deteksi Deteksi," *Angew. Chemie Int. Ed.* 6(11), 951–952., 2019.
- [7] I. A. Fahmy, "5 Pengertian Aplikasi Android Menurut Para Ahli," Ani Ramdhani.
- [8] Suparyanto dan Rosad, "Pengertian Aplikasi Mobile Dan Android," *Suparyanto dan Rosad*, 2020.
- [9] R. A. S and M. Salahudin, "Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek," in *Bandung: Informatika*, 2013.
- [10] Rosa A.S dan M. Shalahudin, "Usecase Diagram," *Contemp. Psychol. A J. Rev.*, 2015.
- [11] A. Paramitha, "Materi 4 - activity diagram," *Mater. 4 - Act. Diag. APSI - 2*, 2018.
- [12] MongoDB, "MongoDB Architecture Guide," *MongoDB White Pap.*, 2016.

