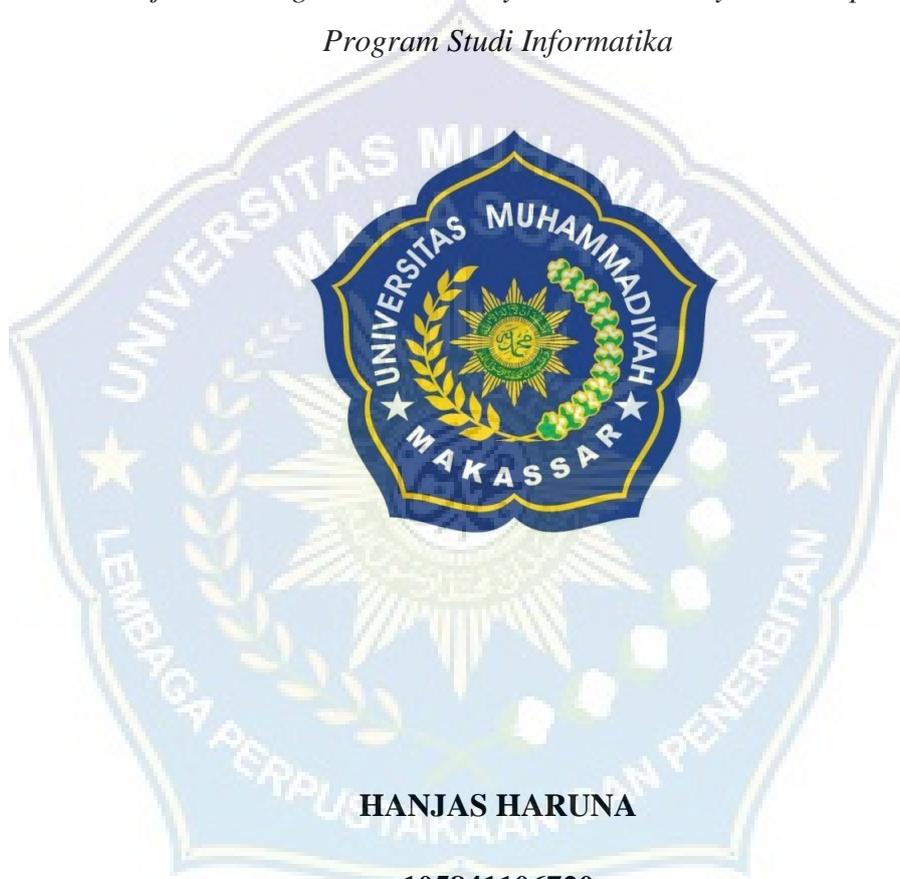


**DETEKSI TINGKAT KEMIRIPAN JUDUL SKRIPSI PRODI
INFORMATIKA MENGGUNAKAN METODE *SUPPORT VECTOR
MACHINE* & *NATURAL LANGUAGE PROCESSING***

SKRIPSI

*Diajukan sebagai Salah Satu Syarat untuk Menyusun Skripsi
Program Studi Informatika*



HANJAS HARUNA

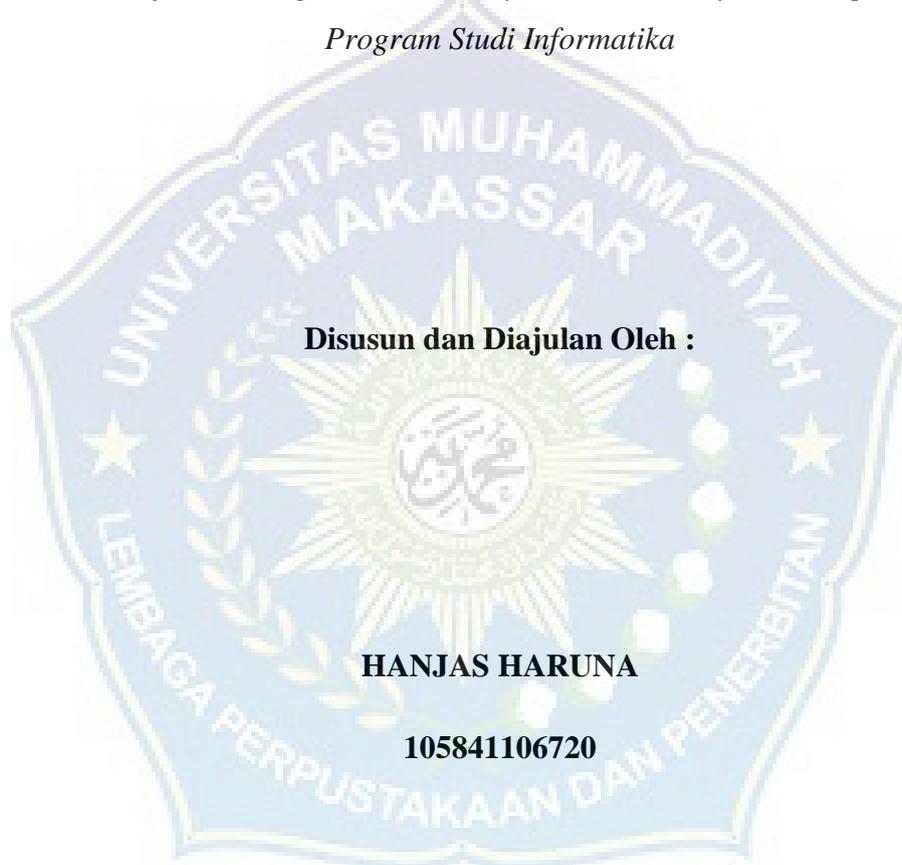
105841106720

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR**

2024

**DETEKSI TINGKAT KEMIRIPAN JUDUL SKRIPSI PRODI
INFORMATIKA MENGGUNAKAN METODE *SUPPORT VECTOR
MACHINE* & *NATURAL LANGUAGE PROCESSING***

*Diajukan sebagai Salah Satu Syarat untuk Menyusun Skripsi
Program Studi Informatika*



Disusun dan Diajukan Oleh :

HANJAS HARUNA

105841106720

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2024**



FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Hanjas Haruna dengan nomor induk Mahasiswa 105 84 11067 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0008/SK-Y/55202/091004/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 26 Agustus 2024

Panitia Ujian :

Makassar,

26 Muharram 1446 H

26 Agustus 2024 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasahuddin

Prof. Dr. Eng. Muhammad Isran Ramli, ST., MT.

2. Penguji

a. Ketua : Dr. Ir. Hj. Hafsa Nirwana, ST., MT.

b. Sekretaris : Fahim Irfan Rahman S.Kom., MT.

3. Anggota

1. Muhyiddin A.M. Hayat, S.Kom., MT.

2. Lukman Anas, S.Kom., MT.

3. Lukman S.Kom., MT.

Mengetahui :

Pembimbing I

Pembimbing II


Rizki Yusliana Bakti ST., MT


Titin Wahyuni S.Pd., M.T

Dekan


Dr. Ir. Hj. Nurnawaty, ST., MT., IPM.

NBM : 795/108



FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **DETEKSI TINGKAT KEMIRIPAN JUDUL SKRIPSI PRODI INFORMATIKA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE & NATURAL LANGUAGE PROCESSING**

Nama : Hanjas Haruna
Stambuk : 105 84 11067 20

Makassar, 26 Agustus 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Rizki Yusliana Bakti ST.,MT

Titin Wahyuni S.Pd.,M.T

Mengetahui,

Ketua Program Studi Informatika



Muhammad A M Hayat, S.Kom., MT.

PRODI INM 1504 577

ABSTRAK

Skripsi merupakan salah satu persyaratan akhir dalam pendidikan akademis mahasiswa yang harus ditulis sesuai dengan bidang ilmunya. Proses awal dalam pembuatan skripsi adalah pengajuan judul, di mana mahasiswa harus memastikan bahwa judul yang diajukan tidak mirip dengan judul skripsi sebelumnya untuk menghindari *plagiarisme*. Deteksi kemiripan judul sangat penting untuk mencegah plagiarisme dan meningkatkan akurasi sistem rekomendasi dengan mengidentifikasi judul yang relevan. Dengan kemajuan teknologi informasi, tantangan seperti plagiarisme menjadi lebih kompleks, memerlukan sistem yang efisien untuk mengecek kemiripan judul. Salah satu metode yang digunakan adalah *Support Vector Machine* (SVM), sebuah teknik pembelajaran mesin yang efektif dalam klasifikasi teks, termasuk judul skripsi, berkat kemampuannya dalam menangani hubungan *non-linear* melalui kernel. Penggunaan *Support vector machine* dalam deteksi kemiripan judul bertujuan untuk memastikan keakuratan dan efisiensi dalam mengidentifikasi potensi *plagiarisme*. Kombinasi metode *Support Vector Machine* (SVM) dan *Natural Language Processing* (NLP), khususnya melalui vektorisasi menggunakan TF-IDF, telah terbukti efektif dalam mendeteksi kemiripan judul skripsi. Proses ini melibatkan pengumpulan data dari database judul_skripsi, diikuti dengan preprocessing yang meliputi tokenisasi, penghapusan kata-kata tidak relevan, dan stemming. Data yang bersih dan terstruktur ini kemudian diubah menjadi representasi numerik melalui TF-IDF, yang memberikan bobot lebih tinggi pada kata-kata yang spesifik namun signifikan. Model SVM yang dilatih dengan data ini berhasil mendeteksi kemiripan dengan tingkat akurasi yang memadai.

Kata kunci : Deteksi Kemiripan.,Judul Skripsi., SVM., NLP, Klasifikasi Teks., Pembelajaran Mesin

ABSTRACT

A thesis is one of the final requirements in a student's academic education that must be written in accordance with their field of study. The initial process in writing a thesis is submitting a title, where students must ensure that the title submitted is not similar to the title of a previous thesis to avoid plagiarism. Detecting title similarity is very important to prevent plagiarism and improve the accuracy of the recommendation system by identifying relevant titles. With the advancement of information technology, challenges such as plagiarism have become more complex, requiring an efficient system to check title similarity. One of the methods used is the Support Vector Machine (SVM), a machine learning technique that is effective in text classification, including thesis titles, thanks to its ability to handle non-linear relationships through kernels. The use of Support Vector Machines in title similarity detection aims to ensure accuracy and efficiency in identifying potential plagiarism. The combination of the Support Vector Machine (SVM) and Natural Language Processing (NLP) methods, especially through vectorization using TF-IDF, has proven effective in detecting the similarity of thesis titles. This process involves collecting data from the thesis_title database, followed by preprocessing including tokenization, removal of irrelevant words, and stemming. This clean and structured data is then transformed into a numeric representation through TF-IDF, which gives higher weight to specific but significant words. The SVM model trained with this data successfully detects similarities with a reasonable level of accuracy.

Keywords: *Similarity Detection., Thesis Title., SVM., NLP, Text Classification., Machine Learning*

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, Tuhan semesta alam, Dzat yang selalu melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian tugas akhir yang berjudul “Deteksi Tingkat Kemiripan Judul skripsi Prodi Informatika Dengan Menggunakan *Metode Support Vector Machine (SVM)* dan *Natural Language Processing*”. Tak lupa sholawat dan salam senantiasa tercurah atas junjungan kita Nabi Muhammad SAW, keluarga, sahabat, dan umatnya hingga akhir zaman. Aamiin. Tugas akhir ini disusun sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang Strata-1 pada program studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Tentunya dalam penyusunan tugas akhir ini tak lepas dari bantuan dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Maka pada kesempatan kali ini, penulis bermaksud untuk mengucapkan terima kasih dan penghargaan kepada:

1. Terima kasih untuk panutanku bapak Haruna. Beliau memang tidak sempat merasakan pendidikan sampai bangku perkuliahan, namun beliau mampu mendidik penulis, memotivasi serta memberi dukungan hingga penulis mampu menyelesaikan studinya sampai sarjana
2. Pintu surgaku, ibunda Hasni. Beliau sangat berperan penting dalam menyelesaikan program study penulis, Beliau juga tidak sempat merasakan pendidikan sampai dibangku perkuliahan, namun semangat, rasa kasih sayangnya serta sujudnya selalu menjadi doa untuk kesuksesan anak anaknya
3. Bapak Dr. Ir. H. Abd. Rakhim Nanda. S.T., M.T., IPU, Sebagai Rektor Perguruan Tinggi Universitas Muhammadiyah Makassar.
4. Ibu Dr. Ir. Hj. Nurnawaty, ST.MT., IPM. Sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
5. Bapak Muhyiddin A M Hayat, S.Kom, M.T. Selaku Ketua Program Studi Informatika Universitas Muhammadiyah Makassar

6. Ibu Rizki Yusliana Bakti S.T., M.T., dan Ibu Titin Wahyuni, S.pd., M.T selaku Pembimbing I dan Pembimbing II, atas kesabarannya dalam memberikan arahan, motivasi, bantuan serta bersedia meluangkan waktu dalam membimbing penulis selama proses penelitian.
7. Segenap Dosen dan Staf Informatika Fakultas Teknik Universitas Muhammadiyah Makassar yang telah membantu dan memberikan banyak ilmu serta dukungan selama masa perkuliahan.
8. Kepada teman seperjuangan saya yakni Imamul muttaqin, Muh syahdan, Wahyudi, Rajwan, Muh syaifullah, Haedir, Syahril dan alfian rahmat.

Terimakasih atas segala bentuk support, canda, tawa dan tangis air mata yang kita lalui bersama-sama dalam menempuh pendidikan di Universitas Muhammadiyah Makassar.

Skripsi ini merupakan sebuah karya saya yang dirangkai sejak awal 2024 dan diselesaikan pada akhir 2024. Skripsi ini merupakan saksi bisu atas perjuangan yang berdarah-darah ini; kegundahan, ketidakontikan, keterasingan, penghianatan, dan ketidakpastian hidup adalah suatu keniscayaan bagi saya, namun (masih) suatu kemungkinan bagi manusia pada umumnya.

Pada fase dewasa ini, kadangkala kita lupa akan arti kebaikan, kebenaran, kesetiaan, persahabatan, ketenangan, dan cinta, karena terlalu banyak penderitaan yang kita peroleh. Tetapi suatu hal yang harus diingat *“Bukan rasa sulitlah yang membuat kita takut, tapi rasa takutlah yang membuat kita sulit”*

Makassar, 21 Mei 2024

Hanjas Haruna

DAFTAR ISI

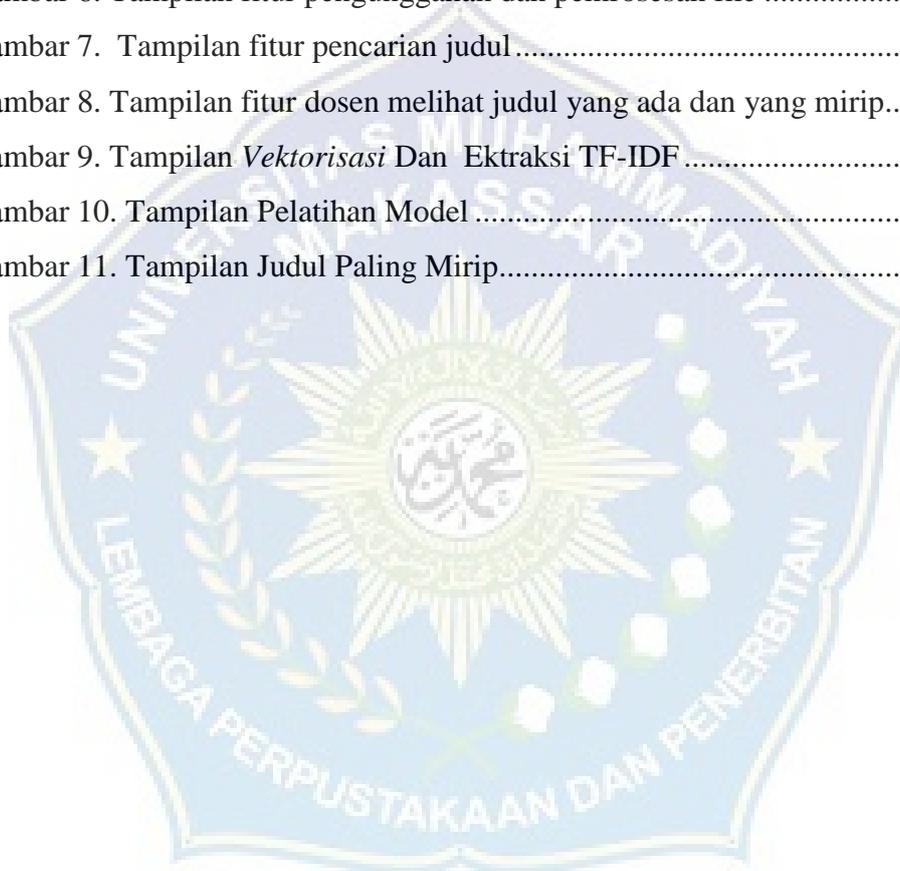
ABSTRAK	iv
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
DAFTAR ISTILAH	xiv
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian.....	2
D. Manfaat Penelitian.....	3
E. Ruang Lingkup Penelitian.....	3
F. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
A. Landasan Teori	5
B. Penelitian Terkait	9
C. Kerangka Pikir.....	12
BAB III METODE PENELITIAN.....	13
A. Tempat dan Waktu Penelitian	13
B. Alat dan Bahan	13
C. Perancangan Sistem.....	13
D. Teknik Pengujian Sistem.....	16
E. Teknik Analisis Data.....	17
BAB IV HASIL DAN PEMBAHASAN	18
A. Hasil Implementasi.....	18
B. Proses Akurasi SVM	28
BAB V PENUTUP.....	34
A. Kesimpulan.....	34
B. Saran.....	34

DAFTAR PUSTAKA	35
LAMPIRAN.....	38



DAFTAR GAMBAR

Gambar 1 Kerangka Berpikir	12
Gambar 2. <i>Flowchart</i> Sistem	14
Gambar 3. Diagram Use Case.....	15
Gambar 4. Tampilan fitur deteksi kemiripan judul.....	22
Gambar 5. Tampilan fitur deteksi kemiripan judul.....	26
Gambar 6. Tampilan fitur pengunggahan dan pemrosesan file	26
Gambar 7. Tampilan fitur pencarian judul.....	27
Gambar 8. Tampilan fitur dosen melihat judul yang ada dan yang mirip.....	28
Gambar 9. Tampilan <i>Vektorisasi</i> Dan Ekstraksi TF-IDF	29
Gambar 10. Tampilan Pelatihan Model	29
Gambar 11. Tampilan Judul Paling Mirip.....	30



DAFTAR TABEL

Tabel 1. Pengumpulan Data	18
Tabel 2. Data sebelum di preprocessing Data	19
Tabel 3. Data sesudah di preprocessing Data.....	20



DAFTAR ISTILAH

<i>Support Vector Machine (SVM)</i>	Algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. SVM bekerja dengan menemukan hyperplane terbaik yang memisahkan data ke dalam kelas yang berbeda.
<i>Natural Language Processing (NLP)</i>	Cabang kecerdasan buatan yang berfokus pada interaksi antara komputer dan bahasa manusia, termasuk pengolahan teks dan suara.
<i>Plagiarisme</i>	Tindakan meniru atau menjiplak karya orang lain dan mengakuinya sebagai karya sendiri, sering terjadi dalam konteks akademis.
<i>Website</i>	Kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet, berisi informasi dalam bentuk teks, gambar, video, atau audio.
<i>Database</i>	Struktur penyimpanan data yang terorganisir, digunakan untuk menyimpan dan mengelola data. Contoh sistem manajemen database adalah MySQL Server.
<i>Tokenisasi</i>	Proses memecah teks menjadi unit-

unit kecil seperti kata atau frasa.

Stop Words

Kata-kata umum yang sering dihapus dalam pemrosesan teks karena tidak memiliki makna penting, seperti "dan", "atau", "adalah".

Stemming

Proses mengurangi kata-kata ke bentuk dasar untuk mengurangi variasi kata yang berbeda dengan makna yang sama.

TF-IDF (Term Frequency-Inverse Document Frequency)

Teknik yang digunakan untuk mengubah teks menjadi representasi numerik berdasarkan frekuensi kemunculan kata dalam dokumen dan seberapa penting kata tersebut di seluruh kumpulan dokumen.

Cosine Similarity

Ukuran kemiripan antara dua vektor yang diukur berdasarkan sudut cosinus antara mereka. Sering digunakan dalam analisis teks untuk mengukur kemiripan dokumen.

Train-Test Split

Teknik dalam pembelajaran mesin di mana dataset dibagi menjadi set pelatihan dan set pengujian untuk melatih dan menguji model.

<i>Accuracy</i>	Ukuran kinerja model yang menunjukkan persentase prediksi yang benar dari keseluruhan prediksi yang dibuat oleh model.
<i>Precision</i>	Ukuran kinerja model yang menunjukkan persentase prediksi positif yang benar-benar positif.
<i>Recall</i>	Ukuran kinerja model yang menunjukkan persentase data positif yang benar-benar terdeteksi oleh model
<i>F1-Score</i>	Ukuran kinerja model yang merupakan harmoni rata-rata dari precision dan recall.
<i>Vektorisasi</i>	Proses mengubah teks menjadi vektor numerik yang dapat digunakan oleh algoritma pembelajaran mesin.
<i>Flowchart</i>	Diagram yang menggambarkan alur proses atau sistem menggunakan simbol-simbol standar.
<i>Use Case Diagram</i>	Diagram yang menggambarkan interaksi antara pengguna dan sistem, menunjukkan fungsi-fungsi utama yang disediakan oleh sistem.

BAB I

PENDAHULUAN

A. Latar Belakang

Skripsi merupakan salah satu persyaratan akhir pendidikan akademis mahasiswa yang ditulis dalam bentuk karya tulis ilmiah serta sesuai bidang ilmunya. Tahapan pertama dalam prosedur skripsi adalah proses pengajuan judul. Mahasiswa wajib mengajukan judul skripsi dengan tingkat kemiripan yang tidak sama dengan judul skripsi sebelumnya agar tidak terjadi *plagiarisme* judul skripsi (Desena & Solichin, 2021)

Deteksi tingkat kemiripan judul adalah proses untuk mengukur seberapa mirip dua atau lebih judul dalam konteks tertentu, seperti penelitian akademis. Pentingnya deteksi kemiripan judul yaitu sebagai pengecekan *plagiarisme* untuk membantu mengidentifikasi judul-judul yang mungkin menjiplak atau sangat mirip dengan karya yang lain serta meningkatkan akurasi sistem rekomendasi dengan menemukan judul yang relevan dan menarik bagi pengguna (Fitrianingsih et al., 2022)

Dengan adanya perkembangan teknologi informasi saat ini membuat pertukaran informasi semakin mudah tidak hanya membawa dampak positif saja. Dampak negatif dari penggunaan pun tidak sedikit, salah satunya sebagai sarana untuk melakukan tindakan *plagiarisme* atau kegiatan mencuri kegiatan hasil karya orang lain dan mengakuinya sebagai karya sendiri dalam proses penyelesaian tugas akhir sehingga diperlukannya sebuah sistem untuk melakukan pengecekan *plagiarisme* pada judul ataupun tugas akhir mahasiswa dengan cepat dan tepat menggunakan yang digunakan salah satunya adalah *Support Vector Machine (SVM)*(Putra Bayu Pratama et al., 2021)

Support Vector Machine (SVM) adalah metode pembelajaran mesin yang efektif untuk klasifikasi teks, termasuk judul penelitian. *Support Vector Machine (SVM)* bekerja dengan baik pada dataset yang

terstruktur dan memiliki kemampuan menangani hubungan non-linear melalui kernel trik. Penggunaan *Support Vector Machine (SVM)* dalam mendeteksi judul dapat mengotomatisasi klasifikasi dan memberikan hasil yang akurat, membantu akademisi dan peneliti dalam mengevaluasi dan memahami perkembangan dibidang teknik Informatika. (Desena & Solichin, 2021)

Natural Language Processing adalah cabang dari kecerdasan buatan (AI) yang berfokus pada interaksi antara komputer dan bahasa manusia. *Natural Language Processing (NLP)* memungkinkan komputer untuk membaca, memahami dan menghasilkan teks atau ucapan manusia dalam cara yang bermanfaat. (Fitrianingsih et al., 2022)

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang dapat disampaikan antara lain:

1. Seberapa efektif kombinasi metode *Support Vectore Machine* dan *Natural Language Processing* dalam mendeteksi tingkat kemiripan judul?
2. Bagaimana tingkat akurat yang diperoleh dalam mengukur keberhasilan metode *support vector machine*?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui seberapa efektif kombinasi metode Support Vector Machine (SVM) dan Natural Language Processing (NLP) dalam mendeteksi tingkat kemiripan judul.
2. Untuk mengukur tingkat akurasi yang diperoleh dalam mendeteksi kemiripan judul menggunakan metode Support Vector Machine (SVM).

D. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi penulis, Meningkatkan pengetahuan dan keterampilan penulis dalam bidang pembelajaran *support vector machine* dan *Natural Language Processing (NLP)*
2. Bagi Universitas, Mengurangi waktu dan usaha yang diperlukan untuk melakukan pengecekan manual terhadap kemiripan judul prodi, berkat adanya alat bantu otomatis.
3. Bagi pembaca, Membantu memahami bagaimana teknologi Support Vector Machine (SVM) dan *Natural Language Processing (NLP)* dapat diterapkan untuk mendeteksi kemiripan judul program studi.

E. Ruang Lingkup Penelitian

Penelitian ini dibuat ruang lingkup penelitian agar penelitian tetap pada jalur yang telah ditentukan di antaranya yaitu:

1. Hasil perbandingan judul penelitian tugas akhir relevan tidak nya judul dilihat dari nilai ukuran tertinggi.
2. Judul Skripsi yang digunakan hanya berektensi (*extension*), doc, txt, dan pdf untuk penelitian perbandingan judul penelitian tugas akhir.
3. Judul Skripsi yang digunakan untuk penelitian adalah judul yang di ambil dari prodi Informatika Universitas Muhammadiyah Makassar dan beberapa referensi lainnya.
4. Pencarian judul menggunakan kata kunci suatu *query*.

F. Sistematika Penulisan

Adapun sistematika penulisan dari penelitian ini adalah sebagai berikut:

BAB I : PENDAHULUAN, Bab ini meliputi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan pada penelitian yang dilakukan.

BAB II : TINJAUAN PUSTAKA, Bab ini menguraikan pembahasan tentang penelitian yang terkait, landasan teori, serta kerangka Pikir.

BAB III : METODE PENELITIAN, Bab ini memberikan gambaran tentang penelitian yang dilakukan dan penjelasan tentang apa yang dilakukan dalam penelitian ini.

BAB IV : HASIL DAN PEMBAHASAN, Bab ini berisi tentang hasil penelitian, pembahasan, dan hasil implementasi sistem.

BAB V : PENUTUP, Bab ini berisi kesimpulan dan saran berdasarkan hasil penelitian.

BAB II TINJAUAN PUSTAKA

A. Landasan Teori

Pada bagian ini berisikan teori–teori yang akan digunakan dalam penelitian ini. Peneliti meninjau teori–teori dari buku, jurnal, dan dari penelitian terdahulu guna mendapatkan informasi yang sesuai dengan penelitian ini

1. Skripsi

Skripsi atau tugas akhir, menurut Kamus Besar Bahasa Indonesia online, adalah karangan ilmiah yang harus ditulis oleh mahasiswa sebagai bagian dari persyaratan akademik. Menurut wikipedia Indonesia, “skripsi / tugas akhir” adalah istilah yang digunakan di Indonesia untuk menunjukkan suatu karya tulis ilmiah yang mencakup paparan tulisan atau fenomena dalam bidang ilmu tertentu dengan menggunakan prinsip-prinsip yang relevan (Nur Alam., 2019).

2. Pencarian judul

Menurut Kamus Besar Bahasa Indonesia online, mencari judul adalah proses, cara atau perbuatan mencari sesuatu. Sedangkan judul adalah nama buku atau bab yang menunjukkan isi atau maksudnya.(Nur Alam., 2019)

3. Website

Website adalah suatu halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan internet. (Hasanah *et al.*, 2020). *Website* atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur

internet.(Sanjaya et al., 2016) Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *browser* seperti *Mozilla Firefox*, *Google Chrome* atau yang lainnya. *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan link dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman kehalaman yang lain, baik diantara halaman yang disimpan dalam server maupun server diseluruh dunia (Desena & Solichin, 2021).

4. *Database*

Pengertian *Database* Berikut pengertian database menurut para ahli:

- a. *Database* adalah struktur penyimpanan data. Untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah database komputer, diperlukan sistem manajemen database seperti *MySql Server* (Kustiyaningsih, 2011)
- b. *Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari fieldI atau kolom. Struktur file yang menyusun sebuah database adalah *Data Recorddan Field* (Anhar, 2010)
- c. *Database* adalah sebagai kumpulan data dari penempatan tenaga kerja yang saling terkait dan mempengaruhi sesuai dengan tingkat kepentinganya sehingga data tersebut terintegrasi dan independence (Martono, 2009).

5. *Support Vector Machine (SVM)*

Ada banyak cara mengetahui tingkat kemiripan suatu data atau judul skripsi, diantaranya dengan menggunakan *metode Support Vector Machine (SVM)* adalah metode pembelajaran mesin yang efektif untuk klasifikasi teks termasuk judul penelitian. *metode Support Vector Machine (SVM)* bekerja dengan baik pada data set yang terstruktur dan memiliki kemampuan menangani hubungan non linear melalui kernel trik. Penggunaan *metode Support Vector*

Machine (SVM) dalam mendeteksi judul dapat mengotomatisasi klasifikasi dan memberikan hasil yang akurat membantu akademisi dan peneliti dalam mengevaluasi dan memahami perkembangan dibidang teknik informatika (Setiawan & Saprudin, 2023).

Support vector machine memanfaatkan optimasi dengan quadratic programming yang apabila digunakan untuk data berdimensi tinggi dan data dengan jumlah besar menjadi kurang efisien. Oleh karena itu para peneliti mengembangkan smoothing technique untuk mengubah optimasi yang terbatas menjadi optimasi yang tanpa batas menggunakan formulasi dari *Support vector machine* standar. Ide yang mendasari *Support vector machine* adalah mengubah formulasi primal *Support vector machine* menjadi problem optimasi yang smooth dan bebas kendala. Karena fungsi objektif dari problem optimasi *Support vector machine* tidak memiliki turunan kedua, maka digunakan smooth function untuk mendapatkan fungsi objektif yang dapat diperoleh turunannya. *Support vector machine* memiliki sifat matematik yang penting yaitu kecembungan yang kuat dan terdiferensialkan tak hingga, dengan menggantikan plus function dengan integral dari fungsi sigmoid neural network (Mutaqim, 2021).

6. *Natural Language Processing*

Natural language Processing adalah area integral dari ilmu komputer antara pembelajaran mesin dan linguistik yang mana komputasi digunakan secara luas (Patel & Prajapati, 2018), yang ditujukan untuk membuat komputer mengerti pernyataan yang ditulis menggunakan bahasa manusia. Pengolahan bahasa alami timbul untuk meringankan pekerjaan user dan untuk memenuhi keinginan terhubung dengan komputer bahasa alami. Karena semua pengguna mungkin tidak fasih dalam bahasa khusus mesin, *Natural Language Processing* melayani pengguna yang tidak memiliki cukup waktu

untuk mempelajari bahasa baru atau mendapatkan kesempurnaan di dalamnya (Processing & Matching, 2022)

Natural Language Processing (NLP) atau bisa disebut juga dengan *NLP* adalah fokus dari bidang ilmu computer dan linguistic yang dikenal sebagai pemrosesan bahasa alami. *Natural Language Processing (NLP)* biasa disebut dengan Komputasional Linguistik serta memiliki segmentasi tuturan (*segmentation of speech*) dan penandaan kelas kata (*part of speech tagging*) adalah beberapa pengembangannya. (Herwin 2019)

Saat ini, *natural language processing* banyak digunakan dalam aplikasi sehari-hari meliputi Siri dan *Google Assistens* sebagai asisten virtual secara pribadi. Didunia Teknologi Industri saat ini menggunakan *Natural Language Processing (NLP)* adalah Langkah yang sangat penting untuk mendapatkan keuntungan dalam kompetitif. *Natural Language Processing (NLP)* dapat membantu dalam berbagai bidang kehidupan dalam menganalisis nilai dari data yang tidak terstruktur. (Rumaisa et al. 2021)

Berdasarkan latar belakang masalah diatas maka dilakukan penelitian yang bertujuan untuk melakukan deteksi tingkat kemiripan judul skripsi dengan dua metode pemodelan data yaitu *Support vector machine* dan *Natural language Processing* metode tersebut digunakan dalam mendeteksi tingkat kemiripan judul skripsi di fakultas teknik Universitas Muhammadiyah Makassar kedua metode ini memiliki karakteristik yang berbeda dan dapat memberikan wawasan yang berharga bagi perguruan tinggi dalam meningkatkan tingkat kelulusan mahasiswa. Berdasarkan penelitian sebelumnya penelitian ini juga mencoba memperoleh pemahaman yang lebih baik tentang kinerja dari kedua metode ini.

B. Penelitian Terkait

1. Nopi Fitriainingsih, “Deteksi Tingkat Kemiripan Judul Menggunakan *Algoritma Oliver* Pada Sistem Informasi Pengajuan Skripsi”, (2022).
Pada penelitian sebelumnya yang dilakukan oleh Nopi Fitriainingsih, yang berjudul “Deteksi Tingkat Kemiripan Judul Menggunakan *Algoritma Oliver* Pada Sistem Informasi Pengajuan Skripsi” Dari hasil pengujian dalam proses pengecekan kemiripan judul skripsi dengan data testing sebanyak 10 data judul skripsi yang diusulkan mahasiswa dan data training sebanyak 217 judul skripsi menghasilkan persentase tingkat kemiripan judul sesuai dengan ketentuan yang telah ditetapkan, didapatkan 2 judul skripsi ditolak karena persentase tingkat kemiripan judul sebesar 70.06% dan 62,89% (lebih besar dari 60%), dan 8 judul skripsi lainnya diterima dengan tingkat kemiripan judul diantaranya sebesar 32.07%, 30.74%, 26.31%, 26.31%, 29.15%, 35.12% dan 37.45% (kurang dari atau sama dengan 60%). (Fitriainingsih et al., 2022)
2. Dina Suci Islamiyati, “Penerapan *Algoritma Knuth-Morris-Pratt* dalam Mendeteksi Tingkat Kemiripan Judul Skripsi Berbasis Web”, (2022).
Pada penelitian sebelumnya dilakukan oleh Dina Suci Islamiyati, yang berjudul” Penerapan *Algoritma Knuth-Morris-Pratt* dalam Mendeteksi Tingkat Kemiripan Judul Skripsi Berbasis Web” Dan Telah didapatkan hasil analisa dari judul yang telah diajukan dengan proses string matching antara string dengan pattern yang mana sistem menampilkan hasil analisa berupa jumlah kemiripan kata dan presentasinya. (Islamiyati et al., 2022)
3. Imam Ahmad, “*Software Development Dengan Extreme Programming (XP)* Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android”, (2020). Penelitian sebelumnya dilakukan oleh Imam Ahmad yang berjudul “*Software Development Dengan Extreme Programming (XP)* Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis

Android” Penelitian ini menghasilkan aplikasi yang mampu menentukan tingkat kemiripan usulan judul skripsi mahasiswa. Pendekatan pencocokan *string Boyer Moore* yang digunakan mampu melakukan pencocokan dengan menemukan pattern yang sama persis dari satu teks. Penggunaan pendekatan *Boyer Moore* dapat melakukan pencocokan pattern dan text dimulai dari karakter paling kanan terlebih dahulu kemudian ke kiri terhadap teks pada judul skripsi. Sedangkan metode XP telah menghasilkan *software* yang sesuai dengan kebutuhan. Hal ini didukung dari hasil black box testing yang menunjukkan bahwa semua fitur dari aplikasi dapat berfungsi dengan baik, artinya aplikasi dapat digunakan.

(Ahmad et al., 2020)

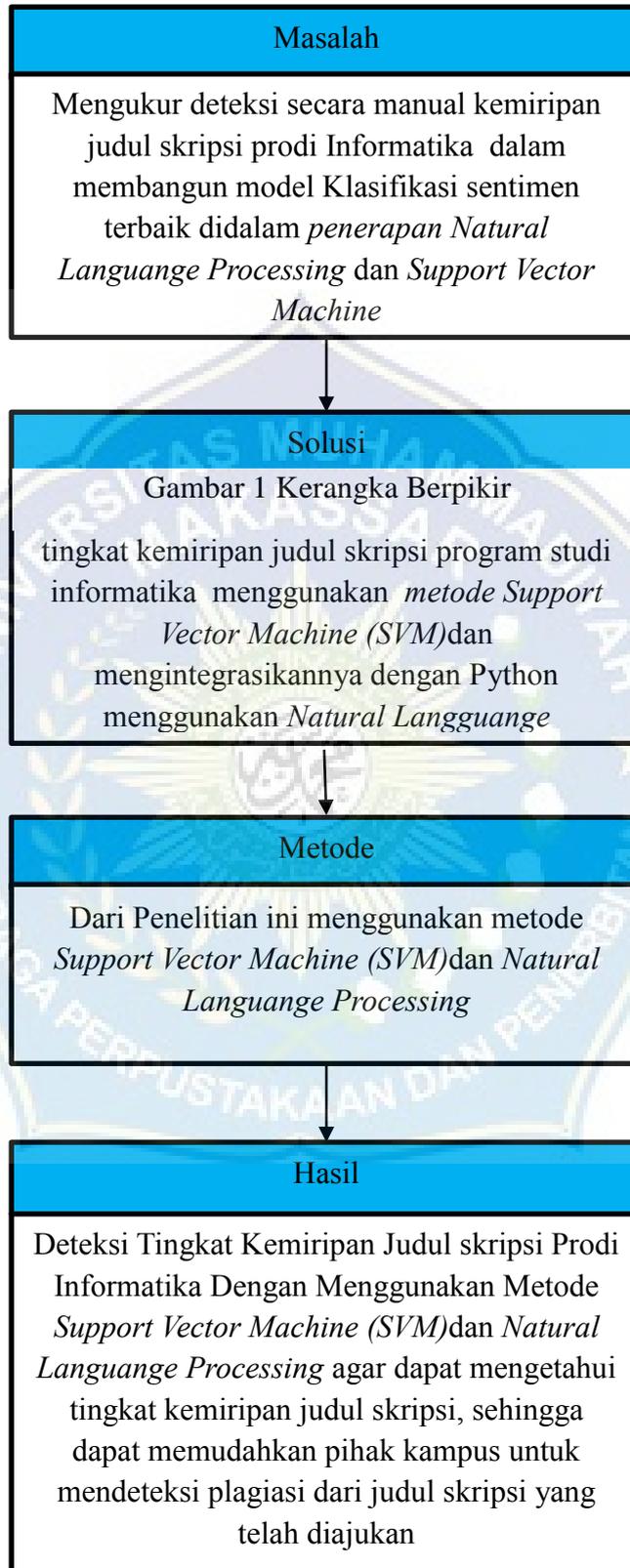
4. Lilik Sugiarto, “Analisa Algoritma *String Matching* dan *Winnowing* untuk deteksi kemiripan judul tugas akhir perguruan tinggi”, (2020). Penelitian sebelumnya dilakukan oleh Lilik Sugiarto, yang berjudul “Analisa Algoritma *String Matching* dan *Winnowing* untuk deteksi kemiripan judul tugas akhir perguruan tinggi” Berdasarkan perhitungan tingkat kemiripan kedua judul tugas akhir yang diterjemahkan ke dalam nilai *fingerprint* menggunakan *algoritma winnowing* maka kemiripan antara judul 1 dengan judul2 memiliki tingkat kemiripan sebesar 50%.(Sugiarto et al., 2021)
5. Nur Alamsyah, “Deteksi *Plagiarisme* tingkat kemiripan judul skripsi pada fakultas teknologi informasi menggunakan algoritma *Winnowing*”, (2019). Penelitian sebelumnya dilakukan oleh Nur Alamsyah yang berjudul “Deteksi *Plagiarisme* tingkat kemiripan judul skripsi pada fakultas teknologi informasi menggunakan algoritma *Winnowing*” hasil Pengujian diatas dalam menentukan tingkat *plagiarisme* dengan menggunakan n-gram, window, bilangan prima menghasilkan 73.86 % tingkat *plagiarisme* tinggi artinya mendekati *plagiarisme* dengan n-gram = 3 , window = 3 dan bilangan prima = 2 sedangkan dengan menggunakan n-gram = 3, window = 3 dan

bilangan prima = 2 menghasilkan 19.82 % artinya hanya mempunyai tingkat *plagiarisme* ringan. (Alamsyah & Rasyidan, 2019)

6. Nur Alam “ Implementasi Perbaikan Ejaan Pada Web Semantik Pencarian Judul Skripsi Ruang Baca Fakultas Teknik yang Menggunakan Algoritma *JARO-WINKLER DISTANCE*”(2019)
Perbaikan ejaan pada pencarian informasi judul skripsi dengan penerapan algoritma jaro-winkler distance pada web semantik ruang baca mampu membantu sebagai dasar pengetahuannya untuk menemukan judul skripsi yang diinginkan oleh mahasiswa yang terdapat pada ruang baca fakultas teknik khususnya. Pada hasil pengujian white box ditemukan bahwa alur sistem dari aplikasi telah bekerja sesuai dengan apa yang peneliti harapkan.
7. Nur Anniza Haris “”



C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Adapun tempat dan waktu penelitian adalah sebagai berikut :

1. Tempat Penelitian

Tempat pelaksanaan penelitian ini adalah di lingkungan kampus tepatnya fakultas teknik prodi informatika universitas muhammadiyah makassar.

2. Waktu Penelitian

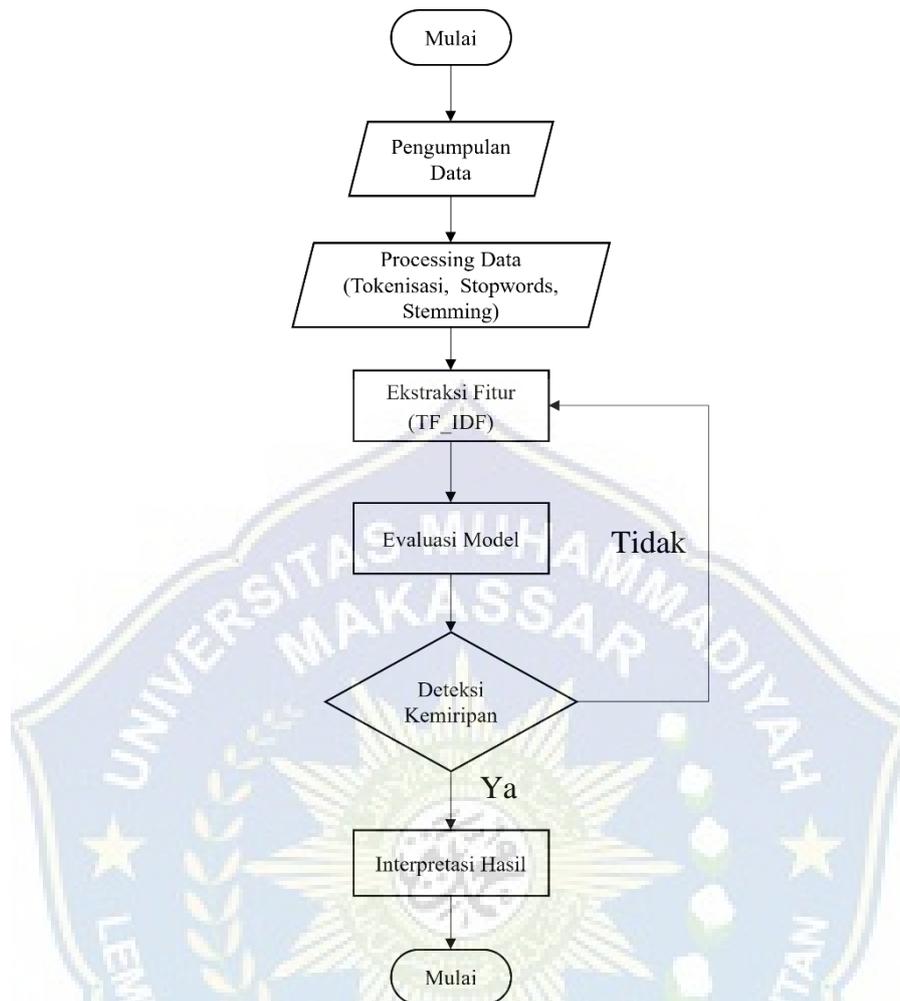
Waktu yang digunakan peneliti untuk penelitian ini di laksanakan dalam kurang lebih dua bulan, dimulai dari bulan Juni sampai Agustus 2024.

B. Alat dan Bahan

Alat yang digunakan dalam penelitian ini adalah seperangkat komputer/ laptop yang dilengkapi perangkat keras dan perangkat lunak pendukung. Sedangkan bahan yang digunakan adalah data yang diambil dari prodi informatika Universitas Muhammadiyah Makassar dan dari *google scholar*

C. Perancangan Sistem

Untuk mempermudah proses perancangan, maka peneliti membuat perancangan sistem dengan menggunakan *Flowchart* seperti pada gambar dibawah ini :



1. Peng

Gambar 2. Flowchart Sistem

umpulan Data

- a. Kumpulkan dataset yang berisi judul-judul skripsi program studi informatika dari berbagai institusi.
- b. Pastikan data sudah di-label jika ada informasi tentang tingkat kemiripan untuk digunakan sebagai data pelatihan.

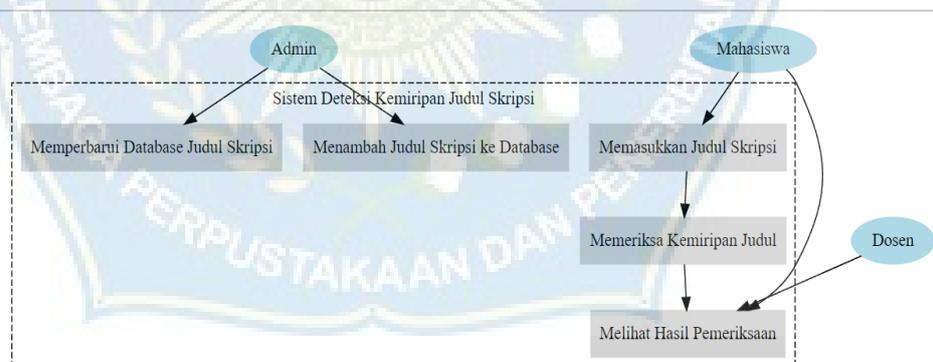
2. Processing Data

Adapun langkah *Processing Data* adalah :

- a. *Tokenisasi* : Memecah teks judul menjadi kata-kata individu.
- b. *Stop Words* : Menghapus kata-kata umum yang tidak memiliki makna penting (seperti "dan", "atau", "adalah").
- c. *Stemming*: Mengurangi kata-kata ke bentuk dasar

3. Ekstraksi fitur
 - a. TF-IDF: Gunakan scikit-learn untuk mengubah teks menjadi vektor fitur.
4. Evaluasi Model
 - a. **Validation Set:** Data yang digunakan untuk mengevaluasi model.
 - b. **Metrik Evaluasi:** Menggunakan akurasi, presisi, recall, dan F1-score untuk menilai kinerja model.
5. Deteksi Kemiripan

Sistem yang menggunakan model terlatih untuk mendeteksi dan mengukur tingkat kemiripan antara judul prodi.
6. Interpretasi Hasil
 - a. Hasil Deteksi: Menampilkan tingkat kemiripan antara judul prodi yang diinput dengan judul prodi yang ada.
 - b. Laporan: Menyediakan laporan atau ringkasan hasil deteksi kemiripan.



Gambar 3. Diagram Use Case

Penjelasan Diagram

1. Mahasiswa: Memasukkan judul skripsi dan melihat hasil pemeriksaan kemiripan.

2. Dosen: Melihat hasil pemeriksaan kemiripan judul skripsi mahasiswa.
3. Admin: Menambah dan memperbarui judul skripsi dalam database.
4. Sistem: Memeriksa kemiripan judul dan menampilkan hasilnya.

D. Teknik Pengujian Sistem

Pengujian sistem akan dilakukan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan analisa dan tujuan dari dibangunnya sistem ini. Untuk mengetahui hasil dari sistem ini apakah sudah sesuai dengan yang diharapkan, dilakukan penghitungan *similarity* dengan inputan judul yang diuji.

Rencana Pengujian Dalam tahapan rencana pengujian akan dilakukan sebagai berikut:

- a. Untuk tahapan pengujian sistem akan dilakukan menggunakan 10 judul yang akan dibandingkan dengan 760 judul yang tersimpan pada *Database* jurnal yang digunakan jurnal ISSN yang di dapat dari *google scholar* pada Universitas Budi Luhur. (Desena & Solichin, 2021)
- b. Dalam tahapan pengujian pertama dilakukan dengan cara membandingkan judul penelitian yang sama dengan judul penelitian yang tersimpan dalam database maka terlihat tingkat kemiripannya bobot cosine similarity berapa persen. (Mawanta et al., 2021)
- c. Dalam tahapan pengujian kedua inputan awal dengan menggunakan judul yang sama dengan konsep mengubah susunan kata atau kombinasi acak kata dalam judul penelitian, maka tingkat nilai cosine similarity hasil akan dibandingkan dengan judul tanpa mengubah susunan kata apakah bobot nilai cosine similarity akan berubah apa akan tetap sama. (Sheva Ricko Apriansyah, 2023)

Dalam pengujian ketiga dilakukan dengan cara penghapusan tiga kata di judul yang sama maka akan dilihat nilai tingkat kemiripan *cosine*

similarity berapa persen apakah nilai *cosine similarity* apakah akan turun bobot kemiripannya.(Lestari & Sumarlinda, 2023)

E. Teknik Analisis Data

Analisis data adalah proses pencarian dan pengaturan sistematis hasil data ulasan dan bahan-bahan yang dikumpulkan untuk meningkatkan pemahaman terhadap semua hal yang dikumpulkan dan memungkinkan menyajikan apa yang ditemukan. Analisis data yang digunakan dalam penelitian adalah melalui tahapan sebagai berikut :

a. Pengumpulan Data

Kumpulkan dataset yang berisi judul-judul program studi informatika dari berbagai institusi. Pastikan data sudah di-label jika ada informasi tentang tingkat kemiripan untuk digunakan sebagai data pelatihan.(Setiawan & Saprudin, 2023)

b. Pembangunan Model

Gunakan *Support Vector Machine (SVM)* sebagai algoritma klasifikasi, bagilah data menjadi set pelatihan dan set pengujian untuk evaluasi model dan Latih model *metode Support Vector Machine (SVM)* menggunakan data pelatihan yang telah diproses. Gunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk menilai performa model.(Ahmad et al., 2020)

c. Deteksi Kemiripan

Terapkan model terlatih untuk mengukur kemiripan judul-judul baru, Gunakan *threshold* tertentu pada skor probabilitas dari model *metode Support Vector Machine (SVM)* untuk menentukan apakah dua judul dianggap mirip atau tidak.(Sanjaya et al., 2016)

d. Interpretasi Hasil

Visualisasikan hasil kemiripan dengan menggunakan diagram atau heatmap untuk memperlihatkan pasangan judul yang mirip. Interpretasikan hasil untuk memahami pola kemiripan yang ditemukan oleh model.(Hasanah et al., 2020)

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Implementasi

1. Pengumpulan Data

Data dikumpulkan dari database judul_skripsi, yang terdiri dari nomor dan judul, dengan total 760 judul skripsi. Proses ini melibatkan ekstraksi data dari sistem basis data MySQL, memastikan bahwa setiap entri berisi informasi yang relevan dan akurat untuk keperluan analisis lebih lanjut. Data ini nantinya akan digunakan sebagai input dalam berbagai tahap *preprocessing* dan pelatihan model untuk mendeteksi kemiripan judul skripsi.

Tabel 1. Pengumpulan Data

No	Judul Skripsi
1	DESAIN DAN IMPLEMENTASI TIME BASED ONE TIME PASSWORD UNTUK MENINGKATKAN KEAMANAN OTENTIKASI PADA WEBSITE TOP UP GAME
2	IMPLEMENTASI AUGMENTED REALITY PADA PEMBUATAN GAME MOBILE DALAM MEMPERKENALKAN SEJARAH KEMERDEKAAN REPUBLIK INDONESIA
3	PEMBUATAN VERIFIKASI SERTIFIKAT DIGITAL SEBAGAI BUKTI KEABSAHAN MENGGUNAKAN ALGORITMA STEGANOGRAFI DENGAN METODE LEAST SIGNIFICANT BIT INSERTION (LSR)
4	PENERAPAN FIKHI SHALAT DENGAN METODE TARJIH PADA APLIKASI MOBILE DAN ANIMASI 3D
5	PENERAPAN WEB SEMANTIK UNTUK PENCARIAN JUDUL SKRIPSI PADA RUANG BACA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH MAKASSAR
6	PENCARIAN BUKU YANG RELEVAN DENGAN JUDUL SKRIPSI PADA RUANG BACA FAKULTAS TEKNIK UNISMUH MENGGUNAKAN ALGORITMA NAÏVE BAYES CLASSIFIER
7	PENERAPAN METODE ROULETE WHEEL SECARA PARALEL UNTUK PENJADWALAN PERKULIAHAN PADA FAKULTAS

TEKNIK UNISMUH MAKASSAR

- 8 IMPLEMENTASI EUCLIDEAN DISTANCE SEBAGAI INFORMASI PENYEWAAN PROPERTI DI BIDANG BANGUNAN BERBASIS ANDROID
 - 9 PEMBUATAN APLIKASI MEDIA PEMBELAJARAN BERBASIS GAME BASED LEARNING PADA MATA PELAJARAN MATEMATIKA KELAS 7 SMPN 8 KOLAKA UTARA
 - 10 PEMBUATAN APLIKASI MEDIA PEMBELAJARAN DALAM PENGENALAN HURUF HIJAIYAH UNTUK ANAK USIA DINI PADA PLATFORM ANDROID
-

2. Preprocessing Teks/Data

Preprocessing data adalah tahap awal dalam pemrosesan data untuk membersihkan dan mengubah data mentah menjadi format yang lebih konsisten dan siap dianalisis. Salah satu langkah penting dalam *preprocessing* adalah mengubah semua teks menjadi huruf kecil. Ini dilakukan untuk menghindari duplikasi, meningkatkan akurasi analisis, dan membuat proses analisis lebih sederhana dan efisien, proses ini masuk kedalam teknik *Natural Language Processing (NLP)*

Tabel 2. Data sebelum di preprocessing Data

No	Judul Skripsi
1	ANALISIS SENTIMEN APLIKASI REKSADANA BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES
2	IMPLEMENTASI ALGORITMA SHUFFLE RANDOM UNTUK AWARENES BOSS PADA GAME EDUKASI BUDAYA ANDROID
3	IMPLEMENTASI ALGORITMA EUCLIDEAN DISTANCE DAN A-STAR PADA GAME EDUKASI MATEMATIKA BERBASIS ANDROID
4	IMPLEMENTASI PERBAIKAN EJAAN PADA WEB SEMANTIK PENCARIAN JUDUL SKRIPSI RUANG BACA FAKULTAS TEKNIK DENGAN MENGGUNAKAN ALGORITMA JARO-WINKLER

5	SISTEM PAKAR PENDETEKSIAN MULUT DAN KUKU PADA SAPI BERBASIS CITRA DI SENTRA TERNAK BANTAENG
6	APLIKASI MONITORING PERTUMBUHAN TANAMAN PADI DI DESA BONTORAMBA KECAMATAN PALLANGGA
7	PERBANDINGAN FUNGSI AKTIVASI HIDDEN LAYER DAN OUTPUT LAYER STUDI KASUS SISTEM PAKAR PENYAKIT AYAM
8	SISTEM PAKAR PENENTUAN JENIS PENYAKIT PADA TANAMAN PISANG DAN CARA MENGATASINYA MELALUI DETEKSI CITRA MENGGUNAKAN METODE CONVOLUTION NEURAL NETWORK
9	SISTEM PENDETEKSI PENGENDARA MOTOR YANG BERLAWANAN ARAH MENGGUNAKAN ALGORITMA YOLOV8
10	PERANCANGAN HOTEL AND CONVENTION CEBTER DENGAN PENDEKATAN ARSITEKTUR
11	PREDIKSI TINGKAT KELULUSAN MAHASISWA MENGGUNAKAN K-MEANS PADA PROGRAM STUDI INFORMATIKA UNISMUH MAKASSAR
12	ANALISIS DAN PEMETAAN AMIL ZAKAT BERDASARKAN DATA DI KECAMATAN RAPPOCINI LAZISMU MAKASSAR
13	APLIKASI REQUEST JADWAL KHUSUS BAGI DOSEN BERHALANGAN PADA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH MAKASSAR
14	PENGARUH OPTIMIZER ADAM DAN SGD TERHADAP ALGORITMA CNN STUDI KASUS SISTEM PAKAR RUMPUT LAUT

Tabel 3. Data sesudah di preprocessing Data

No	Judul Skripsi
1	analisis sentimen aplikasi reksadana berdasarkan ulasan di play store

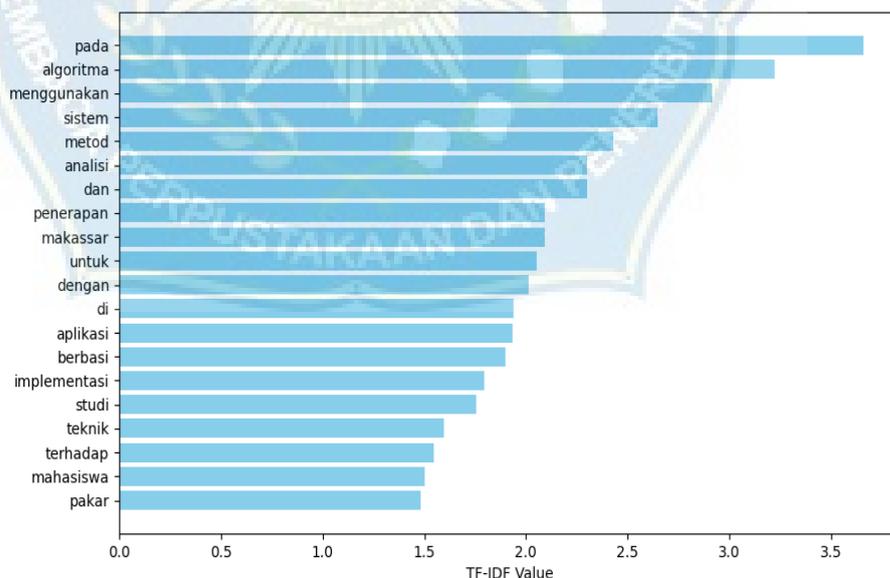
-
- 1 menggunakan metode naive bayes
 - 2 implementasi algoritma shuffle random untuk awareness boss pada game edukasi budaya android
 - 3 implementasi algoritma euclidean distance dan a-star pada game edukasi matematika berbasis android
 - 4 implementasi perbaikan ejaan pada web semantik pencarian judul skripsi ruang baca fakultas teknik dengan menggunakan algoritma jaro-winkler
 - 5 sistem pakar pendeteksian mulut dan kuku pada sapi berbasis citra di sentra ternak bantaeng
 - 6 aplikasi monitoring pertumbuhan tanaman padi di desa bontoramba kecamatan pallangga
 - 7 perbandingan fungsi aktivasi hidden layer dan output layer studi kasus sistem pakar penyakit ayam
 - 8 sistem pakar penentuan jenis penyakit pada tanaman pisang dan cara mengatasinya melalui deteksi citra menggunakan metode convolution neural network
 - 9 sistem pendeteksi pengendaraan motor yang berlawanan arah menggunakan algoritma yolov8
 - 10 perancangan hotel and convention center dengan pendekatan arsitektur
 - 11 prediksi tingkat kelulusan mahasiswa menggunakan k-means pada program studi informatika unismuh makassar
 - 12 analisis dan pemetaan amil zakat berdasarkan data di kecamatan rappocini lazismu makassar
 - 13 aplikasi request jadwal khusus bagi dosen berhalangan pada fakultas teknik universitas muhammadiyah makassar
 - 14 pengaruh optimizer adam dan sgd terhadap algoritma cnn studi kasus sistem pakar rumput laut
-

3. *Vektorisasi* dan Ekstraksi TF-IDF

Setelah teks melalui tahap *preprocessing*, tahap berikutnya adalah vektorisasi menggunakan metode *TF-IDF* (*Term Frequency-Inverse*

Document Frequency). *TF-IDF* adalah teknik yang mengubah teks menjadi representasi numerik. TF mengukur seberapa sering suatu kata muncul dalam dokumen, sementara IDF mengukur seberapa penting kata tersebut di seluruh kumpulan dokumen. Dengan menggabungkan kedua ukuran ini, *TF-IDF* memberikan bobot yang lebih tinggi untuk kata-kata yang sering muncul dalam dokumen tertentu tetapi jarang di seluruh kumpulan dokumen.

Ekstraksi fitur dilakukan dengan menggunakan *Tfidf Vectorizer* untuk mengubah teks menjadi vektor fitur. Proses ini mengubah kata-kata dalam judul skripsi menjadi angka yang menunjukkan seberapa penting kata tersebut dalam satu dokumen dan dalam seluruh koleksi dokumen. Hasil dari proses ini adalah vektor fitur yang digunakan oleh model pembelajaran mesin untuk menganalisis dan mendeteksi kemiripan antara judul-judul skripsi. Diagram di atas menunjukkan nilai *TF-IDF* dari berbagai kata kunci yang dihasilkan dari dataset, memberikan gambaran tentang kata-kata yang paling penting dalam kumpulan data.



Gambar 4. Tampilan fitur deteksi kemiripan judul.

Gambar tersebut menunjukkan grafik batang horizontal yang memvisualisasikan nilai **TF-IDF** dari beberapa kata dalam sebuah

dokumen. Kata "pada" memiliki nilai TF-IDF tertinggi, diikuti oleh "algoritma", "menggunakan", dan "sistem", yang menandakan bahwa kata-kata ini sangat penting dan relevan dalam konteks dokumen tersebut. TF-IDF mengukur pentingnya kata dengan mempertimbangkan frekuensi kemunculannya dalam dokumen serta kelangkaannya dalam kumpulan dokumen lainnya. Oleh karena itu, kata-kata dengan nilai TF-IDF tinggi cenderung menjadi fokus utama atau topik kunci dalam dokumen tersebut.

4. Menghitung Kemiripan

Fungsi `deteksi_kemiripan` melakukan perhitungan kemiripan antara judul baru dengan judul-judul yang ada dalam database menggunakan *Cosine Similarity*. *Cosine Similarity* mengukur kemiripan antara dua vektor (judul skripsi) dengan menghitung sudut kosinus di antara keduanya.

- **Vektorisasi Judul Baru:** Judul baru yang dimasukkan oleh pengguna juga diproses dan diubah menjadi vektor menggunakan TF-IDF.
- **Perhitungan *Cosine Similarity*:** Kemiripan antara vektor judul baru dan setiap vektor judul dalam database dihitung.
- **Pemilihan Judul Paling Mirip:** Dari hasil perhitungan kemiripan, sistem mencari nilai kemiripan tertinggi dan menampilkan judul yang paling mirip.

```
def deteksi_kemiripan(judul_baru):  
    judul_baru_preprocessed =  
preprocess_text(judul_baru)  
    judul_baru_vec =  
tfidf.transform([judul_baru_preprocessed])  
    similarities = cosine_similarity(judul_baru_vec,  
X)  
    max_similarity_idx = similarities.argmax()
```

```

        max_similarity = similarities[0,
max_similarity_idx]
        most_similar_title =
judul.iloc[max_similarity_idx]
        threshold = 0.5
        is_similar = max_similarity >= threshold
        return is_similar, max_similarity,
most_similar_title

```

Hasil dari proses di atas adalah sebuah nilai kemiripan (similarity score). Sistem kemudian memutuskan apakah judul baru tersebut mirip dengan salah satu judul yang ada berdasarkan threshold (batas ambang) yang ditentukan (dalam contoh di atas, threshold-nya adalah 0.5).

5. Pelatihan Model

Setelah *preprocessing* dan vektorisasi, langkah berikutnya adalah melatih model klasifikasi untuk mendeteksi kemiripan. Model SVM dilatih dengan menggunakan data yang telah diubah menjadi vektor TF-IDF. Data ini kemudian dibagi menjadi set pelatihan dan set pengujian untuk mengevaluasi performa model.

Membagi data menjadi set pelatihan dan pengujian

```

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

```

Melatih model Support Vector Machine

```

svm = SVC(kernel='linear', probability=True)

```

```

svm.fit(X_train, y_train)

```

Dengan membagi dataset menjadi dua subset: satu untuk pelatihan (training set) dan satu lagi untuk pengujian (test set) menggunakan fungsi `train_test_split` dari `sklearn.model_selection`. Dataset `X`, yang merupakan representasi vektor *TF-IDF* dari judul skripsi, dan label `y` yang berupa array dummy dengan dua kelas, digunakan sebagai input fungsi ini. Parameter `test_size=0.3` menentukan bahwa 30% dari data akan digunakan untuk pengujian, sementara 70% sisanya digunakan untuk pelatihan. Parameter `random_state=42` memastikan bahwa pembagian data bersifat deterministik, sehingga setiap kali kode dijalankan, hasil pembagian data akan selalu sama, membantu dalam reproduktibilitas hasil. Output dari fungsi ini adalah `X_train` dan `y_train` yang merupakan subset data untuk melatih model, serta `X_test` dan `y_test` yang digunakan untuk menguji model setelah dilatih.

Setelah membagi dataset, model *Support Vector Machine* (SVM) (SVM) didefinisikan menggunakan kelas `SVC` dari `sklearn.svm`. Dalam definisi ini, kernel linear dipilih (`kernel='linear'`) karena cocok untuk dataset yang dapat dipisahkan secara linier. Selain itu, penghitungan probabilitas diaktifkan dengan mengatur parameter `probability=True`, memungkinkan penggunaan metode `predict_proba` untuk mendapatkan probabilitas kelas. Model SVM kemudian dilatih menggunakan metode `fit` dengan data pelatihan `X_train` dan `y_train`. Proses pelatihan ini melibatkan mencari hyperplane terbaik yang memisahkan kelas-kelas dalam data pelatihan, dengan tujuan memaksimalkan margin antara kelas yang berbeda.

6. Proses Utama Aplikasi

a. Proses pemasukan judul

memungkinkan pengguna untuk memasukkan judul skripsi baru dan melihat seberapa mirip judul tersebut dengan judul-judul skripsi yang sudah ada. Aplikasi akan memberikan skor yang menunjukkan seberapa mirip judul baru tersebut dengan yang ada dalam database.

Deteksi Kemiripan Judul Skripsi

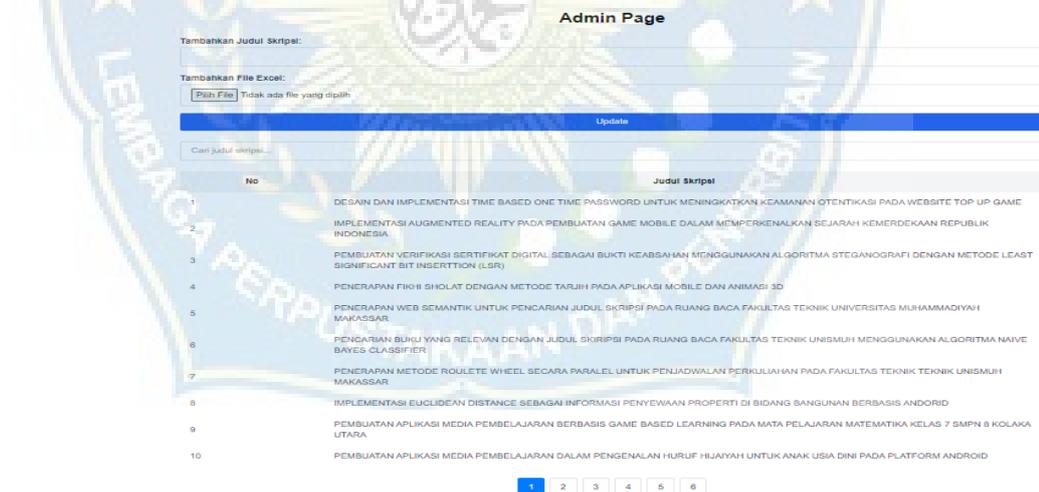
Masukkan Judul Skripsi:

Deteksi

Gambar 5. Tampilan fitur deteksi kemiripan judul.

b. Proses Pengunggahan dan Pemrosesan File

Aplikasi dapat mengunggah file Excel yang berisi daftar judul skripsi dan memprosesnya secara otomatis. Setelah file diunggah, sistem akan mengekstrak data dari file tersebut dan menambahkannya ke database. Proses ini sangat membantu dalam mempercepat dan memudahkan input data, sehingga pengguna tidak perlu memasukkan judul skripsi satu per satu secara manual.



The screenshot shows the 'Admin Page' interface. At the top, there are two input fields: 'Tambahkan Judul skripsi:' and 'Tambahkan File Excel:'. Below the 'Tambahkan File Excel:' field is a 'Pilih File' button and a message 'Tidak ada file yang dipilih'. A 'Uptodate' button is located below these fields. Below the buttons is a search bar labeled 'Cari judul skripsi...'. At the bottom, there is a table with two columns: 'No' and 'Judul Skripsi'. The table contains 10 rows of data, with the first row highlighted in blue. Below the table is a pagination bar with buttons for pages 1, 2, 3, 4, 5, and 6.

No	Judul Skripsi
1	DESAIN DAN IMPLEMENTASI TIME BASED ONE TIME PASSWORD UNTUK MENINGKATKAN KEAMANAN OTENTIKASI PADA WEBSITE TOP UP GAME
2	IMPLEMENTASI AUGMENTED REALITY PADA PEMBUATAN GAME MOBILE DALAM MEMPERKENALKAN SEJARAH KEMERDEKAAN REPUBLIK INDONESIA
3	PEMBUATAN VERIFIKASI SERTIFIKAT DIGITAL SEBAGAI BUKTI KEABSAHAN MENGGUNAKAN ALGORITMA STEGANOGRAFI DENGAN METODE LEAST SIGNIFICANT BIT INSERTION (LSR)
4	PENERAPAN FIKH SHOLAT DENGAN METODE TARJUH PADA APLIKASI MOBILE DAN ANIMASI 3D
5	PENERAPAN WEB SEMANTIK UNTUK PENCARIAN JUDUL SKRIPSI PADA RUANG BACA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH MAKASSAR
6	PENCARIAN BUKU YANG RELEVAN DENGAN JUDUL SKRIPSI PADA RUANG BACA FAKULTAS TEKNIK UNISMUH MENGGUNAKAN ALGORITMA NAIVE BAYES CLASSIFIER
7	PENERAPAN METODE ROULETTE WHEEL SECARA PARALEL UNTUK PENJADWALAN PERKULIAHAN PADA FAKULTAS TEKNIK TEKNIK UNISMUH MAKASSAR
8	IMPLEMENTASI EUCLIDEAN DISTANCE SEBAGAI INFORMASI PENYEWAAN PROPERTI DI BIDANG BANGUNAN BERBASIS ANDROID
9	PEMBUATAN APLIKASI MEDIA PEMBELAJARAN BERBASIS GAME BASED LEARNING PADA MATA PELAJARAN MATEMATIKA KELAS 7 SMPN 8 KOLAKA UTARA
10	PEMBUATAN APLIKASI MEDIA PEMBELAJARAN DALAM PENGENALAN HURUF HUJAYAH UNTUK ANAK USIA DINI PADA PLATFORM ANDROID

Gambar 6. Tampilan fitur pengunggahan dan pemrosesan file

c. Proses Hasil Kemiripan Judul

Deteksi Kemiripan Judul Skripsi

Masukkan Judul Skripsi:

ANALISIS SENTIMEN APLIKASI GOPAY BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES

Deteksi

Hasil Deteksi:

No	Judul Usulan	Judul Skripsi	Similarity Text	Rata-rata Panjang String	Persentase Kemiripan
1	ANALISIS SENTIMEN APLIKASI GOPAY BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES	ANALISIS SENTIMEN APLIKASI REKSADANA BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES	92.93	13	92.93%

Ajukan Judul

Gambar 7. Tampilan fitur pencarian judul

Berdasarkan hasil yang ditampilkan, judul skripsi yang diusulkan "ANALISIS SENTIMEN APLIKASI GOPAY BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES" memiliki kemiripan yang sangat tinggi dengan judul skripsi yang ada di database. Skor kemiripan teks adalah 92.93%, menunjukkan bahwa kedua judul memiliki kesamaan kata-kata yang identik. Rata-rata panjang string adalah 13 karakter, mengindikasikan konsistensi dalam panjang kata-kata. Persentase kemiripan hampir sama yang berarti kedua judul tersebut tidak hanya memiliki kata-kata yang sama, tetapi juga memiliki struktur yang sama.

d. Halaman Dosen

3	Sistem pendukung keputusan untuk menentukan judul skripsi jurusan teknik informatika menggunakan metode Topsis	PENERAPAN WEB SEMANTIK UNTUK PENCARIAN JUDUL SKRIPSI PADA RUANG BACA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH MAKASSAR	0.3351328565943472	<input type="button" value="Terima"/> <input type="button" value="Tolak"/>
4	ANALISIS SENTIMEN APLIKASI WHATSAPP BERDASARKAN ULASAN DI PLAY STORE BERBASIS NATURAL LANGUANGE PROCESSING	ANALISIS SENTIMEN APLIKASI REKSADANA BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES	0.7237490182690437	<input type="button" value="Terima"/> <input type="button" value="Tolak"/>
5	ANALISIS SENTIMEN APLIKASI WHATSAPP BERDASARKAN ULASAN DI PLAY STORE BERBASIS NATURAL LANGUANGE PROCESSING	ANALISIS SENTIMEN APLIKASI REKSADANA BERDASARKAN ULASAN DI PLAY STORE MENGGUNAKAN METODE NAIVE BAYES	0.7237490182690437	<input type="button" value="Terima"/> <input type="button" value="Tolak"/>

Gambar 8. Tampilan fitur dosen melihat judul yang ada dan yang mirip

Halaman tampilan dosen, di mana terlihat daftar judul skripsi yang telah diajukan oleh mahasiswa beserta judul skripsi yang mirip dari database. Setiap baris menunjukkan judul usulan, judul mirip dari database, skor kemiripan, serta dua tombol aksi: "Terima" dan "Tolak". Fitur ini memudahkan dosen untuk menilai dan memutuskan apakah judul usulan memiliki kesamaan yang signifikan dengan judul yang sudah ada, serta memungkinkan mereka untuk menyetujui atau menolak pengajuan dengan cepat.

B. Proses Akurasi SVM

1. Vektorisasi Dan Ekstraksi TF-IDF

Transformasi teks menjadi vektor fitur dilakukan menggunakan teknik *TF-IDF* (*Term Frequency-Inverse Document Frequency*). *TF-*

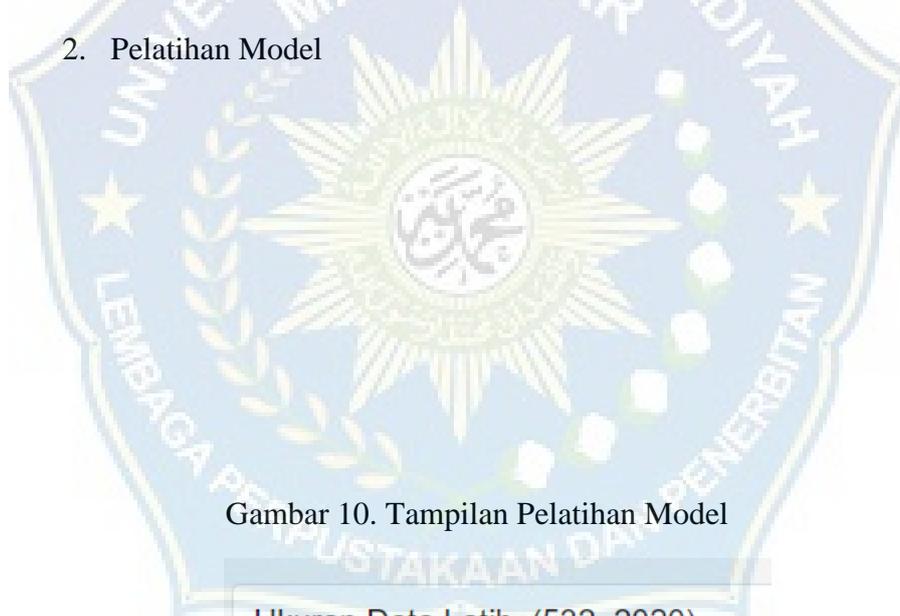
IDF digunakan untuk mengukur seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap seluruh korpus. Hasilnya adalah vektor fitur yang dapat digunakan oleh model machine learning.



Gambar 9. Tampilan *Vektorisasi* Dan Ekstraksi TF-IDF

Jumlah fitur merupakan jumlah semua kata yang telah diproses sebelumnya jadi jumlah judul 54 yang tersedia.

2. Pelatihan Model



Gambar 10. Tampilan Pelatihan Model

Pada tahap ini, model *Support Vector Machine (SVM)*(SVM) dilatih menggunakan data yang telah diproses. SVM adalah algoritma

klasifikasi yang efektif untuk dataset dengan dimensi tinggi. Data dibagi menjadi set pelatihan dan pengujian untuk melatih dan menguji model.

3. Judul yang di ajukan dan Judul Paling Mirip

Judul yang Diajukan
Judul yang diajukan adalah "Pencarian Buku di Perpustakaan". Judul ini akan dibandingkan dengan judul-judul skripsi dalam database untuk menemukan judul yang paling mirip.
Judul yang Paling Mirip
Judul yang paling mirip dengan judul yang diajukan adalah "Sistem Informasi Perpustakaan", dengan skor kemiripan sebesar 0.4789599606584551.

Gambar 11. Tampilan Judul Paling Mirip

Dari total judul yang tersedia dalam database, judul yang diajukan adalah "Pencarian Buku di Perpustakaan" , setelah melakukan *preprocessing*, vektorisasi, dan ekstraksi fitur menggunakan TF-IDF, model *Support Vector Machine (SVM)* dilatih dan diuji. Dari total judul yang tersedia dalam database, judul yang diajukan adalah "Pencarian Buku di Perpustakaan". setelah melakukan *preprocessing*, vektorisasi, dan ekstraksi fitur menggunakan TF-IDF, model *Support Vector Machine (SVM)* dilatih dan diuji. Hasil evaluasi menunjukkan akurasi sebesar 0.7982456140350878, presisi sebesar 0.8260869565217391, recall sebesar 0.7851239669421488, dan *F1-Score* sebesar 0.8050847457627118. Judul yang paling mirip dengan judul yang diajukan adalah "Sistem Informasi Perpustakaan" dengan skor kemiripan sebesar 0.4789599606584551. Perhitungan manual kemiripan kosinus juga menunjukkan berapa besar kemiripannya.

Perhitungan Manual dengan SVM dan TF-IDF

Langkah 1: Preprocessing Data

- Misalkan Anda memiliki dua contoh judul skripsi:
Penerapan Algoritma Support Vector Machine pada Deteksi Kemiripan Judul
- Implementasi Sistem Pengolahan Data dengan Algoritma SVM
Preprocessing melibatkan:
 - Tokenisasi (memecah teks menjadi kata-kata)
 - Penghapusan stop words
 - Stemming (mengubah kata-kata ke bentuk dasar)

Contoh Preprocessing:

- Judul 1: ["penerapan", "algoritma", "support", "vector", "machine", "deteksi", "kemiripan", "judul"]
- Judul 2: ["implementasi", "sistem", "pengolahan", "data", "algoritma", "svm"]

Langkah 2: Ekstraksi Fitur dengan TF-IDF

TF-IDF mengukur pentingnya suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya.

Hitung TF (Term Frequency):

- TF untuk "algoritma" di Judul 1: $TF = 1/8 = 0.125$
- TF untuk "algoritma" di Judul 2: $TF = 1/6 = 0.1667$

Hitung IDF (Inverse Document Frequency):

- IDF untuk "algoritma": $IDF = \log(2/2) = \log(1) = 0$
- Karena muncul di kedua dokumen, IDF menjadi 0. Dalam praktek, sering dilakukan penyesuaian kecil untuk menghindari hasil nol.

Hitung TF-IDF:

- TF-IDF untuk "algoritma" di Judul 1 dan 2 = $TF * IDF = 0$ (karena $IDF = 0$)

Dengan TF-IDF, Anda mendapatkan vektor fitur untuk setiap dokumen. Misalkan Anda memiliki 760 judul, setiap judul akan diwakili oleh sebuah vektor yang berisi nilai TF-IDF untuk setiap kata unik di seluruh dataset.

Langkah 3: Membagi Data dan Melatih Model SVM

Data kemudian dibagi menjadi data pelatihan dan data pengujian. Misalkan Anda membagi data sebagai berikut:

- 70% untuk pelatihan: $0.7 \times 760 = 532$ data pelatihan.
- 30% untuk pengujian: $0.3 \times 760 = 228$ data pengujian.

Latih Model SVM:

Anda melatih model SVM dengan data pelatihan. Proses ini melibatkan optimasi untuk mencari hyperplane terbaik yang memisahkan dua kelas dalam ruang fitur.

Prediksi dan Evaluasi:

Setelah model dilatih, data pengujian digunakan untuk mengevaluasi performa model. Prediksi dilakukan pada data pengujian, dan kemudian dibandingkan dengan label sebenarnya untuk menghitung metrik seperti akurasi, presisi, recall, dan F1-score.

Hasil:

Misalkan setelah pelatihan dan pengujian, Anda mendapatkan hasil sebagai berikut:

- Akurasi: 85%
- Presisi: 82%
- Recall: 78%
- F1-Score: 80%

Contoh Perhitungan Akurasi:

Misalkan dari 228 data pengujian, model Anda benar mengklasifikasikan 194 data.

Akurasi = $194/228 = 0.8509$ atau 85.09%



BAB V

PENUTUP

A. Kesimpulan

1. Kombinasi metode Support Vector Machine (SVM) dan Natural Language Processing (NLP), khususnya melalui vektorisasi menggunakan TF-IDF, telah terbukti efektif dalam mendeteksi kemiripan judul skripsi. Proses ini melibatkan pengumpulan data dari database judul_skripsi, diikuti dengan preprocessing yang meliputi tokenisasi, penghapusan kata-kata tidak relevan, dan stemming. Data yang bersih dan terstruktur ini kemudian diubah menjadi representasi numerik melalui TF-IDF, yang memberikan bobot lebih tinggi pada kata-kata yang spesifik namun signifikan. Model SVM yang dilatih dengan data ini berhasil mendeteksi kemiripan dengan tingkat akurasi yang memadai.
2. Jumlah data yang digunakan memiliki peran krusial dalam meningkatkan akurasi model dalam mendeteksi kemiripan judul. Semakin besar jumlah data yang tersedia, semakin baik model dapat mengenali pola-pola tertentu dalam penulisan judul, sehingga memperkuat kemampuan aplikasi dalam memberikan hasil yang lebih akurat dan andal di berbagai konteks penggunaan.

B. Saran

Memperluas cakupan data judul skripsi dari berbagai disiplin ilmu untuk lebih mengenali berbagai pola dan variasi dalam penulisan judul sehingga meningkatkan akurasi deteksi kemiripan.

DAFTAR PUSTAKA

- Ahmad, I., Borman, R. I., Fakhrurozi, J., & Caksana, G. G. (2020). Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android. *INOVTEK Polbeng - Seri Informatika*, 5(2), 297. <https://doi.org/10.35314/isi.v5i2.1654>
- Alamsyah, N., & Rasyidan, M. (2019). Deteksi *Plagiarisme* Tingkat Kemiripan Judul Skripsi Pada Fakultas Teknologi Informasi Menggunakan Algoritma Winnowing. *Technologia: Jurnal Ilmiah*, 10(4), 197. <https://doi.org/10.31602/tji.v10i4.2361>
- Desena, W., & Solichin, A. (2021). Pencarian Abstrak Tugas Akhir Mahasiswa Berdasarkan Tingkat Kemiripan Menggunakan Algoritma Winnowing dan Jaccard Similarity pada Universitas Budi Luhur. *Informatik : Jurnal Ilmu Komputer*, 17(2), 112. <https://doi.org/10.52958/iftk.v17i2.3628>
- Dillak, R. Y., Laumal, F., & Kadja, L. J. (2016). Sistem Deteksi Dini *Plagiarisme* Tugas Akhir Mahasiswa Menggunakan Algoritma Ngrams Dan Winnowing. *Jurnal Ilmiah Flash*, 2(1), 12. <https://doi.org/10.32511/jiflash.v2i1.19>
- Fitrianingsih, N., Asfi, M., Prasetyo, D., Kusuma, R. P., & Sulhan, M. A. (2022). Deteksi Tingkat Kemiripan Judul Menggunakan Algoritma Oliver Pada Sistem Informasi Pengajuan Skripsi. *Jurnal Media Informatika Budidarma*, 6(4), 2148. <https://doi.org/10.30865/mib.v6i4.4409>
- Hasanah, U. N., Satra, R., & Umar, F. (2020). Deteksi Kemiripan Judul Skripsi Menggunakan Algoritma Smith Waterman. *Buletin Sistem Informasi Dan Teknologi Islam*, 1(1), 56–65. <https://doi.org/10.33096/busiti.v1i1.676>
- Islamiyati, D. S., Fikri, A., Teknik, F., Informatika, T., & Pamulang, U. (2022). Penerapan Algoritma Knuth-Morris-Pratt dalam Mendeteksi Tingkat Kemiripan Judul Skripsi Berbasis Web. 3(2), 58–62. <https://doi.org/10.47065/josh.v3i2.1168>

- Lestari, W., & Sumarlinda, S. (2023). Studi Komparatif Model Klasifikasi Kerentanan Penyakit Jantung Menggunakan Algoritma Machine Learning. *SATIN - Sains Dan Teknologi Informasi*, 9(1), 107–115. <https://doi.org/10.33372/stn.v9i1.918>
- Mawanta, I., Gunawan, T. S., & Wanayumini, W. (2021). Uji Kemiripan Kalimat Judul Tugas Akhir dengan Metode *Cosine Similarity* dan Pembobotan TF-IDF. *Jurnal Media Informatika Budidarma*, 5(2), 726. <https://doi.org/10.30865/mib.v5i2.2935>
- Processing, N. L., & Matching, F. S. (2022). *Natural language processing*.
- Sanjaya, A., Fauzi, I., & Uddin, M. F. (2016). Pencegah plagiasi dengan deteksi kemiripan judul skripsi. *Nusantara Oof Engineering*, 3(2), 7–11.
- Setiawan, A., & Saprudin. (2023). Implementasi Sistem Pendeteksian Kemiripan Judul Skripsi Dengan Algoritma Levenshtein Distance Pada Perpustakaan Universitas Pamulang. *Jurnal Ilmu Komputer Dan Pendidikan*, 2(1), 1–5. <https://journal.mediapublikasi.id/index.php/logic>
- Sheva Ricko Apriansyah. (2023). *Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Judul Skripsi Pada Program Studi Informatika Universitas Teknologi Sumbawa*. 1(2), 1–120.
- Sugiarto, L., Mulyadi, C., & Rihastuti, S. (2021). Analisa Algoritma String Matching Dan Winnowing Untuk Deteksi Kemiripan Judul Tugas Akhir Perguruan Tinggi. *Jurnal Teknologi Informasi*, 6(2), 97–106. <https://doi.org/10.52643/jti.v6i2.1141>
- Wahyuni, M. S., Setiawan, D., & Syahputra, T. (2021). Sistem Temu Kembali Informasi Dengan Latent Semantic Analisis Pada Kesamaan Tugas Akhir Mahasiswa. *Jurnal Teknisi*, 1(2), 46. <https://doi.org/10.54314/teknisi.v1i2.655>

Desena, W., & Solichin, A. (2021). Pencarian Abstrak Tugas Akhir Mahasiswa Berdasarkan Tingkat Kemiripan Menggunakan Algoritma WInnowing dan Jaccard Similarity pada Universitas Budi Luhur. *Informatik : Jurnal Ilmu Komputer*, 17(2), 112. <https://doi.org/10.52958/iftk.v17i2.3628>

Fitrianingsih, N., Asfi, M., Prasetyo, D., Kusuma, R. P., & Sulhan, M. A. (2022). Deteksi Tingkat Kemiripan Judul Menggunakan Algoritma Oliver Pada Sistem Informasi Pengajuan Skripsi. *Jurnal Media Informatika Budidarma*, 6(4), 2148. <https://doi.org/10.30865/mib.v6i4.4409>

Putra Bayu Pratama, N., Mustaqiem, & Minarni. (2021). Pengembangan Sistem Informasi Pengelolaan Judul Skripsi dan Tugas Akhir dengan Fitur Deteksi Kemiripan Menggunakan Algoritma WInnowing. *Terapan Informatika Nusantara*, 2(5), 271–278.



LAMPIRAN

```
from flask import Flask, request, jsonify,
render_template, redirect, url_for, session

import pandas as pd

import numpy as np

from sklearn.feature_extraction.text import
TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score

from sklearn.metrics.pairwise import
cosine_similarity

from sqlalchemy import create_engine, text

import os

import nltk

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

from nltk.tokenize import word_tokenize

nltk.download('punkt')

nltk.download('stopwords')

app = Flask(__name__)

app.secret_key = 'your_secret_key' # Ganti dengan
kunci rahasia Anda
```

```

# Koneksi ke database

engine =
create_engine('mysql+pymysql://root:mariabelajar@if.u
nismuh.ac.id:3388/judul_skripsi')

# Membaca dataset dari database

data = pd.read_sql('SELECT * FROM judul_skripsi',
engine)

print(data.columns) # Tambahkan ini untuk melihat
nama kolom yang sebenarnya

judul = data['judul_skripsi'] # Pastikan nama kolom
sesuai dengan yang ada di database

# Fungsi untuk preprocessing teks

def preprocess_text(text):
    # Tokenisasi
    tokens = word_tokenize(text)

    # Penghapusan stop words
    tokens = [word for word in tokens if word.lower()
not in stopwords.words('english')]

    # Stemming
    stemmer = PorterStemmer()
    tokens = [stemmer.stem(word) for word in tokens]

    return ' '.join(tokens)

```

```

# Preprocessing semua judul skripsi
judul_preprocessed = judul.apply(preprocess_text)

tfidf = TfidfVectorizer(stop_words='english')

# Transformasi teks menjadi vektor fitur
X = tfidf.fit_transform(judul_preprocessed)
X = np.asarray(X.todense()) # Convert the matrix to
an array

# Dummy label untuk tujuan klasifikasi (menggunakan
dua kelas)
y = np.array([1 if i < len(judul) / 2 else 0 for i in
range(len(judul))])

# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=42)

# Melatih model Support Vector Machine
svm = SVC(kernel='linear', probability=True)
svm.fit(X_train, y_train)

# Evaluasi Model
y_pred = svm.predict(X_test)

```

```

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f'Akurasi: {accuracy}')
print(f'Presisi: {precision}')
print(f'Recall: {recall}')
print(f'F1-Score: {f1}')

# Fungsi untuk mendeteksi kemiripan judul baru dengan
# judul dalam database
def deteksi_kemiripan(judul_baru):
    judul_baru_preprocessed = preprocess_text(judul_baru)
    judul_baru_vec = tfidf.transform([judul_baru_preprocessed])
    similarities = cosine_similarity(judul_baru_vec, X)
    max_similarity_idx = similarities.argmax()
    max_similarity = similarities[0, max_similarity_idx]
    most_similar_title = judul.iloc[max_similarity_idx]
    threshold = 0.5
    is_similar = max_similarity >= threshold
    return is_similar, max_similarity, most_similar_title

```

```

@app.route('/')

def home():

    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login():

    username = request.form['username']
    role = request.form['role']
    session['username'] = username
    session['role'] = role

    if role == 'admin':
        return redirect(url_for('admin'))
    elif role == 'mahasiswa':
        return redirect(url_for('mahasiswa'))
    elif role == 'dosen':
        return redirect(url_for('dosen'))

@app.route('/admin')

def admin():

    if 'username' in session and session['role'] ==
    'admin':

        judul_skripsi =
data['judul_skripsi'].tolist()

```

```

        return render_template('admin.html',
judul_skripsi=judul_skripsi)

        return redirect(url_for('home'))

@app.route('/mahasiswa')

def mahasiswa():

    if 'username' in session and session['role'] ==
'mahasiswa':

        return render_template('mahasiswa.html')

        return redirect(url_for('home'))

@app.route('/dosen')

def dosen():

    if 'username' in session and session['role'] ==
'dosen':

        judul_diajukan = pd.read_sql('SELECT * FROM
pengajuan_judul', engine)

        judul_diajukan_list =
judul_diajukan.to_dict(orient='records')

        return render_template('dosen.html',
judul_diajukan=judul_diajukan_list,
enumerate=enumerate)

        return redirect(url_for('home'))

@app.route('/data_uji')

def data_uji():

    if 'username' in session:

        # Creating dummy data for testing purposes

```

```

data_uji_list = [
    {
        "data_uji_ke": f"Judul-{i+1}",
        "judul": judul.iloc[i],
        "similarity_teks": 0, # Replace this
with actual similarity if available
        "rata_rata_panjang":
np.mean([len(word) for word in
word_tokenize(judul.iloc[i])]),
        "persen_kemiripan":
f"{(np.random.rand() - 0.5) * 200:.2f}%", # Dummy
percentage for testing
        "max": f"{np.random.rand() *
100:.2f}%"
    }
    for i in range(len(X_test))
]
return render_template('data_uji.html',
data_uji=data_uji_list, enumerate=enumerate)
return redirect(url_for('home'))

@app.route('/predict', methods=['POST'])
def predict():
    if 'username' in session and session['role'] ==
'mahasiswa':
        judul_baru = request.form['judul_skripsi']
        is_similar, similarity_score,
most_similar_title = deteksi_kemiripan(judul_baru)

```

```

        hasil = "Mirip" if is_similar else "Tidak
Mirip"

        return jsonify({'judul_baru': judul_baru,
'hasil': hasil, 'similarity_score': similarity_score,
'most_similar_title': most_similar_title})

        return redirect(url_for('home'))

@app.route('/update', methods=['POST'])
def update():
    if 'username' in session and session['role'] ==
'admin':
        judul_baru = request.form['judul_skripsi']
        file_excel = request.files.get('file_excel')

        if judul_baru:
            new_data = pd.DataFrame({'JUDUL_SKRIPSI':
[jjudul_baru]}) # Pastikan nama kolom sesuai
            new_data.to_sql('judul_skripsi',
con=engine, if_exists='append', index=False)

        if file_excel:
            file_path = os.path.join('./uploads',
file_excel.filename)

            file_excel.save(file_path)

            new_data = pd.read_excel(file_path)

            new_data.columns = ['NO',
'JUDUL_SKRIPSI'] # Sesuaikan nama kolom dengan yang
ada di database

```

```

        new_data.to_sql('judul_skripsi',
con=engine, if_exists='append', index=False)

        return jsonify({'message': 'Data berhasil
diperbarui'})

        return redirect(url_for('home'))

@app.route('/ajukan', methods=['POST'])
def ajukan():
    if 'username' in session and session['role'] ==
'mahasiswa':
        data = request.get_json()
        judul_baru = data['judul_baru']
        similarity_score = data['similarity_score']
        most_similar_title =
data['most_similar_title']

        df_ajukan = pd.DataFrame({
            'judul_baru': [judul_baru],
            'similarity_score': [similarity_score],
            'most_similar_title':
[most_similar_title],
            'status': ['pending'] # Tambahkan status
awal sebagai pending
        })

        df_ajukan.to_sql('pengajuan_judul',
con=engine, if_exists='append', index=False)

```

```

        return jsonify({'message': 'Judul berhasil
diajukan untuk diperiksa oleh dosen.'})

        return redirect(url_for('home'))

@app.route('/handle_action', methods=['POST'])
def handle_action():
    data = request.get_json()
    judul_baru = data['judul_baru']
    status = data['status']

    with engine.connect() as conn:
        query = text("UPDATE pengajuan_judul SET
status = :status WHERE judul_baru = :judul_baru")
        conn.execute(query, {'status': status,
'judul_baru': judul_baru})
        print(f'Judul {judul_baru} berhasil
di{status}')

    return jsonify({'success': True})

if __name__ == '__main__':
    app.run(debug=True)

```



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Alamat Kantor: Jl. Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Hanjas Haruna
Nim : 105841106720
Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10 %	10 %
2	Bab 2	25 %	25 %
3	Bab 3	9 %	10 %
4	Bab 4	10 %	10 %
5	Bab 5	4 %	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 21 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Hanjaz haruna 105841106720

BAB I

by Tahap Tutup

Submission date: 21-Aug-2024 10:16AM (UTC+0700)

Submission ID: 2435336025

File name: BAB-I.docx (20.57K)

Word count: 691

Character count: 4510



Dipindai dengan CamScanner

ranjas haruna 105841106720 BAB I

ORIGINALITY REPORT

10%

SIMILARITY INDEX

8%

INTERNET SOURCES

6%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	Asvarizal Filcha, Mardhiya Hayaty. "Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa", JUITA : Jurnal Informatika, 2019 Publication	2%
2	etheses.uin-malang.ac.id Internet Source	2%
3	repository.radenintan.ac.id Internet Source	2%
4	repository.ummat.ac.id Internet Source	2%
5	text-id.123dok.com Internet Source	2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On

Hanjas haruna 105841106720

BAB II

by Tahap Tutup

Submission date: 21-Aug-2024 10:17AM (UTC+0700)

Submission ID: 2435336443

File name: BAB-II.docx (47.74K)

Word count: 1630

Character count: 10792

 Dipindai dengan CamScanner

Hanjas haruna 105841106720 BAB II

ORIGINALITY REPORT

25%

SIMILARITY INDEX

22%

INTERNET SOURCES

13%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	jurnal.teknokrat.ac.id Internet Source	5%
2	repository.unhas.ac.id Internet Source	5%
3	anyflip.com Internet Source	4%
4	ejurnal.seminar-id.com Internet Source	2%
5	staffnew.uny.ac.id Internet Source	2%
6	Sri Winiarti, Desy Widayanti, Ulaya Ahdiani, Taufiq Ismail. "Klasifikasi Jenis Buku Berdasarkan Cover dan Judul Buku Menggunakan Metode Support Vector Machine dan Cosine Similarity", Sainteks, 2022 Publication	2%
7	Submitted to Universitas Lancang Kuning Student Paper	2%





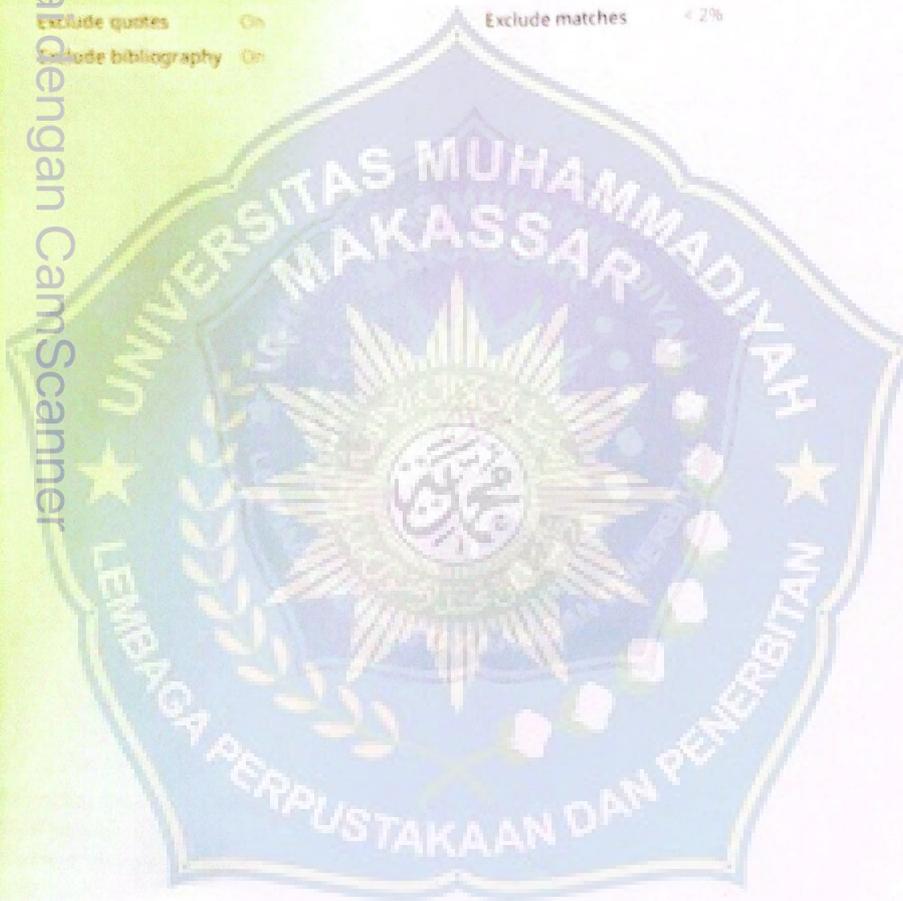
Dipindai dengan CamScanner

digilib.uinsby.ac.id
Internet Source

2%

Exclude quotes On
Exclude bibliography On

Exclude matches < 2%



Hanjas haruna 105841106720

BAB III

by Tahap Tutup



Submission date: 21-Aug-2024 10:18AM (UTC+0700)

Submission ID: 2435336658

File name: BAB-III.docx (130.02K)

Word count: 710

Character count: 4553

ORIGINALITY REPORT

9% SIMILARITY INDEX **7%** INTERNET SOURCES **4%** PUBLICATIONS **2%** STUDENT PAPERS

PRIMARY SOURCES

1	id.123dok.com Internet Source	3%
2	123dok.com Internet Source	2%
3	Muhamad Ziaul Haq, Cut Susan Octiva, Ayuliana Ayuliana, Uli Wildan Nuryanto, Dikky Suryadi. "Algoritma Naïve Bayes untuk Mengidentifikasi Hoaks di Media Sosial", Jurnal Minfo Polgan, 2024 Publication	2%
4	simki.unpkediri.ac.id Internet Source	1%
5	jurnal.radenfatah.ac.id Internet Source	1%



Exclude quotes On Exclude matches < 1%
Exclude bibliography On

Hanjas haruna 105841106720

BAB IV

by Tahap Tutup

Submission date: 21-Aug-2024 10:18AM (UTC+0700)

Submission ID: 2435336844

File name: BAB-IV.docx (730.91K)

Word count: 1260

Character count: 8479

 Dipindai dengan CamScanner

anjas haruna 105841106720 BAB IV

ORIGINALITY REPORT

10%

SIMILARITY INDEX

5%

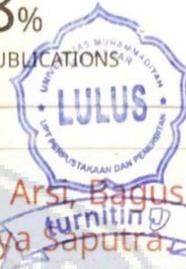
INTERNET SOURCES

8%

PUBLICATIONS

6%

STUDENT PAPERS



PRIMARY SOURCES

- 1 Haris Junianto, Primandani Arsi, Bagus Adhi Kusuma, Dhanar Intan Surya Saputra. "Evaluasi Aplikasi Raileo Melalui Analisis Sentimen Ulasan Playstore Dengan Metode Naive Bayes", SINTECH (Science and Information Technology) Journal, 2024. Publication. 2%
- 2 Submitted to Universitas Negeri Semarang - iTh. Student Paper. 1%
- 3 jurnal.undhirabali.ac.id. Internet Source. 1%
- 4 Reza Zulfiqri, Betha Nurina Sari, Tesa Nur Padilah. "ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI MEDIA SOSIAL INSTAGRAM PADA SITUS GOOGLE PLAY STORE MENGGUNAKAN NAÏVE BAYES CLASSIFIER", Jurnal Informatika dan Teknik Elektro Terapan, 2024. Publication. 1%

5 Submitted to University of Muhammadiyah Malang Student Paper 1 %

6 eprints.ums.ac.id Internet Source 1 %

7 Nur Aisyah Wahyuni, Dinda Putri Ayu, Hafidz Irsyad. "Analisis Sentimen di Youtube Terhadap Kenaikan UKT Menggunakan Metode Support Vector Machine", Arcitech: Journal of Computer Science and Artificial Intelligence, 2024 Publication 1 %

8 Submitted to Universitas Sultan Ageng Tirtayasa Student Paper 1 %

9 Khairul Huda, Sry Dhina Pohan, Youfih Herlina. "PENERAPAN PEMBOBOTAN TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY DAN ALGORITMA K-NEAREST NEIGHBOR UNTUK ANALISIS ULASAN HOTEL DI SITUS TRIPADVISOR", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication 1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 1%

Hanjas haruna 105841106720

BAB V

by Tahap Tutup

Submission date: 21-Aug-2024 10:19AM (UTC+0700)

Submission ID: 2435337233

File name: BAB-V.docx (14.7K)

Word count: 177

Character count: 1159

 Dipindai dengan CamScanner

anjas haruna 105841106720 BAB V

ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

core.ac.uk
Internet Source



4%

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off