# PENERAPAN PEMBOBOTAN KATA DENGAN ALGORITMA TF-IDF DAN COSINE SIMILARITY UNTUK MENGOPTIMALKAN PENCARIAN TUGAS AKHIR TEKNIK INFORMATIKA

**SKRIPSI**

**Oleh:**
**MUHAMMAD FARIS WAFDA**
**220411100039**

**Dosen Pembimbing 1: Dr. Arik Kurniawati, S.Kom., M.T.**
**Dosen Pembimbing 2: Arik Kurniawati, S.Kom., M.T.**

**PROGRAM STUDI TEKNIK INFORMATIKA**
**JURUSAN TEKNIK INFORMATIKA**
**FAKULTAS TEKNIK**
**UNIVERSITAS TRUNOJOYO MADURA**
**2024**

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Dalam era digital saat ini, informasi tersedia dalam jumlah yang sangat besar dan tersebar di berbagai platform. Institusi pendidikan, seperti universitas, menghasilkan banyak karya ilmiah, termasuk tugas akhir, setiap tahunnya. Namun, masalah utama yang muncul adalah bagaimana menemukan informasi yang relevan secara cepat dan efisien dari kumpulan data yang besar ini. Sistem pencarian yang ada sering kali tidak mampu memberikan hasil yang optimal, terutama jika pengguna tidak mengetahui istilah yang tepat atau menggunakan kata kunci yang berbeda dari yang digunakan dalam dokumen. Universitas Trunojoyo Madura, sebagai institusi yang aktif dalam pengembangan riset, memiliki portal tugas akhir yang menjadi sarana penyimpanan dan akses karya ilmiah mahasiswanya. Namun, sistem pencarian yang tersedia di portal tersebut belum mampu memenuhi kebutuhan pengguna secara optimal. Ketika pengguna memasukkan kata kunci tertentu, hasil pencarian yang muncul sering kali tidak relevan dengan keinginan mereka. Hal ini disebabkan oleh kurangnya optimalisasi dalam proses pencocokan kata kunci, di mana variasi istilah atau penggunaan kata kunci yang berbeda dari dokumen yang dicari menyebabkan dokumen yang muncul tidak sesuai dengan ekspektasi pengguna. Akibatnya, pengguna harus menghabiskan lebih banyak waktu untuk menyaring dan meninjau setiap dokumen yang dihasilkan oleh sistem pencarian.

Sistem pencarian yang tidak efisien ini menunjukkan adanya kebutuhan akan solusi yang lebih baik, khususnya dalam hal penerapan algoritma yang mampu meningkatkan relevansi hasil pencarian. Salah satu pendekatan yang dapat diterapkan adalah penggunaan algoritma TF-IDF (Term Frequency-Inverse Document Frequency) yang terbukti efektif dalam mengukur tingkat kepentingan kata dalam dokumen secara lebih akurat. Algoritma ini dapat memberikan bobot pada kata-kata yang sering muncul dalam dokumen tertentu, namun jarang muncul dalam dokumen lainnya, sehingga membantu mengidentifikasi dokumen yang lebih relevan dengan kata kunci yang dimasukkan pengguna. Dengan implementasi algoritma TF-IDF, diharapkan sistem pencarian di portal tugas akhir Universitas Trunojoyo Madura dapat dioptimalkan, sehingga mampu memberikan hasil pencarian yang lebih sesuai dengan kebutuhan pengguna, serta

mempermudah akses terhadap informasi yang dibutuhkan secara cepat dan efisien.

Algoritma TF-IDF (Term Frequency-Inverse Document Frequency) merupakan salah satu teknik yang telah banyak digunakan dalam bidang Information Retrieval (IR). TF-IDF digunakan untuk mengubah data yang berupa kata-kata menjadi numerik. Nilai TF diperoleh dari banyaknya kemunculan kata (term) pada sebuah dokumen. Nilai TF akan besar jika kata tersebut sering muncul dan akan kecil jika kata tersebut jarang muncul [1]. Algoritma ini memungkinkan sistem untuk mengukur seberapa penting sebuah kata dalam satu dokumen dibandingkan dengan seluruh dokumen dalam koleksi. TF-IDF tidak hanya mengukur frekuensi kemunculan sebuah kata (Term Frequency - TF), tetapi juga mempertimbangkan seberapa jarang kata tersebut muncul dalam seluruh dokumen (Inverse Document Frequency - IDF), sehingga dapat memberikan bobot lebih kepada kata-kata yang unik dan relevan dalam konteks tertentu.

Selain TF-IDF, penerapan Cosine Similarity juga menjadi langkah penting untuk menghitung tingkat kemiripan antara kata kunci dengan dokumen dan antar dokumen. Konsep utama dalam kesamaan kosinus adalah mengubah dua string menjadi vektor dalam ruang multidimensi [2]. Semakin besar nilai relevansi, maka kata kunci dengan data yang ditampilkan akan semakin sama atau relevan [3]. Dengan penerapan Cosine Similarity, sistem pencarian dapat melakukan pengukuran kesamaan secara lebih mendalam. Selain hanya melihat frekuensi kemunculan kata seperti pada algoritma TF-IDF, Cosine Similarity memperhitungkan konteks keseluruhan dokumen dan kata kunci, sehingga mampu menghasilkan tingkat relevansi yang lebih akurat. Dalam kasus portal tugas akhir Universitas Trunojoyo Madura, metode ini dapat membantu mempersempit hasil pencarian ke dokumen-dokumen yang tidak hanya mengandung kata kunci yang sesuai, tetapi juga memiliki konteks yang lebih dekat dengan maksud pencarian pengguna.

Penggunaan TF-IDF dalam sistem pencarian tugas akhir menjadi penting karena dapat meningkatkan relevansi hasil pencarian. Algoritma ini mengatasi masalah pencarian sederhana seperti Sequential Search atau pencarian berbasis kecocokan kata kunci sederhana, yang hanya mengandalkan kemunculan kata tanpa mempertimbangkan relevansi kata tersebut dalam seluruh dokumen. Terdapat penelitian pada tahun 2017 yang membandingkan TF-IDF dengan query biasa [4]. Dalam pengujiannya, peneliti membagi kedalam 3 test Case, yaitu Hasil pencarian, urutan tampilan data, dan kebutuhan user. Dari perbandingan tersebut, TF-IDF memiliki beberapa keunggulan seperti mengurutkan nilai kemiripan dari

yang lebih besar ke yang terkecil, dan juga data yang paling sesuai dengan kebutuhan user maka akan ditampilkan dipaling atas. Sedangkan untuk query biasa, data hanya diurutkan berdasarkan alfabet secara ascending dan juga user masih harus memilah kembali data yang benar benar sesuai.

Beberapa algoritma lain, seperti Boolean Retrieval yang hanya mempertimbangkan kehadiran atau ketiadaan kata, juga memiliki keterbatasan dalam hal relevansi. Algoritma ini sering kali menghasilkan hasil pencarian yang tidak efektif karena tidak mampu membedakan antara kata-kata yang umum dengan yang spesifik dalam konteks pencarian. Sebaliknya, TF-IDF mampu mengatasi keterbatasan ini dengan memprioritaskan kata-kata yang jarang muncul tetapi relevan dengan topik yang dicari.

Namun dalam beberapa penelitian, algoritma Sequential search dan boolean Retrieval masih banyak digunakan. Seperti pada tahun 2018 [5], terdapat penelitian tentang Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik. Dalam penelitian ini, penulis memfokuskan Algoritma sequential search untuk digunakan dalam pencarian nomor surat pada sistem arsip elektronik. Penelitian ini berhasil mengatasi masalah pencarian nomor surat pada sistem arsip elektronik ini. Namun, terdapat kekurangan dari algoritma Sequential Search ini, yaitu hanya user harus mengetahui kode surat yang akan dicari, jika tidak mengetahui maka mesin pencari tidak akan menampilkan data apapun. Kekurangan yang lain adalah untuk data yang banyak, maka proses pencarian juga akan semakin lama karena Sequential Search akan mencari satu persatu dari awal hingga ke akhir data.

Penelitian oleh S. Al-Otaibi, dkk. [2] menunjukkan bahwa TF-IDF dan Cosine Similarity mendapatkan akurasi sebesar 90% untuk menampilkan data yang sesuai dengan kata kunci. Hal ini menunjukkan potensi besar dari algoritma TF-IDF dan Cosine Similarity dalam mengatasi permasalahan relevansi pencarian, terutama pada dokumen yang melibatkan isi rangkuman yang relevan terhadap kata kunci yang dimasukkan.

Dengan demikian, penerapan algoritma TF-IDF dan Cosine Similarity diharapkan dapat memberikan solusi yang lebih baik dalam mengoptimalkan pencarian tugas akhir. Algoritma ini tidak hanya mampu mempercepat pencarian, tetapi juga meningkatkan akurasi hasil pencarian dibandingkan dengan algoritma sederhana seperti Sequential Search, sehingga memudahkan mahasiswa atau peneliti dalam menemukan informasi yang relevan dengan topik penelitian

mereka. Dalam jangka panjang, penerapan kedua metode ini dapat menjadi solusi yang kuat dalam menangani masalah pencarian pada kumpulan data teks yang besar dan tidak terstruktur di institusi pendidikan, seperti yang dialami oleh Universitas Trunojoyo Madura.

## 1.2 Rumusan Masalah

### 1.2.1 Permasalahan

Dalam pencarian tugas akhir di PTA Trunojoyo, seringkali pengguna mengalami kesulitan menemukan dokumen yang relevan meskipun telah menggunakan kata kunci yang dianggap tepat, dan pada kenyataannya pencarian berdasarkan fakultas dan program studi tidak menampilkan apapun meskipun sudah memasukkan kata kunci yang sesuai. Hal ini terjadi karena kurangnya maintenance dengan banyaknya dokumen yang tersedia, variasi penggunaan kata kunci, dan kurangnya optimasi dalam sistem pencarian yang hanya berbasis pencocokan kata kunci kemunculan tanpa perangkingan kecocokan terhadap kata kunci. Algoritma sederhana tersebut sering menghasilkan hasil pencarian yang kurang relevan karena hanya mempertimbangkan kemunculan kata kunci tanpa memperhitungkan pentingnya kata dalam konteks seluruh dokumen. Pada beberapa penelitian yang sudah dilakukan, juga masih menggunakan algoirtma sequential dan boolean search untuk mencari dokumen yang sangat banyak. Hal tersebut membuat proses pencarian menjadi cukup lama dan kurang relevan untuk mencari dokumen berdasarkan kemiripan kata kunci.

### 1.2.2 Solusi Permasalahan

Untuk mengatasi permasalahan ini, solusi yang diusulkan adalah algoritma TF-IDF dan Cosine Similarity. Algoritma ini dapat diterapkan dalam sistem pencarian tugas akhir di Prodi Teknik Informatika Universitas Trunojoyo. Algoritma ini menggabungkan dua faktor penting yaitu frekuensi kata dalam dokumen Term Frequency (TF) dan seberapa jarang kata tersebut muncul di seluruh dokumen Inverse Document Frequency (IDF), sehingga menghasilkan bobot yang lebih relevan pada kata-kata yang signifikan dalam konteks pencarian. Cosine Similarity diterapkan untuk mengukur seberapa mirip sebuah dokumen dengan kata kunci yang diberikan. Cosine Similarity bekerja dengan menghitung sudut antara dua vektor dokumen dalam ruang multidimensi. Solusi ini menggabungkan kekuatan TF-IDF untuk pengukuran bobot kata yang lebih akurat dengan kemampuan Cosine Similarity untuk mengukur kemiripan keseluruhan, sehingga sistem pencarian tugas akhir dapat dioptimalkan.

### 1.2.3 Pertanyaan Penelitian

Berdasarkan permasalahan yang telah di jelaskan di atas, dapat dirumuskan pertanyaan dalam penelitian sebagai berikut:

- Bagaimana performa waktu pemrosesan algoritma TF-IDF dibandingkan dengan metode pencarian berbasis kata kunci sederhana pada kumpulan data tugas akhir Prodi Teknik Informatika PTA Trunojoyo?
- Berapa banyak jumlah data yang dimasukkan dan Bagaimana proses pengumpulan dan preprocessing data tugas akhir PTA Trunojoyo untuk memenuhi kebutuhan algoritma TF-IDF?
- Berapa tingkat keberhasilan atau akurasi dari Algoritma TF-IDF dan Cosine similarity dalam mencari dokumen berdasarkan kata kunci yang dimasukkan?
- Bagaimana cara mengimplementasikan algoritma TF-IDF pada sistem pencarian tugas akhir yang sudah ada?

## 1.3 Tujuan dan Manfaat

Skripsi atau proyek akhir memiliki manfaat yang sangat penting bagi mahasiswa dan lingkungan akademik, antara lain:

### 1.3.1 Tujuan

- Untuk mengetahui waktu yang diperlukan untuk memproses pencarian berdasarkan kata kunci
- Mengembangkan sistem pencarian tugas akhir menggunakan algoritma TF-IDF dan Cosine Similarity untuk meningkatkan relevansi hasil pencarian.
- Untuk mengetahui tingkat keberhasilan atau akurasi dari algoritma TF-IDF dan Cosine Similarity dalam mencari dokumen berdasarkan kata kunci yang dimasukkan
- Membandingkan kinerja algoritma TF-IDF dengan metode pencarian konvensional pada kumpulan data tugas akhir Prodi Teknik Informatika PTA Trunojoyo.

### 1.3.2 Manfaat

Dengan demikian, penerapan algoritma TF-IDF dan Cosine Similarity pada sistem pencarian tugas akhir diharapkan dapat meningkatkan efisiensi dan efektivitas pengguna dalam menemukan dokumen yang relevan. Penerapan algoritma juga bertujuan untuk memberikan pengalaman pencarian yang lebih baik bagi pengguna. Sehingga Mahasiswa dan peneliti dapat lebih mudah menemukan

sumber referensi yang relevan, sehingga meningkatkan produktivitas dalam kegiatan akademik.

## 1.4 Batasan Masalah

Batasan masalah yang diterapkan dalam penelitian tugas akhir ini antara lain :

- Penelitian ini fokus pada dokumen tugas akhir khususnya dalam judul dan abstract dokumen PTA Trunojoyo Prodi Teknik Informatika
- Proses preprocessing teks meliputi penghapusan simbol yang tidak dapat dibaca, tokenisasi, stop word removal, dan stemming.
- Evaluasi kinerja sistem dilakukan dengan menggunakan akurasi dan perbandingan dengan algoritma sebelumnya.

## 1.5 Sistematika Penulisan Skripsi

Pada penelitian ini sistematika penulisan skripsi yang digunakan adalah sebagai berikut:

- **Bab 1 PENDAHULUAN**. Pada bab pertama, pendahuluan memberikan gambaran umum mengenai topik yang diangkat dalam penelitian ini, mulai dari pemaparan masalah pencarian tugas akhir yang kurang optimal hingga solusi yang ditawarkan melalui penerapan algoritma TF-IDF dan Cosine Similarity. Penulis menyusun pemaparan dan ringkasan dari setiap sub bab yang terbagi dalam lima bagian utama, yaitu latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, serta sistematika penulisan.

- **Bab 2 KAJIAN PUSTAKA**. Bab dua berisi kajian penelitian terdahulu yang menjadi dasar bagi peneliti dalam menyusun proposal ini, serta memuat kajian pustaka mengenai teori-teori yang relevan. Kajian pustaka mencakup landasan teori tentang TF-IDF, Cosine Similarity, pencarian teks, pengolahan dokumen, dan algoritma pencarian berbasis vektor. Selain itu, bab ini juga membahas metode evaluasi yang digunakan untuk mengukur efektivitas dan akurasi sistem pencarian yang diusulkan. Teori-teori ini memberikan pemahaman mendalam mengenai bagaimana algoritma TF-IDF dan Cosine Similarity bekerja serta bagaimana penerapannya dapat mengoptimalkan pencarian tugas akhir.

- **Bab 3 METODE USULAN**. Bab tiga menjelaskan tentang metode penelitian yang mencakup jenis dan sumber data yang digunakan dalam

penelitian ini. Bab ini juga menguraikan teknik pengolahan data, tahapan-tahapan penelitian, serta evaluasi hasil dari analisis yang dilakukan. Selain itu, dibahas juga uji coba yang dilakukan selama proses analisis data untuk memastikan keakuratan serta efektivitas metode yang diterapkan dalam penelitian ini.

# BAB II

# KAJIAN PUSTAKA

## 2.1 Mesin Pencari

Mesin pencari adalah sebuah sistem yang dirancang untuk mencari informasi dari berbagai sumber di dunia digital, khususnya di internet. Dengan menggunakan kata kunci atau frasa yang dimasukkan pengguna, mesin pencari akan mencari dan menyajikan dokumen, halaman web, gambar, video, atau jenis konten lainnya yang relevan dengan query tersebut. Algoritma dari mesin pencari beragam, seperti Sequential Search, Boolean Search, sampai TF-IDF dan Cosine Similarity. Algoritma Mesin pencarian juga harus melihat jenis jenis dari Mesin Pencarinya. Contohnya seperti pencarian nomor surat, Metode Sequential Search atau disebut pencarian beruntun dapat digunakan untuk melakukan pencarian data baik pada array yang sudah terurut maupun yang belum terurut [5].

Mesin pencari yang relevan bisa kita lihat dengan algoritma yang dimiliki dari mesin pencari tersebut. Mesin pencari yang relevan tidak hanya melihat tersedianya dokumen dengan kata kunci yang diberikan, namun mesin pencari juga harus menampilkan kesesuaian dengan kata kunci. Kesesuaian tersebut diurutkan berdasarkan nilai kemiripan hasil perhitungannya, dari nilai terbesar ke nilai terkecil dalam bentuk persentase [1]. Untuk mencari kemiripan dan kesesuaian dengan kata kunci diperlukan sebuah metode, metode TF-IDF dapat digunakan untuk mengatasi permasalahan similaritas [6]. Metode TF-IDF diterapkan pada penelitian pencarian informasi untuk menemukan teks berita yang sesuai dengan kata kunci yang diberikan. Penelitian ini berhasil menampilkan artikel yang relevan atau sesuai dengan sistem yang dibuat [3].

## 2.2 Temu Kembali informasi *(Information Retrieval)*

Pencarian menggunakan query yang langsung mencari kedalam basis data sudah jarang, namun tidak menutup kemungkinan jika pencarian kedalam basis data masih digunakan untuk beberapa kasus. Saat ini, aplikasi pencarian bukan lagi sekedar kueri ke dalam basisdata (data retrieval) [7]. Untuk meningkatkan relevansi pencarian dibutuhkan sebuah sistem Temu Kembali Informasi(Information Retrieval).

Temu kembali informasi adalah pengambilan informasi dokumen dokumen dari isi dokumen itu sendiri. Sederhananya, sistem temu kembali adalah proses

menemukan informasi yang relevan dengan permintaan pengguna dari suatu koleksi dokumen. Proses dari temu kembali informasi adalah indexing, query processing, ranking, dan sistem akan menampilkan pencarian kepada pengguna. Indexing disini merupakan pemberian bobot terhadap kata-kata yang relevan dengan dokumen tersebut. Query Proccessing merupakan kata kunci yang dimasukkan oleh pengguna yang juga akan diberi bobot. Setelah pemberian bobot, query dan dokumen akan dibandingkan dan diurutkan berdasarkan tingkat kemiripan. Lalu sistem akan menampilkan dokumen yang sudah diranking dan relevan berdasarkan kata kunci pengguna.

### 2.2.1 Web Scraping

Untuk membuat mesin pencarian, diperlukan kumpulan kumpulan dokumen yang akan dimasukkan kedalam mesin pencarian tersebut. Metode untuk mengumpulkannya ada berbagai macam cara. Cara yang mudah adalah dengan web scraping dengan tools yang tersedia. Web scraping akan mengumpulkan informasi yang berkaitan dengan berita-berita terkait standar, prosedur dan aturan yang tersebar pada berbagai website [8]. Sebuah website memiliki akses data untuk tampilan antar-muka dari susunan HTML beserta CSS.

Web scraping berguna untuk mengambil data antar muka dari website tersebut, peneliti memanfaatkan dari akses data yang tersedia untuk diambil data data yang berguna untuk pengambilan informasi dokumen dokumen yang tersedia. Berbagai macam tools yang tersedia di internet, dalam penelitian ini menggunakan tools Python yaitu menggunakan library BeautifulSoup. Library ini akan mengambil informasi HTML dan XML yang tersedia didalam website menjadi data dengan rangkaian elemen yang mudah dibaca. Dalam penelitian ini, website yang digunakan untuk diambil data HTML nya adalah PTA Trunojoyo yang dapat diakses secara gratis.

### 2.2.2 TF-IDF

TF-IDF atau Term Frequency Inverse Document Frequency merupakan algoritma untuk pembobotan setiap kata dalam dokumen. Dalam penelitian ini, data sudah diambil dengan Web Scraping. Data yang sudah diambil, akan dilakukan pembobotan kata disetiap dokumennya. Semakin banyak kemunculan suatu term, maka akan mempengaruhi besarnya bobot dan nilai kesesuaiannya [**?**]. Proses dari TF-IDF ini dibagi menjadi tiga, yaitu untuk menghitung TF, IDF, lalu menghitung bobot TF-IDF.

Term Frequency (TF) adalah banyaknya term didalam kalimat disetiap

dokumen. Dari perhitungan TF ini, akan menghasilkan Dokumen Frequency (DF). Rumus Persamaan Term Frequency (TF) bisa dilihat pada persamaan dibawah persamaan (2.1).

$$W(d,t) = TF(d,t) \tag{2.1}$$

Setelah menentukan dan menghitung jumlah kata disetiap dokumen dan menghitung banyaknya term yang muncul disetiap dokumen. Maka langkah selanjutnya dapat menghitung IDF dengan persamaan (2.2). IDF menghitung log setiap dokumen dibagi dokumen frequensi yang sudah dihitung.

$$idf = log(D/df) \tag{2.2}$$

Untuk menentukan bobot setiap dokumen, dapat dilakukan menggunakan persamaan (2.3). Perhitungannya adalah term frequency yang muncul akan dikali dengan idf yang sudah dihitung sebelumnya.

$$Wdt = tf * idf \tag{2.3}$$

Setelah Tf-IDF sudah ditemukan, langkah terakhir dalam TF-IDF adalah perankingan sesuai dengan nilai paling besar atau perankingan secara descending untuk melihat dokumen mana yang memiliki nilai kesamaan paling tinggi atau mirip.

### 2.2.3 Cosine Similarity

Cosine Similarity merupakan skema untuk mengukur kesamaan berbasis vektor. Algoritma ini menghitung sudut antara dua vektor yakni query dan dokumen untuk menentukan kemiripannya. Dokumen yang paling mirip dengan query maka ditempatkan pada peringkat atas dalam pencarian. Rumus persamaan Cosine similarity sebagai berikut persamaan (2.4).

$$sim(Q,D_i) = \frac{\sum_i W_{Q,j} W_{i,j}}{\sqrt{\sum_j W_{Q,j}^2} \sqrt{\sum_i W_{i,j}^2}} \tag{2.4}$$

## 2.3 Penelitian Terkait

Mesin Pencarian sudah banyak digunakan dalam diberbagai kasus, contohnya seperti mesin pencarian Tugas Akhir Trunojoyo, Mesin pencarian dokumen resmi, dan lain-lain. Akan tetapi, masih banyak Mesin Pencarian yang terbukti kurang efektif untuk menampilkan dokumen yang sesuai dengan kata kunci. Ada banyak cara untuk membuat Mesin Pencarian dengan algoritma yang efisien, mulai dari Sequence Search, Query SQL, sampai dengan melakukan pembobotan menggunakan berbagai Algoritma. Seperti pada tahun 2023 [9]. Melakukan pembuatan aplikasi pencarian tugas akhir menggunakan MySQL. Dalam penelitian ini, peneliti mengungkapkan bahwa sebelumnya penyimpanan Judul Tugas Akhir masih disimpan dalam buku yang disediakan. Sehingga, pencarian judul tugas akhir akan menjadi sangat lama dan seringkali terjadinya salah penulisan atau human error lainnya. Hal tersebut menjadi tujuan peneliti membuat aplikasi pencarian tugas akhir dengan Mysql. Untuk mencari judul tugas akhir, peneliti membuat dengan query sederhana tanpa melihat relevansi setiap dokumen lainnya.

Pada penelitian tahun 2024 [10]. Penelitian ini membahas Algoritma Boyer-Moore pada Aplikasi Pencarian dan Repositori Skripsi. Penulis mengusulkan menggunakan alogiritma Boyer-Moore karena dalam Repositori Skripsi yang terlalu banyak membuat pencarian membutuhkan waktu yang sangat lama. Algoritma ini menggunakan metode string matching tersebut, pencocokan data yang ada dalam repositori dengan kata keyword dapat dilakukan dengan lebih efisien. Algoritma ini sangat cepat dalam mencocokkan keyword dengan dokumen, sehingga algoritma ini cocok dalam pencarian Skripsi. Namun terdapat kekurangan juga dalam algoritma ini, yaitu tidak memperhatikan relevansi dan juga hanya menampilkan ketika pengguna sudah tahu keyword yang akan dimasukkan.

Selain itu, terdapat penelitian pada tahun 2021 [11]. Penelitian ini membahas Implementasi String Matching dengan Algoritma Boyer-Moore. Masalah dalam penelitian ini adalah banyaknya kemiripan tugas Akhir dan Skripsi antar Mahasiswa. Sehingga Program Studi perlu melakukan pengecekan untuk menghindari plagiat. Pada Universitas tersebut, tidak ada sistem otomatis yang mempermudah pencocokan judul, sehingga proses pengecekan bisa memakan waktu. Untuk mengatasi masalah tersebut, digunakan algoritma string matching Boyer-Moore untuk memeriksa kemiripan judul secara otomatis. Algoritma tersebut sangat cepat dan efisien, sehingga pencarian judul Skripsi untuk

menghindari kesamaan dalam pemilihan judul menjadi lebih mudah. Namun, meskipun kecepatan dalam pencarian Judul Skripsi sangat efisien, tetapi algoritma Boyer-Moore tidak memperhatikan relevansi. Sehingga algoritma tersebut hanya mencocokkan pada kata kunci yang dimasukkan tanpa melihat relevansi dokumennya.

Pada tahun 2017 [12] terdapat penelitian tentang Implementasi Algoritma Knuth-Morris-Pratt Pada Fungsi Pencarian Judul Tugas Akhir Repository. Pada pencarian tugas akhir sebelumnya, hanya menggunakan query sederhana, sehingga kecepatan dan efisiensi waktu yang dibutuhkan relatif lebih lama. Penulis mengusulkan pencarian tugas akhir menggunakan algoritma Knuth Morris Pratt. Dengan algoritma tersebut, pencarian tugas akhir dengan jumlah repositori sangat besar hanya memakan waktu 0.0138 detik. Waktu tersebut sangat cepat dan efisien untuk mencari judul Skripsi. Namun algoritma tersebut hanya memperhatikan kecepatan dan efisiensi dari waktu saja, sehingga tidak bisa menampilkan dokumen yang relevan dari kata kunci yang dimasukkan.

Pada tahun 2018, [13] terdapat penelitian tentang mekanisme pencarian judul skripsi dengan metode TF-IDF. Pada penelitian ini, penulis memiliki Masalah yang dihadapi dalam proses pengajuan judul skripsi adalah kesulitan dalam mencari dan menentukan judul yang tepat serta menghindari duplikasi dengan judul skripsi terdahulu. Untuk itu, mahasiswa memerlukan referensi dari judul-judul skripsi sebelumnya agar dapat memperkirakan topik mana yang belum banyak dikembangkan dan memperbesar kemungkinan diterima. Namun, pencarian manual melalui dokumen-dokumen judul yang sudah ada memakan waktu dan tidak efisien. Oleh karena itu, penulis memilih algoritma TF-IDF dalam membuat mesin pencarian Judul Skripsi. Hasil penelitian ini menunjukkan bahwa penerapan metode ini menghasilkan pencarian yang lebih efisien, cepat, dan akurat dalam sistem informasi, dengan tingkat kemiripan yang lebih tinggi dibandingkan metode lain.

Terdapat penelitian tahun 2021, [6] yang membahas Sistem Pencarian Similaritas Judul Tugas Akhir Menggunakan Metode TF-IDF. Fokus dari penelitian ini adalah Untuk menghindari judul yang sama dengan penelitian sebelumnya, sehingga diperlukan sebuah sistem pencarian kemiripan judul. Solusi yang diusulkan adalah menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency) untuk membangun sistem pencarian similaritas judul tugas akhir. Metode ini menghitung pembobotan kata kunci untuk menemukan kesamaan. Hasil ujicoba dengan 384 data judul tugas akhir menunjukkan bahwa

sebanyak 99 judul memiliki kemiripan berdasarkan kata kunci yang sama. TF-IDF terbukti efektif sebagai solusi dalam membantu mahasiswa dan dosen menentukan judul tugas akhir yang unik.

Pada tahun 2024 [14], terdapat penelitian tentang Penerapan Information Retrieval dalam Sistem Analisis Kemiripan Proposal Skripsi menggunakan Cosine Similarity. Penelitian ini terdapat masalah yaitu plagiarisme dalam karya tulis ilmiah yang di mana mahasiswa sering hanya mengubah lokasi penelitian tanpa menawarkan kebaruan. Penelitian ini menawarkan solusi berupa sistem yang dapat menganalisis kemiripan proposal skripsi. Sistem ini menggunakan input berupa judul dan ringkasan proposal, menggabungkan information retrieval dengan text mining menggunakan metode cosine similarity setelah pembobotan dengan TF-IDF. Hasil penelitian menunjukkan bahwa sistem mampu mengenali input pengguna dan memberikan peringkat berdasarkan tingkat kemiripan.

Pada tahun 2021 [15] terdapat penelitian tentang Uji Kemiripan Kalimat Judul Tugas Akhir dengan Metode Cosine Similarity dan Pembobotan TF-IDF. Penulis mengungkapkan terdapat permasalahan yaitu kemiripan judul laporan tugas akhir karena proses pengecekan dilakukan secara manual oleh program studi yang tidak efektif dan menyebabkan banyak judul yang serupa. Peneliti mengusulkan menggunakan algoritma TF-IDF untuk melakukan pembobotan setiap kata dalam dokumen, dan menggunakan algoritma Cosine Similarity untuk melihat similaritas atau kesamaan dokumen dengan kata kunci yang dimasukkan. Hasil uji coba menunjukkan bahwa 43% judul tidak layak diajukan kembali karena memiliki kemiripan yang tinggi, sementara 53% layak diajukan. Rata-rata waktu yang dibutuhkan untuk uji kemiripan adalah 0.12117 menit per judul. Algoritma TF-IDF dan Cosine Similarity sangat efektif ketika dipadukan untuk menghasilkan Search Engine yang optimal dalam mencari Judul tugas akhir dengan memperhatikan relevansi dan kesamaan tiap dokumen dan kata kunci yang diberikan.

**Tabel 2.1** Penelitian Terkait - Bagian 1

| No | Peneliti, Tahun | Permasalahan | Model/Solusi | Hasil |
|---|---|---|---|---|
| 1 | S. T. Siska, dkk. 2023 [9] Link Jurnal | Penyimpanan Judul Tugas Akhir dan Skripsi masih menggunakan buku | Membuat aplikasi untuk menyimpan repositori kedalam database dan membuat Pencarian menggunakan Query sederhana | Pencarian Tugas Akhir sangat efisien dan relatif cepat, meskipun tidak memperhatikan relevansi pada dokumen |
| 2 | D. N. I. Basoeki, dkk. 2024 [10] Link Jurnal | Pencarian judul tugas akhir terlalu lama, sedangkan permintaan pencarian Judul Tugas Akhir sangat banyak | Membuat aplikasi Repositori Judul Tugas Akhir dengan algoritma pencariran Boyer-Moore | Pencarian Judul Tugas Akhir sangat efisien dan sangat cepat, namun hasil dokumen tidak memperhatikan relevansi |
| 3 | I. Ahmad, R. I. Borman, dkk. 2021 [11] Link Jurnal | Pengecekan Judul Tugas Akhir dan skripsi manual dan sangat lama, sehingga terdapat kemiripan dengan judul tugas akhir terdahulu | Mengembangkan sistem Pencarian dengan algoritma Boyer-Moore | Menghasilkan Pencarian judul tugas akhir dan skripsi yang sangat cepat, namun dokumen tidak melihat kemiripan dengan kata kunci |
| 4 | H. T. Sadiah, dkk. 2021 [12] Link Jurnal | Pencarian pada sistem repository Tugas Akhir dan Skripsi belum optimal | Implementasi Algoritma KMP untuk mempercepat proses pencarian tugas Akhir | Performa algoritma KMP untuk menampilkan dokumen adalah 0.0138 detik, sudah cukup cepat, namun masih belum melihat relevansi antar dokumen |

**Tabel 2.2** Penelitian Terkait - Bagian 2

| No | Peneliti, Tahun | Permasalahan | Model/Solusi | Hasil |
|---|---|---|---|---|
| 5 | M. A. Ariyanti, dkk. 2018 [13] Link Jurnal | Kesulitan dalam mencari dan menentukan judul yang tepat serta menghindari duplikasi dengan judul skripsi terdahulu | Merancang dan membangun mekanisme pencarian dengan metode TF-IDF | Kinerja Metode TF-IDF mendapatkan akurasi sebesar 100% dalam pengujian black box |
| 6 | A. Amrulloh, dkk. 2021 [6] Link Jurnal | Kesulitan untuk mengetahui apakah judul tugas akhir yang diajukan sudah pernah diteliti sebelumnya atau belum | membangun sebuah sistem pencarian similaritas judul tugas akhir dengan menerapkan metode TF-IDF | Sebanyak 384 data ditemukan judul tugas akhir yang memiliki kesamaan berdasarkan kata kunci sebanyak 99 data |
| 7 | M. D. Afandi, dkk. 2024 [14] Link Jurnal | Plagiarisme dalam Proposal Skripsi | Membuat mesin pencarian menggunakan TF-IDF dan Cosine Similarity | Sistem Pencarian sangat efisien karena memperhatikan pembobotan kata, Kemiripan dan relevansi antar dokumen dengan kata kunci |
| 8 | I. Mawanta, dkk. 2021 [15] Link Jurnal | Pengecekan Kemiripan Tugas Akhir dan Skripsi secara manual | Uji Kemiripan judul tugas akhir dengan menggunakan algoritma TF-IDF dan Cosine Similarity | Pengecekan judul tugas akhir sangat efisien dan menghemat waktu dengan memperhatikan relevansi dan kemiripan antar Tugas Akhir terdahulu |

# BAB III
# METODOLOGI

## 3.1 Datasets

Dataset yang digunakan dalam penelitian ini diambil dari portal tugas akhir Universitas Trunojoyo Madura. Data yang diambil terdiri dari link, judul tugas akhir, dan abstrak. Jumlah data yang diambil adalah sebanyak 852 tugas akhir mahasiswa Teknik Informatika Universitas Trunojoyo Madura. Pengumpulan data dilakukan melalui teknik web scraping dengan menggunakan alat BeautifulSoup dari pustaka bs4 pada bahasa pemrograman Python.

Proses web scraping dilakukan untuk mengekstrak data secara langsung dari halaman web portal tugas akhir. Dengan menggunakan BeautifulSoup, data yang relevan seperti link tugas akhir, judul, dan abstrak diambil dari setiap halaman detail tugas akhir. Data ini kemudian disusun dalam bentuk dataset yang akan digunakan sebagai sumber utama dalam penelitian ini.

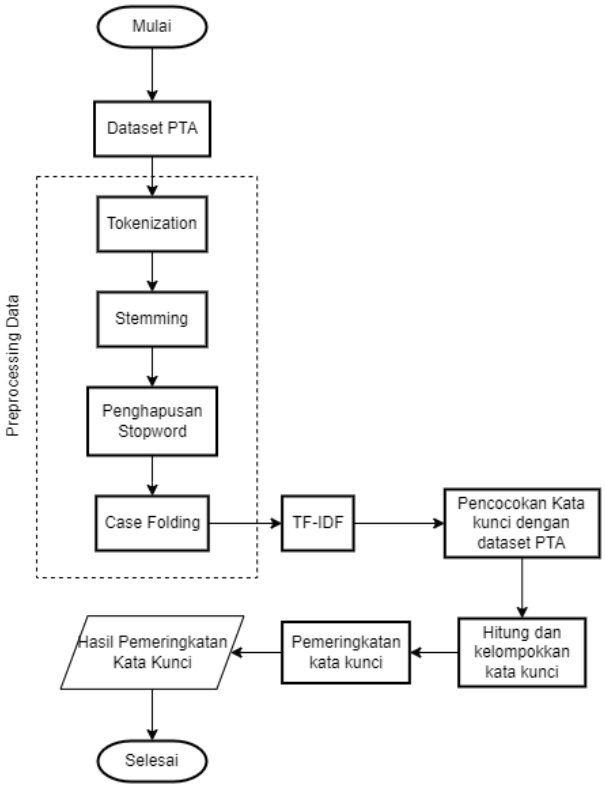
Metode ini memungkinkan pengambilan data secara efisien dan sistematis tanpa harus melakukan input manual terhadap tiap tugas akhir, sehingga mempermudah analisis terhadap tugas akhir yang diambil. Hasil dari scraping ini adalah dataset berjumlah 852 entri, yang mencakup informasi penting dari setiap tugas akhir. Dataset dapat di akses pada https://pta.trunojo.ac.id.

**Tabel 3.1** Contoh Dataset Tugas Akhir PTA Trunojoyo

| **Judul** | **Abstrak** | **Link** |
|---|---|---|
| PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE... | Sistem informasi akademik (SIAKAD) merupakan sistem informasi... | pta.trunojoyo.ac.id/68 |
| APLIKASI KONTROL DAN MONITORING JARINGAN... | Berjalannya koneksi jaringan komputer dengan lancar... | pta.trunojoyo.ac.id/76 |
| RANCANG BANGUN APLIKASI PROXY SERVER... | Web server adalah sebuah perangkat lunak server yang berfungsi... | pta.trunojoyo.ac.id/80 |

## 3.2 Tahapan Penelitian

Alur dari penelitian dapat dilihat pada Diagram Alir pada Gambar 0.1.




**Gambar 3.1** Diagram Alir Penelitian

Dari Diagram alir pada Gambar 0.1. tahapan penelitian yang akan dilakukan adalah sebagai berikut.

1. Dataset PTA Trunojoyo dilakukan preprocessing terlebih dahulu. Tahapan dari preprocessing meliputi:

   (a) Tokenization, tokenization merupakan pemecahan teks berdasarkan karakter spasi menjadi kata-kata. Dalam penelitian ini, tokenization akan diterapkan pada Judul dan Abstrak.

   (b) Stemming, stemming merupakan perubahan kata ke dalam bentuk dasar. Contohnya, ’Kemanusiaan’ menjadi ’Manusia’.

   (c) Penghapusan Stopword, stopwords merupakan kata-kata umum yang sering muncul tetapi tidak memiliki nilai informasi yang besar. Contohnya seperti kata ’dan’, ’di’, ’adalah’.

   (d) Case Folding, case folding merupakan konversi semua huruf dalam teks menjadi huruf kecil (lowercase) untuk memastikan konsistensi saat melakukan analisis teks.

2. Pembobotan TF-IDF. Dari preprocessing yang sudah dilakukan, semua kata dalam satu dokumen akan dimasukkan ke dalam array. Lalu, TF-IDF akan memberi bobot pada setiap kata dalam dokumen.

3. Pencocokan Kata Kunci dengan Dataset. Dari pembobotan dengan TF-IDF, akan dimasukkan kata kunci dari dokumen yang akan dicari. Kata kunci tersebut akan dicocokkan menggunakan Cosine Similarity dengan dataset yang sudah diberi bobot.

4. Perankingan Dokumen berdasarkan Kata Kunci. Setelah kata kunci dicocokkan dengan dataset, maka akan dihitung skor ranking dari dataset, lalu akan diranking berdasarkan skor yang paling tinggi. Dari hasil ranking tersebut, akan ditampilkan dokumen dari yang skornya tertinggi ke yang terendah, berdasarkan kemiripan kata kunci dengan dokumen.

## 3.3 Metode yang Digunakan

Metode yang digunakan untuk mengolah dan mencari kemiripan dokumen adalah TF-IDF (Term Frequency - Inverse Document Frequency) dan Cosine Similarity. Kedua metode ini sering digunakan dalam pemrosesan teks untuk menganalisis hubungan dan kemiripan antar dokumen.

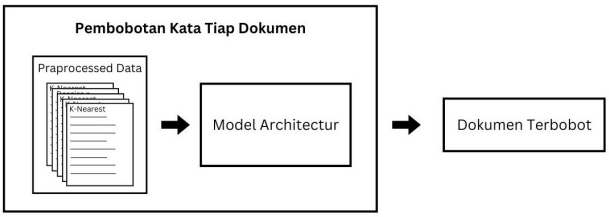
TF-IDF adalah metode pembobotan kata yang bertujuan untuk menghitung seberapa penting sebuah kata dalam suatu dokumen, relatif terhadap keseluruhan dokumen yang ada. Kata yang sering muncul dalam suatu dokumen akan memiliki bobot tinggi, namun jika kata tersebut sering muncul di banyak dokumen lain, bobotnya akan berkurang. Dalam penelitian ini, TF-IDF digunakan untuk memberi bobot pada kata-kata yang terdapat dalam judul dan abstrak tugas akhir yang telah diambil dari portal tugas akhir Universitas Trunojoyo Madura.

Setelah bobot setiap kata dihitung menggunakan TF-IDF, langkah berikutnya adalah menghitung kemiripan antara dokumen dengan menggunakan metode Cosine Similarity. Metode ini digunakan untuk mengukur seberapa mirip suatu dokumen dengan kata kunci yang dimasukkan pengguna. Cosine Similarity bekerja dengan mengukur kesamaan sudut antara dua dokumen dalam ruang vektor. Semakin mirip kata-kata yang ada di kedua dokumen, semakin tinggi nilai kemiripannya.

## 3.4 Rancangan Sistem

Sistem akan dirancang secara umum, yaitu data setelah preprocessing akan dimasukkan kedalam arsitektur model untuk dilakukan indexing. Indexing ini

merupakan penyimpanan dokumen atau dataset PTA kedalam format pickle. Dari format pickle tersebut, dapat dihasilkan dokumen dokumen yang telah dilakukan pembobotan dengan TF-IDF. Setelah dilakukan proses indexing proses selanjutnya adalah pencocokan kata kunci dengan dataset yang sudah diindexing menggunakan Cosine Similarity. Score dari pencocokan tersebut akan diranking dan akan ditampilkan hasil peranking an berdasarkan kata kunci. Diagram Rancangan sistem secara umum dapat dilihat pada Gambar0.2

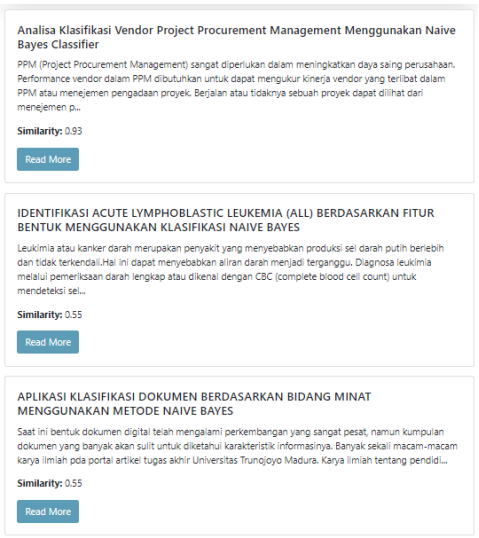



**Gambar 3.2** Diagram Rancangan Sistem

Dari rancangan sistem diatas, harapannya semua kata dalam dokumen berhasil diberi bobot dan akan dimasukkan kedalam format pickle. Dari format pickle tersebut maka akan diuji kemiripan dengan query yang dimasukkan.

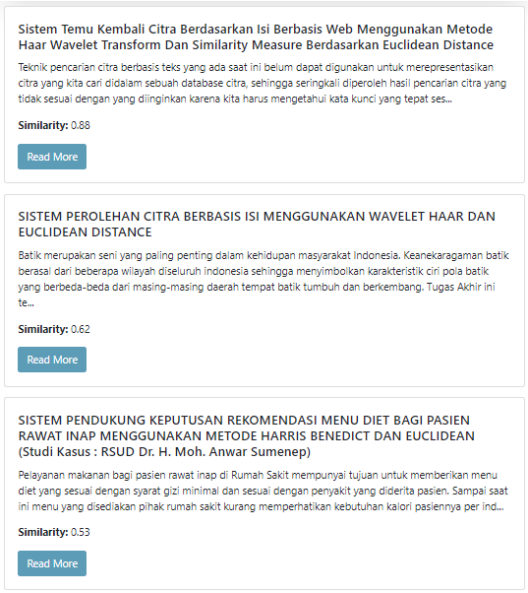
## 3.5 Skenario Uji Coba

Skenario uji coba ini menggunakan query yang mirip sepenuhnya dengan data yang ada di data set. Pada pengujian pertama dilakukan pencarian menggunakan query dari judul tugas akhir yaitu ”Analisa Klasifikasi Vendor Project Procurement Management Menggunakan Naive Bayes Classifier”. Hasilnya ada pada gambar 0.3



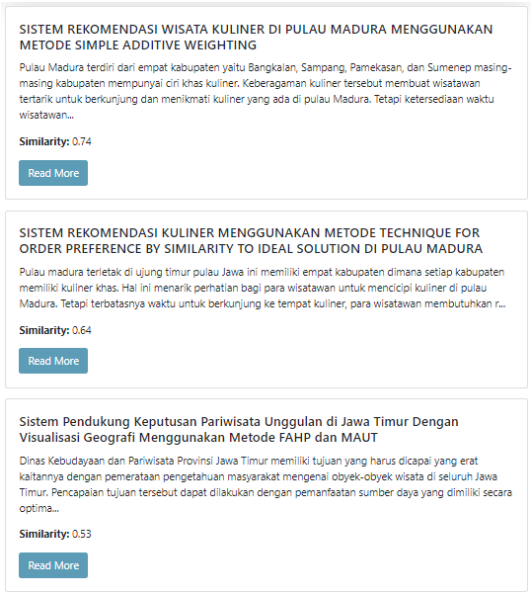

**Gambar 3.3** Skenario 1

Pada skenario pertama gambar 0.3, query disamakan sepenuhnya menggunakan judul tugas akhir. Hasil yang diberikan sesuai dengan query dimana dokumen yang dicari berada pada urutan pertama dengan score similarity 0.93. Selanjutnya akan dilakukan scenario uji kedua yang dimana menggunakan masukan query yang mirip sebagian dengan dataset tugas akhir. Query judul tugas akhir yaitu ”Sistem Temu Kembali Metode Haar Wavelet Transform”. Hasil dari masukan query tersebut dapat dilihat pada gambar 0.4




**Gambar 3.4** Skenario 2

Pada Skenario kedua gambar 0.4, hasil yang diberikan masih sesuai dengan query dimana dokumen yang mirip dengan query nya berada pada urutan pertama dengan score similarity sebesar 0.88. Terakhir, akan dilakukan scenario uji coba ketiga yang dimana menggunakan masukan query yang sama sekali berbeda pada dataset tugas akhir. Query judul tugas akhir yaitu ”Rekomendasi Wisata di Pulau Madura Jawa Timur”. Hasil dari masukan query tersebut dapat dilihat pada gambar 0.5




**Gambar 3.5** Skenario 3

Pada skenario terakhir gambar 0.5, hasil yang diberikan masih terdapat didalam dataset dengan dokumen yang paling mirip dengan query berada pada urutan pertama dan mendapatkan score similiarity sebesar 0.74. Hal ini menunjukkan bahwa saat memasukkan query yang berbeda, sistem mampu mencari dokumen yang relevan dan memiliki kesesuaian dengan query tersebut. Dengan demikian, arsitektur dan metode yang telah dibangun terbukti efektif dalam menyajikan dokumen yang sesuai dengan query yang diajukan.

# DAFTAR PUSTAKA


[1] A. B. Arifa, G. F. Fitriana, A. R. Hasan *et al.*, "Temu kembali informasi pada soal ujian dengan rencana pembelajaran menggunakan vector space model," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 63–68, 2021.

[2] S. Al-Otaibi, N. Altwoijry, A. Alqahtani, L. Aldheem, M. Alqhatani, N. Alsuraiby, S. Alsaif, and S. Albarrak, "Cosine similarity-based algorithm for social networking recommendation," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1881–1892, 2022.

[3] W. Yulita, M. C. Untoro, M. Praseptiawan, I. F. Ashari, A. Afriansyah, and A. Pee, "Automatic scoring using term frequency inverse document frequency document frequency and cosine similarity," *Scientific Journal of Informatics*, vol. 10, no. 2, pp. 93–104, 2023.

[4] A. Rokhim *et al.*, "Implementasi metode term frequency inversed document frequence (tf-idf) dan vector space model pada aplikasi pemberkasan skripsi berbasis web," *SPIRIT*, vol. 9, no. 1, 2018.

[5] A. Sonita and M. Sari, "Implementasi algoritma sequential searching untuk pencarian nomor surat pada sistem arsip elektronik. pseudocode, 5 (1), 1–9," 2018.

[6] A. Amrulloh and I. F. Adam, "Sistem pencarian similaritas judul tugas akhir menggunakan metode tf-idf," *Jurnal CoreIT*, vol. 7, no. 2, 2021.

[7] S. Suprianto, A. Fadlil, and S. Sunardi, "Aplikasi sistem temu kembali angket mahasiswa menggunakan metode generalized vector space model," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 6, no. 1, pp. 33–40, 2019.

[8] N. Ani, D. Y. Sinaga, N. Junior, and M. D. Munggaran, "Penerapan algoritma term frequency-inverse document frequency (tf-idf) untuk fitur pencarian dokumen standar nasional indonesia," *JSAI (Journal Scientific and Applied Informatics)*, vol. 6, no. 3, pp. 517–522, 2023.

[9] S. T. Siska, A. Budiman, and H. Fenia, "Aplikasi pencarian judul tugas akhir mahasiswa berbasis visual studio 2012 dan mysql," *Rang Teknik Journal*, vol. 6, no. 2, pp. 277–284, 2023.

[10] D. N. I. Basoeki, A. P. Sari, and F. A. Akbar, "Penggunaan metode boyer moore pada aplikasi pencarian dan repositori skripsi berbasis web," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 3946–3954, 2024.

[11] I. Ahmad, R. I. Borman, G. G. Caksana, and J. Fakhrurozi, "Implementasi string matching dengan algoritma boyer-moore untuk menentukan tingkat kemiripan pada pengajuan judul skripsi/ta mahasiswa (studi kasus: Universitas xyz)," *SINTECH (Science and Information Technology) Journal*, vol. 4, no. 1, pp. 53–58, 2021.

[12] H. T. Sadiah, "Implementasi algoritma knuth-morris-pratt pada fungsi pencarian judul tugas akhir repository," *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, vol. 14, no. 1, pp. 115–124, 2017.

[13] M. A. Ariyanti, A. P. Wibawa, and U. Pujianto, "Metode term frequency-invers document frequency pada mekanisme pencarian judul skripsi," *Metode*, vol. 28, no. 2, 2018.

[14] M. D. Afandi, A. Homaidi, A. Ghofur, and A. Zubairi, "Penerapan information retrieval dalam sistem analisis kemiripan proposal skripsi menggunakan cosine similarity," *Swabumi*, vol. 12, no. 1, pp. 39–46, 2024.

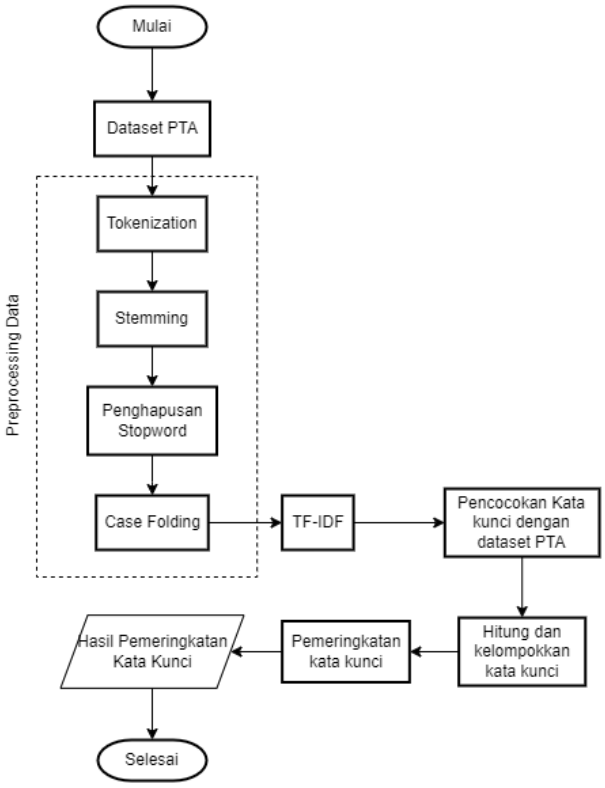
[15] I. Mawanta, T. Gunawan, and W. Wanayumini, "Uji kemiripan kalimat judul tugas akhir dengan metode cosine similarity dan pembobotan tf-idf," *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, pp. 726–738, 2021.

# LAMPIRAN A

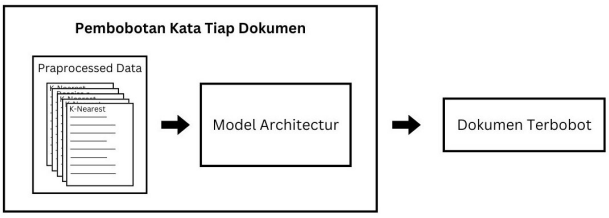
# KODE PROGRAM

# LAMPIRAN B

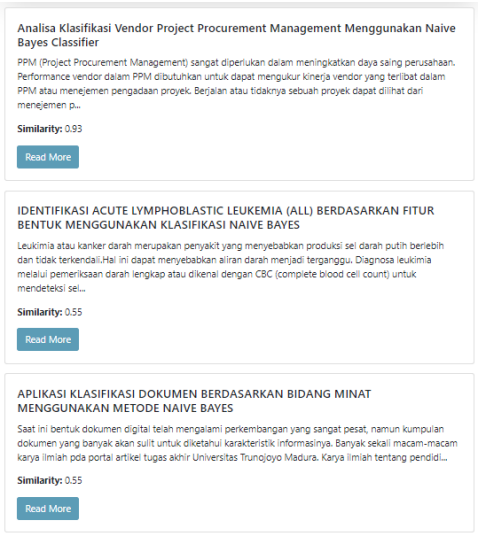
# GAMBAR-GAMBAR



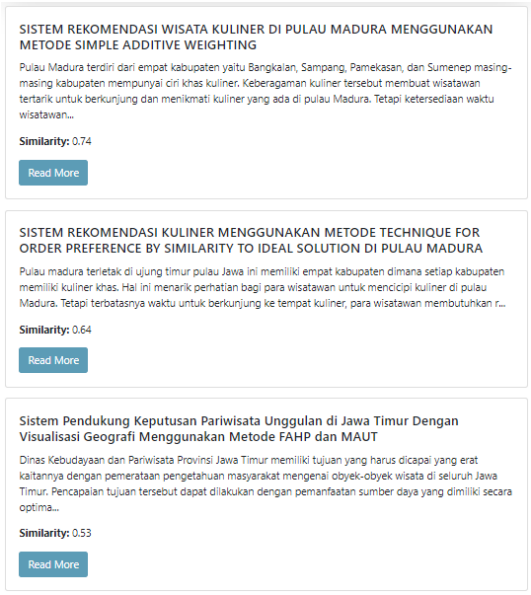

**Gambar 0.1** Diagram Alir Penelitian



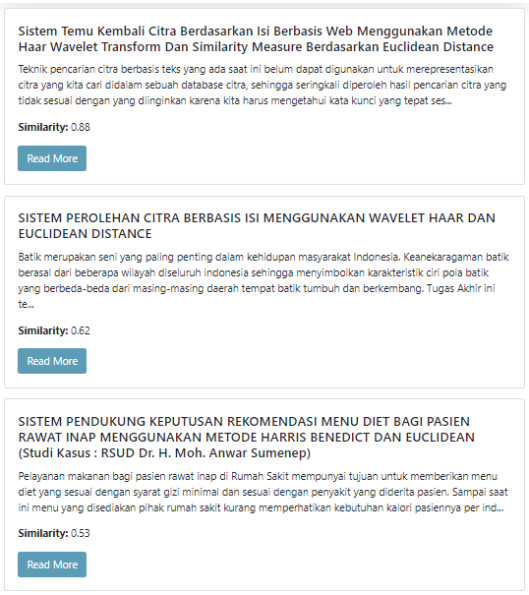

**Gambar 0.2** Diagram Rancangan Sistem




**Gambar 0.3** Skenario 1




**Gambar 0.5** Skenario 3




**Gambar 0.4** Skenario 2