

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Tugas akhir memiliki peran penting dalam pendidikan tinggi karena menjadi sarana mahasiswa menerapkan pengetahuan secara praktis, terutama dalam pengembangan aplikasi. Sebagai puncak dari proses pembelajaran, tugas akhir mendorong mahasiswa untuk mengintegrasikan teori dengan praktik, serta melatih kemampuan berpikir kritis, pemecahan masalah, dan inovasi [1].

Namun, duplikasi ide dalam pengembangan aplikasi proyek akhir menjadi fenomena yang umum terjadi. Banyak proyek memiliki fitur yang mirip satu sama lain, sehingga menghasilkan keluaran yang redundan dan kurang inovatif. Kondisi ini umumnya disebabkan oleh kecenderungan mahasiswa memilih topik yang familiar atau dianggap mudah [2]. Akses yang terbatas terhadap data proyek sebelumnya dan kurangnya dokumentasi turut memperburuk situasi ini, sehingga mahasiswa tanpa sadar mengajukan topik yang serupa dan berulang [1].

Sebelumnya sistem telah dikembangkan untuk mengatasi masalah ini, seperti sistem Discover Idea yang menyediakan fitur deteksi kesamaan topik serta rekomendasi dosen pembimbing, reviewer, dan penguji. Namun, sistem tersebut masih memiliki berbagai kelemahan. Skor kesamaan yang dihasilkan belum akurat secara semantik karena hanya mengandalkan pencocokan pemodelan topik, sehingga gagal menangkap makna sebenarnya dari sebuah judul [2][3]. Akibatnya, rekomendasi dosen yang dihasilkan pun sering kali tidak konsisten. Di samping itu, proses pengisian tautan Open Library ke dalam sistem masih dilakukan secara manual oleh dosen, sehingga menghambat kelengkapan dokumentasi.

Permasalahan tersebut menunjukkan perlunya sistem yang mampu mengevaluasi kemiripan topik secara semantik untuk menemukan keterkaitan makna yang lebih mendalam. Pendekatan semantik semacam ini diharapkan dapat meningkatkan akurasi pencarian topik, memperbaiki konsistensi rekomendasi dosen, serta mengotomatisasi proses pendataan seperti integrasi tautan Open Library. Sistem yang dikembangkan dalam proyek ini dirancang untuk menggantikan arsitektur sebelumnya dengan kerangka kerja yang lebih efisien, terintegrasi, dan mendukung pengelolaan tugas akhir secara berkelanjutan.

## 1.2 Rumusan Masalah

Rumusan masalah yang diajukan dalam proyek akhir ini adalah:

1. Bagaimana merancang sebuah aplikasi yang memudahkan mahasiswa dalam menemukan topik proyek akhir yang mirip?

2. Bagaimana mengimplementasikan sistem deteksi kemiripan topik yang relevan secara konseptual?
3. Bagaimana mempermudah proses penambahan tautan Open Library yang selama ini dilakukan secara manual oleh dosen?
4. Bagaimana merancang API yang dapat menggantikan endpoint API pada versi sebelumnya tanpa mengganggu fungsionalitas web Monitor PA?

### 1.3 Tujuan

Tujuan akhir dari proyek ini adalah sebagai berikut:

1. Merancang aplikasi agar dapat memudahkan mahasiswa dalam menemukan topik proyek akhir yang sesuai.
2. Mengimplementasikan algoritma deteksi kesamaan berbasis *semantic similarity* untuk memberikan referensi topik terkait.
3. Membuat otomatisasi penambahan tautan Open Library
4. Membangun API baru yang kompatibel dengan sistem web Monitor PA untuk memastikan kontinuitas layanan prediksi pembimbing, reviewer, dan penguji.

### 1.4 Batasan Masalah

Adapun batasan masalah dalam proyek ini meliputi:

1. Aplikasi ini hanya akan mencakup topik proyek akhir yang diajukan oleh mahasiswa di prodi D3 Rekayasa Perangkat Lunak Aplikasi.
2. Data yang digunakan untuk mengembangkan aplikasi akan berasal dari proyek akhir mahasiswa di tahun ajaran 2016/2017 sampai dengan tahun ajaran 2023/2024.
3. Metode deteksi kesamaan topik yang akan digunakan adalah teknik-teknik *Natural Language Processing*.

### 1.5 Metode Penyelesaian Masalah

Berikut adalah metodologi penyelesaian masalah yang digunakan dalam proyek akhir ini.

1. **Planning & Design:** Tentukan fitur machine learning yang akan ditambahkan dan desain halaman website untuk mendukung fitur tersebut.
2. **Data Preparation:** Data teks yang relevan dikumpulkan dan dibersihkan. Proses ini termasuk cleaning, tokenisasi, dan stemming .
3. **Model Training & Testing:** Latih model machine learning di lingkungan terpisah dan lakukan pengujian untuk memastikan akurasi.

4. **API Development:** Kembangkan API untuk menghubungkan model machine learning ke website (misalnya, menggunakan Flask).
5. **Integration with Front-End:** Integrasikan API dengan halaman website yang relevan. Setiap halaman dapat menangani berbagai fungsi atau interaksi dengan model (misalnya, input data pengguna untuk prediksi).
6. **Deployment:** Gunakan CI/CD pipeline untuk secara otomatis menguji dan mendeploy aplikasi ke server atau cloud platform.
7. **Monitoring & Maintenance:** Monitor performa model dan halaman website untuk perbaikan dan pembaruan.

### 1.6 Pembagian Tugas Anggota

Berikut adalah pembagian tugas tim pada tugas akhir ini:

#### 1. Adrian Maulana

Peran : UI/UX Designer, Front End Developer

Tanggung Jawab :

- a. Membuat antarmuka aplikasi
- b. Membuat mockup aplikasi
- c. Implementasi *Front End*
- d. Intergrasi dengan *Back End*

#### 2. Andre Fransiscus Masalle

Peran : Data Scientist, AI Engineer, Backend Developer

Tanggung Jawab :

- a. Pengumpulan Data
- b. Pengembangan dan Implementasi Model
- c. Penyediaan dan Pembuatan API