

Sistem Rekomendasi Konversi Program MSIB Dengan Mata Kuliah Prodi Informatika ITN Malang Menggunakan *Cosine Similarity*

Difa Fisabilillah^{1*}, Karina Auliasari², Yosep Agus Pranoto³

^{1,2,3}Fakultas Teknologi Industri, Teknik Informatika, Institut Teknologi Nasional, Malang, Indonesia

E-mail: ^{1*}difafisabilillah91@gmail.com, ²karina.auliasari@lecturer.itn.ac.id, ³yoa@lecturer.itn.ac.id

(* difafisabilillah91@gmail.com)

Abstrak

Program Magang Studi Independen Bersertifikat (MSIB) adalah bagian dari inisiatif Merdeka Belajar Kampus Merdeka (MBKM), yang memungkinkan mahasiswa mengonversi hingga 20 SKS dari kegiatan yang diikuti. Namun, proses ini terkendala karena perbedaan silabus MSIB dan kurikulum Prodi Teknik Informatika ITN Malang. Penelitian ini bertujuan untuk mengembangkan sistem berbasis kecerdasan buatan yang mampu memberikan rekomendasi konversi mata kuliah secara otomatis. Dataset terdiri dari 54 deskripsi mata kuliah serta 17 silabus MSIB. Penelitian ini menggunakan pendekatan *open testing*, dimana setiap silabus MSIB dicocokkan secara langsung dengan seluruh dataset mata kuliah tanpa membagi data menjadi bagian *train* dan *test*. Proses perhitungan kesamaan teks dilakukan menggunakan model *Natural Language Processing* (NLP) berbasis *Transformers* dan algoritma *cosine similarity*. Evaluasi dilakukan dengan membandingkan rekomendasi sistem dengan validasi pakar. Metode pengujian melibatkan *confusion matrix* untuk mengevaluasi performa sistem. Hasil evaluasi menunjukkan akurasi 85%, *precision* 87%, *recall* 85%, dan *f1-score* rata-rata tertimbang sebesar 85%. Sistem ini tidak hanya menampilkan skor kesamaan, tetapi juga memberikan rekomendasi konkret berupa daftar mata kuliah yang relevan berdasarkan tingkat kesamaan tertinggi. Selain itu, sistem dikembangkan menjadi aplikasi yang dapat langsung digunakan oleh pihak kampus untuk mempermudah proses konversi SKS mahasiswa.

Kata kunci: Sistem Rekomendasi, Pemrosesan Bahasa Alami, *Cosine Similarity*, *Transformers*, Kecerdasan Buatan

Abstract

The Certified Independent Study and Internship Program (MSIB), part of the Merdeka Belajar Kampus Merdeka (MBKM) initiative, allows students to convert up to 20 credits (SKS) from their activities. However, challenges arise due to differences between the MSIB syllabus and the Informatics Engineering curriculum at ITN Malang. This research develops an AI-based system to recommend course conversions automatically. The dataset comprises 54 course descriptions and 17 MSIB syllabi. Using an open testing approach, each MSIB syllabus is compared with all courses without splitting data into training and testing sets. Text similarity is calculated with a Transformer-based NLP model and cosine similarity. Evaluation compares system recommendations with expert validation, utilizing a confusion matrix to assess accuracy (85%), precision (87%), recall (85%), and f1-score (85%). The system provides similarity scores and recommends courses based on the highest scores. Additionally, it has been developed into an application to assist the university in streamlining the SKS conversion process.

Keywords: Recommendation System, Natural Language Processing, Cosine Similarity, Transformers, Artificial Intelligence

1. PENDAHULUAN

Perguruan tinggi yang terdaftar dan telah bergabung menjadi mitra dalam program Merdeka Belajar Kampus Merdeka (MBKM) memiliki peran dalam menyediakan pengalaman belajar, keterampilan praktis, disiplin ilmu, membangun jaringan profesional, dan mempersiapkan diri untuk kebutuhan industri dan dunia kerja saat ini. Kesiapan karir mahasiswa sangat dipengaruhi oleh pengalaman yang diperoleh melalui kegiatan kampus merdeka, yang memberikan kesempatan untuk memanfaatkan pengetahuan yang telah diperoleh untuk diterapkan dalam lingkungan kerja profesional. Dalam kerangka program MSIB, mahasiswa memiliki kesempatan untuk mengonversi mata kuliah sebagai pengganti 20 SKS yang ditempuh dalam program MSIB yang sedang dijalani. Namun, proses konversi ini rumit dan memakan waktu, karena memerlukan pencocokan antara silabus kegiatan MSIB dan silabus mata kuliah yang relevan di beberapa

perguruan tinggi yang memiliki ketentuan khusus. Proses ini tidak bisa langsung dilakukan karena program yang diikuti harus *linear* dengan mata kuliah yang sedang ditempuh.

Bagi pihak kampus, pencocokan manual silabus memakan banyak waktu dan sumber daya, terutama jika terdapat ratusan mahasiswa yang mengikuti program MSIB. Oleh karena itu, diperlukan sistem yang mampu memberikan solusi otomatis, dan efisien untuk mencocokkan deskripsi silabus, sehingga mengurangi beban administratif kampus dan membantu mahasiswa dalam menyelesaikan proses konversi dengan lebih mudah. Penggunaan algoritma *cosine similarity* memungkinkan sistem untuk secara objektif dan konsisten mengukur kesamaan antara dua teks[1], sehingga meminimalkan potensi kesalahan dalam proses konversi. Algoritma ini dipilih karena kesederhanaan dan efisiensinya dalam menghitung kesamaan teks berbasis distribusi kata[2], yang sangat relevan untuk mencocokkan deskripsi silabus dalam format tidak terstruktur.

Beberapa studi sebelumnya telah dilakukan untuk menghitung kesamaan teks menggunakan metode *cosine similarity*. Salah satunya, studi yang dilakukan Febriyanti, Rini, dan Arsalan menggunakan metode *Cosine Similarity Measure* untuk mendeteksi tingkat kesamaan teks antara dokumen pengajuan hibah penelitian. Hasil penelitian menunjukkan bahwa nilai *Cosine Similarity* terendah yang dihasilkan adalah 0, yang berarti tidak ada kesamaan sama sekali antara dokumen yang dibandingkan. Sementara itu, nilai tertinggi yang dihasilkan adalah 0.763 [3]. Penelitian lain dilakukan oleh Kurniawan dan Arif (2022) untuk mengklasifikasikan berita berdasarkan tema dan isi. Hasilnya menunjukkan bahwa bobot kata memengaruhi akurasi klasifikasi, dimana konten berita tanpa kategori dihitung kesamaannya dengan berita yang sudah memiliki kategori untuk menentukan kemiripan tertinggi[4]. Penelitian oleh Heri Sutikno dan Saniati meneliti penggunaan algoritma *Cosine Similarity* dalam mendeteksi kesamaan topik pada judul skripsi. Hasilnya menunjukkan tingkat kemiripan 100% untuk judul identik, 85% untuk judul dengan tiga kata dihapus, dan 92% untuk judul dengan sinonim[5]. Teknologi *Text Mining* dan NLP memungkinkan sistem untuk membaca, menganalisis, dan mengekstraksi informasi penting[6] dari teks silabus yang tidak terstruktur.

Dengan pendekatan *cosine similarity*, sistem ini memproses data silabus MSIB dan mata kuliah kampus. Proses dimulai dengan *preprocessing* teks, konversi ke vektor menggunakan *Transformers*, dan perhitungan tingkat kesamaan antar dokumen berdasarkan algoritma *cosine similarity*. Dataset yang digunakan terdiri dari 17 silabus MSIB dan 54 deskripsi mata kuliah. Sistem ini akan menghasilkan skor kesamaan untuk setiap pasangan dokumen dan merekomendasikan mata kuliah dengan skor tertinggi untuk dikonversi.

2. METODE PENELITIAN

Metode penelitian ini dirancang untuk mengatasi tantangan dalam mencocokkan silabus kegiatan MSIB dengan mata kuliah kampus. Sistem rekomendasi dirancang berdasarkan analisis kesamaan deskripsi silabus menggunakan pendekatan NLP dan algoritma *cosine similarity*, adapun tahapan dalam penelitian ini tertera pada gambar 1.



Gambar 1. Alur Metode Penelitian

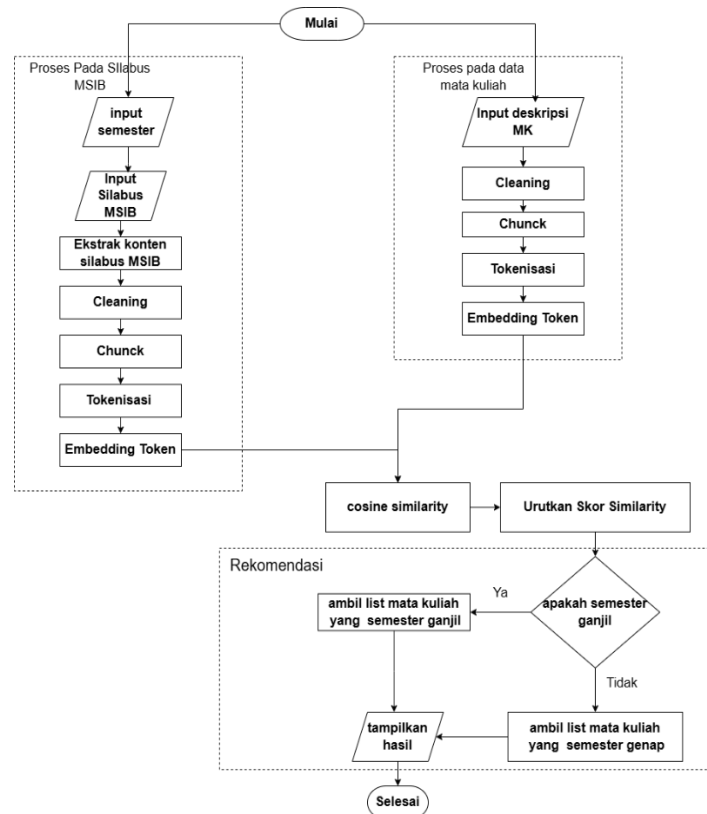
2.1. Pengumpulan Data

Sumber utama data dalam penelitian ini terdiri dari dua bagian. Pertama, data terkait silabus program MSIB dikumpulkan dari total 17 mahasiswa yang mengikuti program tersebut pada periode Agustus–Desember 2023 dan Februari–Juni 2024. Proses pengumpulan data dilakukan melalui wawancara dengan pihak mahasiswa yang memberikan informasi mengenai dokumen silabus MSIB serta mata kuliah yang berhasil mereka konversi. Kedua, data mata kuliah (MK) yang digunakan untuk rekomendasi sistem konversi yang diambil dari kurikulum Prodi Teknik

Informatika ITN Malang tahun 2019–2024, yang mencakup mata kuliah yang terstruktur dari tahun pertama hingga keempat.

2.2. Pengolahan Data

Alur pengolahan data untuk sistem rekomendasi konversi digambarkan pada gambar 2. *Flowchart* ini bertujuan untuk memberikan gambaran langkah-langkah atau prosedur program[7]. Selain bertujuan untuk memberikan gambaran alur, *flowchart* juga digunakan untuk memecahkan masalah kedalam segmen kecil[8].



Gambar 2. *Flowchart* Metode

Penjelasan untuk tahapan memperoleh rekomendasi konversi mata kuliah :

a. Input Data

Tahap pertama yang dilakukan adalah penulis menginputkan data yang dibutuhkan. Data Silabus MSIB dan data deskripsi Mata Kuliah untuk dibandingkan. Kemudian menginputkan data semester mahasiswa.

b. Cleaning

Tahap yang bertujuan untuk mengonversi data mentah menjadi format yang dapat langsung digunakan untuk analisis[9]. Proses ini mencakup beberapa langkah: menghapus karakter khusus seperti URL, email, tanda baca, dan emoji. Selanjutnya, dilakukan *case folding* untuk mengonversi seluruh huruf menjadi *lowercase*[10]. Tujuan dari langkah ini adalah untuk menyamakan format teks sehingga teks yang diproses memiliki format yang seragam[11]. Kemudian, tahap terakhir adalah menghapus *stopword*, yaitu kata umum yang dianggap tidak memiliki makna penting[12].

c. Chunk

Proses ini merupakan metode untuk membagi setiap kalimat menjadi beberapa *segmen*[13]. *Chunking* pada tahap ini dapat membantu model memahami teks input dengan lebih baik[14]. Hal ini penting karena model *transformer* yang digunakan memiliki batas maksimal pemrosesan

hingga 384 token. Oleh karena itu, metode ini sangat dibutuhkan untuk mengolah data silabus MSIB yang umumnya memiliki konten dengan lebih dari 1000 kata.

d. Tokenisasi

Tokenisasi adalah proses memecah string *input* menjadi bagian-bagian yang lebih kecil. Langkah ini dilakukan untuk memisahkan kata dalam kalimat disebuah dokumen[15]. *Tokenizer* berperan dalam menganalisis struktur dan konten teks dengan lebih rinci [16]. Tujuannya agar sistem mudah mengubah tiap kata/sub-kata menjadi representasi *vector*.

e. *Embedding Token*

Token embedding dilakukan dengan tujuan mengubah token menjadi representasi numerik, yaitu dalam bentuk vektor. Tahap ini dilakukan menggunakan model *Transformer*, yang merupakan bagian dari *Deep Neural Networks* (DNN) dan menggunakan mekanisme *self-attention* untuk menangkap hubungan kontekstual dalam data sekuensial pada teks[17]. Arsitektur *transformer* terdiri dari pengkode (*encoder*) dan pengurai (*decoder*)[18]. Yang pada penelitian ini hanya *encoder* saja yang digunakan karena tujuannya hanya untuk *embedding* (representasi vektor) tiap token.

f. *Cosine Similarity*

Hasil *embedding* dari deskripsi silabus MSIB dan mata kuliah dibandingkan tingkat kesamaan menggunakan *Cosine Similarity*. *Cosine similarity* mengukur kesamaan antara dua vektor berdasarkan sudutnya. Semakin kecil sudut, semakin mirip keduanya. Ini sering digunakan untuk mengukur kesamaan makna antar objek[19]. *Cosine similarity* dijabarkan dengan rumus(1) berikut [20]:

$$Similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (1)$$

g. Rekomendasi *Output*

Skor kesamaan dari perhitungan *cosine similarity* digunakan untuk merekomendasikan mata kuliah yang paling relevan dengan kegiatan MSIB. Hasil perhitungan ini disusun secara berurutan berdasarkan skor tertinggi. Mata kuliah dengan tingkat kesamaan tertinggi ditempatkan diposisi teratas sebagai rekomendasi utama.

2.3. Valiasi

Validasi dilakukan dengan membandingkan hasil rekomendasi sistem dengan hasil yang sudah dievaluasi oleh pakar dalam bidang terkait. Validasi dilakukan dengan memanfaatkan dataset yang sudah diberi label oleh pakar, dimana label tersebut menunjukkan apakah rekomendasi yang diberikan oleh sistem benar atau tidak [21]. Label 0 menunjukkan bahwa rekomendasi sistem tidak sesuai, sementara label 1 menunjukkan bahwa rekomendasi sistem sesuai dengan harapan.

2.4. Evaluasi

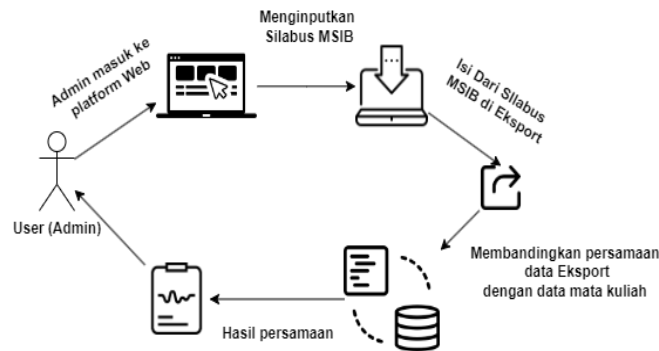
Evaluasi sistem rekomendasi ini dilakukan untuk mengukur seberapa baik kinerja model [22] dalam memberikan rekomendasi mata kuliah yang sesuai dengan menggunakan *confusion matrix*. Metrik yang digunakan dalam evaluasi adalah akurasi, *precision*, *recall*, dan *F1-score*.

- Presisi mengukur sejauh mana rekomendasi yang relevan (sesuai) dari seluruh rekomendasi yang diberikan oleh sistem.
- Recall* mengukur seberapa banyak rekomendasi relevan yang berhasil ditemukan dari semua mata kuliah yang seharusnya direkomendasikan
- F1-score* memberikan memberikan gambaran seimbang antara *presisi* dan *recall* untuk menilai keseimbangan keduanya
- Akurasi mengukur persentase hasil rekomendasi yang benar dari keseluruhan prediksi.

3. HASIL DAN PEMBAHASAN

3.1 Arsitektur Sistem

Sistem terdiri dari berbagai fungsi yang saling terintegrasi, yang ditunjukkan dengan garis atau alur data antara setiap blok.

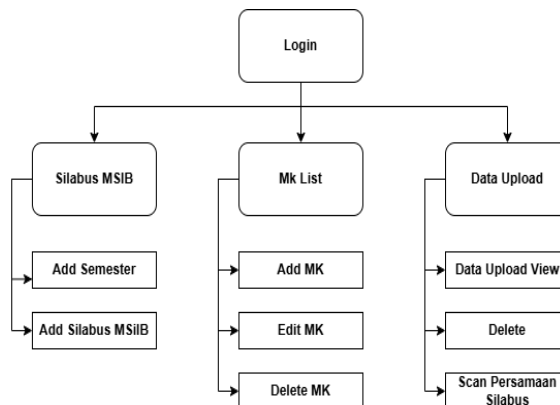


Gambar 3. Design Arsitektur Sistem

Pada gambar 3 arsitektur sistem terbagi menjadi tiga bagian utama, diantaranya: *input*, proses, dan *output*. Pertama, pengguna diminta untuk menginputkan data silabus MSIB dan semester yang diikuti. Data tersebut kemudian diproses oleh sistem untuk dihitung tingkat kesamaan antara silabus MSIB dengan data mata kuliah yang ada. Terakhir, hasil dari proses perbandingan tersebut ditampilkan sebagai *output* berupa rekomendasi mata kuliah

3.2 Struktur Menu

Struktur menu pada sistem ini digambarkan pada gambar 4. Menu Silabus MSIB mencakup tindakan tambah semester dan tambah silabus MSIB, sedangkan Menu Mk List berisi tindakan tambah, edit, dan hapus mata kuliah. Menu Data Upload menyediakan fitur lihat data, hapus, dan *scan* persamaan silabus. Pada menu *scan* persamaan silabus inilah metode dijalankan untuk mendapat rekomendasi konversi.



Gambar 4. Struktur Menu

3.3 Input Data

Seperti yang telah dijelaskan pada *flowchart* metode pada gambar 2 bahwa dibutuhkan 3 inputan yaitu inputan semester, data silabus MSIB dan data deskripsi MK. Data-data ini disimpan dalam *database* MongoDB Atlas.

a. Data Silabus MSIB

Data disimpan dalam koleksi bernama *upload* yang terdiri dari beberapa *field*. *Field _id* digunakan sebagai identifikasi unik, sedangkan semester mencatat semester mahasiswa. Field lainnya meliputi *file_name* untuk nama *file* PDF, *file_size* untuk jumlah kata dalam dokumen, *date* untuk waktu unggah, dan *extracted_text* untuk teks yang diekstrak dari *file*.

```
_id: ObjectId('67550b1a30e7d3b482763a18')
semester : "genap"
filename : "Silabus Program Studi Independen MSIB 6 - Educourse.id.pdf"
content_type : "application/pdf"
file_size : 3697
date : "2024-12-08"
extracted_text : "Silabus Pembelajaran
                  PT Maleo Edukasi Teknologi
                  2024
                  Program Studi Ind..."
page_count : 16
```

Gambar 5. Data Silabus MSIB

Pada gambar 5 adalah contoh salah satu tampilan data silabus MSIB yang berhasil disimpan. Pada koleksi *upload* ini data yang paling penting adalah data semester dan *extracted_text*. *Field* semester digunakan sebagai indikator untuk menentukan rekomendasi mata kuliah sesuai dengan semester mahasiswa tersebut.

b. Data Mata Kuliah

Data mata kuliah disimpan dalam koleksi silabus yang mencakup beberapa *field*, seperti *id_mk* untuk ID unik dan *mk* untuk nama mata kuliah. *Field* lainnya mencatat semester pengajaran, *sks* sebagai bobot studi, dan deskripsi mengenai materi yang dipelajari. Selain itu, *field date* digunakan untuk mencatat tanggal input mata kuliah.

```
_id: ObjectId('671e98180bb3e9b9ae9a64a3')
id_mk : "IF 1202"
mk : "Kalkulus"
semester : 1
sks : 2
deskripsi : "<p><span data-sheets-root='1'>Deskripsi </span><span data-sheets-root='..."
image_path : "https://img.freepik.com/free-photo/learning-education-mathematics-calc..."
date : "2024-10-28"
```

Gambar 6. Koleksi Database Mata Kuliah

Pada Gambar 6 menunjukkan salah satu tampilan data mata kuliah yang telah diinputkan dan tersimpan dalam *database*. *Field* deskripsi merupakan data utama yang digunakan untuk perbandingan *similarity* dengan silabus MSIB. Deskripsi mata kuliah berisi penjelasan mengenai isi, tujuan, dan cakupan materi yang diajarkan dalam mata kuliah tersebut.

3.4 Cleaning

Hasil proses ini adalah teks yang telah dibersihkan dari karakter khusus, tanda baca, angka, dan *stopwords*, serta diubah menjadi huruf kecil. Hasil *preprocessing* ditampilkan pada tabel 1 pada kolom sebelah kanan. Hasil tersebut menunjukkan bahwa elemen seperti *stopwords*, karakter khusus, dan tanda baca berhasil dihilangkan dari teks asli.

Tabel 1. Hasil Proses Cleaning

Data Silabus MSIB atau Mata Kuliah	Hasil data setelah preprocessing
"""Program Description \nIntro to Python for Data Science. Program berbasis keterampilan ini ditujukan untuk learner, yang ingin belajar \npython dan ingin menerapkan statistik, machine learning, """	['program description intro to python for data science program berbasis keterampilan learner belajar python menerapkan statistik machine learning']

3.5 Chunk

Setelah tahap *cleaning* selesai, dilakukan pemisahan teks menjadi beberapa *segmen* melalui metode yang disebut *chunking*. Sebagai contoh, pada table 2 *chunking* dilakukan dengan membagi teks yang awalnya berupa satu kalimat menjadi empat kalimat terpisah, menggunakan tanda koma sebagai pemisah.

Tabel 2. Hasil Proses Chunk

Text Hasil Cleaning	Hasil Chunk
'program description intro to python for data science program berbasis keterampilan learner belajar python menerapkan statistik machine learning'	['program description intro to python', 'for data science program berbasis', 'keterampilan learner belajar python menerapkan', 'statistik machine learning']

3.6 Tokenisasi

Kemudian tiap kalimat diproses lagi untuk dipecah menjadi token, agar lebih mudah dianalisis oleh model *transformers*. Pada gambar 7 hasil tokenisasi menunjukkan daftar token dan subtoken dari tiap kalinat. Token-token tersebut selanjutnya digunakan untuk perhitungan *embedding*.

```
Kalimat 1: ['<s>', '_program', '_description', '_intro', '_to', '_py', 'thon', '</s>', '<pad>']
Kalimat 2: ['<s>', '_for', '_data', '_science', '_program', '_berbasis', '</s>', '<pad>', '<pad>']
Kalimat 3: ['<s>', '_keterampilan', '_learn', 'er', '_belajar', '_py', 'thon', '_menerapkan', '</s>']
```

Gambar 7. Hasil Proses Tokenisasi

3.7 Embedding Token

Setelah proses tokenisasi, setiap token dipetakan ke dalam ruang vektor berdimensi tinggi menggunakan model berbasis *Transformer*. Proses ini bertujuan menangkap makna dan hubungan antar kata dalam konteks tertentu. Gambar 8 di bawah ini menunjukkan representasi numerik berupa vektor yang merepresentasikan konteks dan isi teks.

```
tensor([[ -0.0077,  0.2112, -0.1019, -0.0933, -0.0326,  0.0303,  0.1121,  0.0521,
          -0.0410,  0.2969, -0.0253, -0.1964,  0.2676, -0.1299, -0.4213, -0.1816,
           0.1301,  0.3261, -0.0791, -0.3912, -0.2421,  0.0180,  0.0198,  0.0025,
          -0.1456, -0.0133, -0.0794,  0.0588,  0.2481, -0.0183,  0.1731,  0.1331,
           0.6201,  0.2004, -0.2118,  0.3534, -0.1934,  0.1907, -0.2173,  0.1714,
           0.0945, -0.1209,  0.1179,  0.1874, -0.2397, -0.2039, -0.1711, -0.2045,
          -0.1072, -0.2360, -0.4510, -0.0749, -0.2231, -0.5838,  0.0794,  0.0204,
           0.2615,  0.1300, -0.2959,  0.0522, -0.2361,  0.0744, -0.1123,  0.0956,
           0.3222,  0.3295,  0.0613,  0.3524,  0.2623, -0.2351, -0.2194,  0.2377,
          -0.2847,  0.0623, -0.3337,  0.3334,  0.0159,  0.2155,  0.0680, -0.0428,
          -0.0331,  0.0626,  0.0156,  0.0681,  0.1124,  0.0532,  0.1200, -0.0007,
          -0.0861, -0.0788,  0.0156,  0.0096, -0.0770,  0.2026, -0.3063,  0.1887,
          -0.0101, -0.3665,  0.1277,  0.1421, -0.2180, -0.1130, -0.3079,  0.1088,
           0.0466, -0.0979,  0.0729, -0.1844,  0.0373, -0.2680,  0.3310,  0.0675,
          -0.1320, -0.0335,  0.1030, -0.0306, -0.0264, -0.0433, -0.0400,  0.3566,
           0.0969,  0.0395,  0.0031,  0.0359,  0.2581, -0.2613, -0.0939, -0.1809,
          -0.0325, -0.3957,  0.4587, -0.0580,  0.0419,  0.0666, -0.0799, -0.1169,
           0.1375,  0.0822, -0.2109,  0.3517,  0.1272,  0.1041, -0.0363,  0.0206,
```

Gambar 8. Hasil Embedding Token

3.8 Rekomendasi Output

a. Pengujian Sistem

Pengujian dilakukan menggunakan metode *open testing* dengan data silabus MSIB sebagai input. Sistem ini tidak membagi data ke dalam *train* dan *test set* karena berbasis *text similarity* yang tidak memerlukan proses pelatihan. Sebaliknya, seluruh data langsung digunakan untuk menguji performa sistem. Pengujian melibatkan 17 silabus MSIB, dimana sistem menghasilkan rekomendasi mata kuliah berdasarkan perhitungan *cosine similarity* terhadap hasil *embedding* token silabus MSIB dengan deskripsi mata kuliah. Data yang diuji bersifat *unlabeled*, dan hasil rekomendasi dinilai berdasarkan kecocokan yang dihasilkan oleh algoritma.

Semester Learning and Assessment Plan (SLAP)	
	Minggu ke- : 1
	Topik : Introduction to Educational Platform
	Waktu : 60 jam
Target Hasil Pembelajaran	
- Tinjauan teknologi pendidikan dan perannya dalam lingkungan pembelajaran modern. - Memahami komponen-komponen dari sebuah platform pendidikan. - Pengenalan tentang HTML, CSS, dan JavaScript untuk pengembangan web. - Menyiapkan lingkungan pengembangan dasar.	
Detail Materi Pembelajaran	
- Pengenalan tentang HTML - Memahami struktur dasar HTML dan tag-tag umum yang digunakan dalam pembuatan halaman web. - Praktik membuat halaman web sederhana dengan elemen dasar menggunakan HTML. - Pengantar CSS	

Gambar 9. Tampilan Silabus Program MSIB Berformat PDF

Pada Gambar 9 adalah salah satu contoh data silabus program MSIB yang berformat PDF yang diuji. Setelah data silabus dan deskripsi mata kuliah melalui proses *cleaning* hingga *embedding*, langkah selanjutnya adalah membandingkan hasil *embedding* silabus MSIB dengan setiap mata kuliah dalam *database*. Perbandingan ini dilakukan satu per satu untuk mencari skor kemiripan tertinggi menggunakan *cosine similarity*. Dengan demikian, sistem dapat menentukan mata kuliah mana yang paling relevan dengan silabus MSIB yang telah diinputkan.

IF 8113 : Technopreneurship similarity yang didapatkan adalah 0.8879173985830914	IF 4306 : Pemrograman Web similarity yang didapatkan adalah 0.8837049225993576	IF 2109 : Pendidikan Bahasa Indonesia similarity yang didapatkan adalah 0.8697914488527072
IF 8609 : Manajemen Proyek Perangkat Lunak similarity yang didapatkan adalah 0.8676006882089715	IF 4602 : Rekayasa Perangkat Lunak similarity yang didapatkan adalah 0.8439059556036612	IF 6607 : Penjaminan Mutu Perangkat Lunak similarity yang didapatkan adalah 0.8196824309934969

Gambar 10. Hasil Rekomendasi konversi

Pada Gambar 9 menampilkan hasil rekomendasi *output* sistem berupa daftar mata kuliah dan skor *similarity* yang direkomendasikan untuk konversi. Rekomendasi diurutkan berdasarkan skor *similarity* dari yang terbesar hingga terkecil. *Output* ini menghasilkan 23 rekomendasi konversi mata kuliah pada semester genap sesuai dengan input semester mahasiswa. Daftar lengkapnya disajikan pada tabel 3.

Tabel 3. Hasil Lengkap Rekomendasi

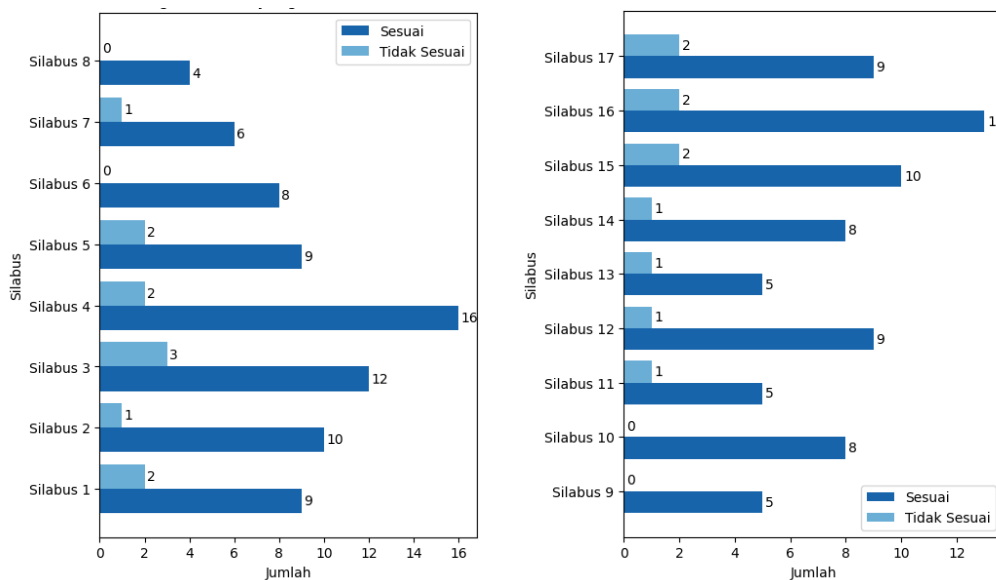
id_mk	Mata Kuliah	Semester	Skor <i>Similarity</i>
IF 8113	Technopreneurship	8	0,8879174168
IF 4306	Pemrograman Web	4	0,8837049059
IF 2109	Pendidikan Bahasa Indonesia	2	0,8697914507
IF 8609	Manajemen Proyek Perangkat Lunak	8	0,8676006329
IF 4602	Rekayasa Perangkat Lunak	4	0,8439060075
IF 6607	Penjaminan Mutu Perangkat Lunak	6	0,8196824176
IF2601	Basis Data	2	0,8135758366
IF 6404	Data Mining	6	0,7818078098
IF2202	Aljabar Linear	2	0,7692150941
IF 4802	Manajemen Jaringan	4	0,7672217096
IF 6109	Metodologi Penelitian	6	0,7397450709
IF 2202	Struktur Data	2	0,7206240149
IF 6505	Komputer Vision	4	0,7183035278
IF 4401	Kecerdasan Buatan	4	0,6930530565
IF 6706	Internet of Things (Kapita Selecta)	6	0,56223286
IF 6504	Animasi dan Game	6	0,5404691758
IF 4603	Sistem Informasi	4	0,5330104461
IF 4206	Metode Numerik	4	0,5326382737

IF 2703	Arsitektur Komputer	2	0,5115580292
IF2303	Teknik Pemrograman	2	0,4947961572
IF 6403	Sistem Pakar	6	0,4828341902
IF 2704	Sistem Operasi	2	0,4727890617
IF 4305	Pemrograman Visual	4	0,3048355315

Seperti yang tercantum pada table 3, tabel ini menyajikan hasil rekomendasi sistem beserta informasi terkait, seperti ID mata kuliah, nama mata kuliah, semester, dan skor *similarity*. Pada kolom skor *similarity*, semakin tinggi nilainya maka semakin relevan deskripsi mata kuliah dengan silabus MSIB yang sedang diujikan, dan sebaliknya.

3.9 Validasi Pakar

Untuk memastikan bahwa model benar-benar sesuai dengan kebutuhan dan memberikan hasil yang relevan, hasil rekomendasi dari 17 data silabus yang telah diuji dibandingkan dengan validasi dari pakar. Hasil validasi dapat diamati pada visualisasi gambar 11.



Gambar 11. Hasil Validasi Pakar

Visualisasi pada gambar 11 menunjukkan bahwa hasil validasi dari 17 silabus yang diuji menunjukkan tingkat kecocokan yang tinggi dengan rata-rata jumlah rekomendasi yang sesuai sebesar 8,59 per silabus dan rata-rata tidak sesuai sebesar 1,12. Silabus dengan tingkat kecocokan tertinggi adalah *Silabus 4* dengan 16 rekomendasi yang sesuai, sedangkan yang terendah adalah *Silabus 8* dengan hanya 4 rekomendasi yang sesuai. Data ini mengindikasikan bahwa sistem mampu menghasilkan rekomendasi yang relevan dengan validasi pakar.

3.10 Evaluasi

Penulis melakukan evaluasi akurasi sistem rekomendasi dengan menggunakan *confusion matrix* untuk mengukur sejauh mana sistem mampu memberikan rekomendasi mata kuliah yang dapat dikonversi bagi mahasiswa yang mengikuti program MSIB. Evaluasi ini dilakukan dengan membandingkan hasil rekomendasi sistem dengan data validasi dari pakar yang telah diberikan label 0 dan 1. Label 0 menunjukkan kelas untuk hasil rekomendasi yang tidak sesuai, sedangkan label 1 menunjukkan kelas untuk hasil rekomendasi yang sesuai. Evaluasi dilakukan menggunakan *threshold similarity* sebesar 0.7, yang berarti hanya pasangan teks dengan skor kemiripan 0.7 atau lebih tinggi dianggap relevan atau sesuai. Detail hasil evaluasi untuk seluruh silabus dapat ditemukan pada tabel 4.

Tabel 4. Hasil Evaluasi

Silabus	Precision (0)	Recall (0)	F1-Score (0)	Precision (1)	Recall (1)	F1-Score (1)	Akurasi
Silabus 1	0.88	0.79	0.83	0.69	0.82	0.75	0.80
Silabus 2	0.90	0.75	0.82	0.77	0.91	0.83	0.83
Silabus 3	0.89	0.67	0.76	0.71	0.91	0.80	0.78
Silabus 4	0.67	0.75	0.71	0.86	0.80	0.83	0.78
Silabus 5	0.85	0.92	0.88	0.94	0.89	0.91	0.90
Silabus 6	0.85	0.92	0.88	0.90	0.82	0.86	0.87
Silabus 7	1.00	0.60	0.75	0.57	1.00	0.73	0.74
Silabus 8	0.92	0.75	0.83	0.60	0.86	0.71	0.78
Silabus 9	0.89	0.89	0.89	0.82	0.82	0.82	0.87
Silabus 10	1.00	0.86	0.93	0.73	1.00	0.84	0.90
Silabus 11	1.00	0.86	0.93	0.73	1.00	0.84	0.90
Silabus 12	0.94	1.00	0.97	1.00	0.83	0.91	0.96
Silabus 13	0.92	0.92	0.92	0.90	0.90	0.90	0.91
Silabus 14	0.96	0.92	0.94	0.71	0.83	0.77	0.90
Silabus 15	0.90	0.64	0.75	0.62	0.89	0.73	0.74
Silabus 16	0.88	0.83	0.86	0.77	0.83	0.80	0.83
Silabus 17	0.88	0.93	0.90	0.93	0.87	0.90	0.90

Tabel 4 menunjukkan hasil evaluasi sistem rekomendasi untuk 17 silabus berdasarkan metrik *precision*, *recall*, *F1-score*, dan akurasi untuk dua kategori kelas yaitu kelas 0 (tidak sesuai) dan kelas 1 (sesuai). Berdasarkan hasil evaluasi, akurasi keseluruhan sistem adalah 85%. Rata-rata performa sistem secara keseluruhan pada metrik *precision* dan *recall* masing-masing mencapai 87% dan 85%, dengan nilai rata-rata *F1-score* sebesar 85%. Hasil evaluasi juga menunjukkan bahwa rata-rata *F1-score* untuk kelas 0 lebih tinggi yaitu 86% dibandingkan dengan kelas 1 yang sedikit lebih rendah yaitu 82%. Hal ini mengindikasikan bahwa secara keseluruhan, sistem lebih baik dalam mengenali kelas 0 dibandingkan kelas 1. Performa yang lebih rendah pada kelas 1 disebabkan oleh ketidakseimbangan data, dimana jumlah sampel kelas 1 lebih sedikit dibandingkan dengan kelas 0. Ketidakseimbangan ini membuat model lebih cenderung memprioritaskan pola dari kelas 0, sehingga memengaruhi kemampuan model dalam memprediksi kelas 1 dengan baik.

4. KESIMPULAN DAN SARAN

Sistem rekomendasi berbasis *cosine similarity* yang dikembangkan dalam penelitian ini berhasil membantu mahasiswa MSIB dalam memilih mata kuliah yang dapat dikonversi berdasarkan deskripsi silabus. Sistem ini mampu memberikan rekomendasi dengan akurasi sebesar 85%, didukung oleh validasi pakar yang menunjukkan tingkat kesesuaian hasil rekomendasi mencapai 82%. Dengan kemampuan mencocokkan deskripsi teks secara otomatis, sistem ini mempermudah proses konversi mata kuliah sebelumnya dikerjakan dengan cara manual dan memerlukan waktu yang lama. Hasil ini mengindikasikan bahwa sistem mampu memberikan rekomendasi yang andal, meskipun masih diperlukan optimasi untuk meningkatkan performa pada kelas minoritas (kelas 1). Untuk mengatasi kekurangan ini, disarankan untuk menangani ketidakseimbangan data, misalnya melalui teknik *oversampling* atau *undersampling*. Selain itu, penggunaan model yang lebih kompleks atau *tuning parameter* lebih lanjut dapat membantu memperbaiki performa sistem, khususnya dalam mendeteksi kelas 1 dengan lebih akurat. Penelitian selanjutnya juga dapat memperluas sistem dengan menerapkan metode pembelajaran mesin berbasis *supervised* untuk meningkatkan akurasi dan relevansi rekomendasi. Selain itu, penambahan dataset dari program MSIB yang lebih beragam dapat memperkaya kemampuan sistem dalam mencocokkan deskripsi mata kuliah secara lebih luas dan komprehensif.

DAFTAR PUSTAKA

- [1] Y. Januzaj and A. Luma, "Cosine Similarity – A Computing Approach to Match Similarity Between Higher Education Programs and Job Market Demands Based on Maximum

- Number of Common Words,” *International Journal of Emerging Technologies in Learning*, vol. 17, no. 12, pp. 258–268, 2022.
- [2] T. P. Rinjeni, A. Indriawan, and N. A. Rakhmawati, “Matching Scientific Article Titles using Cosine Similarity and Jaccard Similarity Algorithm,” *Procedia Computer Science*, vol. 234, no. 2023, pp. 553–560, 2024.
- [3] N. Febriyanti, D. P. Rini, and O. Arsalan, “Text Similarity Detection Between Documents Using Case Based Reasoning Method with Cosine Similarity Measure (Case Study SIMNG LPPM Universitas Sriwijaya),” *Sriwijaya Journal of Informatics and Applications*, vol. 3, no. 2, pp. 85–94, 2022.
- [4] Rio Feriangga Kurniawan, “Implementasi Text Mining Menggunakan Metode Cosine Similarity Untuk Klasifikasi Konten Berita Di Postingan Grup Facebook Info Lantas Dan Kriminal Pasuruan,” *JAMI: Jurnal Ahli Muda Indonesia*, vol. 3, no. 1, pp. 9–17, 2022.
- [5] H. S. Sutikno, “Implementasi Algoritma Cosine Similarity Untuk Mendeteksi Kemiripan Topik Judul,” *Journal of Engineering, Computer Science and Information Technology (JECSIT)*, vol. 1, no. 1, pp. 51–61, 2021.
- [6] M. J. Farrell, L. Brierley, A. Willoughby, A. Yates, and N. Mideo, “Past and future uses of text mining in ecology and evolution,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 289, no. 1975, 2022, pp. 1-19.
- [7] M. A. Julyananda, T. Yulianti, and D. Pasha, “Rancang Bangun Media Pembelajaran Matematika Menggunakan Metode Demonstrasi Untuk Kelas 1 Sekolah Dasar,” *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 3, no. 3, pp. 366–375, 2022.
- [8] A. Sutanti, M. Komaruddin, P. Damayanti, and P. U. Studi Sistem Informasi Metro, “Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur,” *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 9, no. 1, pp. 366-375, 2022.
- [9] A. Q. Surbakti, R. Hayami, and J. A. Amien, “Analisa Tanggapan Terhadap Psbb Di Indonesia Dengan Algoritma Decision Tree Pada Twitter,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 2, no. 2, pp. 91–97, 2021.
- [10] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, “Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19),” *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, pp. 406-416, 2021.
- [11] J. Mantik, Y. R. Saputri, and H. Februariyanti, “Sentiment Analysis on Shopee E-Commerce Using the Naïve Bayes Classifier Algorithm,” *Jurnal Mantik*, vol. 6, no. 2, pp. 1349–1357, 2022.
- [12] F. Amin, E. Nur Wahyudi, and B. Hartono, “Penerapan Kemiripan Dokumen pada Mesin Pencari Menggunakan Metode Hellinger,” *Jurnal Rekayasa Industri (JRI)*, vol. 5, no. 2, pp. 118–125, 2023.
- [13] M. D. Putri, A. Rahmat, and Y. Sanjaya, “The Use of Chunking Technique Combined with the Writing Is Thinking Technique to Control Students’ Cognitive Load When Learning About the Human Reproductive System,” *KnE Social Sciences: International Conference on Matematics and Science Education (ICMScE 2022): Learning Materials and Modules*, 2024, pp. 83–90.
- [14] J. Risakotta, “Penerapan Chunking Strategy Untuk Meningkatkan Kemampuan Memahami Teks Dalam Bahasa Inggris Pada Smk Kesehatan Nusaniwe Ambon,” *VOCATIONAL: Jurnal Inovasi Pendidikan Kejuruan*, vol. 2, no. 4, pp. 327–334, 2023.
- [15] D. Soyusiawaty, “Buku Ajar Pemrosesan Bahasa Alami,” *Universitas Ahmad Dahlan*, pp. 1–99, 2023.
- [16] A. N. Oktavia, *at al.*, “Implementasi Metode Natural Language Processing Dalam Studi Analisis,” *Jurnal Ilmiah Ilmu Komputer dan Multimedia*, vol. 2, no. 1, pp. 154–159, 2024.
- [17] S. Islam *et al.*, “A comprehensive survey on applications of transformers for deep learning tasks,” *Expert Syst Appl*, vol. 241, 2024.
- [18] D. H. Putra, and G. W. I. Noto, “Implementasi Generative Pre-Trained Transformers 2 Untuk Mengoreksi Kesalahan Penulisan Bahasa Indonesia Pada Dokumen Jurnal,” *Seminar Nasional Technopex-2023, Institute Teknologi Indonesia*, 2023, pp. 318–323.

- [19] H. Steck, C. Ekanadham, and N. Kallus, "Is Cosine-Similarity of Embeddings Really About Similarity?," *WWW 2024 Companion-Proceedings of the ACM Web Conference*, no. Llm, pp. 887–890, 2024.
- [20] L. Meilina, I. N. S. Kumara, and I. N. Setiawan, "Literature Review Klasifikasi Data Menggunakan Metode Cosine Similarity dan Artificial Neural Network," *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 2, pp. 307-314, 2021.
- [21] D. G. S. Gogahu and T. Prasetyo, "Pengembangan Media Pembelajaran Berbasis E-Bookstory untuk Meningkatkan Literasi Membaca Siswa Sekolah Dasar," *Jurnal Basicedu*, vol. 4, no. 4, pp. 1004–1015, 2020.
- [22] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci Rep*, vol. 12, no. 1, pp. 1–9, 2022.