

Sistem Rekomendasi Konversi Mata Kuliah Berbasis NLP Menggunakan Metode BERT Di Universitas Islam Balitar

Moh Hafid Nur Firmansyah¹,Saiful Nur Budiman², Sabitul kirom³

^{1,2,3}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Islam Balitar

E-mail: ¹hafidnurfirmansyah10@gmail.com, ²sync.saifulnb@gmail.com, ³sabitulkirrom@gmail.com

Abstrak – Konversi mata kuliah antar institusi seringkali menjadi tantangan karena adanya ketidaksesuaian dalam deskripsi mata kuliah serta proses manual yang memakan waktu yang cukup lama karena tingkat dengan kompleks nya deskripsi yang harus dicocokan untuk melakukan konversi. Penelitian ini bertujuan untuk menghadirkan solusi melalui pengembangan sistem rekomendasi berbasis teknologi Bidirectional Encoder Representations from Transformers (BERT). Tahapan penelitian dimulai dengan pengumpulan data deskripsi mata kuliah dari dokumen akademik resmi, diikuti dengan proses pengolahan data, seperti cleaning, tokenisasi, penghapusan kata-kata umum (stopwords), dan stemming. Model BERT yang telah disesuaikan digunakan untuk menghasilkan representasi vektor dari deskripsi mata kuliah. Tingkat kesesuaian antar mata kuliah diukur menggunakan metode cosine similarity. Evaluasi dilakukan dengan membandingkan hasil rekomendasi otomatis dengan hasil manual yang diberikan oleh staf akademik..Hasil penelitian menunjukkan bahwa sistem berbasis BERT mampu memberikan rekomendasi kesesuaian mata kuliah yang relevan dan konsisten dengan penilaian manual, sekaligus secara signifikan mengurangi waktu yang dibutuhkan. Dengan demikian, sistem ini tidak hanya memberikan solusi praktis untuk konversi mata kuliah, tetapi juga membuka peluang penerapan lebih luas dari teknologi NLP di dunia pendidikan.

Kata Kunci — BERT ,Machine Learning ,NLP, Sistem Rekomendasi

1. PENDAHULUAN

Dalam era globalisasi pendidikan tinggi, mobilitas mahasiswa antar institusi pendidikan semakin meningkat. Fenomena ini mencakup transfer mahasiswa antar universitas, program magang atau studi independent mandiri, dan kolaborasi antar institusi pendidikan. Sebagai konsekuensinya, proses konversi matakuliah menjadi aspek krusial dalam menjamin kelancaran transisi akademik mahasiswa. Namun, proses ini seringkali rumit, memakan waktu, dan rentan terhadap inkonsistensi karena umumnya dilakukan secara manual oleh staf akademik.

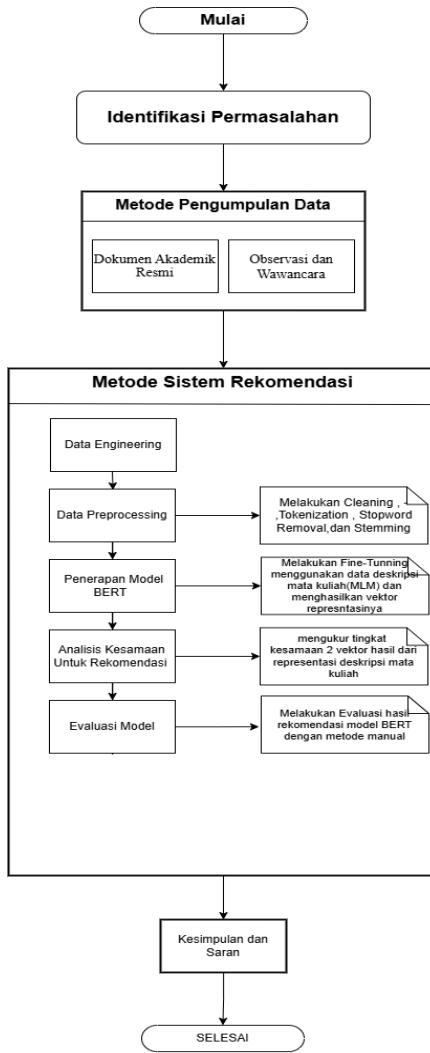
Proses konversi mata kuliah merupakan suatu langkah yang penting dalam menentukan kesesuaian antara modul yang ingin dikonversi yang berasal dari institusi yang berbeda.Dalam proses ini banyak tantangan yang harus dihadapi mulai variasi deskripsi modul yang sangat beragam,standar capaian pembelajaran , serta waktu yang harus digunakan dalam proses konversi yang memakan waktu cukup lama.Penelitian ini akan menerapkan sebuah metode *Natural Language Processing (NLP)* berbasis model BERT untuk melakukan analisis kesesuaian antar deskripsi mata kuliah dengan modul yang akan dilakukan konversi[1].Pendekatan ini selain untuk meningkatkan efisiensi dan konsistensi dalam proses konversi ,tetapi juga memberikan solusi inovatif yang menerapkan kemampuan model BERT dalam menangkap makna antar teks. Dengan menganalisis (deskripsi) silabus secara mendalam, penelitian ini berkontribusi pada pengembangan sistem rekomendasi berbasis NLP yang dapat mendukung proses akademik dengan lebih efektif. [2]

Penelitian ini memiliki tujuan untuk mengembangkan sistem rekomendasi berbasis NLP dengan menggunakan metode BERT , dengan data Data yang digunakan dalam penelitian ini terdiri dari deskripsi silabus mata kuliah dan deskripsi kegiatan mahasiswa yang telah diajukan untuk proses konversi, dengan fokus pada analisis kesesuaian berdasarkan isi deskripsi tersebut .Sistem ini diharapkan dapat dapat meningkatkan efisiensi dan efektifitas staff akademik di Universitas Islam Balitar dalam melakukan konversi mata kuliah.

2. METODE PENELITIAN

Untuk mencapai tingkat akurasi tertinggi dalam sistem rekomendasi konversi mata kuliah, penulis menggunakan satu pendekatan utama. Teknologi itu adalah *Natural Language Processing* atau NLP dengan model BERT. Prosesor Bahasa Alami ini memungkinkan penekanan maksimal pada deskripsi berbagai mata kuliah dan membantu menarik rekomendasi yang sesuai. NLP sendiri memiliki implementasi yang cukup mahal. Dalam penulisan ini, NLP memerlukan sejumlah besar tahap yang dimulai dengan prapemrosesan data teks, seperti *data cleaning*, tokenisasi, penghapusan *stopword*, dan langkah terakhir representasi vektor model BERT. Dalam hal ini, setiap proses dirancang untuk memastikan data teks dapat diambil dengan mudah, dan bahkan metode mesin

besar bisa mendapatkan hasil yang lebih tinggi. Gambar 1 menunjukkan langkah-langkah bagaimana penelitian ini dibuat :



Gambar 1.Alur Penelitian

Penjelasan dari gambar 1 diatas adalah sebagai berikut :

2.1 Pengumpulan Data

Metode pengumpulan data dalam penelitian ini mencakup pengumpulan dokumen akademik resmi dan observasi langsung terhadap proses konversi mata kuliah. Dokumen akademik resmi, seperti silabus, katalog kurikulum, dan panduan pembelajaran dari Universitas Islam Balitar, digunakan untuk memahami struktur mata kuliah, capaian pembelajaran, serta kebijakan konversi kegiatan seperti MBKM, magang, dan pelatihan sertifikasi. Observasi langsung dilakukan terhadap proses manual yang dijalankan oleh Kaprodi dan staf akademik, meliputi pengumpulan dokumen kegiatan mahasiswa, analisis kesesuaian silabus, hingga pengambilan keputusan akhir. Selain itu, wawancara mendalam dengan Kaprodi memberikan perspektif lebih rinci mengenai tantangan dan peluang optimalisasi proses konversi, yang menjadi masukan penting dalam perancangan sistem otomatisasi yang diusulkan

2.2 Data Engineering dan Preprocesing

Preprocessing data teks merupakan tahap awal yang penting dalam pengolahan data berbasis *Natural Language Processing* (NLP). Tahap ini bertujuan untuk mempersiapkan data teks mentah menjadi format yang lebih terstruktur dan siap digunakan dalam proses analisis atau pemodelan. Dalam konteks penelitian ini, *preprocessing* diperlukan untuk memastikan deskripsi mata kuliah yang diolah memiliki kualitas yang konsisten, bebas dari data tidak relevan, dan sesuai dengan kebutuhan model BERT. Proses preprocessing ini mencakup serangkaian langkah sistematis yang dirancang untuk meningkatkan akurasi dan efisiensi algoritma NLP. Berikut adalah jenis-jenis *preprocessing* data teks yang digunakan:

1. *Cleaning*: Menghapus karakter atau data yang tidak relevan.
 2. *Case Folding*: Mengonversi semua teks menjadi huruf kecil.
 3. *Tokenization*: Memecah teks menjadi unit analisis.
 4. *Stopword Removal*: Menghilangkan kata yang tidak bermakna (misalnya, "dan", "di").
 5. *Stemming*: Mengubah kata menjadi bentuk dasar.

2.3 Penerapan Model BERT

Implementasi model BERT (Bidirectional Encoder Representations from Transformers) dilakukan untuk mengoptimalkan proses analisis teks dalam sistem rekomendasi konversi mata kuliah. BERT dipilih karena kemampuannya memahami konteks teks secara bidireksional, sehingga dapat mengenali hubungan antara kata dalam sebuah kalimat dengan lebih baik dibandingkan metode konvensional. Hasil representasi *vektor* ini kemudian dibandingkan menggunakan teknik seperti *cosine similarity* untuk menentukan tingkat kesamaan antar mata kuliah. Implementasi BERT diharapkan dapat meningkatkan akurasi sistem dalam merekomendasikan kesepadan mata kuliah, mengurangi inkonsistensi, dan mempercepat proses konversi secara keseluruhan. Selain itu, model ini dirancang agar dapat beradaptasi dengan berbagai format deskripsi mata kuliah, memberikan fleksibilitas dalam pengembangan sistem rekomendasi di masa mendatang [3].

2.4 Hasil Rekomendasi Menggunakan *Cosine Similarity*

Setelah vektor deskripsi mata kuliah yang telah dihasilkan oleh model BERT, Metode ini membantu dalam menentukan tingkat kesesuaian antara deskripsi mata kuliah dari institusi yang berbeda, sehingga memungkinkan pencocokan mata kuliah yang relevan. Keunggulan utama *cosine similarity* adalah kemampuannya untuk fokus pada arah hubungan antar vektor, yang sangat berguna ketika membandingkan data teks yang mungkin memiliki jumlah kata atau istilah yang tidak seimbang. Dengan metode ini, sistem dapat memberikan rekomendasi konversi mata kuliah yang lebih akurat dan relevan, mengurangi potensi kesalahan yang disebabkan oleh pendekatan berbasis kata sederhana. Berikut adalah rumus yang digunakan untuk menghitung *cosine similarity* secara manual :

Keterangan :

$A = \text{Vektor } A$, yang akan dibandingkan kemiripannya

B = Vektor *B*, yang akan dibandingkan kemiripannya

$A \bullet B = \text{Cross product antara } |A| \text{ dan } |B|$

$|A|$ = Panjang vektor A

$|B|$ = Panjang vektor B

$|A||B| = \text{Cross product antara } |A| \text{ dan } |B|$

2.5 Evaluasi Model

Evaluasi sistem rekomendasi (yang dilakukan) melibatkan perbandingan antara hasil rekomendasi yang dihasilkan oleh sistem dan keputusan manual yang dikerjakan oleh staf akademik. Dalam proses ini, deskripsi mata kuliah yang telah diproses oleh sistem dievaluasi: seberapa besar tingkat kesesuaiannya dengan hasil yang diperoleh dari metode manual. Para staf akademik akan menilai apakah pasangan mata kuliah yang direkomendasikan oleh sistem sesuai dengan kebijakan konversi yang berlaku, (ini) berdasarkan kesamaan konten, capaian pembelajaran, dan beban studi. Pendekatan ini bertujuan untuk mengukur akurasi sistem dalam mencerminkan proses manual; namun, juga untuk mengidentifikasi sejauh mana sistem dapat diandalkan untuk menggantikan atau melengkapi metode konvensional. Meskipun demikian, tantangan tetap ada.

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Dataset

Pengumpulan data dalam penelitian ini dilakukan dengan memanfaatkan teks dari modul pembelajaran yang telah dikonversi menjadi deskripsi mata kuliah. Data hasil konversi ini kemudian diolah dan dipersiapkan untuk diproses menggunakan model BERT guna mengukur kesamaan deskripsi antar mata kuliah. Pada Gambar 2 merupakan dataset mulai dari nama mata kuliah , jumlah sks dari mata kulih , serta penjelasan silabus apa saja yang dapat dikonversi kedalam mata kuliah.

subject	skls	keywords
Teknologi Blockchain	3	teknologi blockchain, cryptocurrency, kontrak pintar, buku besar terdistribusi, aplikasi blockchain, algoritma konsensus, keamanan blockchain
Kecerdasan Buatan	3	sistem cerdas, kecerdasan buatan, sistem pakar, sistem pendukung keputusan, logika fuzzy, jaringan syaraf tiruan, agen cerdas, sistem adaptif
Otomasi Sistem	3	otomasi sistem, otomatisasi proses, RPA, otomatisasi proses robotik, alat otomatisasi, otomatisasi alur kerja, skrip otomatisasi, pengujian otomatisasi
Keamanan Sistem Cyber-Physical dan Kota Cerdas	3	sistem cyber-physical, keamanan kota pintar, keamanan IoT, infrastruktur kritis, serangan cyber-physical, keamanan smart grid, keamanan penerbangan
Internet of Things	3	IoT, perangkat terhubung, platform IoT, keamanan IoT, jaringan sensor, komunikasi M2M, perangkat pintar, aplikasi IoT, analitik IoT, infrastruktur IoT
Pemrograman Web Lanjut	4	Laravel Framework,JavaScript dan jQuery,HTML, CSS, JavaScript,Critical Thinking,Logika dan Algoritma Pemrograman,komputasi awan, layanan web
Managemen Teknologi Informasi	2	Meningkatkan Keterampilan Berpikir Kreatif,Professional Softskill , Design Thinking,
Pengujian Perangkat Lunak	3	User Experience Design,User Interface Designing & Prototyping
Kriptografi	3	Research & Usability Testing
Kuliah Kerja Nyata	3	Final Project - Group Project,Capstone Project,Final Project and Showcase,Machine Learning Specialization by Andrew Ng
Sistem Pendukung Keputusan	2	Machine Learning Specialization by Andrew Ng,privasi data, perlindungan data, GDPR, keamanan data, privasi pengguna, privasi informasi, keamanan data
Desain Analisis Algoritma	3	Tensorflow Data and Deployment,Belajar Dasar Pemrograman Web,Belajar Membuat Front-End Web untuk Pemula,Belajar Fundamental TensorFlow dan Python
Managemen Strategik	3	Preparing for TensorFlow Developer Certification,Memulai Dasar Pemrograman untuk Menjadi Pengembang Software ke Dasar,Project management
Praktek Kerja Lapangan	3	Soft skill & Career Development,Capstone / Final Project,
Jaringan Komputer Lanjut	3	Belajar Dasar Pemrograman Web,Belajar Membuat Front-End Web untuk Pemula,Belajar Fundamental Front-End Web Development,Dasar Jaringan Komputer Lanjut
Pengantar E-Business	3	Soft skill & Career Development,Dasar Ilmu Pemasaran Digital,
Etika Profesi Teknologi Informasi	3	Foundations for Data Science,privasi data, perlindungan data, GDPR, keamanan data, privasi pengguna, privasi informasi, kebijakan privasi, e-commerce
Multimedia	3	Data Visualization,Data Preprocessing,Could Computing
Managemen Teknologi Informasi	2	Machine Learning ,Model Performance Analytics,Professional Softskill

Gambar 2.Dataset Deskripsi Mata Kuliah

3.2. Preprocesing data

Setelah mengumpulkan data, langkah berikutnya adalah melakukan persiapan dan pengolahan data yang telah diperoleh. Tahapan awal ini mencakup pemrosesan data untuk memastikan data siap digunakan dalam langkah-langkah selanjutnya, seperti analisis kesamaan deskripsi. Pemrosesan data dilakukan melalui beberapa tahapan, termasuk tokenisasi, pembersihan teks, penghapusan *stopword*, *stemming*, dan *normalisasi*, sehingga data dapat digunakan secara optimal dalam model berbasis BERT

a) Cleaning text

Pada bagian cleaning text, proses ini akan menghapus elemen-elemen yang tidak relevan atau dapat mengganggu analisis, seperti tanda baca, angka, karakter khusus, serta spasi berlebih. Selain itu, teks juga akan dinormalisasi dengan menghilangkan simbol-simbol yang tidak diperlukan, sehingga menghasilkan data yang lebih bersih dan siap digunakan untuk pemrosesan lebih lanjut menggunakan model BERT. Hasil dari Tabel 1 menghilangkan selain huruf dan angka.

Tabel 1. Cleaning Text dan Case Folding

	Before	After
Detail Program Studi Independen: RevoU Tech Academy - Learning Data & Software Development with AI Detail Program Studi Independen: 16 Februari - 30 Juni 2024 Detail Program Studi Independen: 20 Detail Program Studi Independen: Daring Detail Program Studi Independen: Program ini didesain untuk memberikan gambaran kepada peserta mengenai karir yang dapat mereka Detail Program Studi Independen: tempuh di perusahaan teknologi, terutama di 2 bidang besar yaitu software engineering dan data Detail Program Studi Independen: analytics. Detail Program Studi Independen: Kedua posisi ini merupakan posisi kunci dari perusahaan teknologi, dimana keduanya hampir selalu ada Detail Program Studi Independen: di setiap perusahaan. Setiap perusahaan teknologi memerlukan perangkat lunak (software) yang mumpuni Detail Program Studi Independen: dalam proses pengembangan produk dan setiap produk dikembangkan berdasarkan keputusan berbasis Detail Program Studi Independen: data dengan bantuan tim data analis. Detail Program Studi Independen: Mahasiswa semester 5 keatas dari semua jurusan/program studi Detail Program Studi Independen: Lulus seleksi dari RevoU Detail Program Studi Independen: Certificate of completion - diberikan bagi murid yang menyelesaikan seluruh program dan lulus dari Detail Program Studi Independen: semua modul yang ada	detail program studi independen revou tech academy learning data software development with ai detail program studi independen februari juni detail program studi independen detail program studi independen daring detail program studi independen program ini didesain untuk memberikan gambaran kepada peserta mengenai karir yang dapat mereka detail program studi independen tempuh di perusahaan teknologi terutama di bidang besar yaitu software engineering dan data detail program studi independen analytics detail program studi independen kedua posisi ini merupakan posisi kunci dari perusahaan teknologi dimana keduanya hampir selalu ada detail program studi independen di setiap perusahaan setiap perusahaan teknologi memerlukan perangkat lunak software yang mumpuni detail program studi independen dalam proses pengembangan produk dan setiap produk dikembangkan berdasarkan keputusan berbasis detail program studi independen data dengan bantuan tim data analis detail program studi independen mahasiswa semester keatas dari semua jurusan program studi detail program studi independen lulus seleksi dari revou detail program studi independen certificate of completion diberikan bagi murid yang	

menyelesaikan seluruh program dan lulus dari detail program studi independen semua modul yang ada

b) *Case Folding*

Mengubah seluruh teks menjadi huruf kecil dilakukan untuk menormalkan data teks agar konsisten, terutama ketika terdapat variasi penggunaan huruf besar dan kecil di awal, tengah, atau akhir teks. Proses ini memastikan keseragaman data, sehingga dapat meningkatkan keakuratan dan keandalan model BERT dalam analisis. Sebagai contoh, jika sebagian teks menggunakan huruf kapital dan sebagian lainnya huruf kecil, model akan kesulitan mengidentifikasi pola yang tepat, yang dapat memengaruhi hasil analisis dan rekomendasi. Hasil dari *case folding* terlihat pada Table 1 diatas

c) *Tokenization*

Pada tahapan ini teks akan dipecah menjadi beberapa kata untuk memudahkan proses . Proses ini melibatkan pemisahan teks berdasarkan tanda baca, spasi, atau aturan linguistik lainnya. *Tokenization* membantu dalam mempersiapkan teks untuk analisis lebih lanjut, seperti dalam aplikasi pemrosesan bahasa alami (NLP), di mana setiap token dapat dianalisis secara individual untuk tugas-tugas seperti pengenalan entitas, analisis sentimen, atau pembuatan model bahasa [4]. Hasil *Tokenization* bisa terlihat pada table 2 dibawah

Tabel 2. *Tokenization*

Before	After
detail program studi	['detail','program','studi','independen','revou','tech
independen revou tech	', 'academy', 'learning', 'data', 'software', 'developme
academy learning data	nt', 'with', 'ai', 'detail', 'program', 'studi', 'independen', '
software development with	februari', 'juni', 'detail', ']
ai detail program studi	
independen februari juni	
detail program studi	
independen detail program	
studi independen daring	
detail program studi	
independen program ini	

d) *Stopword Removal*

Menghapus *stopwords* adalah langkah dalam preprocessing data teks untuk meningkatkan efisiensi analisis dengan mengurangi jumlah kata tidak informatif, seperti 'dan', 'di', atau 'yang', yang tidak berkontribusi signifikan pada hasil analisis [9] . Langkah ini mendukung pengurangan kosakata, normalisasi untuk mengurangi variabilitas dalam data, serta meningkatkan konsistensi dengan mengurangi variasi dalam teks. Dalam konteks sistem rekomendasi berbasis *Natural Language Processing* (NLP), seperti yang diterapkan menggunakan metode BERT, penghapusan *stopwords* membantu memperbaiki kualitas analisis semantik, menghasilkan model yang lebih akurat dan relevan. Hasil *stopword* ditampilkan pada Table 3 dibawah

Tabel 3. Stopword Removal

Before	After
<code>['detail','program','studi','independen','revou','tech','academy','learning','data','software','development','with','ai','detail','program','studi','independen','februari','juni','detail',]</code>	<code>['detail','program','studi','independen','revou','tech','academy','learning','data','software','development','with','ai','detail','program','studi','independen','februari','juni','detail',]</code>

e) *Stemming*

Stemming adalah proses dalam *preprocessing* teks yang bertujuan untuk mengubah kata ke bentuk dasarnya atau akar kata, seperti mengubah 'berjalan', 'berjalanlah', dan 'berjalan-jalan' menjadi 'jalan'. Langkah ini mengurangi jumlah variasi kata yang berasal dari infleksi atau imbuhan, sehingga mempermudah analisis *semantik* dan meningkatkan efisiensi dalam pengolahan data teks [10]. Dalam sistem rekomendasi berbasis *Natural Language Processing* (NLP), seperti yang menggunakan metode BERT, *stemming* membantu meningkatkan konsistensi data, memungkinkan model untuk memahami konteks secara lebih akurat, dan menghasilkan analisis yang lebih relevan [5]

Tabel 4. Stemming

Before	After
<code>['detail','program','studi','independen','lulus','seleksi','revou','detail','program','studi','independen','certificate','completion','murid','menyel esaikan','program','lulus','detail','program','studi','independen','modul']</code>	<code>detail program studi independen lulus seleksi revou detail program studi independen certificate completion murid selesai program lulus detail program studi independen modul</code>

3.3. Penerapan Model BERT

Penerapan model BERT (*Bidirectional Encoder Representations from Transformers*) bertujuan untuk menghasilkan representasi *vektor* atau *embedding* dari teks [6]. Dalam proses ini, deskripsi teks, seperti silabus atau deskripsi mata kuliah, melalui tahap *preprocessing* dan kemudian dimasukkan ke dalam model BERT. Model ini menganalisis konteks teks secara bidireksional, memperhatikan hubungan antar kata sebelum dan sesudahnya. Hasilnya adalah *vektor embedding* dengan dimensi tinggi yang merepresentasikan makna semantik dari teks secara mendalam [7]. Vektor *embedding* ini digunakan untuk membandingkan teks, misalnya, melalui teknik *cosine similarity*, untuk mengukur tingkat kesamaan antara deskripsi mata kuliah. Penerapan *embedding* BERT meningkatkan akurasi dalam analisis teks dan mendukung sistem rekomendasi yang lebih presisi [8]. Pada gambar 3 Teks diubah menjadi token numerik yang dapat dimengerti oleh BERT. token [CLS] ditambahkan di awal teks untuk merepresentasikan keseluruhan teks. token [SEP] ditambahkan di akhir teks untuk memisahkan teks (jika ada), output dari gambar 3 akan berupa tensor berisi token numerik teks input seperti pada gambar 4.

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

inputs1 = tokenizer(text1, return_tensors='pt', padding=True, truncation=True, max_length=512)
inputs2 = tokenizer(text2, return_tensors='pt', padding=True, truncation=True, max_length=512)
```

Gambar 3. Input Token BERT

```
{'input_ids': tensor([[101, 3698, 4083, 2003, 6429, 102]]),  
'attention_mask': tensor([[1, 1, 1, 1, 1, 1]])}
```

Gambar 4. Output tensor BERT

Token inilah yang akan dimasukkan kedalam model BERT pada Gambar 5 untuk diproses sehingga menghasilkan output *vector* representasi kalimat dari teks seperti pada Gambar 6.

```
with torch.no_grad():  
    outputs1 = model(**inputs1)  
    outputs2 = model(**inputs2)  
embedding1 = outputs1.last_hidden_state[:, 0, :]  
embedding2 = outputs2.last_hidden_state[:, 0, :]
```

Gambar 5. Output tensor BERT

```
tensor([[-2.5033e-01,  1.3012e-01, -3.8201e-01,  3.8649e-01,  1.6175e-01,  
       -2.5572e-01, -3.0991e-01,  1.3857e+00,  1.1244e+00, -4.8777e-01,  
       1.7366e-01, -3.7290e-01, -4.4814e-02, -7.6610e-02, -7.6988e-01,  
       1.8130e-02,  3.8515e-01, -6.1484e-01, -3.3556e-01, -2.9973e-02,  
       5.5050e-01, -1.1543e-01, -3.6565e-01, -1.6887e-03,  2.6016e-01,  
      -7.9536e-02,  4.0510e-01,  5.4062e-01,  4.3256e-01, -2.3833e-01,  
       2.9949e-01, -3.7151e-01,  2.6821e-01,  2.9137e-01, -9.8852e-01,  
       4.9364e-01, -8.4000e-01,  8.9625e-01, -4.1110e-01, -4.8938e-01,  
      -5.0511e-01,  1.0812e+00, -6.8253e-01, -2.2077e-01,  7.0079e-01,  
       2.9547e-02,  3.6990e-01, -1.0801e-01,  3.6276e-01,  6.9556e-01,  
       1.1434e+00, -4.0136e-01,  3.5511e-01,  2.4459e-01, -1.0371e-01,  
       3.6916e-01,  2.3770e-01,  4.8049e-02,  8.8130e-02, -2.9171e-01,  
       2.4802e-01, -7.1895e-02, -3.4910e-01,  6.1174e-01,  9.4992e-02,  
      -3.1110e-01, -6.2308e-02,  4.1943e-01,  8.5514e-01, -3.0806e-01,  
      -1.3570e+00, -2.7202e-01, -1.0608e+00, -2.4658e-01, -1.1412e+00,  
       2.4400e-01, -5.0970e-01, -5.9332e-01,  3.7062e-01, -3.5068e-01,  
       3.8448e-01, -8.6641e-01,  2.0227e-01,  7.3911e-01,  7.1543e-01,  
      -5.6802e-01, -7.9089e-01,  2.9237e-01,  2.1087e-02, -1.2692e+00,  
      -1.3763e-01,  9.9146e-02, -6.6002e-01, -9.4604e-01, -8.4765e-01,  
      -8.1069e-02,  3.2166e-01,  4.9380e-02,  9.1598e-01,  3.2429e-01,  
      -1.9593e-01, -4.2442e-01, -3.1762e-01,  1.4285e-02, -5.9095e-01,  
      -1.0387e+00, -3.5789e-02,  3.2671e-01, -3.9746e-02, -1.3198e-01,  
       8.5653e-02,  3.5857e-01, -1.0753e-01,  6.8202e-01,  5.2331e-01,  
      -1.1848e-01,  1.2025e-01,  2.0891e-01, -2.9350e-02,  3.1782e-01,  
      -4.3851e-01, -6.9907e-01,  3.0799e-01,  2.3552e-01,  6.9304e-02,  
       5.9319e-01,  1.3742e-01,  4.8464e-01, -2.2016e-01, -2.2961e-01,  
      -6.2267e-01, -1.8914e-01, -3.0105e-02, -4.1832e-01, -1.0108e+00])
```

Gambar 6. Vector Embedding BERT

3.4. Melakukan Rekomendasi dengan *Cosine Similarity*

Menerapkan rekomendasi melalui *cosine similarity* melibatkan pengukuran derajat kesamaan antara dua representasi vektor teks (contohnya, deskripsi mata kuliah) yang telah diproses menggunakan model BERT. *cosine similarity* menghitung sudut di antara vektor-vektor itu dalam ruang multidimensi; hasilnya adalah nilai yang berkisar antara -1 hingga 1. Semakin dekat nilai tersebut ke 1, semakin tinggi tingkat kesamaan antara teks yang dibandingkan. Namun, perlu dicatat bahwa nilai -1 menunjukkan ketidakcocokan yang sempurna. Pada Gambar 10 hasil rekomendasi konversi berupa nilai persentase yang menampilkan mata kuliah yang paling cocok untuk dilakukan konversi sesuai silabus yang dimasukkan kedalam model BERT sebelumnya. Rekomendasi yang dihasilkan akan berfungsi sebagai referensi bagi staf akademik, dalam menentukan kesetaraan atau konversi mata kuliah secara objektif dan akurat.

```
[69] # Hasil dalam bentuk persentase
similarities = [(subjects[i], sks_values[i], round(cosine_similarities[0][i] * 100, 2)) for i in range(len(subjects))]
sorted_similarities = sorted(similarities, key=lambda x: x[2], reverse=True)

print("Hasil Rekomendasi ")
sorted_similarities
```

Hasil Rekomendasi
[('Data Mining', 3, 53.07),
 ('Jaringan Syaraf Tiruan(SC)', 4, 51.89),
 ('Kecerdasan Buatan', 3, 30.08),
 ('Grafika Komputer', 3, 14.36),
 ('Penjaminan Kualitas TI', 3, 6.48),

Gambar 7. Hasil Rekomendasi

3.5. Melakukan Evaluasi Model

Hasil rekomendasi akan divaluasi dengan membandingkan dengan rekomendasi staff akademik yang dilakukan secara manual sebelumnya. Dalam evaluasi kali ini menggunakan 5 modul yang sudah dikonversi oleh staf akademik sebelumnya dengan hasil rekomendasi dari model BERT yang di buat. Pada gambar 8 dilakukan evaluasi dengan hasil rata-rata 53 persen, Data silabus terbaru cenderung memiliki persentase kecocokan yang rendah karena belum banyak kesesuaian dengan dataset yang ada.

Modul	Hasil Rekomendasi	Konversi Manual	Percentase Kecocokan
RevoU Tech Academy: Learning Data Analytics & Software Development with AI	Data Mining	Data Mining	66.66666667
	Aplikasi Perangkat Bergerak	Manajemen Proyek Perangkat Lunak	
	Pemrograman Web Lanjut	Sistem Pendukung Keputusan lanjut	
	Jaringan Syaraf Tiruan		
Silabus Infinite Learning Mobile Development batch 6	Sistem Pendukung Keputusan lanjut		75
	Aplikasi Perangkat Bergerak	Aplikasi Perangkat Bergerak	
	Sistem Pendukung Keputusan lanjut	Sistem Pendukung Keputusan lanjut	
	Data Mining	Manajemen Proyek Perangkat Lunak	
Back End Javascript Binar Academy batch 5	Kuliah Kerja Nyata	Kuliah Kerja Nyata	66.66666667
	Manajement Proyek Perangkat Lunak		
	Desain Analisis Algoritma	Pemrograman Web Lanjut	
	Pemrograman Web Lanjut	Praktek Kerja Lapangan	
StartUp Campus Data Scient Batch 7	Kriptografi	Managemen Strategik	25
	Managemen Strategik	Pengantar E-Business	
	Jaringan Komputer Lanjut	Desain Analisis Algoritma	
	Multimedia	Jaringan Komputer Lanjut	
Back End Javascript Binar Academy batch 7	Jaringan Syaraf Tiruan	Manajement Teknologi Informasi	33.33333333
	Pemrograman Mobile	Pengujian Perangkat Lunak	
	Data Mining	Multimedia	
	Desain Analisis Algoritma	Jaringan Syaraf Tiruan	
Rata-Rata Akurasi			53.33333333

Gambar 8. Evaluasi Rekomendasi

4. SIMPULAN

Penelitian ini berhasil mengimplementasikan model BERT untuk mengukur kesamaan deskripsi mata kuliah, yang menghasilkan rekomendasi yang relevan dan mendukung efisiensi dalam penyusunan kurikulum berbasis teknologi. Hasil menunjukkan bahwa pendekatan ini mampu memahami dan menangkap konteks dari teks deskripsi secara akurat. Sistem ini memiliki kelebihan dalam kemampuannya untuk memproses data teks secara mendalam menggunakan representasi berbasis vektor, yang memungkinkan pengukuran kesamaan yang lebih kontekstual dibandingkan pendekatan berbasis kata kunci konvensional. Kelemahan dari pendekatan ini adalah ketergantungan pada kualitas data yang digunakan. Jika data deskripsi tidak terstruktur atau mengandung informasi yang ambigu, akurasi hasil dapat terpengaruh. Selain itu, pemrosesan data yang kompleks membutuhkan waktu dan sumber daya komputasi yang tinggi. Hasil penelitian ini penting untuk mendukung pengembangan teknologi pendidikan, khususnya dalam pengelolaan dan penyelarasan kurikulum secara otomatis. Aplikasi ini dapat meningkatkan efisiensi kerja institusi pendidikan dan memastikan kesesuaian antara mata kuliah yang ditawarkan

5. SARAN

Untuk pengembangan lebih lanjut, disarankan untuk mengeksplorasi penggunaan model lain selain BERT, seperti RoBERTa atau GPT, guna membandingkan performa dan hasil analisis kesamaan. Selain itu, penggunaan dataset yang lebih besar dan beragam dapat membantu meningkatkan akurasi sistem rekomendasi

DAFTAR PUSTAKA

- [1] R. Fatmasari, R. K. Septiani, T. H. Pinem, D. Fabiyanto, and W. Gata, “Implementasi Algoritma BERT Pada Komentar Layanan Akademik dan Non Akademik Universitas Terbuka di Media Sosial,” *Sains, Aplikasi, Komputasi dan Teknologi Informasi*, vol. 5, no. 2, p. 96, Jan. 2024, doi: 10.30872/jsakti.v5i2.13915.
- [2] A. S. Yazid and E. Winarko, “Fine-Tuning BERT untuk Menangani Ambiguitas Pada POS Tagging Bahasa Indonesia,” *Jurnal Linguistik Komputasional (JLK)*, vol. 6, no. 2, pp. 57–64, Sep. 2023, doi: 10.26418/jlk.v6i2.148.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.
- [4] I. Huda, “IMPLEMENTASI NATURAL LANGUAGE PROCESSING (NLP) UNTUK APLIKASI PENCARIAN LOKASI,” *Jurnal Nasional Teknologi Terapan (JNTT)*, vol. 3, no. 2, p. 15, Oct. 2021, doi: 10.22146/jntt.35036.
- [5] D. O. Sihombing, “Implementasi Natural Language Processing (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 2, p. 396, Dec. 2022, doi: 10.30865/json.v4i2.5374.
- [6] D. Sebastian, H. D. Purnomo, and I. Sembiring, “BERT for Natural Language Processing in Bahasa Indonesia,” in *2022 2nd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, IEEE, Dec. 2022, pp. 204–209. doi: 10.1109/ICICyTA57421.2022.10038230.
- [7] R. Fatmasari, R. K. Septiani, T. H. Pinem, D. Fabiyanto, and W. Gata, “Implementasi Algoritma BERT Pada Komentar Layanan Akademik dan Non Akademik Universitas Terbuka di Media Sosial,” *Sains, Aplikasi, Komputasi dan Teknologi Informasi*, vol. 5, no. 2, p. 96, Jan. 2024, doi: 10.30872/jsakti.v5i2.13915.
- [8] G. Girinoto, D. A. Alwan, G. A. N. Gde K.T. D, O. G. Nabila, A. Arizal, and D. F. Priambodo, “Implementasi Deteksi Judul Berita Clickbait Berbahasa Indonesia dengan pre-trained model Multilingual BERT Pada Aplikasi Berbasis Chrome Extension,” *Jurnal Ilmiah SINUS*, vol. 20, no. 2, p. 25, Jul. 2022, doi: 10.30646/sinus.v20i2.624.
- [9] M. Prasetya, M. Wulandari, and S. A. Nikmah, “Implementasi NLP(Natural Language Processing) Dasar pada Analisis Sentimen Review Spotify”, *Stains*, vol. 3, no. 1, pp. 145–153, Jan. 2024.
- [10] R. Azhar and M. F. . Wijayanto, “Analisis Sentimen di Twitter: Mengungkap Persepsi dan Emosi Publik Seputar Konflik Palestina-Israel”, *Stains*, vol. 3, no. 1, pp. 118–124, Jan. 2024.