

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1 Sistem Rekomendasi

Menurut Anggoro & Izzatillah (2022) dalam penelitian yang berjudul “Sistem Rekomendasi Musik Dengan Metode *Collaborative Filtering* Berbasis *Android*”. Sebuah sistem rekomendasi berfungsi memberikan suatu saran produk atau item yang sesuai dengan kebutuhan penggunanya. Dengan memanfaatkan sebuah input berupa minat dan preferensi pengguna, sistem ini akan menerapkan algoritma khusus untuk mengidentifikasi produk-produk yang paling relevan. Ketika pengguna akan dihadapkan dengan informasi yang begitu banyak, sistem rekomendasi memiliki peran sebagai penyaring yang memisahkan konten relevan dari yang tidak sesuai kebutuhan. Efektivitas sistem ini terletak pada kemampuannya mengolah preferensi pengguna menjadi rekomendasi yang tepat sasaran. Pada dasarnya, sistem ini bekerja layaknya suatu asisten pribadi yang akan mencoba memahami selera penggunanya dengan menganalisis keinginan mereka melalui berbagai metode rekomendasi untuk menyajikan pilihan-pilihan yang paling sesuai.

Menurut Theodorus (2022) pada penelitian yang berjudul “*Machine Learning* Rekomendasi Produk dalam Penjualan Menggunakan Metode *Item-Based Collaborative Filtering*” tujuan dari sistem rekomendasi adalah untuk meningkatkan pengalaman pengguna dan efisiensi dalam menemukan informasi atau produk yang relevan. Sistem ini dirancang untuk menganalisis preferensi dan

perilaku pengguna, kemudian menyajikan saran yang personal dan sesuai dengan minat mereka. Dalam konteks penjualan, tujuan ini mencakup peningkatan konversi penjualan dan kepuasan pelanggan melalui penyajian produk yang paling mungkin diminati oleh setiap individu. Lebih lanjut, sistem rekomendasi bertujuan untuk mengatasi masalah kelebihan informasi yang sering dihadapi pengguna di era digital. Dengan menyaring dan memprioritaskan informasi atau produk berdasarkan profil pengguna, sistem ini membantu mengurangi waktu dan usaha yang diperlukan dalam proses pencarian. Selain itu, sistem rekomendasi juga bertujuan untuk memperkenalkan pengguna pada opsi-opsi baru yang mungkin belum mereka ketahui sebelumnya, namun berpotensi menarik minat mereka. Hal ini tidak hanya menguntungkan pengguna dengan memperluas wawasan mereka, tetapi juga membantu penyedia layanan atau penjual dalam mengoptimalkan inventaris dan meningkatkan *engagement* pengguna.

2.1.2 Konversi Mata Kuliah

Konversi mata kuliah merupakan proses penyetaraan atau pengakuan terhadap mata kuliah yang telah ditempuh oleh mahasiswa di institusi pendidikan sebelumnya atau program studi lain. Menurut Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 3 Tahun 2020 tentang Standar Nasional Pendidikan Tinggi, konversi mata kuliah dilakukan dengan mempertimbangkan kesetaraan capaian pembelajaran, substansi, dan beban studi. Proses ini menjadi krusial dalam sistem pendidikan tinggi modern, terutama dengan meningkatnya mobilitas mahasiswa dan kebutuhan akan fleksibilitas kurikulum. Di Universitas Islam Balitar, konversi mata kuliah memainkan peran penting dalam memfasilitasi

transfer mahasiswa antar program studi, pengakuan kredit dari pendidikan non-formal, dan penyesuaian kurikulum sesuai dengan perkembangan ilmu pengetahuan dan teknologi.

Menurut Wahanani (2022) pada penelitian yang berjudul “Rancang Bangun Sistem Konversi Mata Kuliah (Studi Kasus : Prodi Informatika, Fasilkom, UPN "Veteran" Jawa Timur)” konversi mata kuliah merupakan proses penting dalam sistem akademik perguruan tinggi. Proses ini melibatkan evaluasi dan penyetaraan mata kuliah yang telah diambil oleh mahasiswa di institusi atau program studi sebelumnya dengan kurikulum yang berlaku di institusi atau program studi baru. Tujuan utama dari konversi mata kuliah adalah untuk memastikan kesinambungan pembelajaran mahasiswa dan mengoptimalkan waktu studi mereka. Sistem konversi yang efektif memungkinkan mahasiswa untuk melanjutkan studi tanpa harus mengulang materi yang sudah dikuasai, sekaligus memastikan bahwa mereka memenuhi standar kompetensi yang ditetapkan oleh program studi baru. Implementasi sistem konversi mata kuliah menghadapi beberapa tantangan, terutama dalam hal kompleksitas proses dan kebutuhan akan akurasi menekankan pentingnya pendekatan sistematis dalam merancang sistem konversi, yang melibatkan analisis mendalam terhadap struktur kurikulum, capaian pembelajaran, dan konten mata kuliah. Penggunaan teknologi informasi, seperti sistem berbasis web, dapat meningkatkan efisiensi dan transparansi proses konversi. Sistem ini memungkinkan otomatisasi sebagian besar tahapan konversi, mulai dari pengajuan permohonan oleh mahasiswa hingga evaluasi oleh tim akademik. Selain itu, pendekatan ini juga memfasilitasi penyimpanan dan pengelolaan data historis

konversi, yang dapat digunakan untuk analisis dan perbaikan berkelanjutan terhadap kebijakan dan prosedur konversi mata kuliah di institusi.

2.1.3 *Python*

Python adalah bahasa pemrograman yang sangat fleksibel dan digunakan secara luas di berbagai industri, menggabungkan pengembangan web, analisis data, dan komputasi ilmiah, terutama karena kemudahan penggunaannya dan kemudahan pembacaannya. *Python* telah menjadi salah satu bahasa pemrograman yang paling populer dalam pengembangan aplikasi *machine learning* (ML) dan *Deep Learning* (DL). *Python* menawarkan berbagai fitur dan pustaka yang membuatnya ideal untuk pengembangan solusi ML dan DL. *Python* dikenal dengan sintaksnya yang sederhana dan mudah dipelajari, memungkinkan para pengembang untuk fokus pada penyelesaian masalah dan pengembangan algoritma tanpa terhambat oleh kompleksitas bahasa pemrograman itu sendiri. Fitur-fitur ini menjadikan *Python* sebagai pilihan utama baik bagi para pemula maupun ahli dalam bidang machine learning dan deep learning. Selain kemudahan penggunaan, *Python* memiliki ekosistem pustaka yang kaya dan beragam yang mendukung pengembangan ML dan DL. *Library* seperti *NumPy*, *Pandas*, dan *Matplotlib* menyediakan dasar yang kuat untuk manipulasi data dan visualisasi, sementara *Scikit-learn* menawarkan berbagai algoritma machine learning yang siap digunakan. Untuk deep learning, pustaka seperti *TensorFlow* dan *PyTorch* memberikan dukungan yang kuat untuk pembangunan dan pelatihan model *neural network* yang kompleks. *TensorFlow* dan *PyTorch* memungkinkan para peneliti dan praktisi untuk mengembangkan model *deep learning* dari konsep hingga implementasi dengan efisiensi tinggi.

Keunggulan ini membuat *Python* tidak hanya relevan dalam penelitian akademis, tetapi juga dalam aplikasi industri yang membutuhkan solusi *machine learning* dan *deep learning* yang tangguh dan dapat diskalakan. (Alfarizi dkk, 2023)

2.1.4 Streamlit Framework

Streamlit merupakan *framework* yang dirancang untuk membuat tampilan website berbasis *machine learning* dengan menggunakan bahasa pemrograman python (Walingkas & Saian, 2023). *Framework* ini banyak digunakan dalam pengembangan website yang interaktif dengan pengembangan dengan mengedepankan konsep “*script to app*” sehingga memungkinkan mengembangkan website tanpa pemahaman teknologi *front-end*, sehingga pengembang dapat lebih fokus dalam membuat model *machine learning* atau *deep learning*. (Syafarina & Zaenuddin, 2023)

2.1.5 Google Colab

Google Colab merupakan sebuah platform pengembangan secara daring yang menyediakan sebuah fitur untuk menjalankan dan menulis kode *Python* secara langsung tanpa perlu menginstal program tambahan di komputer. Platform ini unggul karena menghadirkan kemampuan kelancaran dalam komputasi yang mumpuni serta terintegrasi dengan layanan *Google Drive*. Karakteristik tersebut menjadikan *Colab* sebagai sarana yang ideal untuk mempelajari, mengeksplorasi, dan mengimplementasikan berbagai metode numerik. (Sihombing, 2022a)

Colaboratory, yang lebih dikenal sebagai *Google Colab*, hadir sebagai solusi komputasi berbasis web yang memungkinkan penulisan dan eksekusi kode *Python* secara langsung melalui peramban. Platform ini menghadirkan pengalaman

pengembangan yang mumpuni tanpa perlu pemasangan aplikasi tambahan, memanfaatkan sepenuhnya kemampuan komputasi awan, termasuk dukungan akselerasi GPU dan TPU secara cuma-cuma. (Nirracca dkk, 2023)

Keunggulan utama *platform* ini terletak pada kemudahan akses dan fleksibilitas penggunaannya. Pengguna dapat langsung memulai proyek pemrograman *Python* hanya dengan koneksi internet dan aplikasi web. Selain itu, fitur kolaborasi *real-time* memungkinkan berbagai pihak untuk berkontribusi dan mengedit *notebook* secara bersamaan, menciptakan lingkungan kerja yang dinamis dan efisien. Sistem ini menjadi pilihan utama bagi beragam kalangan, mulai dari mahasiswa, peneliti, hingga pengembang profesional. Tanpa perlu mengkhawatirkan spesifikasi perangkat keras atau *setup environment* yang rumit, pengguna dapat fokus pada pengembangan proyek mereka. Dukungan infrastruktur komputasi awan yang handal menjamin kelancaran eksekusi kode, sementara antarmuka yang intuitif memudahkan proses pembelajaran dan eksperimentasi dalam pemrograman *Python*. (Nirracca dkk, 2023)

2.1.6 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) merupakan sebuah cabang dari kecerdasan buatan (AI) interdisipliner yang menggabungkan ilmu komputer, kecerdasan buatan, dan linguistik. NLP berfokus pada pengembangan sistem yang mampu memahami, menginterpretasikan, dan menghasilkan bahasa manusia dalam bentuk yang alami. Tujuan utama NLP adalah menjembatani kesenjangan antara bahasa manusia yang kompleks dan ambigu dengan bahasa mesin yang terstruktur dan eksplisit. Dalam praktiknya, NLP melibatkan berbagai teknik dan algoritma

untuk menganalisis struktur bahasa, makna kata, konteks, dan nuansa linguistik lainnya, memungkinkan komputer untuk "memahami" dan berinteraksi dengan bahasa manusia secara lebih efektif. (Hadinata & Stianingsih, 2024)

Menurut Huda (2021) dalam artikel yang berjudul “Implementasi *Natural Language Processing* (NLP) Untuk Aplikasi Pencarian Lokasi” Implementasi NLP mencakup berbagai aplikasi, mulai dari analisis sentimen dan ekstraksi informasi hingga penerjemahan otomatis dan pemrosesan *query* dalam bahasa alami. Dalam konteks pencarian lokasi, NLP memungkinkan sistem untuk menginterpretasikan permintaan pengguna yang diungkapkan dalam bahasa sehari-hari, mengekstrak informasi kunci seperti nama tempat, jenis lokasi, atau atribut geografis, dan mengubahnya menjadi *query* yang dapat diproses oleh mesin. Hal ini meningkatkan keakuratan dan relevansi hasil pencarian, serta meningkatkan pengalaman pengguna dengan memungkinkan interaksi yang lebih intuitif dan alami dengan sistem pencarian. Dengan kemajuan dalam pembelajaran mesin dan *deep learning*, kemampuan NLP terus berkembang, membuka peluang baru untuk aplikasi yang lebih canggih dan responsif dalam berbagai domain, termasuk sistem pencarian lokasi.

A. *Text Preprocessing*

Input awal dalam proses ini adalah dokumen utuh, tahap *text preprocessing* melibatkan persiapan teks menjadi data yang akan diolah pada tahap berikutnya. (Mustaqhfi dkk, 2012). Proses *text preprocessing* dalam penelitian ini terdiri dari beberapa langkah, seperti *cleaning*, *case folding*, memecahkan kalimat menjadi kata (*tokenizing*), menghilangkan kata *stopword*, dan *stemming*.

1. *Cleaning*

Cleaning merupakan tahap penting yang bertujuan untuk memastikan kualitas dan integritas data yang akan digunakan dalam analisis atau pemodelan. Langkah ini melibatkan identifikasi dan penghapusan data yang tidak relevan, seperti duplikasi atau entri yang tidak sesuai, memperbaiki data yang salah atau tidak lengkap dengan cara mengisi nilai yang hilang atau memperbaiki kesalahan penulisan, serta memastikan konsistensi format data. Melalui proses *cleaning*, data yang dihasilkan akan lebih bersih, akurat, dan dapat diandalkan, sehingga analisis atau model yang dibangun berdasarkan data tersebut akan lebih *valid* dan bermakna.

2. *Case Folding*

Case folding adalah proses dalam *preprocessing* teks yang mengubah semua karakter teks menjadi huruf kecil (*lowercase*). Tujuan utama dari *case folding* adalah untuk mengurangi variasi kata yang disebabkan oleh perbedaan huruf besar dan kecil, sehingga "Data", "data", dan "DATA" dianggap sebagai satu kata yang sama. Ini membantu dalam standarisasi teks dan meningkatkan konsistensi dalam analisis teks, seperti dalam aplikasi pemrosesan bahasa alami (*NLP*) atau *text mining*.

3. *Tokenization*

Tokenization adalah proses dalam *preprocessing* teks yang membagi teks menjadi token, yang dapat berupa kata, frasa, atau bahkan karakter, tergantung pada tujuan analisis. 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. *Stopword Removal*

Stopword Removal merupakan penghapusan kata-kata umum yang sering terlihat dalam teks tetapi tidak mengandung informasi yang signifikan untuk analisis, seperti "dan", "di", "yang", atau "a", "the", "is" dalam bahasa Inggris. Kata-kata ini, yang disebut *stopwords*, biasanya tidak memiliki makna signifikan dalam konteks analisis teks karena frekuensinya yang tinggi dan kontribusinya yang rendah terhadap pemahaman konten. Dengan menghapus *stopwords*, kita dapat mengurangi ukuran data dan meningkatkan efisiensi serta fokus analisis pada kata-kata yang lebih bermakna dan informatif.

5. *Stemming*

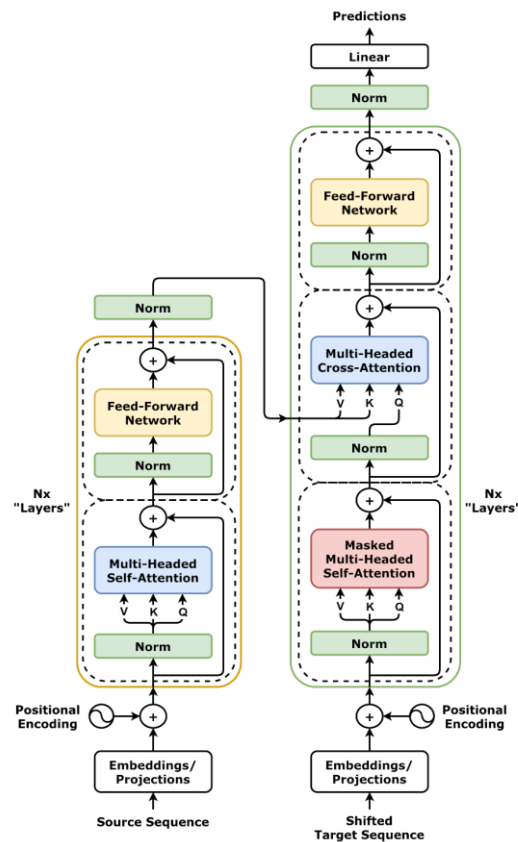
Stemming adalah proses untuk mengurangi kata-kata ke bentuk dasarnya atau akar katanya (*stem*). Proses ini menghilangkan akhiran atau imbuhan dari kata-kata untuk mengubahnya menjadi bentuk dasar yang lebih umum. Misalnya, kata-kata seperti "running", "runner", dan "ran" akan diubah menjadi "run". *Stemming* membantu dalam mengurangi variasi kata yang disebabkan oleh *infleksi* atau *derivasi*, sehingga meningkatkan konsistensi data teks dan mempermudah analisis lebih lanjut seperti pencarian informasi, klasifikasi teks, atau pemrosesan bahasa alami (NLP).

B. *Processing*

Tahap terpenting dari proses *text mining* adalah proses pengolahan. Tujuan dari tahap ini adalah menemukan pola atau pengetahuan tentang keseluruhan teks. Pada titik ini, pendekatan yang digunakan adalah dengan membatasi term yang dihasilkan dari tahap *preprocessing*. Banyak aplikasi menggunakan pembobotan kombinasi, yang terdiri dari perkalian bobot lokal *term* frekuensi dan dokumen *inversi* frekuensi global yang diproses dengan metode *BERT*.

2.1.7 Transformer

Transformer adalah model deep learning yang diperkenalkan pada tahun 2017 melalui makalah "*Attention is All You Need*" oleh peneliti dari *Google*. Model ini dirancang khusus untuk menangani data berurutan, seperti teks, dengan memanfaatkan arsitektur berbasis mekanisme perhatian (*attention*). Struktur *Transformer* terdiri dari dua elemen utama, yaitu *encoder* yang bertugas memproses data input, dan *decoder* yang menghasilkan output berdasarkan hasil dari *encoder*. Dengan mekanisme perhatian, model ini dapat memahami hubungan antar kata dalam teks, bahkan jika kata-kata tersebut berjauhan. Berbeda dengan model berulang seperti *RNN* atau *LSTM*. *Transformer* mendukung pemrosesan paralel, sehingga pelatihannya menjadi lebih cepat dan sangat cocok untuk dataset berukuran besar. (Hasan, 2022)



(Sumber : Devlin dkk, 2019)

Gambar 2.1 Arsitektur *Transformer*

Arsitektur *Transformer* terdiri dari dua komponen utama: *encoder* dan *decoder*, yang berfungsi untuk memproses data input dan menghasilkan output secara efisien. *Encoder* bertugas memahami konteks dari data input dengan menerapkan lapisan *self-attention* dan jaringan saraf *feed-forward*. *Self-attention* memungkinkan model untuk mempertimbangkan hubungan antara kata-kata dalam teks, bahkan jika kata-kata tersebut berjauhan, sehingga mampu memahami konteks yang lebih luas. Sementara itu, jaringan *feed-forward* pada *encoder* membantu dalam pemrosesan informasi untuk setiap kata secara individual. *Positional encoding* juga digunakan untuk memberikan informasi tentang urutan kata dalam input, karena arsitektur *Transformer* tidak memiliki elemen berulang seperti pada *RNN*. (Zhixing Tan dkk, 2020)

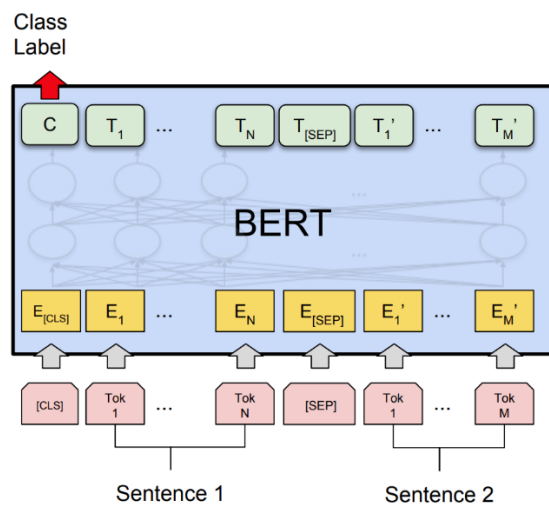
Di sisi lain, *decoder* juga menggunakan *self-attention* untuk memahami hubungan dalam output yang dihasilkan, serta mekanisme perhatian tambahan yang menghubungkannya dengan output dari *encoder*. Lapisan ini memungkinkan *decoder* untuk fokus pada bagian input yang relevan saat menghasilkan kata berikutnya dalam urutan output. Arsitektur ini juga mendukung pemrosesan paralel, karena tidak membutuhkan input dalam urutan tertentu, yang membuatnya lebih cepat dan efisien dibandingkan model-model sebelumnya seperti *LSTM* atau *RNN*. Kombinasi *encoder* dan *decoder* ini membuat *Transformer* sangat efektif untuk berbagai tugas *NLP* seperti terjemahan, penjawaban pertanyaan, dan teks otomatisasi. (Zhixing Tan dkk, 2020)

2.1.8 BERT (*Bidirectional Encoder Representations from Transformers*)

BERT, kependekan dari *Bidirectional Encoder Representation from Transformer*, merupakan terobosan dalam bidang pemrosesan bahasa alami. Model ini dirancang untuk memahami konteks kata secara mendalam, dengan mempertimbangkan hubungan antar kata dalam suatu kalimat secara dua arah. Keunikan *BERT* terletak pada kemampuannya menganalisis teks secara menyeluruh. Tidak seperti model sebelumnya yang hanya membaca teks dari satu arah, *BERT* mampu memahami makna kata berdasarkan konteks sebelum dan sesudahnya. Ini memungkinkan interpretasi yang lebih akurat dan kompleks dari bahasa manusia. (Sebastian dkk., 2022)

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model yang menggunakan arsitektur *Transformer bidirectional*, yang memungkinkan pemrosesan teks dengan memperhitungkan konteks dari kedua arah

(kiri ke kanan dan kanan ke kiri) secara bersamaan. Arsitektur ini terdiri dari beberapa lapisan *encoder transformer* yang dirancang untuk menghasilkan representasi kontekstual yang mendalam dari kata-kata dalam sebuah kalimat. Tidak seperti model sebelumnya yang menggunakan pendekatan *unidirectional*, *BERT* memanfaatkan teknik *Masked Language Model* (MLM), di mana sebagian kata dalam teks masukan disembunyikan secara acak, dan model dilatih untuk memprediksi kata yang hilang berdasarkan konteks di sekitarnya. Selain itu, *BERT* juga menggunakan tugas *Next Sentence Prediction* (NSP), yang dirancang untuk memahami hubungan antara dua kalimat, membuatnya efektif untuk tugas seperti klasifikasi teks, pemahaman pertanyaan, dan inferensi bahasa alami. (Fatmasari dkk., 2024)



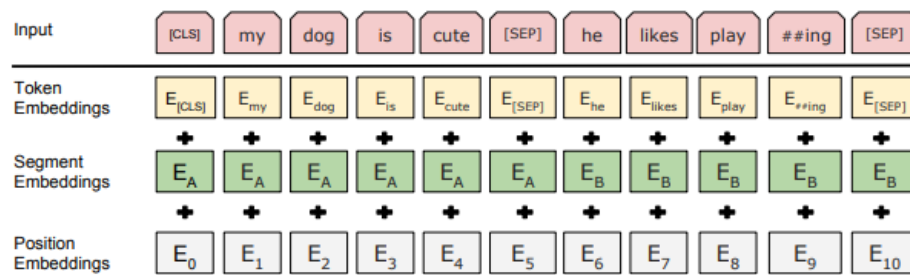
(Sumber : Devlin dkk, 2019)

Gambar 2.2 Arsitektur *BERT*

Dalam proses kerjanya, *BERT* melakukan dua tahap utama: *pre-training* dan *fine-tuning*. Selama *pre-training*, model dilatih menggunakan data teks yang tidak berlabel melalui tugas *MLM* dan *NSP* untuk menghasilkan representasi yang kaya secara kontekstual. Setelah *pre-training* selesai, model ini dapat disesuaikan untuk

berbagai tugas spesifik dengan menambahkan lapisan keluaran tambahan dan melatih ulang seluruh parameter model menggunakan data berlabel dari tugas tertentu. Misalnya, untuk tugas tanya-jawab, model menerima input berupa kombinasi pertanyaan dan paragraf konteks, yang kemudian diproses untuk menemukan rentang jawaban terbaik. Pendekatan ini membuat *BERT* fleksibel dan mampu memberikan hasil terbaik tanpa perlu modifikasi arsitektur yang signifikan untuk berbagai jenis tugas pemrosesan bahasa alami. (Fatmasari dkk, 2024)

Input pada model *BERT* (*Bidirectional Encoder Representations from Transformers*) dirancang untuk mendukung berbagai jenis tugas pemrosesan bahasa alami melalui pendekatan yang fleksibel dan terstruktur. Model ini menerima masukan berupa urutan token yang dihasilkan melalui proses tokenisasi menggunakan algoritma *WordPiece*, dengan kosakata sebanyak 30.000 token. Setiap urutan token dapat mencakup satu atau dua kalimat yang dipisahkan oleh token khusus [SEP]. Untuk merepresentasikan input secara lebih mendalam, *BERT* menggabungkan tiga jenis *embedding*: token *embeddings*, yang merepresentasikan setiap token sebagai *vektor*; *segment embeddings*, yang membedakan kalimat pertama dan kedua jika ada; serta *position embeddings*, yang menyandikan posisi relatif token dalam urutan teks. Selain itu, token khusus [CLS] ditempatkan di awal setiap urutan untuk digunakan sebagai representasi *agregat* dari seluruh input, menjadikannya sangat penting untuk tugas klasifikasi. (Yazid & Winarko, 2023)



(Sumber : Devlin dkk, 2019)

Gambar 2.3 Arsitektur *Layer BERT*

Output pada *BERT* bergantung pada tugas yang sedang diselesaikan. Untuk tugas klasifikasi, seperti analisis sentimen atau pengenalan hubungan antar kalimat, keluaran dari token *[CLS]* digunakan sebagai representasi keseluruhan teks, yang kemudian diteruskan ke lapisan klasifikasi untuk memprediksi label. Pada tugas token-level, seperti pengenalan entitas bernama (*NER*), keluaran dari setiap token digunakan untuk memprediksi label pada tingkat token. Dalam tugas tanya-jawab, seperti pada dataset *SQuAD*, model memprediksi posisi awal dan akhir jawaban dalam teks dengan menghitung probabilitas berdasarkan keluaran dari token-token yang relevan. Pendekatan ini memungkinkan *BERT* untuk menangkap hubungan kontekstual yang lebih baik, baik pada tingkat token maupun urutan, menjadikannya model yang sangat fleksibel untuk berbagai tugas pemrosesan bahasa alami. (Girinoto dkk, 2022)

Multilingual BERT membawa dimensi baru dalam dunia pemrosesan bahasa alami. Berbeda dengan model *BERT* standar, varian ini memiliki kemampuan unik untuk memahami dan mengolah teks dalam beragam bahasa. Dikembangkan dengan memanfaatkan dataset yang mencakup lebih dari 100 bahasa, termasuk Bahasa Indonesia, model ini menjadi pionir dalam mengatasi tantangan lintas bahasa dalam *NLP*. (Girinoto dkk., 2022)

Metode ini mampu memahami secara mendalam deskripsi dan konten mata kuliah, memungkinkan pencocokan yang lebih akurat antara mata kuliah yang akan dikonversi. Misalnya, dalam menganalisis silabus dan capaian pembelajaran, *BERT* dapat mengenali kemiripan konseptual bahkan ketika deskripsi mata kuliah menggunakan terminologi yang berbeda. Keunggulan *BERT* dalam sistem rekomendasi ini tidak hanya terletak pada akurasinya, tetapi juga pada kemampuannya beradaptasi dengan berbagai format deskripsi mata kuliah. Model ini dapat memproses informasi dari berbagai sumber, seperti katalog mata kuliah, silabus, atau bahkan materi pembelajaran, memberikan fleksibilitas yang tinggi dalam implementasinya di lingkungan akademik. Hal ini membuka peluang untuk pengembangan sistem konversi mata kuliah yang lebih canggih dan responsif terhadap kebutuhan mahasiswa dan institusi.

2.1.9 *Cosine Similarity*

Cosine similarity merupakan teknik canggih dalam analisis data yang memungkinkan perbandingan efektif antara dua atau lebih entitas yang direpresentasikan sebagai *vektor*. Metode ini telah membuktikan kegunaannya yang luas, terutama dalam bidang penambangan data dan pencarian dokumen. Keunggulannya terletak pada kemampuannya untuk mengukur tingkat kemiripan antar objek tanpa terpengaruh oleh perbedaan ukuran atau panjang *vektor*. Dalam konteks pencarian dokumen, *cosine similarity* mampu dengan cepat dan akurat mengidentifikasi dokumen-dokumen yang memiliki konten serupa, bahkan dalam kumpulan data yang sangat besar. Prinsip kerjanya yang berdasarkan sudut antara

vektor memungkinkan pendeteksian kemiripan yang lebih halus dan kontekstual, melampaui pendekatan pencocokan kata sederhana. (Kurniadi dkk., 2020)

Hal ini menjadikan *cosine similarity* sebagai pilihan yang sangat relevan dan efisien untuk berbagai aplikasi, mulai dari sistem rekomendasi hingga analisis plagiarisme, di mana pemahaman mendalam tentang kemiripan konten sangat diperlukan. Penulisan fungsi *cosine similarity* dapat dinyatakan dengan rumus :

$$\text{Similarity}(A,B) = \frac{(A \cdot B)}{(\|A\| \cdot \|B\|)} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}} \dots \dots \dots (2.1)$$

Keterangan :

- A = Vektor A, yang akan dibandingkan kemiripannya
- B = Vektor B, yang akan dibandingkan kemiripannya
- A • B = Cross product antara |A| dan |B|
- |A| = Panjang vektor A
- |B| = Panjang vektor B
- |A||B| = Cross product antara |A| dan |B|

2.1.10 Evaluasi Hasil Rekomendasi

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 relevansi sistem dalam mencerminkan proses manual. Namun, juga untuk mengidentifikasi sejauh mana sistem dapat diandalkan untuk menggantikan atau melengkapi metode konvensional.

2.2.Kajian Penelitian

Sebelum penelitian ini dilakukan, peneliti melakukan kajian terkait dengan penelitian menganalisis Sistem rekomendasi konversi mata kuliah berbasis nlp dengan menggunakan metode *BERT*. Tabel 2.2 adalah beberapa penelitian yang mendasari dalam melakukan penelitian.

Tabel 2.1 Kajian Penelitian Terdahulu

No.	Penulis	Tahun	Judul
1	Dwivedi & Roshni	2017	<i>Recommender system for big data in education</i>
2	Chen & McLeod	2021	<i>Collaborative Filtering for Information Recommendation Systems</i>
3	Henni Endah Wahanani, Made Hanindia Prami Swari, Fawwaz Ali Akbar	2022	Rancang Bangun Sistem Konversi Mata Kuliah (Studi Kasus : Prodi Informatika, Fasilkom, UPN "Veteran" Jawa Timur)
4	Glenn Ferio, Rolly Intan, Silvia Rostianingsih	2019	Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode <i>User Based Collaborative Filtering</i> Berbasis Algoritma <i>Adjusted Cosine Similarity</i>
5	Abdus Salam , Fauzan Putraga Albahri , Fathurrahmad	2022	Sistem Rekomendasi Tugas Akhir Mahasiswa pada AMIK Indonesia untuk Mendukung Merdeka Belajar-Kampus Merdeka Menggunakan Metode <i>Collaborative Filtering (CF)</i>
6	Hans Satria Kusuma, Aina Musdholifah	2021	<i>Recommendation System for Thesis Topics Using Content-based Filtering</i>
7	Dedi Darwis, Nery Siskawati & Zaenal Abidin	2020	Implementasi Algoritma Cosine Similarity Dan Metode TF-IDF Berbasis PHP Untuk Menghasilkan Rekomendasi Seminar
8	Akhmad Muzaki & Arita Witanti	2021	Sentiment Analysis Of The Community In The Twitter To The 2020 Election In Pandemic Covid-19 By Method Naive Bayes Classifier
9	Siti Saidah & Joanna Mayary	2022	Implementasi <i>Natural Language</i> dan Algoritma <i>Cosine Similarity</i> dalam Penilaian Ujian Esai Otomatis
10	Gerry Putra Fresnando	2024	Implementasi Metode Cosine Similarity Untuk Sistem Rekomendasi Konversi SKS MBKM Pada Departemen Teknologi Informasi ITS

Penelitian yang dilakukan Dwivedi & Roshni (2017) berjudul "*Recommender system for big data in education*" mengeksplorasi potensi integrasi teknologi big data dalam sistem rekomendasi pendidikan. Studi ini mengusulkan

sebuah pendekatan inovatif yang menggabungkan metode *collaborative filtering* dan *content-based filtering*, didukung oleh teknologi *big data* seperti *Hadoop* dan *MapReduce*. Sistem yang dirancang bertujuan untuk memberikan rekomendasi yang personal dan relevan kepada peserta didik, mencakup aspek-aspek seperti materi pembelajaran, pemilihan kursus, dan perencanaan karir. Melalui pengolahan dan analisis data pendidikan dalam skala besar, penelitian ini mendemonstrasikan potensi peningkatan akurasi dan relevansi rekomendasi. Para peneliti juga mengidentifikasi tantangan implementasi, termasuk masalah privasi data dan kebutuhan pembaruan model yang berkelanjutan. Penelitian studi ini menyoroti bagaimana pemanfaatan *big data* dalam sistem rekomendasi dapat membuka jalan bagi personalisasi pendidikan yang lebih efektif, memberikan kontribusi signifikan terhadap peningkatan kualitas dan efisiensi proses pembelajaran di era digital.

Penelitian kedua yang dilakukan oleh Chen & McLeod (2022) mengkaji penerapan teknik *Collaborative Filtering* dalam konteks sistem rekomendasi informasi. Studi ini mengeksplorasi bagaimana *Collaborative Filtering* dapat meningkatkan akurasi dan relevansi rekomendasi dengan memanfaatkan pola perilaku dan preferensi pengguna. Para peneliti menyelidiki berbagai algoritma *Collaborative Filtering*, termasuk pendekatan berbasis memori dan berbasis model, serta menganalisis kelebihan dan tantangan masing-masing metode. Mereka juga membahas isu-isu seperti *cold start problem*, skalabilitas, dan privasi data dalam implementasi sistem rekomendasi berbasis *Collaborative Filtering*. Hasil penelitian menunjukkan bahwa teknik *Collaborative Filtering*, ketika diimplementasikan dengan tepat, dapat secara signifikan meningkatkan kualitas rekomendasi

informasi, menyediakan pengalaman yang lebih personal dan relevan bagi pengguna. Studi ini juga menyoroti pentingnya *hybrid approaches* yang menggabungkan *Collaborative Filtering* dengan teknik lain untuk mengatasi keterbatasan inherent dari masing-masing metode.

Berdasarkan berbagai penelitian sebelumnya, sistem rekomendasi telah berkembang dengan berbagai pendekatan, seperti *Collaborative Filtering* yang digunakan untuk merekomendasikan mata kuliah pilihan atau tugas akhir mahasiswa berdasarkan kesamaan pengguna, serta *Content-Based Filtering* yang diterapkan pada topik skripsi. Sedangkan *Cosine Similarity* digunakan dalam sistem rekomendasi untuk konversi SKS atau penilaian otomatis ujian esai. Namun, pendekatan-pendekatan tersebut umumnya hanya mempertimbangkan satu dimensi data atau konteks tertentu, sehingga terbatas dalam menangkap hubungan yang lebih kompleks dalam data berukuran besar. Kebaruan penelitian ini terletak pada pemanfaatan *BERT (Bidirectional Encoder Representations from Transformers)* sebagai model rekomendasi berbasis pemahaman konteks mendalam, penelitian ini diharapkan dapat menghadirkan sistem rekomendasi konversi mata kuliah yang lebih presisi dan relevan untuk mendukung pengambilan keputusan di Prodi Teknik Informatika Universitas Islam Balitar.