BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan suatu proses dalam data mining yang bertujuan untuk memahami sosial sentimen pada teks tersebut dengan mengindentifikasi dan mengekstrak suatu informasi [12]. Fakta dan opini merupakan bagian dari informasi tersebut. Opini adalah pandangan subjektif yang mencerminkan cara seseorang menyampaikan pendapatnya tentang berbagai hal, berdasarkan persepsi dan asumsi pribadi. Opini ini dapat digunakan sebagai analisa untuk dilakukan Analisis sentimen tentang suatu produk dan pelayanan [13].

Analisis sentimen teknologi khususnya pada Chatbot cenderung ke persepsi pengguna mengenai ChatGPT. Dalam pembahasan ini, aplikasi ChatGPT memiliki banyak sekali topik yang di bicarakan melalui media sosial seperti Twitter. Pengguna Twitter umumnya sering membahas mengenai performa, fitur, dan pengalaman mereka dalam penggunaan aplikasi ChatGPT. Contohnya pembahasan topik penggunaan ChatGPT untuk edukasi, kebanyakan pengguna sangat terbantu dengan ChatGPT karena dapat menyeelesaikan tugas mereka dengan cepat, seperti tugas membuat essay, skrip, kode dan lain-lain. Namun, hal ini menimbulkan opini negatif, seperti menganggap menggunakan ChatGPT untuk menyelesaikan tugasnya adalah hal yang curang, dan juga mengurangi tingkat kreatifitas karena pengguna ChatGPT hanya perlu mengetik *prompt* di ChatGPT untuk mendapatkan hasil yang diinginkan penggunanya [6].

2.2 Text Preprocessing

Text preprocessing adalah langkah penting yang bertujuan untuk membersihkan data dari kata-kata tertentu yang dapat memengaruhi hasil analisis sentimen. Dengan melakukan *preprocessing* data terlebih dahulu, peningkatan kinerja model BERT ditunjukkan dan ada juga penelitian tentang berbagai teknik preprocessing yang diterapkan dan bagaimana hal tersebut mengubah kinerja masing-masing pengklasifikasi [13]. Pada penelitian ini, Text Preprocessing memiliki beberapa tahap.

- *Case Folding*

  Case folding adalah metode pengubahan huruf kapital menjadi huruf kecil,

karena Penggunaan huruf kapital pada sebuah teks tidak selalu konsisten dan dapat memengaruhi performa pada training data.

- *Normalization*
  Normalization adalah proses menghapus tanda baca seperti titik, koma, tanda seru, tanda petik, serta angka numerik dan simbol lainnya dari teks.

- *Tokenization*
  Tokenization merupakan proses pememecahan tanggapan menjadi satuan kata dengan melihat spasi antar kalimat [14].

2.3 Bidirectional Encoder Representations from Transformers (BERT)

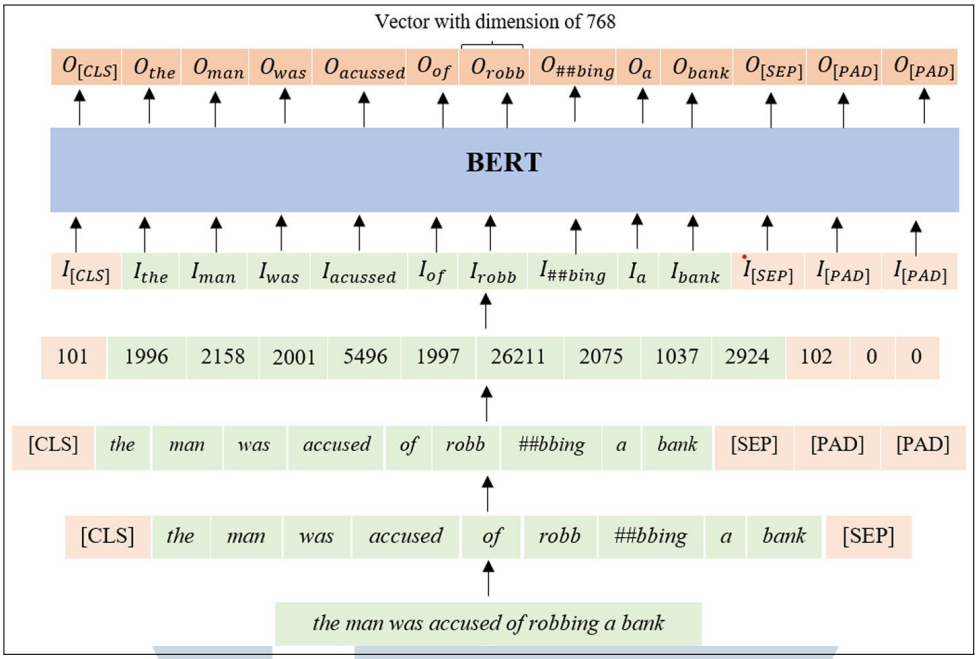
Bidirectional Encoder Representations from Transformers (BERT) adalah model bidirectional Natural Language Processing (NLP) yang di train dengan encoder berdasarkan model Transformer.

2.3.1 Representasi input/Output model BERT

Untuk melakukan proses klasifikasi menggunakan BERT, dibutuhkan tahap pra proses teks yaitu *tokenization*, *padding*, dan *encoding*. *Tokenization* pada BERT dilakukan dengan menggunakan model yang tersedia, lalu ditambahkan token [CLS] dan [SEP] pada awal dan akhir teks [15].

Padding dilakukan untuk memastikan bahwa setiap teks di dalam data tersebut memiliki ukuran token yang sama. Misalkan apabila jumlah token pada sebuah data tersebut 25 token, setiap teks dengan jumlah token lebih sedikit dari 25 akan di *padding* dengan token spesial [PAD] sampai ukuran tersebut mencapai 25 token. Di saat yang sama, teks yang memiliki lebih dari 25 token akan dipotong hanya sampai dengan 25 token dan token setelahnya akan menjadi token [SEP] [15].

Langkah selanjutnya adalah *encoding*, tujuannya adalah untuk memetakan token menjadi integer untuk memproses dokumen dengan BERT. Proses *encoding* token di lakukan dengan membuat peta dengan token model yang disediakan sebagai key dan integer yang sesuai sebagai value. Token pada setiap dokumen akan di petakan ke dalam ineger yang sesuai sehingga integer yang lain dapat merepresentasikan setiap token. Berikut merupakan ilustrasi input dan output model BERT [15].

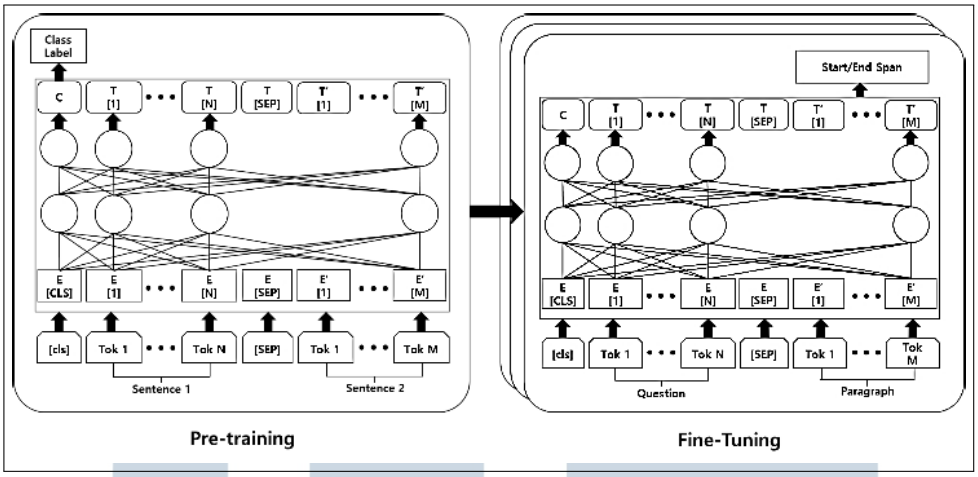


Gambar 2.1. Ilustrasi *input* dan *output* BERT

Sumber: [15]

2.3.2 Proses training model BERT

Model BERT membagi proses Training ke dalam 2 tahap, yaitu Pretraining dan Fine-Tuning. Pretraining adalah tahap inisiasi bobot suatu model yang sebelumnya di inisiasi ke dalam value yang acak ke bobot yang di train di dalam masalah lain. Fine-tuning adalah tahap untuk training model secara lebih dalam dengan menambah minimum bobot untuk downstream task ke semua bobot pre-trained. Berikut adalah Gambar prosedur pre-trainning dan fine-tuning [16].



Gambar 2.2. Ilustrasi *pre-training* dan *fine-tuning BERT*
Sumber: [16]

2.4 Confusion Matrix

Confusion Matrix adalah metode yang digunakan untuk mengukur tingkat akurasi suatu model. Confusion Matrix disajikan seperti pada tabel berikut [17].

Tabel 2.1. Confusion Matrix

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP	FP
Negatif	FN	TN

Keterangan:

- TP (True Positive): Data aktual Positif yang terklasifikasi positif.
- TN (True Negative): Data aktual Negatif yang terklasifikasi Negatif.
- FN (False Negative): Data aktual Positif yang terklasifikasi negatif.
- FP (False Positive): Data aktual Negatif yang terklasifikasi positif.

Terdapat beberapa rumus yang dipakai Confusion Matrix untuk menghitung Nilai akurasi, yaitu *Accuracy*, *Precicion*, *Recall*, dan *F1-Score*, berikut dijelaskan rumus-rumus yang digunakan.

1. *Accuracy*

Accuracy adalah indikator ukuran yang menunjukkan sejauh mana model dapat memprediksi label dengan tepat secara keseluruhan. Nilai ini diperoleh dengan membandingkan jumlah prediksi yang benar dengan total prediksi yang dilakukan. Berdasarkan confusion matrix, *accuracy* dapat dihitung dengan menjumlahkan elemen-elemen pada diagonal utama, lalu membaginya dengan jumlah seluruh elemen lainnya. Rumus untuk menghitung *accuracy* adalah sebagai berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2. *Precision*

   Precision adalah performance metrik yang digunakan untuk mengukur performa. *Precision* didefinisikan sebagai rasio pada positif yang sebenarnya dengan total angka prediksi positif [9].

$$Presisi = \frac{TP}{TP + FP} \tag{2.2}$$

3. *Recall*

   Recall juga digunakan untuk mengukur performa suatu model. *Recall* dihitung dengan membagi jumlah positif yang benar dengan jumlah positif yang benar dan juga positif yang salah [9].

$$Presisi = \frac{TP}{TP + FN} \tag{2.3}$$

4. F1-Score

   F1-score merupakan metrik yang lebih baik dibandingkan metriks lain dalam kondisi dimana kelas-kelas tidak seimbang karena mempertimbangkan *precision* dan *recall* lalu memberikan pemahaman yang lebih baik tentang kinerja model [9].

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \tag{2.4}$$