BAB 2
LANDASAN TEORI

2.1 TransJakarta

Berdasarkan situs Transjakarta.co.id, TransJakarta merupakan layanan transportasi publik berbasis *Bus Rapid Transit* yang pertama kali beroperasi di kawasan Asia Tenggara dan Asia Selatan pada 1 Februari 2004, dengan tujuan melayani kebutuhan mobilitas masyarakat di DKI Jakarta dan wilayah sekitarnya [15]. Seiring perkembangannya, layanan TransJakarta terus ditingkatkan melalui perluasan rute, penambahan armada, dan modernisasi sistem operasional. Secara kelembagaan, TransJakarta yang awalnya berstatus badan pengelola kemudian resmi bertransformasi menjadi Badan Usaha Milik Daerah bernama PT Transportasi Jakarta sejak 27 Maret 2014 [15]. Selain menyediakan layanan transportasi publik, TransJakarta juga aktif memanfaatkan media sosial X (sebelumnya Twitter) melalui akun resmi @PT_Transjakarta sebagai sarana komunikasi dan informasi pengaduan masyarakat.

2.2 Emosi

Dalam kajian psikologi, *American Psychological Association* (APA) mendefinisikan emosi sebagai suatu pola reaksi kompleks yang mencakup pengalaman, perilaku, serta respons fisiologis. Emosi dipahami sebagai bentuk respons individu terhadap situasi yang memiliki makna penting secara personal, yang umumnya terdiri dari tiga komponen, yaitu pengalaman subjektif, reaksi fisiologis, dan ekspresi perilaku [16].

Dalam kategori emosi, *Basic Emotions Theory* yang dikemukakan oleh Paul Ekman (1992) menjadi salah satu kerangka utama dalam studi emosi, terutama untuk pengenalan emosi berbasis teks di media sosial [17]. Teori tersebut mengklasifikasikan enam emosi dasar manusia, yaitu *joy*, *sadness*, *anger*, *fear*, *surprise*, dan *disgust* [18]. Keenam emosi tersebut dianggap bersifat universal karena dapat dikenali pada berbagai budaya melalui ekspresi maupun respons emosional manusia. Maka demikian, penelitian ini menggunakan keenam emosi dasar tersebut sebagai kelas utama dalam proses klasifikasi emosi.

Karakteristik data media sosial tidak selalu mengandung emosi yang jelas. Banyak unggahan hanya berupa informasi, pertanyaan, laporan kondisi, atau

pernyataan objektif yang tidak menunjukkan dominasi salah satu emosi dasar. Apabila seluruh data dipaksakan ke dalam enam emosi Ekman, maka *tweet* yang sebenarnya tidak mengandung emosi berpotensi memperoleh label yang kurang akurat. Maka demikian, penelitian ini menambahkan satu kategori, yaitu kelas *neutral* untuk data teks yang tidak memiliki kecenderungan terhadap salah satu emosi dasar atau tidak ditemukan emosi yang dominan, sehingga penelitian ini menggunakan tujuh kelas emosi, yaitu *joy*, *sadness*, *anger*, *fear*, *surprise*, *disgust*, dan *neutral*.

2.3 Media Sosial X (sebelumnya Twitter)

X merupakan platform *microblogging* yang memfasilitasi pengguna dalam berbagi informasi, opini, dan pengalaman pribadi dalam bentuk pesan singkat atau *tweet* [1]. Platform ini menyediakan berbagai fitur seperti *retweet*, *reply*, dan *like* yang mendukung interaksi antar pengguna serta mempercepat penyebaran informasi secara luas. Media sosial X juga bersifat *real-time*, terbuka, dan mencerminkan persepsi publik secara spontan yang digunakan sebagai sumber data dalam penelitian analisis opini publik [1]. Karakteristik teks pada media sosial X umumnya bersifat singkat, informal, dan banyak menggunakan singkatan, *slang*, maupun bahasa tidak baku. platform ini dapat dimanfaatkan sebagai sumber data yang representatif untuk mengeksplorasi berbagai opini dan tanggapan masyarakat mengenai kualitas layanan publik, termasuk layanan transportasi umum [3].

2.4 *Natural Language Processing*

Natural Language Processing adalah cabang ilmu kecerdasan buatan yang mempelajari cara agar komputer dapat mengenali, menafsirkan, mengolah, dan menghasilkan bahasa alami yang digunakan manusia dalam komunikasi sehari-hari [19]. Bidang ini menggabungkan berbagai linguistik komputasional, statistik, dan *machine learning* untuk mengambil informasi dan makna dari data berbentuk teks. Dalam hal ini, *Natural Language Processing* dapat berperan sebagai penghubung antara manusia dan mesin melalui penggunaan bahasa alami dalam komunikasi sehari-hari. Perkembangan teknologi yang semakin pesat juga mendorong penggunaan model berbasis *deep learning*, terutama arsitektur *transformer*, seperti BERT dan GPT, yang saat ini banyak digunakan untuk menyelesaikan berbagai tugas dalam *Natural Language Processing* [19].

2.5 Klasifikasi Teks

Klasifikasi teks adalah proses pemberian label pada dokumen atau kalimat secara otomatis dengan memanfaatkan pola dan informasi yang terdapat dalam isi teks, sehingga dapat diproses dan dianalisis secara otomatis dalam skala besar. Proses ini umumnya bekerja melalui pendekatan *machine learning*, di mana sebuah model dilatih menggunakan data berlabel untuk mengenali pola dari setiap kategori teks. Teknik ini merupakan bagian dasar dalam *Natural Language Processing* yang membantu komputer memahami struktur dan isi teks secara lebih mendalam, sehingga memungkinkan otomatisasi berbagai tugas yang sebelumnya memerlukan keterlibatan manusia. Melalui klasifikasi teks, sistem dapat merespons masukan teks dengan tepat, menyesuaikan layanan dengan kebutuhan pengguna, dan mengalisis opini dan sentimen secara efektif [19].

2.6 *Tweet Harvest*

Tweet Harvest merupakan sistem *open-source* berbasis *command line* yang dikembangkan oleh Helmi Satria untuk mendapatkan data dari Twitter. Alat ini memanfaatkan *library Playwright* untuk melakukan *web scraping* pada halaman pencarian X berdasarkan *keyword* dan rentang waktu tertentu [20]. *Tweet Harvest* dimanfaatkan sebagai alat untuk mengambil data *tweet* yang terkait dengan TransJakarta yang kemudian digunakan sebagai *dataset* dalam proses analisis klasifikasi emosi.

2.7 *Preprocessing*

Preprocessing adalah proses beberapa langkah untuk membersihkan dan mengubah teks dalam *dataset* yang tidak terstruktur menjadi format yang dapat dipahami oleh model NLP [19]. Dengan *preprocessing*, data menjadi bersih dan terstruktur, serta disederhanakan untuk memudahkan pemrosesan selanjutnya. Proses ini memastikan model hanya menggunakan informasi yang relevan dalam mengidentifikasi kalimat, sehingga dapat lebih efektif dalam melakukan analisis. Tahapan ini memiliki beberapa proses sebagai berikut:

1. ***Cleaning*** merupakan tahapan pembersihan data dengan menghapus berbagai elemen yang tidak diperlukan dalam proses analisis. Pada tahap ini dilakukan

penghapusan simbol, angka, tanda baca, URL, *mention*, *hashtag*, serta karakter selain huruf dan spasi sehingga teks menjadi terstruktur.

2. ***Case Folding*** merupakan tahapan untuk menyeragamkan semua teks menjadi huruf kecil.

3. ***Normalization*** merupakan tahapan untuk merubah format kata yang tidak baku menjadi kata baku. Ini termasuk memperbaiki ejaan, mengubah angka ke dalam bentuk kata, dan memperbaiki singkatan.

4. ***Tokenization*** merupakan proses memecah teks menjadi unit yang lebih kecil yang disebut token.

5. ***Stopword Removal*** merupakan tahapan menghilangkan kata-kata yang terlalu sering muncul dalam teks tetapi tidak memiliki makna yang signifikan dalam analisis.

2.8 *NRC Emotion Lexicon*

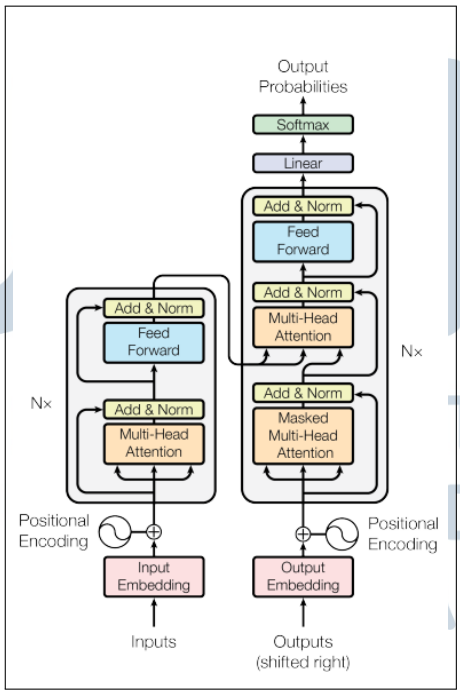
National Research Council Emotion Lexicon atau *NRC Emotion Lexicon* merupakan leksikon emosi yang digunakan dalam analisis teks pada bidang NLP untuk mengidentifikasi emosi yang terkandung dalam suatu kata. Leksikon ini dikembangkan oleh Mohammad dan Turney yang mengelompokkan kata ke dalam delapan emosi menurut teori Plutchik, yaitu *anger*, *fear*, *anticipation*, *trust*, *surprise*, *sadness*, *joy*, dan *disgust*, serta dua kategori sentimen, yaitu positif dan negatif [21]. Meskipun *NRC Emotion Lexicon* mengadopsi delapan kategori emosi menurut teori Plutchik, penelitian ini hanya memanfaatkan enam kategori emosi yang selaras dengan teori emosi dasar Paul Ekman, yaitu *joy*, *sadness*, *anger*, *fear*, *surprise*, dan *disgust*. Dua kategori lainnya, yaitu *trust* dan *anticipation*, tidak digunakan sebagai kelas keluaran karena keduanya tidak termasuk ke dalam enam emosi dasar Ekman yang menjadi landasan penelitian ini.

NRC Emotion Lexicon ini digunakan untuk melakukan pelabelan emosi secara otomatis melalui metode *string matching*, yaitu dengan mencocokkan setiap kata dalam satu kalimat dengan kamus *NRC Emotion Lexicon*. Setiap kata yang ditemukan akan memberikan skor pada kategori emosi tersebut, sedangkan kata yang tidak terdapat dalam leksikon akan diabaikan. Skor tersebut kemudian diakumulasikan untuk menentukan emosi yang paling dominan dalam kalimat tersebut. Metode ini digunakan untuk menghasilkan label emosi secara

otomatis pada data dalam jumlah besar tanpa memerlukan pelabelan manual yang memerlukan waktu banyak. Label emosi yang dihasilkan berasal dari skor leksikon dan bukan dari anotasi manusia secara langsung, sehingga label tersebut merupakan *proxy emotion labels* dan bukan *ground-truth emotional states* [22]. Pada penelitian ini, *NRC Emotion Lexicon* akan digunakan sebagai metode pelabelan awal untuk menentukan label emosi pada *dataset* sebelum dilakukan proses pelatihan dan klasifikasi model.

2.9 Model *Transformer*

Model *Transformer* pertama kali diperkenalkan oleh Vaswani et al. (2017) melalui publikasi *Attention is All You Need* [23]. Arsitektur ini dianggap sebagai terobosan dalam pengembangan model *deep learning* karena menggunakan mekanisme *self-attention* untuk menangkap hubungan antar kata dalam teks tanpa menggunakan proses sekuensial seperti pada *Recurrent Neural Network* dan *Long Short-Term Memory*. Pendekatan tersebut memungkinkan model memperoleh pemahaman kontekstual yang lebih baik terhadap suatu kalimat. Selain itu, *transformer* mampu memproses data secara paralel sehingga proses pelatihan menjadi lebih efisien dibandingkan model berbasis sekuensial.



Gambar 2.1. Arsitektur Model *Transformer*

Sumber: [23]

Berdasarkan Gambar 2.1, arsitektur dasar *transformer* terdiri atas dua komponen utama, yaitu *encoder* dan *decoder*. Arsitektur ini dikembangkan untuk menyelesaikan tugas *sequence-to-sequence* seperti penerjemahan mesin, di mana *encoder* bertugas mengubah urutan token masukan menjadi representasi kontekstual, sedangkan *decoder* menggunakan representasi tersebut untuk menghasilkan urutan token keluaran secara bertahap, yaitu satu token demi satu token berdasarkan token yang telah dihasilkan sebelumnya. Setiap komponen dibangun dari beberapa *layer* yang disusun secara berulang untuk meningkatkan kemampuan model dalam mempelajari representasi data [23].

Pada bagian *encoder*, setiap lapisan terdiri atas *multi-head self-attention* dan *feed forward network* yang masing-masing diikuti oleh mekanisme *Add & Norm*. Seluruh token pada masukan dapat saling memperhatikan (*self-attention*) sehingga model mampu memahami hubungan antar kata dalam satu kalimat secara menyeluruh. Sementara itu, pada bagian *decoder*, selain itu terdapat juga lapisan *encoder-decoder attention* yang memungkinkan *decoder* memanfaatkan informasi yang telah dipelajari oleh *encoder*. Penggunaan *masked self-attention* memastikan bahwa proses prediksi hanya bergantung pada token-token sebelumnya sehingga keluaran dapat dihasilkan secara berurutan [23].

Transformer memanfaatkan *positional encoding* untuk memberikan informasi mengenai posisi atau urutan setiap token dalam suatu kalimat sehingga model tetap dapat memahami struktur urutan teks. Arsitektur ini kemudian menjadi fondasi bagi pengembangan berbagai model bahasa modern *Large Language Models*, termasuk BERT dan GPT [10]. Perbedaan utama keduanya terletak pada arsitektur yang digunakan, di mana BERT hanya memanfaatkan bagian encoder untuk menghasilkan representasi kontekstual dua arah (*bidirectional*), sehingga setiap token dapat memperhatikan konteks dari sisi kiri maupun kanan secara bersamaan [10]. Sebaliknya, GPT hanya menggunakan *decoder* dengan mekanisme *masked self-attention* sehingga proses prediksi dilakukan secara sekuensial berdasarkan token-token sebelumnya. Oleh karena itu, BERT lebih sesuai untuk tugas pemahaman bahasa, sedangkan GPT lebih sesuai untuk tugas pembangkitan teks (*text generation*) [23].

Selain sebagai arsitektur model, *transformer* tersedia dalam pustaka *open source* yang dikembangkan oleh *Hugging Face*. Pustaka ini dirancang untuk mendukung implementasi berbagai model berbasis *transformer* dan menyediakan akses terhadap model *pre-trained* untuk kebutuhan penelitian [24]. Pustaka ini menyediakan alur kerja standar dalam pemodelan, mulai dari memproses data,

menerapkan model, dan menghasilkan prediksi. Melalui pustaka ini, peneliti dapat melakukan proses tokenisasi, *fine-tuning*, dan inferensi model pada berbagai *downstream task* seperti klasifikasi teks atau analisis sentimen.

2.10 *Bidirectional Encoder Representations from Transformers* (BERT)

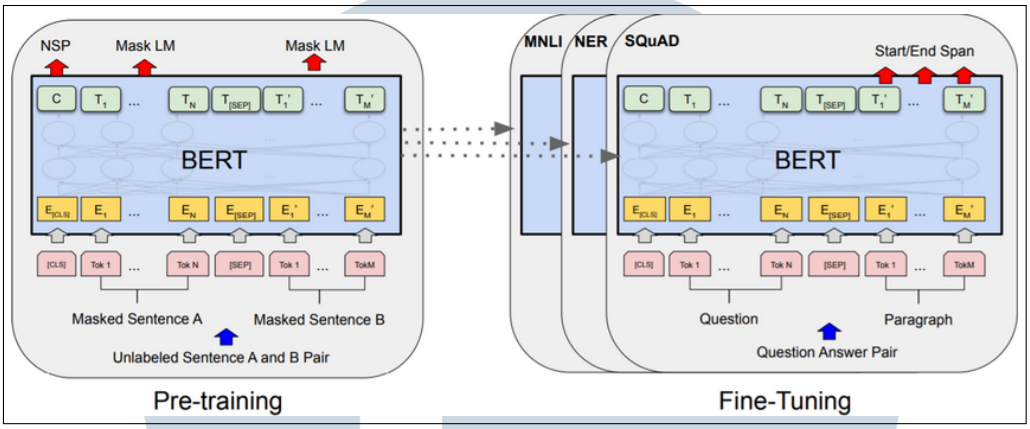
BERT adalah model *pre-trained* berbasis arsitektur *transformer* yang diperkenalkan Google AI pada tahun 2018 yang membawa revolusi pemahaman bahasa alami [10]. BERT dirancang menggunakan *bidirectional* yang dimana model dapat memahami konteks sebuah kata berdasarkan keseluruhan kalimat dengan memanfaatkan *self-attention*, sehingga setiap token dapat dipahami berdasarkan hubungan dan informasi yang berasal dari token sebelum maupun sesudahnya dalam suatu urutan teks.

Arsitektur BERT memanfaatkan serangkaian lapisan transformer *encoder* yang bekerja untuk menghasilkan representasi kontekstual dari setiap token. Setiap lapisan mengimplementasikan mekanisme *Multi-Head Self-Attention* yang memungkinkan model untuk memahami berbagai aspek dari teks dengan menangkap informasi dari perspektif yang berbeda. Representasi yang dihasilkan kemudian diproses oleh *Feed Forward Neural Network* untuk mentransformasikan dan memahami informasi yang telah diperoleh sebelum diteruskan ke lapisan *encoder* berikutnya. Pelatihan BERT dilakukan dua tahapan pembelajaran, yaitu *pre-training* dan *fine-tuning*. Pada tahapan *pre-training*, model mempelajari karakteristik bahasa dari kumpulan data teks dalam jumlah besar menggunakan metode *self-supervised learning*, yaitu:

1. ***Masked Language Modeling*** **(MLM)**: dilakukan dengan menutupi token pada teks input secara acak menggunakan token $[MASK]$. Kemudian model harus memprediksi token yang ditutup berdasarkan informasi konteks dari kedua arah kalimat [10].
2. ***Next Sentence Prediction*** **(NSP)**: merupakan tugas pelatihan yang bertujuan untuk memahami apakah dua kalimat memiliki hubungan atau tidak [10].

Setelah melalui proses *pre-training*, model memasuki tahap *fine-tuning*. Pada tahap ini, *ouput layer* ditambahkan untuk kebutuhan tugas baru(*downstream task*). Seluruh parameter model kemudian dilatih ulang secara *end-to-end* menggunakan *dataset* yang lebih spesifik. Pendekatan ini memberikan fleksibilitas

tinggi sehingga BERT mampu mencapai performa yang unggul pada berbagai tugas. Ilustrasi proses *pre-training* dan *fine-tuning* ditunjukkan pada Gambar 2.2.



Gambar 2.2. Ilustrasi *Pre-training* dan *Fine-tuning* BERT
Sumber: [10]

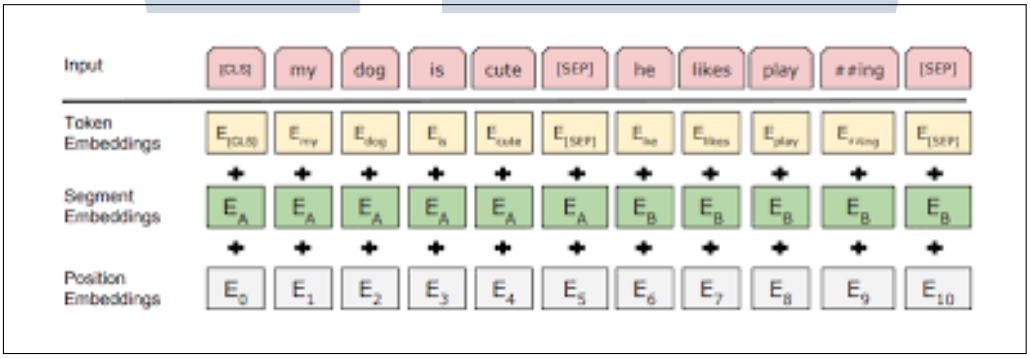
BERT memiliki dua konfigurasi utama, yaitu BERT-*base* dan BERT-*large*. BERT-*base* terdiri atas 12 *layer* dengan sekitar 110 juta parameter, sedangkan BERT-*large* terdiri atas 24 *layer* dengan sekitar 340 juta parameter [10]. Perbedaan jumlah *layer* dan parameter ini mempengaruhi kemampuan model dalam menangkap konteks, BERT-*large* memiliki kemampuan representasi yang lebih baik tetapi membutuhkan komputasi yang lebih besar.

Selain itu, salah satu karakteristik dalam BERT adalah penggunaan *embedding* sebagai representasi input. Setiap urutan input diawali dengan token khusus [*CLS*] yang digunakan sebagai representasi keseluruhan kalimat dalam tugas klasifikasi, serta token [*SEP*] yang berfungsi sebagai pemisah antar kalimat. Setiap token dalam input diproses melalui tiga jenis *embedding*, yaitu:

1. ***Token Embedding***, yaitu representasi numerik yang dibentuk untuk setiap token pada teks. Representasi ini memungkinkan model menangkap informasi semantik dasar yang terkandung dalam kata atau sub-kata yang diproses [10].
2. ***Segment Embedding***, yaitu komponen yang menandai asal suatu token berdasarkan segmen kalimatnya. Hal ini membantu model mengidentifikasi apakah token tersebut berasal dari kalimat pertama atau kalimat kedua dalam pasangan kalimat [10].

3. ***Position Embedding***, yaitu representasi yang menyimpan informasi mengenai letak token dalam urutan teks. Dengan adanya informasi posisi, model dapat mempertahankan pemahaman terhadap susunan dan hubungan antar kata dalam suatu kalimat [10].

Kombinasi ketiga jenis *embedding* tersebut memungkinkan model untuk memahami identitas token, struktur segmentasi, serta urutan kemunculan kata dalam teks. Informasi tersebut kemudian diolah oleh *transformer encoder* sehingga menghasilkan representasi yang mempertimbangkan konteks kalimat. Representasi dari token `[CLS]` digunakan untuk melakukan klasifikasi dan menghasilkan label kelas. Ilustrasi proses representasi input BERT pada Gambar 2.3.

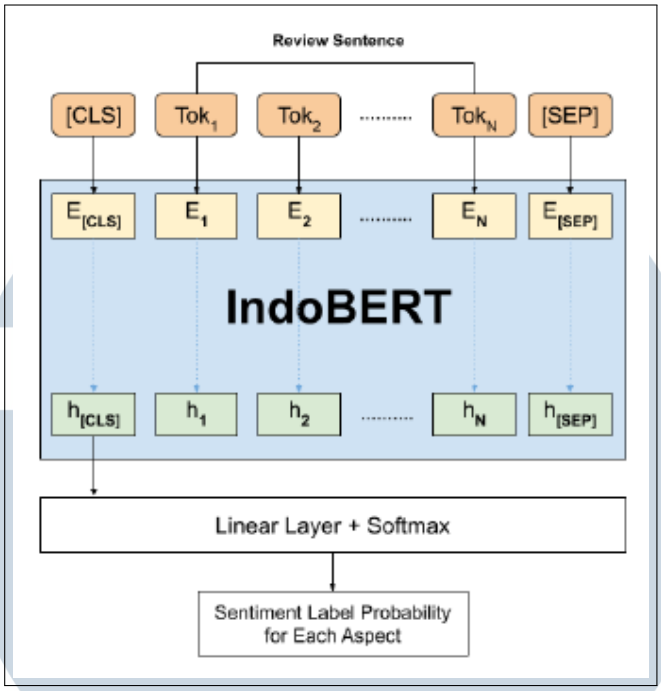


Gambar 2.3. Representasi Input Tiga *Embedding* pada BERT

Sumber: [10]

2.11 IndoBERT

Sebagai adaptasi BERT untuk bahasa Indonesia, IndoBERT dibangun dengan arsitektur *transformer* yang serupa dengan BERT. Pada proses pelatihannya, model memanfaatkan tugas MLM untuk memprediksi token yang disembunyikan dan NSP untuk mempelajari hubungan antar kalimat. Setelah proses *pre-training*, model dapat disesuaikan dengan berbagai tugas melalui tahap *fine-tuning*. Berbeda dari BERT yang dilatih menggunakan data berbahasa Inggris, model ini dilatih menggunakan korpus berskala besar yang terdiri lebih dari 220 juta kata yang dikumpulkan dari Wikipedia bahasa Indonesia (74 juta kata), media daring Kompas, Tempo, Liputan6 (55 juta kata), dan Indonesian Web Corpus (90 juta kata) [25]. Ilustrasi arsitektur dan alur pemrosesan IndoBERT ditampilkan pada Gambar 2.4.



Gambar 2.4. Arsitektur dan Alur Kerja IndoBERT

Sumber: [26]

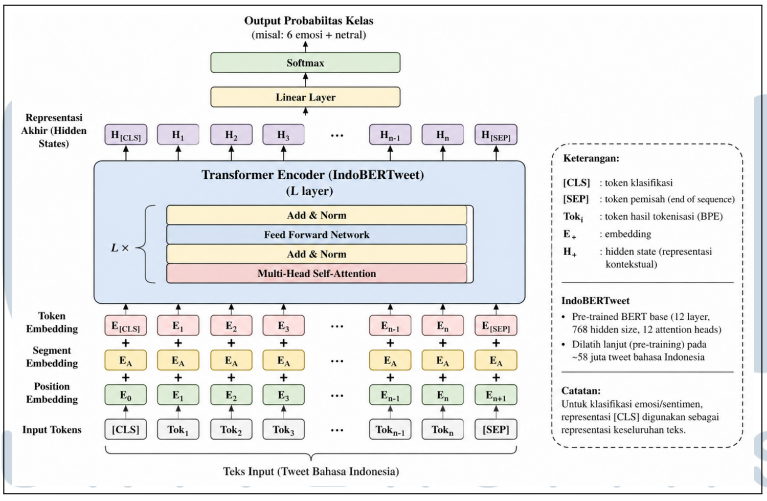
Gambar 2.4 memperlihatkan alur kerja IndoBERT dalam melakukan klasifikasi sentimen. Teks ulasan diproses melalui tahap tokenisasi dan *embedding*, kemudian direpresentasikan secara kontekstual oleh *transformer encoder* IndoBERT. Token [*CLS*] berfungsi sebagai ringkasan informasi kalimat dan diteruskan ke lapisan klasifikasi tambahan berbasis *linear layer* dan *softmax* untuk menghasilkan probabilitas kelas. Melalui pendekatan *fine-tuning* pada lapisan klasifikasinya, IndoBERT sangat efektif digunakan untuk tugas NLP data bahasa Indonesia [12].

2.12 IndoBERTweet

Pada tahun 2021, tim Koto et al. mengembangkan IndoBERTweet sebagai pengembangan dari IndoBERT. IndoBERTweet merupakan model pre-trained skala besar pertama untuk Twitter berbahasa Indonesia yang dilengkapi dengan kosakata khusus media sosial X (sebelumnya Twitter) [11]. Berbeda dengan IndoBERT yang dilatih menggunakan korpus bahasa Indonesia umum, IndoBERTweet dilatih menggunakan kumpulan tweet berbahasa Indonesia dari empat topik, yaitu ekonomi, kesehatan, pemerintahan, dan edukasi [11]. Model ini menerapkan metode *vocabulary adaptation* untuk menangani perbedaan kosakata

antara bahasa formal dan bahasa di media sosial, sehingga mampu memahami bahasa tidak baku, istilah slang, singkatan, serta pola penulisan yang umum ditemukan pada media sosial X. Selain itu, IndoBERTweet digunakan dengan pendekatan transfer learning, yaitu memanfaatkan model yang telah melalui proses pre-training kemudian disesuaikan dengan tugas tertentu melalui proses fine-tuning [1].

Dibandingkan dengan IndoBERT, IndoBERTweet lebih sesuai digunakan dalam penelitian ini karena dataset yang digunakan berupa tweet mengenai layanan TransJakarta. Data pada media sosial X memiliki karakteristik yang berbeda dengan teks formal, seperti penggunaan bahasa tidak baku, singkatan, *slang*, *hashtag*, dan variasi penulisan lainnya. Sementara itu, IndoBERT dilatih menggunakan korpus bahasa Indonesia umum sehingga belum secara khusus mempelajari karakteristik bahasa di media sosial [25]. Sebaliknya, IndoBERTweet telah melalui proses pre-training pada kumpulan tweet berbahasa Indonesia, sehingga mampu menghasilkan representasi kontekstual yang lebih sesuai untuk data media sosial. Maka demikian, IndoBERTweet dipilih dalam penelitian ini untuk meningkatkan kinerja klasifikasi emosi pada data *tweet* mengenai layanan TransJakarta. Arsitektur dan alur kerja IndoBERTweet ditunjukkan pada Gambar 2.5.



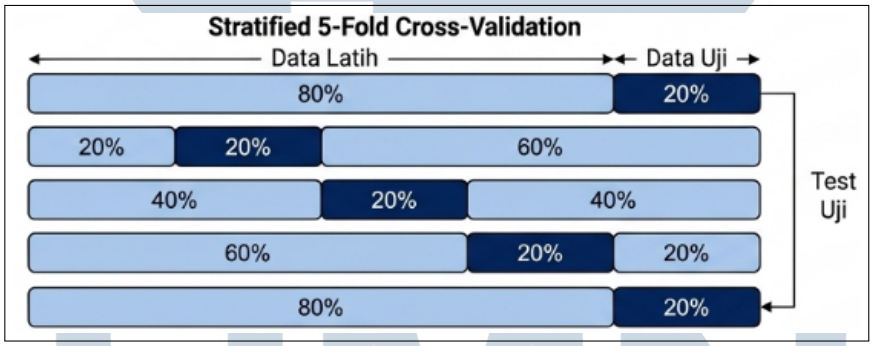
Gambar 2.5. Arsitektur dan alur kerja IndoBERTweet

2.13 *Stratified K-Fold Cross-Validation*

Cross-Validation merupakan teknik validasi model yang dilakukan dengan membagi *dataset* ke dalam beberapa *fold* untuk mengevaluasi kemampuan generalisasi model. Pada metode ini, setiap *fold* secara bergantian digunakan

sebagai data validasi, sedangkan *fold* lainnya digunakan sebagai data latih. Proses tersebut memungkinkan seluruh data memperoleh kesempatan yang sama untuk digunakan dalam tahap validasi sehingga evaluasi model tidak bergantung pada satu skenario pembagian data saja sehingga performa model dapat diukur secara lebih akurat pada data yang belum pernah dilihat sebelumnya, serta membantu mengurangi risiko *overfitting* [27].

Salah satu teknik validasi yang digunakan adalah *Stratified K-Fold Cross-Validation*. Metode ini membagi data menjadi beberapa *fold* dengan tetap menjaga proporsi setiap kelas pada masing-masing bagian agar sesuai dengan distribusi data awal [27]. Pendekatan tersebut sangat bermanfaat untuk menangani dataset yang memiliki ketidakseimbangan kelas, termasuk pada kasus klasifikasi emosi [28]. Selain itu, metode ini mampu menghasilkan evaluasi yang lebih konsisten karena setiap *fold* merepresentasikan distribusi label yang serupa dengan data keseluruhan, sehingga potensi bias akibat pembagian data dapat diminimalkan [29].



Gambar 2.6. Ilustrasi *Stratified K-Fold Cross-Validation*

Gambar 2.6 menunjukkan penerapan *Stratified K-Fold Cross-Validation* dengan lima *fold*. *Dataset* dibagi menjadi lima bagian dengan proporsi kelas yang tetap terjaga pada setiap *fold*. Pada setiap iterasi, satu *fold* berperan sebagai data validasi dan *fold* lainnya sebagai data pelatihan. Proses ini berlangsung hingga seluruh *fold* digunakan sebagai data validasi, sehingga evaluasi model dapat dilakukan secara lebih menyeluruh dengan tetap mempertahankan distribusi kelas pada data asli.

2.14 *Confusion matrix* dan Metrik Evaluasi

Confusion matrix merupakan tabel evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya [9]. Melalui tabel

ini, dapat diketahui jumlah prediksi yang benar maupun yang salah sehingga kinerja model klasifikasi dapat dianalisis secara lebih rinci. *Confusion matrix* terdiri atas empat komponen utama, yaitu:

1. *True Positive* (TP), yaitu jumlah data yang berhasil diklasifikasikan dengan benar ke dalam kelas yang seharusnya.
2. *True Negative* (TN), yaitu jumlah data yang berhasil dikenali dengan benar sebagai bukan anggota kelas yang diamati.
3. *False Positive* (FP), yaitu jumlah data yang sebenarnya tidak termasuk ke dalam suatu kelas, tetapi diprediksi sebagai anggota kelas tersebut.
4. *False Negative* (FN), yaitu jumlah data yang termasuk ke dalam suatu kelas, tetapi diprediksi sebagai kelas yang berbeda.

Nilai TP, TN, FP, dan FN kemudian digunakan untuk menghitung berbagai metrik evaluasi, seperti *accuracy*, *precision*, *recall*, dan *F1-score* [12, 9]. Metrik tersebut digunakan untuk menilai kemampuan model dalam menghasilkan prediksi yang sesuai dengan label sebenarnya. Perhitungan masing-masing metrik dijelaskan pada bagian berikut:

1. ***Accuracy***

   Accuracy merupakan metrik yang menunjukkan tingkat ketepatan model dengan membandingkan jumlah prediksi *True Positive* (TP) dan *True Negative* (TN), terhadap total seluruh data yang dievaluasi.

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

2. ***Precision***

   Precision adalah rasio antara jumlah data yang diprediksi positif secara benar (*True Positive*) terhadap seluruh data yang diklasifikasikan sebagai kelas positif.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

3. ***Recall***

   Recall merupakan metrik yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi data yang benar-benar termasuk ke dalam suatu kelas.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.3}$$

4. ***F1-Score***

   F1-Score merupakan metrik evaluasi yang mengukur keseimbangan antara nilai *Precision* dan *Recall* dalam menilai kinerja model.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

Di antara metrik evaluasi yang digunakan, Penelitian ini menggunakan *macro F1-score* sebagai metrik utama untuk menentukan performa model. Metrik tersebut dihitung dari rata-rata *F1-score* seluruh kelas emosi sehingga setiap kelas memberikan kontribusi yang setara terhadap hasil evaluasi. Dengan demikian, penilaian model menjadi lebih representatif pada *dataset* yang memiliki ketidakseimbangan distribusi kelas [1].

2.15 *Explainable Artificial Intelligence* (XAI)

Explainable Artificial Intelligence (XAI) merupakan pendekatan yang bertujuan untuk membantu manusia memahami dan menjelaskan keputusan yang dihasilkan oleh model [13]. XAI berperan dalam meningkatkan transparansi model dengan menjelaskan faktor-faktor yang memengaruhi hasil prediksi, terutama pada model yang bersifat *black-box* dan sulit diinterpretasikan secara langsung [13]. Sebagian besar penelitian klasifikasi teks berfokus pada peningkatan performa model, sementara aspek interpretabilitas masih jarang dibahas. Penerapan XAI dalam penelitian ini bertujuan untuk mendukung *trustworthy artificial intelligence*, yaitu model yang tidak hanya memiliki tingkat akurasi yang tinggi, tetapi juga transparan, interpretatif, dan dapat dipercaya oleh manusia [13].

2.16 *SHapley Additive exPlanations* (SHAP)

SHAP merupakan metode XAI yang digunakan untuk menginterpretasikan hasil prediksi model dengan menunjukkan besarnya kontribusi setiap fitur terhadap keputusan yang dihasilkan. Metode ini diperkenalkan oleh Lundberg dan Lee (2017) berdasarkan konsep *Shapley Value* dari teori permainan kooperatif yang digunakan menghitung kontribusi setiap fitur secara adil terhadap keluaran model [30]. SHAP dikembangkan untuk mengatasi keterbatasan interpretabilitas pada model pembelajaran mesin yang kompleks, seperti *ensemble learning* dan *deep learning*, yang umumnya bersifat *black-box* sehingga proses pengambilan keputusan model sulit dipahami secara langsung. SHAP digunakan untuk mengukur seberapa besar kontribusi masing-masing token terhadap probabilitas kelas yang diprediksi sehingga dapat diketahui token-token yang paling memengaruhi keputusan model.

Kontribusi setiap token dihitung menggunakan konsep *Shapley Value*, yaitu rata-rata kontribusi marginal suatu fitur terhadap keluaran model pada seluruh kemungkinan kombinasi fitur lainnya. Pendekatan ini memungkinkan kontribusi setiap token dihitung secara adil dan konsisten terhadap hasil prediksi model [30]. Secara matematis, nilai SHAP untuk fitur ke-i dinyatakan sebagai berikut:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{2.5}$$

Keterangan:

- ϕ_i : nilai kontribusi SHAP dari fitur atau token ke-i.
- F : himpunan seluruh fitur atau token pada input.
- S : subset fitur dari F yang tidak memuat fitur ke-i.
- $|F|$: jumlah seluruh fitur pada input.
- $|S|$: jumlah fitur dalam subset S.
- x : representasi input.
- $f_S(x_S)$: keluaran model berdasarkan subset fitur S.
- $f_{S \cup \{i\}}(x_{S \cup \{i\}})$: keluaran model ketika fitur ke-i ditambahkan ke subset S.

Menurut Lundberg dan Lee [30], SHAP memiliki tiga karakteristik utama yang menjamin kualitas interpretasi yang dihasilkan, yaitu *local accuracy*, *missingness*, dan *consistency*. Karakteristik *local accuracy* memastikan bahwa jumlah kontribusi seluruh fitur sama dengan nilai prediksi yang dihasilkan model, sehingga penjelasan model dapat merepresentasikan keluaran model asli secara akurat. Karakteristik *missingness* menjamin bahwa fitur yang tidak berkontribusi terhadap prediksi akan memperoleh nilai atribusi nol. Karakteristik *consistency* memastikan bahwa ketika kontribusi suatu fitur terhadap keluaran model meningkat, nilai atribusi yang diberikan kepada fitur tersebut tidak akan menurun. Ketiga karakteristik tersebut menjadikan SHAP sebagai metode interpretabilitas yang memiliki landasan teoritis kuat dan konsisten dalam menjelaskan keputusan model pembelajaran mesin.

Dalam klasifikasi teks, nilai SHAP digunakan untuk mengidentifikasi token yang meningkatkan maupun menurunkan probabilitas suatu kelas prediksi tertentu. Nilai SHAP positif menunjukkan bahwa token tersebut mendukung prediksi terhadap suatu kelas, sedangkan nilai SHAP negatif menunjukkan bahwa token tersebut mengurangi probabilitas kelas tersebut. Melalui nilai SHAP, dapat diketahui token-token yang paling berkontribusi terhadap prediksi kelas emosi beserta arah pengaruhnya terhadap hasil klasifikasi. Pada penelitian ini, SHAP digunakan untuk menginterpretasikan hasil prediksi model IndoBERTweet sehingga faktor-faktor yang memengaruhi keputusan model dalam mengklasifikasikan emosi dapat dijelaskan secara lebih transparan, konsisten, dan interpretatif [13].

2.17 Penelitian Terdahulu

Tabel 2.1. Ringkasan Penelitian Terkait

Penulis	**Judul Penelitian**	**Metode**	**Hasil**	**Kontribusi**
Madyatmadja, E. D. dkk, 2025	*Sentiment Analysis of Transjakarta User Feedback Using Machine Learning Models*	*Naive Bayes*, *Support Vector Machine*, *Decision Tree*	*Support Vector Machine accuracy* 93%, *macro F1-score* 86%	Perbandingan algoritma *machine learning* untuk analisis sentimen TransJakarta
A. Zaki, L. Muflikhah, dan T. N. Fatyanosa, 2025	Analisis Sentimen Layanan PT Kereta Api Indonesia pada Twitter Menggunakan *Fine-Tuned* IndoBERTweet	*Fine-Tuned* IndoBERT Tweet	*Accuracy* 83,33%, *macro F1-score* 77,02%.	Penerapan IndoBERTweet pada analisis sentimen Twitter
Raihan, H. A. dkk, 2025	Model Klasifikasi Emosi Berbasis Teks dengan Algoritma Decision Tree dan Support Vector Machine	*Support Vector Machine*, *Decision Tree*, TF-IDF, *Stratified K-Fold Cross Validation*	*Support Vector Machine accuracy* 89%, *macro F1-score* 86%	Perbandingan *Support Vector Machine* dan *Decision Tree* untuk klasifikasi emosi

Lanjut pada halaman berikutnya

Author, Tahun	Judul Penelitian	Metode	Hasil	Kontribusi
Yefta Tolla & Kusrini, 2025	Deteksi Stres dan Depresi dari Media Sosial dengan Machine Learning	TF-IDF, *Support Vector Machine*, SHAP	*Accuracy* 96,44%, *macro F1-score* 96%, mean CV 95,04%.	Penerapan SHAP untuk interpretabilitas model.
C. Saputra, A. S. Saragih, dan D. Ronaldo, 2025	Prediksi Emosi dalam Teks Bahasa Indonesia Menggunakan Model IndoBERT	*Fine-Tuning* IndoBERT	*Accuracy* 73%	Penerapan IndoBERT untuk klasifikasi emosi teks

Berdasarkan penelitian terdahulu pada Tabel 2.1, dapat diketahui bahwa berbagai metode telah digunakan untuk analisis sentimen dan klasifikasi emosi pada data teks, mulai dari algoritma *machine learning* tradisional seperti *Naive Bayes*, *Support Vector Machine*, dan *Decision Tree* hingga model berbasis *transformer* seperti IndoBERT dan IndoBERTweet. Hasil penelitian menunjukkan bahwa model berbasis *transformer* umumnya memberikan performa yang lebih baik karena mampu memahami konteks kalimat secara lebih efektif, khususnya pada data media sosial. Selain itu, beberapa penelitian juga telah menerapkan SHAP sebagai metode *Explainable Artificial Intelligence* (XAI) untuk meningkatkan interpretabilitas hasil prediksi model.

Meskipun demikian, masih terdapat beberapa celah penelitian. Sebagian besar penelitian pada layanan transportasi publik masih berfokus pada analisis sentimen, sedangkan penelitian mengenai klasifikasi emosi masih relatif terbatas. Selain itu, penerapan IndoBERTweet yang dikombinasikan dengan SHAP untuk klasifikasi emosi pada data media sosial X terkait layanan TransJakarta belum banyak dilakukan. Oleh karena itu, penelitian ini menerapkan model IndoBERTweet untuk klasifikasi emosi pengguna media sosial X terhadap layanan TransJakarta serta memanfaatkan SHAP untuk memberikan interpretasi terhadap hasil prediksi model.