BAB 2
LANDASAN TEORI

Penelitian ini menggunakan algoritma IndoBERTweet untuk melakukan analisis sentimen terhadap Program Makan Bergizi Gratis (MBG). Bab ini menguraikan teori dan penelitian terdahulu yang menjadi landasan penelitian, meliputi konsep analisis sentimen, karakteristik Platform X, teknik pengumpulan data, praproses teks, metode *heuristic labeling* dan *pseudo labeling*, serta model-model *Natural Language Processing* (NLP) yang relevan.

2.1 Analisis Sentimen

Analisis sentimen adalah cabang studi yang mempelajari opini, sentimen, evaluasi, sikap, dan emosi masyarakat terhadap berbagai entitas seperti produk, layanan, organisasi, individu, isu, atau peristiwa [8]. Pada konteks media sosial, analisis sentimen bertujuan mengklasifikasikan teks menjadi kategori seperti positif, negatif, atau netral untuk memahami persepsi publik secara kuantitatif dan kualitatif.

2.2 Platform X (sebelumnya Twitter)

Platform X (sebelumnya dikenal sebagai Twitter) adalah platform media sosial berbasis *microblogging* yang menonjolkan pesan singkat, interaksi *real-time*, dan mekanisme distribusi informasi melalui *retweet*, *mention*, dan *hashtag* [9, 10]. Sifatnya yang terbuka dan cepat membuat Platform X sering digunakan sebagai saluran publik untuk menyampaikan dukungan, kritik, serta mobilisasi opini terhadap kebijakan publik. Oleh karena itu, Platform X sering menjadi sumber data yang kaya untuk studi opini publik dan analisis sentimen.

2.3 *Tweet Harvest*

Tweet Harvest adalah alat berbasis *command-line* yang digunakan untuk melakukan *crawling* data dari Platform X. Alat ini memungkinkan pengguna melakukan pencarian berdasarkan kata kunci dan menyimpan hasilnya dalam format CSV untuk analisis lebih lanjut [11]. Dalam penelitian ini, *Tweet Harvest*

digunakan untuk mengumpulkan *tweet* yang mengandung kata kunci terkait Program MBG sebagai bahan *input* untuk *pipeline* analisis sentimen.

2.4 *Text Processing*

Text processing merupakan tahap krusial dalam analisis sentimen, terutama ketika sumber data berasal dari media sosial yang bersifat tidak terstruktur. Pada penelitian ini, *dataset* awal terdiri dari sekitar 17.480 *tweet* berbahasa Indonesia, yang umumnya mengandung emotikon, singkatan, tanda baca berlebih, dan bahasa tidak formal [12].

Tahapan praproses yang diterapkan meliputi:

- ***Case folding***: Mengubah seluruh huruf menjadi huruf kecil untuk menyamakan bentuk kata.
- **Penghapusan tanda baca dan simbol**: Menghapus tanda baca, URL, angka (kecuali yang kontekstual), serta simbol seperti `@` dan `#`.
- **Penghapusan *mention* dan *hashtag***: *Mention* (`@username`) dan *hashtag* (`#topik`) dihapus atau dinormalisasi bila tidak menambah konteks emosional.
- **Tokenisasi**: Memecah kalimat menjadi *token* menggunakan *tokenizer* IndoBERT dari HuggingFace yang sesuai dengan struktur bahasa Indonesia.
- ***Stopword removal***: Menghapus kata-kata umum yang tidak memberikan informasi sentimen seperti “yang”, “dan”, “di”, “ke”, “itu”.
- **Normalisasi kata tidak baku**: Mengubah singkatan dan *slang* misalnya (“gk”, “bgt”, “tdk”, “ga”) menjadi bentuk baku (“tidak”, “banget”) untuk meningkatkan kualitas representasi.
- **Mengatasi kata pengulangan**: Mereduksi karakter berulang yang berlebihan misalnya (“bangeeeet” menjadi “banget”) guna menstandardisasi kata dan mengurangi variansi kosakata yang tidak perlu.

2.5 *Heuristic Labeling*

Heuristic labeling adalah pendekatan pelabelan programatik yang menggunakan sekumpulan aturan atau *labeling functions* berbasis kata kunci, pola reguler, emotikon, atau sinyal metadata (misalnya jumlah *retweet*, keberadaan URL) untuk memberikan label awal pada data tanpa anotasi manual. Metode ini berguna untuk menghasilkan *dataset* berlabel secara cepat dan ekonomis, serta sering dipakai sebagai tahap awal sebelum pelatihan model atau penerapan teknik semi-terawasi lainnya [13]. Dalam penelitian ini, *heuristic labeling* dapat digunakan sebagai sumber label tambahan atau untuk memfilter data sebelum proses anotasi manual dan *pseudo-labeling*, namun pelabelan akhir pada studi ini mengandalkan kombinasi anotasi manual (*seed*) dan *pseudo-labeling* otomatis untuk menjaga kualitas label.

2.6 *Pseudo Labeling*

Pseudo labeling adalah teknik pembelajaran semi-terawasi yang diperkenalkan oleh Lee (2013). Metode ini memberikan label sementara (*pseudo-label*) pada data tanpa label berdasarkan prediksi model, lalu melatih model menggunakan gabungan data berlabel dan data ber-*pseudo-label*. Pendekatan ini efektif untuk memanfaatkan data tak berlabel dan dapat meningkatkan generalisasi model bila *pseudo-label* yang dihasilkan memiliki tingkat kepercayaan yang tinggi [14].

Dalam implementasi penelitian ini, langkah umum *pseudo-labeling* adalah:

1. Melakukan anotasi manual pada 750 sampel sebagai *seed labeled set*.
2. Melatih model awal (*teacher*) pada *seed* tersebut.
3. Memprediksi label pada data tak berlabel dan menerima *pseudo-label* dengan probabilitas tertinggi di atas ambang τ.
4. Menggabungkan *pseudo-label* terpilih dengan data manual untuk melatih model akhir (*student*).

2.7 *Class Weight*

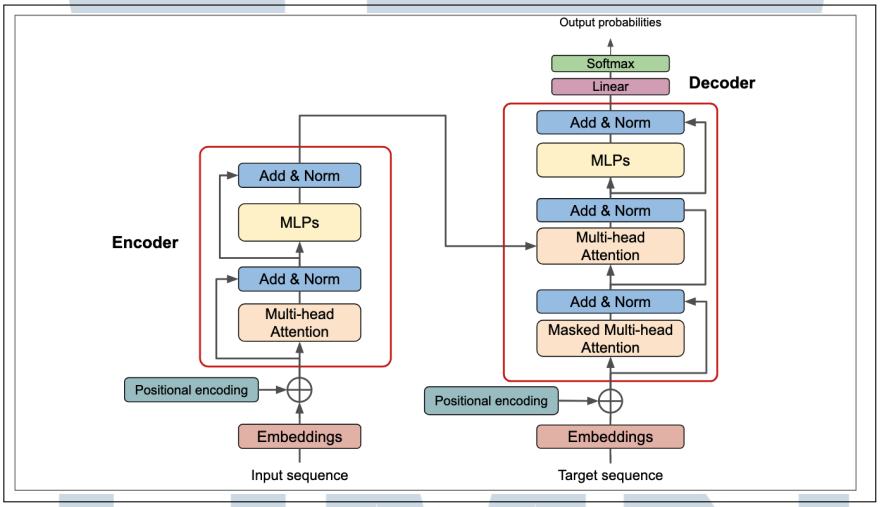
Class Weight adalah teknik yang digunakan dalam mengatasi ketidakseimbangan kelas pada *dataset* klasifikasi. Dalam konteks analisis sentimen, ketidakseimbangan kelas sering terjadi ketika distribusi sampel antar kelas sentimen (positif, negatif, atau netral) tidak merata, yang mengakibatkan model cenderung bias terhadap kelas yang memiliki jumlah sampel lebih banyak [15]. Prinsip dasar *class weight* adalah meningkatkan bobot kelas minoritas dan mengurangi bobot kelas mayoritas selama proses pembelajaran. Dengan cara ini, model dilatih dengan menggunakan penalti yang lebih tinggi ketika melakukan kesalahan prediksi pada kelas minoritas dibandingkan pada kelas mayoritas. Hal ini mendorong model untuk lebih fokus mempelajari pola dari kelas yang kurang terwakili. Berbeda dengan teknik seperti *oversampling* dan *undersampling*, *class weight* tidak mengubah komposisi *dataset* asli melainkan mengubah cara model memproses kesalahan prediksi selama *training* [16].

2.8 *Natural Language Processing*

Natural Language Processing (NLP) merupakan cabang multidisiplin dari kecerdasan buatan yang menjembatani kesenjangan antara komunikasi manusia dan pemahaman komputer melalui penggabungan pendekatan linguistik komputasi, model statistik, dan pembelajaran mesin. Teknologi ini memungkinkan sistem komputer untuk tidak hanya memproses teks secara harfiah, tetapi juga memahami struktur sintaksis, makna semantik, serta nuansa emosional yang terkandung di dalamnya. Dalam konteks spesifik analisis sentimen, NLP memegang peran vital dalam mentransformasi data teks tidak terstruktur menjadi format yang dapat dikalkulasi secara matematis. Hal ini mencakup penerapan teknik representasi teks (*word embedding*) untuk menangkap hubungan antar kata, pemilihan arsitektur model yang tepat untuk mempelajari pola bahasa yang kompleks, serta penggunaan metode evaluasi yang ketat, yang semuanya diperlukan untuk membangun sistem klasifikasi sentimen yang akurat dan andal [17].

2.9 Model *Transformer*

Transformer adalah arsitektur jaringan neural yang menggunakan mekanisme *self-attention* untuk memproses urutan secara paralel, sehingga mampu menangkap dependensi jarak jauh antar *token* dengan efisien. Arsitektur ini menjadi dasar bagi model-model besar seperti BERT, GPT, dan T5, yang merevolusi banyak tugas NLP [18]. Struktur *Transformer* terdiri dari lapisan *encoder* dan *decoder* yang masing-masing mengandung blok *multi-head self-attention* dan *feed-forward network*. *Positional encoding* ditambahkan untuk mempertahankan informasi urutan *token* [19]. Ilustrasi arsitektur ini dapat dilihat pada Gambar 2.1.



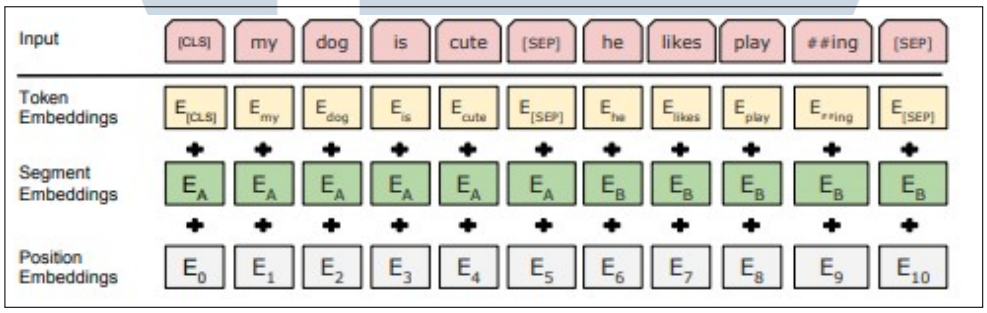
Gambar 2.1. Arsitektur model *Transformer*. Sumber: [19]

2.10 Model BERT

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model berbasis *Transformer* yang melakukan *pre-training* secara *bidirectional* menggunakan *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Setelah *pre-training*, BERT dapat di-*fine-tune* untuk tugas spesifik seperti klasifikasi teks dan analisis sentimen dengan menambahkan lapisan *output* khusus [20]. Cara kerja BERT dapat dibagi menjadi tiga tahapan utama, yaitu representasi *input*, arsitektur model, dan mekanisme *output*. BERT menerima *input* berupa urutan *token* yang dapat merepresentasikan satu kalimat atau sepasang kalimat (misalnya, [*Question, Answer*]). Representasi *input* pada BERT merupakan

penjumlahan dari tiga *embedding* utama, seperti yang diilustrasikan pada Gambar 2.2:

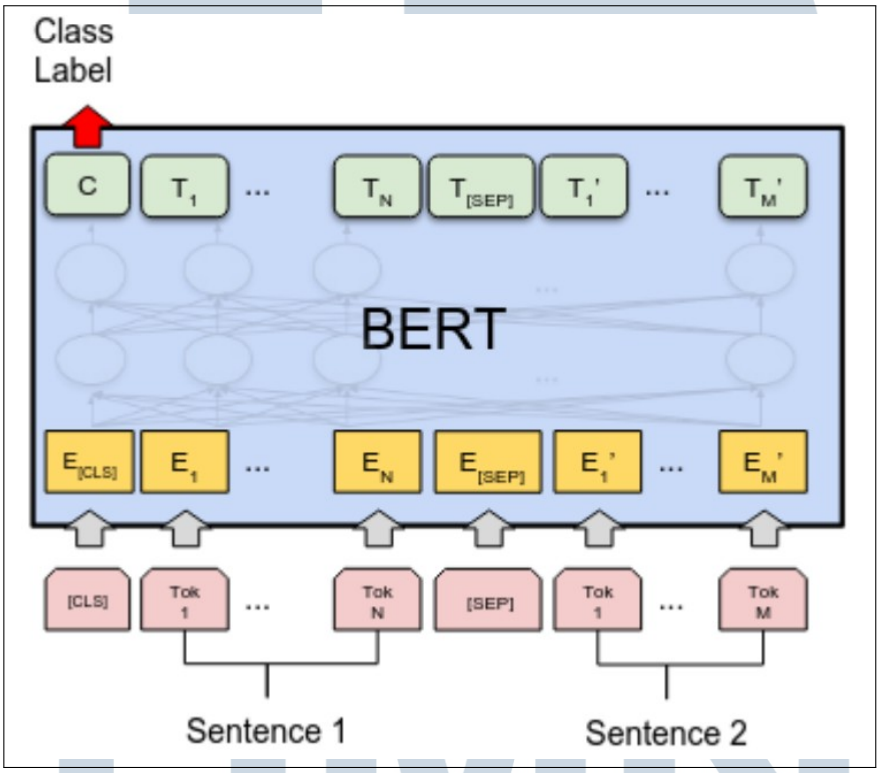
1. ***Token Embeddings***: Menggunakan representasi kata *WordPiece*. Token khusus `[CLS]` ditambahkan di awal setiap urutan *input* untuk tugas klasifikasi, dan token `[SEP]` digunakan untuk memisahkan dua kalimat.
2. ***Segment Embeddings***: Menandakan apakah sebuah token termasuk dalam kalimat pertama (A) atau kalimat kedua (B).
3. ***Position Embeddings***: Menandakan posisi atau urutan token dalam kalimat, karena arsitektur *Transformer* tidak memproses data secara sekuensial (urutan waktu) seperti RNN.



Gambar 2.2. Visualisasi representasi *input* pada BERT yang terdiri dari *Token*, *Segment*, dan *Position Embeddings*. Sumber: [20]

Arsitektur dasar BERT adalah *multi-layer bidirectional Transformer encoder*. Berbeda dengan model sebelumnya yang membaca teks dari kiri-ke-kanan atau kanan-ke-kiri, *Transformer encoder* pada BERT membaca seluruh urutan kata sekaligus (*bidirectional*). Untuk melatih pemahaman bahasa yang mendalam, BERT menggunakan dua strategi *pre-training* sebagaimana ditunjukkan pada Gambar 2.3:

- ***Masked Language Modeling* (MLM)**: Sekitar 15% token dalam *input* ditutupi (*masked*) secara acak, dan model dilatih untuk memprediksi token asli berdasarkan konteks di sekitarnya.
- ***Next Sentence Prediction* (NSP)**: Model menerima pasangan kalimat (A dan B) dan dilatih untuk memprediksi apakah kalimat B adalah kelanjutan logis dari kalimat A.



Gambar 2.3. Ilustrasi proses *pre-training* BERT menggunakan *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Sumber: [20]

2.10.1 Mekanisme *Output* dan *Fine-tuning*

Hasil keluaran dari BERT adalah vektor representasi untuk setiap token *input*. Untuk tugas klasifikasi seperti analisis sentimen, representasi dari token `[CLS]` pada lapisan terakhir digunakan sebagai *input* untuk lapisan klasifikasi tambahan (misalnya, *Softmax layer*). Pada tahap *fine-tuning*, seluruh parameter model yang telah dilatih (*pre-trained*) disesuaikan kembali menggunakan *dataset* spesifik tugas yang lebih kecil.

2.11 IndoBERT

IndoBERT adalah adaptasi BERT yang dilatih khusus pada korpus bahasa Indonesia untuk menangkap kekhasan linguistik lokal, termasuk variasi gaya bahasa formal dan informal. *Pre-training* IndoBERT menggunakan sumber seperti Wikipedia Indonesia, korpus berita, dan konten media sosial sehingga model ini lebih sensitif terhadap nuansa bahasa Indonesia dibandingkan model multibahasa generik [21, 22]. Beberapa studi melaporkan performa tinggi IndoBERT pada tugas klasifikasi teks berbahasa Indonesia, termasuk klasifikasi emosi dan berita. IndoBERT sering dijadikan titik awal adaptasi domain untuk model yang ditujukan pada data media sosial [6, 7].

2.12 IndoBERTweet

IndoBERTweet adalah model *pretrained* yang diadaptasi khusus untuk korpus Twitter berbahasa Indonesia. Model ini dikembangkan dengan strategi *domain-adaptive pretraining* pada korpus *tweet* besar (Desember 2019–Desember 2020) dan penambahan kosakata *domain-specific* berbasis *WordPiece* (*vocab* $\approx$ 32K) [5]. Tantangan utama adaptasi domain adalah tokenisasi kata baru yang tidak ada pada kosakata model dasar; untuk itu IndoBERTweet mengeksplorasi beberapa strategi inisialisasi *embedding token* baru, termasuk inisialisasi acak, proyeksi *fastText*, dan inisialisasi dengan rata-rata *embedding subword* dari model dasar. Penelitian menunjukkan bahwa inisialisasi dengan rata-rata *embedding subword* memberikan keseimbangan terbaik antara efisiensi pelatihan dan performa *downstream*, sehingga memungkinkan adaptasi domain yang jauh lebih cepat (hingga $\sim 5\times$ lebih cepat dibanding *pretraining* dari awal) dengan performa yang kompetitif pada tugas-tugas Twitter seperti analisis sentimen, klasifikasi emosi, deteksi ujaran kebencian, dan NER [5].

2.13 Evaluasi

Evaluasi kinerja model dilakukan menggunakan metrik standar klasifikasi: *Precision*, *Recall*, *F1-Score*, dan *Accuracy*. Metrik tersebut dihitung dari nilai pada *confusion matrix* yang terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Precision mengukur tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem, seperti yang ditunjukkan pada Persamaan 2.1.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi, sebagaimana didefinisikan dalam Persamaan 2.2.

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

F1-Score merupakan rata-rata harmonis dari *precision* dan *recall*, yang dihitung menggunakan Persamaan 2.3.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.3}$$

Sedangkan *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan data dengan benar secara keseluruhan, seperti terlihat pada Persamaan 2.4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$