BAB II

KAJIAN LITERATUR

2.1 Program Makan Bergizi Gratis

Program Makan Bergizi Gratis (MBG) merupakan salah satu program pemerintahan Presiden dan Wakil Presiden Prabowo-Gibran, yang mulai dilaksanakan pada 6 Januari 2025 [19]. Program ini dirancang sebagai upaya untuk mengatasi masalah *stunting* dan gizi buruk, yang masih menjadi tantangan besar di Indonesia [20], [21]. Berdasarkan data dari Kementerian Kesehatan RI (2023), angka *stunting* nasional mencapai 21,6% dan pada tahun 2025 mengalami penurunan hingga 19,8% [22], [23]. Program ini dialokasikan untuk anak-anak sekolah dengan memberikan menu makanan yang seimbang seperti karbohidrat, protein, vitamin dan mineral, sehingga diharapkan dapat memperoleh asupan gizi yang cukup [21].

Realitas di lapangan implementasi program ini mengalami berbagai kendala seperti keterbatasan anggaran, ketidaksiapan infrastruktur distribusi pangan, kualitas pengawasan gizi yang belum optimal, serta koordinasi yang belum maksimal antara pemerintah pusat, daerah, sekolah, dan penyedia layanan. Kendala tersebut menjadi hambatan signifikan untuk mencapai tujuan utama program dengan memastikan seluruh anak-anak sekolah mendapatkan asupan gizi yang cukup dan berkualitas [24].

Selain itu, hal yang perlu diperhatikan adalah kualitas rasa, kebersihan makanan, kecukupan porsi, variasi menu, ketepatan waktu penyajian, serta tata kelola distribusi dapat memengaruhi penerimaan peserta didik. Tanggapan peserta didik terhadap kualitas makanan ini sangat penting karena mereka merupakan penerima manfaat langsung. Ketika makanan yang diberikan dianggap enak, bersih, dan memadai, peserta didik akan merasa dihargai dan termotivasi untuk mengikuti program [25].

2.2 Kualitas Makanan

Kualitas makanan merupakan standar yang harus dipenuhi, mencakup keamanan pangan, nilai gizi, mutu bahan baku, dan cita rasa. Untuk menjamin mutu tersebut, bahan baku perlu memenuhi spesifikasi kualitas yang ditetapkan oleh pemasok. Penentuan kualitas makanan juga memerlukan metode penilaian objektif serta evaluasi sensorik oleh masyarakat untuk mengetahui tingkat penerimaan produk. Penilaian kualitas makanan didasarkan pada kriteria seperti variasi

makanan, kesesuaian penggunaan bahan dalam menu, kesehatan makanan, cara penyajian, dan ukuran porsi [20]. Secara menyeluruh, kualitas makanan mencakup aspek warna, aroma, tekstur, dan tampilan yang dipengaruhi oleh penampilan, bentuk, suhu, tingkat kematangan, serta cita rasa. Dalam bidang *food and beverage*, dimensi kualitas tersebut tercermin melalui kesegaran, penyajian, tingkat kematangan yang tepat, serta variasi menu. Konsep kualitas makanan ini menjadi landasan dalam menganalisis persepsi pengguna terhadap konten makan bergizi gratis pada platform TikTok.

Kualitas makanan dapat diukur melalui beberapa indikator utama. Indikator tersebut meliputi kualitas rasa yang harus sesuai dengan preferensi konsumen, serta kuantitas yang berkaitan dengan jumlah makanan yang disajikan. Selain itu, variasi menu menjadi faktor penting karena menunjukkan keberagaman jenis makanan yang ditawarkan. Kualitas juga ditentukan oleh adanya cita rasa khas yang menjadi pembeda suatu produk dengan produk lainnya. Aspek higienitas berperan dalam menjamin keamanan dan kelayakan makanan untuk dikonsumsi. Di samping itu, inovasi dalam pengembangan menu baru turut memengaruhi persepsi terhadap kualitas makanan. Seluruh indikator ini digunakan untuk memberikan gambaran yang lebih terukur dalam menilai kualitas makanan.

Menjaga kualitas makanan penting untuk mencegah cemaran biologis, kimia, dan benda lain yang berbahaya bagi kesehatan manusia [21]. Kualitas makanan yang baik berkaitan langsung dengan pemenuhan gizi yang diterima oleh individu. Apabila kualitas maupun kuantitas porsi tidak diperhatikan, dampaknya dapat berupa kekurangan gizi. Kondisi ini, terutama pada anak-anak, dapat menyebabkan defisiensi protein, vitamin, dan mineral sehingga pertumbuhan fisik dan kognitif terganggu. Dampak lanjutan dari kondisi tersebut adalah menurunnya kemampuan belajar dan prestasi sekolah [22], [23]. Oleh karena itu, pemenuhan gizi seimbang berperan penting dalam menjaga kesehatan, meningkatkan daya tahan tubuh, serta mendukung kecerdasan. Status gizi yang baik berkontribusi pada kualitas hidup dan produktivitas, sedangkan ketidakseimbangan gizi dapat menimbulkan gangguan kesehatan dan menghambat perkembangan anak.

2.3 Media Sosial

Bagian ini membahas konsep dasar media sosial, karakteristiknya, serta berbagai platform yang berperan dalam proses pembentukan opini publik. Media sosial adalah sarana yang digunakan pengguna untuk menampilkan diri, berinteraksi, berkolaborasi, berbagi informasi, serta menjalin hubungan dengan pengguna lain. Di dalamnya, terdapat tiga makna utama dalam

bersosial, yaitu pengenalan (*cognition*), komunikasi (*communication*), dan kerja sama (*cooperation*) [26]. Tidak mengherankan jika kehadiran media sosial menjadi sangat populer. TikTok, Facebook, Twitter, YouTube, Instagram, hingga Path merupakan beberapa jenis platform media sosial yang banyak diminati oleh masyarakat luas [27]. Media sosial tidak dilakukan secara tatap muka, tetapi melalui perantara teknologi yang disebut *Computer Mediated Communication* (CMC), sehingga pengguna tetap dapat berinteraksi secara langsung (*real-time*). Media sosial memberikan kemudahan karena bisa digunakan kapan saja dan di mana saja, namun juga membuat pengguna cenderung menghabiskan lebih banyak waktu setiap harinya [28].

2.3.1 Definisi dan Konsep Dasar

Media sosial telah mengubah lanskap komunikasi modern secara fundamental. Menurut Qadri, media sosial memberikan dampak positif dalam proses interaksi sosial, politik, maupun ekonomi, serta memberikan kemudahan dalam mengakses informasi dan berkomunikasi jarak jauh. Andreas Kaplan dan Michael Haenlein mendefinisikan media sosial sebagai sekelompok aplikasi berbasis internet yang dibangun di atas dasar ideologi dan teknologi Web 2.0, yang memungkinkan penciptaan dan pertukaran *User-Generated-Content* (UGC). Definisi ini menegaskan bahwa media sosial tidak hanya berfungsi sebagai sarana komunikasi, tetapi juga sebagai wadah partisipasi aktif pengguna dalam menghasilkan dan menyebarkan informasi [29]. Dalam perkembangannya, media sosial telah menjadi "ruang publik baru" yang memungkinkan interaksi, distribusi informasi, serta pembentukan opini publik terjadi secara cepat dan luas [30]. Fenomena ini juga didukung oleh temuan Suratman et al. (2025), yang mencatat bahwa tingkat kepercayaan publik terhadap berita di media sosial kini bersaing dengan media konvensional, meskipun validitas informasinya sering kali menjadi tantangan tersendiri [31].

2.3.2 Karakteristik Media Sosial

Karakteristik media sosial membedakannya dari media massa tradisional. Media sosial memiliki ciri utama yaitu memfasilitasi komunikasi dua arah atau banyak arah (*many-to-many*) [29]. Selain itu, Karakteristik utama komunikasi digital di media sosial meliputi penyebaran informasi yang cepat (*velocity*) dan kemampuan untuk berbagi (*shareability*) yang luas, yang secara signifikan memengaruhi pandangan dan sikap masyarakat [32]. Lebih lanjut, media sosial memiliki karakteristik *interactivity* yang memungkinkan terjadinya dialog intensif secara *real-time* antara pembuat kebijakan dan masyarakat luas [33]. Selain itu, terdapat aspek *connectedness* yang

mampu menghubungkan berbagai kelompok kepentingan dalam satu ruang digital tanpa hambatan jarak fisik [34]. Secara teknis, platform modern kini sangat bergantung pada *algorithmic curation* yang menentukan relevansi konten berdasarkan perilaku pengguna, yang secara tidak langsung membentuk ruang gema (*echo chambers*) dalam opini publik [35]. Karakteristik *user-generated content* juga menjadi pembeda utama, di mana validitas informasi sangat bergantung pada partisipasi aktif pengguna dalam memverifikasi narasi yang beredar [36]. Terakhir, sifat *persistency* atau ketetapan data digital memungkinkan setiap rekam jejak opini masyarakat tersimpan secara permanen, sehingga menjadi aset penting dalam analisis data besar (*big data*) untuk keperluan riset kebijakan [37].

2.3.3 Peran Media Sosial dalam Pembentukan Opini Publik

Media sosial memiliki peran sentral dalam mengonstruksi opini publik, terutama terkait kebijakan pemerintah. Media sosial memainkan peran penting dalam membentuk opini masyarakat serta mempengaruhi sudut pandang individu terhadap isu-isu politik, meskipun juga menghadirkan tantangan bagi demokrasi seperti polarisasi [38]. Hal senada diungkapkan oleh Hidayat et al. (2025), yang menemukan bahwa platform digital telah menjadi arena utama untuk mobilisasi opini publik, di mana sering terjadi disonansi antara agenda pemerintah dengan wacana yang berkembang di media sosial [39]. Selain itu, pemanfaatan teknologi *Big Data* pada media sosial juga mendukung pemerintah dalam memantau isu serta merespons sentimen publik secara lebih efektif dalam situasi krisis [40]. Peran tersebut selanjutnya dapat diamati melalui berbagai platform media sosial populer yang digunakan masyarakat dalam menyampaikan dan membentuk opini publik [41]. Dalam dinamika komunikasi politik di Indonesia, media sosial telah bertransformasi menjadi arena utama bagi warga untuk menyuarakan ketidakpuasan secara kolektif, yang pada akhirnya menuntut pemerintah untuk lebih transparan dan responsif dalam setiap kebijakan yang diambil [42]. Selain itu, analisis wacana di platform digital memungkinkan identifikasi polarisasi opini yang terjadi di masyarakat, sehingga menjadi indikator penting dalam mengukur legitimasi program pemerintah di ruang publik [43].

2.3.4 TikTok

TikTok merupakan sebuah aplikasi yang memungkinkan penggunanya untuk membuat, menonton, serta mengeksplorasi berbagai konten video, seperti *lip-sync* dan meme [44]. TikTok pertama kali diluncurkan pada September 2016 di Tiongkok dengan nama Douyin dan dengan

cepat memperoleh popularitas luas di dalam negeri. Sebelum akhirnya ByteDance sebagai perusahaan pengembang meluncurkan TikTok ke pasar internasional pada tahun berikutnya [45], [46]. Menurut Data Reportal, pada awal tahun 2025 iklan TikTok dapat menjangkau 108 juta atau 53,5 persen pengguna berusia 18 tahun keatas di Indonesia, dan dapat menjangkau 50,7 persen pengguna internet *local* tanpa memandang usia. Dengan tingkat kepopuleran yang tinggi, khususnya di kalangan generasi muda, TikTok menjadi salah satu platform media sosial utama untuk mengonsumsi dan menyampaikan *respons* dan opini terhadap berbagai isu publik [47]. Saat mengunggah konten, pengguna dapat membagikan aktivitas, pandangan maupun tanggapan terhadap suatu topik tertentu. Konten ini kemudian ditampilkan kepada pengguna lain melalui fitur "untuk Anda" [46]. dan Pengguna juga dapat menambahkan tagar untuk memudahkan pencarian dan pengelompokan video [48]. Sebagai platform dengan tingkat respons pengguna yang tinggi, TikTok membentuk pola interaksi yang berbeda, dimana opini publik dipengaruhi oleh penyajian konten, Teknik penyampaian kreator, dan emosional pengguna. Hal ini menjadikan TikTok sebagai media yang berperan dalam membentuk dan mencerminkan opini masyarakat [47], [49].

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan salah satu bagian dari kecerdasan buatan yang berfokus pada pemrosesan bahasa manusia agar dapat dipahami oleh komputer [50]. Melalui NLP, sistem komputer dapat memahami, menafsirkan, dan menghasilkan bahasa yang menyerupai komunikasi manusia [51]. NLP juga mampu menangani berbagai permasalahan dalam bahasa alami, seperti kesalahan ejaan dan perbedaan gaya penulisan. Kemampuan tersebut membantu meningkatkan akurasi model dalam mengenali sentimen subjektif yang terkandung dalam suatu teks [52]. NLP memiliki dua komponen utama yaitu, *Natural Language Understandin*g (NLU) dan *Natural Language Generation* (NLG). NLU berperan dalam memahami makna dan konteks dari teks atau pertanyaan pengguna, sedangkan NLG berfungsi dalam menghasilkan teks atau respons yang sesuai dengan konteks yang dipahami [53].

Lingkup NLP sangat luas dan aplikasinya terus mengalami perkembangan seiring dengan kemajuan teknologi, khususnya dengan semakin melimpahnya data teks pada berbagai platform media sosial, situs web, dan email. Dalam bidang bisnis, NLP dimanfaatkan pada sistem *chatbot* untuk layanan pelanggan yang mampu berkomunikasi dengan konsumen menggunakan bahasa alami. Di dunia digital, NLP berperan penting dalam sistem pencarian informasi, di mana mesin pencari seperti Google memanfaatkan teknik ini untuk memahami kueri pengguna dan

menampilkan hasil yang relevan. Selain itu, NLP juga digunakan dalam sistem rekomendasi, dengan menganalisis teks yang dihasilkan pengguna, seperti ulasan atau komentar, untuk memberikan saran produk atau layanan yang lebih tepat sasaran. Di sisi lain, NLP juga digunakan dalam analisis opini publik berbasis teks yang membantu perusahaan atau organisasi memahami persepsi serta respons masyarakat terhadap suatu produk, layanan, maupun isu yang sedang berkembang [54].

Dalam pengembangan sistem NLP, data berbasis teks menjadi sumber utama yang diperlukan agar sistem dapat memahami, mengolah, dan menghasilkan bahasa manusia secara akurat [51]. NLP memungkinkan data teks yang tidak terstruktur, khususnya dari media sosial, diproses menjadi informasi kuantitatif yang dapat menggambarkan kecenderungan sentimen pengguna. Dalam analisis sentimen, teks diubah menjadi representasi numerik agar dapat diproses oleh model pembelajaran mesin atau *deep learning*. Tahapan dalam NLP meliputi pengumpulan data, pembersihan teks (*text cleaning*), tokenisasi, *stemming* atau *lemmatisasi*, serta penghapusan *stopwords* [55]. Setiap tahapan memiliki peran penting dalam meningkatkan kualitas data sebelum dilakukan proses klasifikasi. Apabila tahapan NLP tidak dilakukan secara optimal, maka hasil analisis sentimen yang diperoleh berpotensi kurang akurat serta sulit untuk dipahami secara tepat [56].

2.5 Analisis Sentimen

Analisis sentimen merupakan metode yang digunakan untuk mengetahui opini publik terhadap produk dan layanan publik [57]. Cara kerja analisis sentimen adalah mengklasifikasikan teks ke dalam beberapa kategori sentimen seperti positif, negatif, atau netral dengan memanfaatkan informasi berupa teks. Metode ini banyak diterapkan pada data tidak terstruktur seperti ulasan pengguna, percakapan media sosial, dan artikel berita, sehingga dapat dijadikan dasar pengambilan keputusan di berbagai bidang [58]. Banyaknya informasi menyebabkan pemantauan secara manual kurang efektif, terutama topik yang dibahas bersifat sensitif dan mendapatkan perhatian publik yang luas. Proses pembacaan data satu per satu berpotensi menghasilkan penilaian yang tidak konsisten serta menimbulkan bias, karena sangat bergantung pada sampel data yang teramati secara kebetulan. Oleh sebab itu, diperlukan pendekatan analisis otomatis agar pola opini masyarakat dapat diidentifikasi secara lebih sistematis dan terukur. Dalam konteks kebijakan sosial, kemampuan untuk memahami masukan publik secara cepat dapat mendukung proses evaluasi program serta meningkatkan kualitas pelaksanaan kebijakan. Namun demikian,

keakuratan hasil analisis tetap bergantung pada pengelolaan data dan tahapan pembersihan data yang tepat, agar tidak memunculkan *bias* baru dalam proses analisis [59]. Penerapan metode ini terbukti efektif dalam mengekstraksi opini publik terhadap suatu isu atau layanan, khususnya pada data berbahasa Indonesia yang bersumber dari media sosial [60].

Analisis sentimen tidak hanya terbatas pada proses klasifikasi teks, tetapi juga dapat dibedakan berdasarkan jenis analisisnya [61]. Pertama, analisis sentimen bertingkat digunakan untuk menilai tingkat sentimen, misalnya melalui skala bintang, di mana nilai tinggi menunjukkan sentimen positif dan nilai rendah menunjukkan sentimen negatif. Kedua, pengenalan emosi digunakan untuk mengidentifikasi emosi tertentu seperti kebahagiaan, kemarahan, kesedihan, dan frustrasi. Ketiga, analisis berbasis aspek berfokus pada identifikasi sentimen terhadap aspek atau karakteristik yang memiliki emosi positif, negatif, atau netral. Keempat, analisis sentimen multilingual digunakan untuk mengenali dan memproses berbagai bahasa dalam teks sesuai dengan konteks yang digunakan [61].

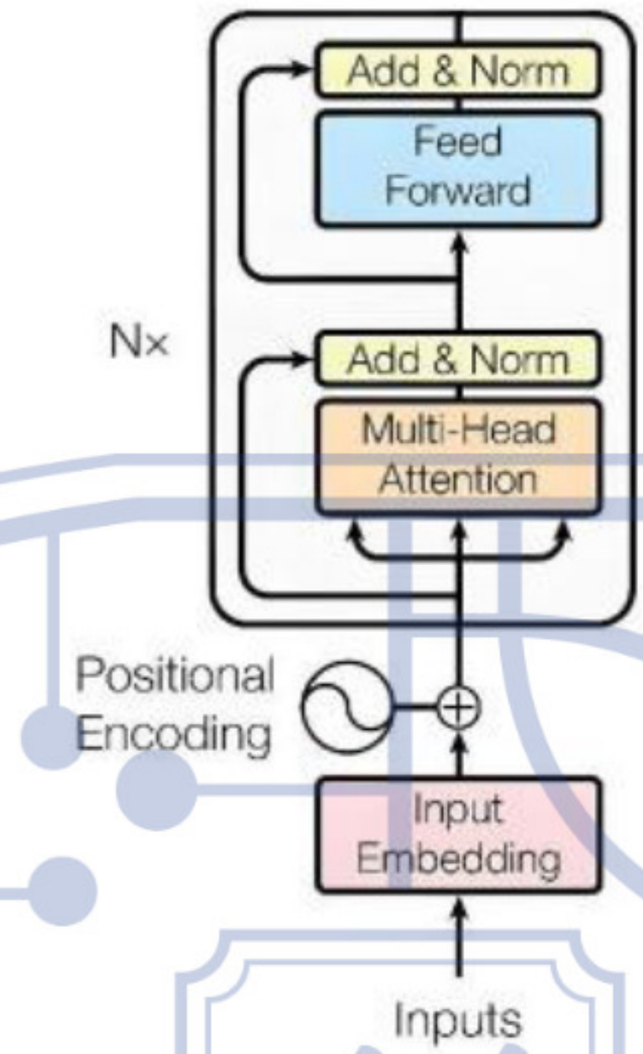
Ditinjau dari tingkat analisisnya, analisis sentimen dapat dikelompokkan menjadi tingkat dokumen, tingkat kalimat, dan tingkat aspek. Analisis pada tingkat dokumen menilai keseluruhan isi dokumen sebagai satu representasi sentimen. Sementara itu, analisis tingkat kalimat berfokus pada penentuan sentimen pada setiap kalimat secara terpisah, sehingga mampu memberikan hasil yang lebih detail dibandingkan tingkat dokumen. Adapun analisis tingkat aspek menitikberatkan pada identifikasi sentimen terhadap aspek-aspek tertentu dari suatu objek atau entitas yang dianalisis [58]. Dalam penerapannya, analisis sentimen juga dapat dibedakan berdasarkan pendekatan atau metode yang digunakan dalam proses analisis. Terdapat tiga pendekatan utama yang sering digunakan, yaitu:

1. Pendekatan dengan *lexicon* memanfaatkan kamus kata yang memiliki skor polaritas atau daftar kata dengan label tertentu. Pada pendekatan ini, teks terlebih dahulu dipecah menjadi token, kemudian setiap token dicocokkan dengan kamus sentimen untuk memperoleh skor atau label yang selanjutnya digabungkan guna menentukan sentimen keseluruhan.
2. Pendekatan *machine learning* adalah pendekatan yang menggunakan *machine learning* sebagai tugas klasifikasi dengan data berlabel, lalu model dilatih dengan memetakan teks ke kelas sentimen tertentu berdasarkan pola yang dipelajarin dari data latih. Beberapa algoritma yang biasanya digunakan adalah *logistic regresion*, SVM.

3. Pendekatan *deep learning* memanfaatkan model *pre-train* berbasis arsitektur *transformer*, seperti IndoBERT, yang telah mempelajari pola bahasa Indonesia dengan skala besar. Proses kerja IndoBERT meliputi tahap tokenisasi, pembentukan representasi kontekstual, proses klasifikasi, serta prediksi label sentimen [59].

2.6 Bidirectional Encoder Representations from Transformers (BERT)

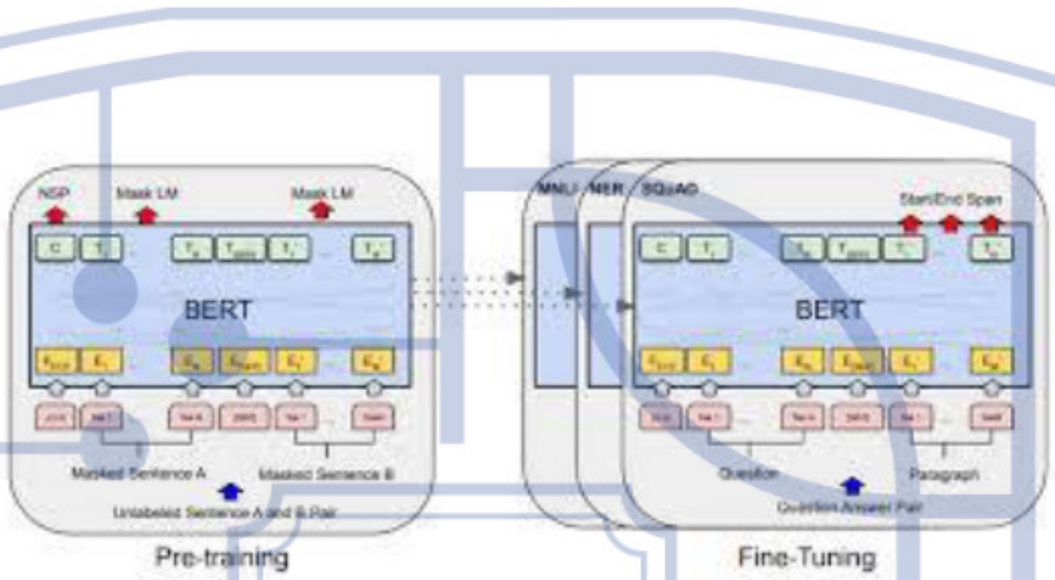
BERT merupakan model bahasa berbasis *deep learning* yang dikembangkan oleh Google dan Jacob Devlin pada tahun 2018. BERT dilatih menggunakan data teks tanpa label, sehingga mampu mempelajari hubungan kontekstual antar kata secara lebih akurat [62]. Model ini banyak digunakan dalam berbagai tugas *Natural Language Processing* (NLP), termasuk klasifikasi teks dan analisis sentimen. Penerapan BERT umumnya dilakukan memalui proses *fine-tuning*, yaitu menambahkan lapisan klasifikasi sesuai dengan kebutuhan tugas tertentu tanpa perlu melatih model dari awal. Pendekatan ini dilakukan karena lebih efisien dalam penggunaan sumber daya komputer dibandingkan dengan melatih model BERT dari awal yaitu melalui tahap *pre-training* [63]. Dalam proses masukan, BERT menggunakan teknik *WordPiece embedding* untuk mengubah teks menjadi token yang sesuai dengan kosakata model. Selain itu, terdapat token khusus seperti [CLS] yang digunakan untuk klasifikasi dan [SEP] untuk memisahkan kalimat. Representasi akhir dari gabungan token *embedding*, *segment embedding,* dan *positional embedding,* yang kemudian menjadi masukan bagi model BERT [63].



Gambar II.1 Arsitektur BERT [64]

Arsitektur BERT sebagai model bahasa berbasis *Transformer* ditunjukkan pada Gambar 2.1 BERT menerima masukan berupa urutan teks dan menghasilkan keluaran berupa pengkodean makna yang mempertimbangkan konteks untuk setiap token dalam urutan tersebut. Model ini bersifat *bidirectional*, artinya dalam memahami suatu kata, BERT memperhatikan kata-kata yang berada sebelum dan sesudahnya sehingga mampu menangkap makna yang lebih kompleks. BERT menggunakan arsitektur *Transformer encoder* untuk memproses teks. Setiap lapisan *encoder* terdiri dari dua komponen utama, yaitu *multi-head self-attention* dan *feed-forward neural network.* Pada tahap *self-attention*, model menganalisis hubungan antar kata dalam satu urutan untuk memperoleh informasi yang relevan. Hasil pemrosesan kemudian digabungkan kembali dengan input sebelumnya agar informasi awal tetap terjaga, lalu diteruskan ke lapisan berikutnya. Proses ini dilakukan berulang pada beberapa lapisan *encoder* sehingga pemahaman konteks menjadi semakin mendalam. Mekanisme *self-attention* memungkinkan setiap token dipahami dengan mempertimbangkan seluruh token lain dalam urutan yang sama [65]. Selain mekanisme tersebut, arsitektur BERT juga memiliki lapisan *Add & Norm* yang berfungsi menjaga proses pelatihan tetap stabil. Lapisan ini bekerja dengan menambahkan kembali informasi dari input sebelumnya melalui *residual connection* dan melakukan normalisasi data agar nilai yang diproses tetap seimbang.

Setelah itu, informasi diproses oleh *feed-forward neural network*, yaitu jaringan saraf yang terdiri dari dua lapisan *dense* yang membantu model mempelajari pola bahasa yang lebih kompleks. Hasil dari proses ini kemudian kembali melewati lapisan *Add & Norm* kedua untuk memastikan data tetap stabil sebelum diteruskan ke lapisan berikutnya. Kombinasi antara *self-attention, feed-forward neural network*, dan *Add & Norm* membuat BERT mampu memahami teks secara lebih kaya dan kontekstual sehingga efektif digunakan dalam berbagai tugas pemrosesan bahasa alami [66].



Gambar II.2 Ilustrasi tahapan *pre-training* [63]

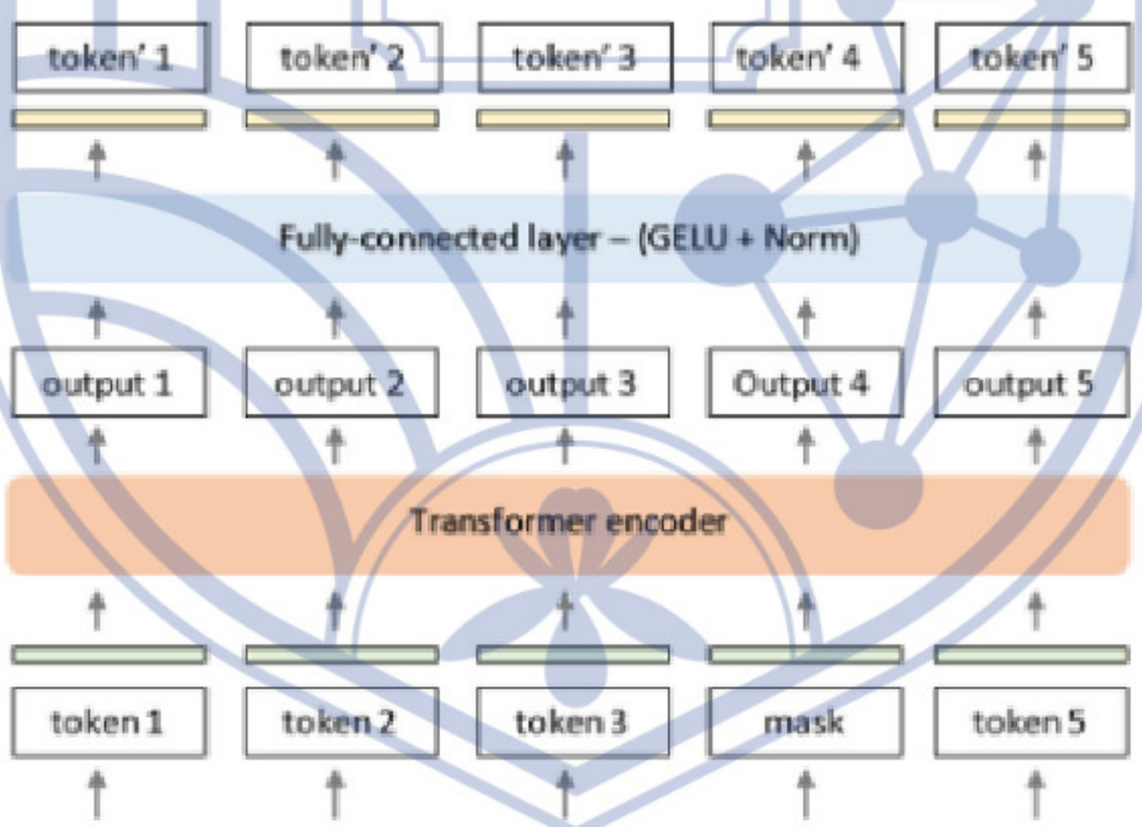
Pada Gambar 2.2 juga ditunjukkan bahwa proses pelatihan BERT terdiri dari dua tahap utama, yaitu *pre-training* dan *fine-tuning*. Pada tahap *pre-training,* model dilatih menggunakan dua tugas utama, yaitu *Masked Language Modelling* (MLM) dan *Next Sentence Prediction* (NSP). MLM memungkinkan model memahami konteks kata dari sisi kiri dan kanan dalam suatu kalimat sehingga mendukung pembelajaran dua arah. Sementara itu, NSP digunakan untuk mempelajari hubungan antara dua kalimat dalam satu pasangan teks. Setelah *pre-training* selesai, model dapat disesuaikan melalui proses *fine-tuning* untuk berbagai tugas tertentu, seperti analisis sentimen [63].

2.7 Indonesian Bidirectional Encoder Representation From Transformers (IndoBERT)

IndoBERT merupakan model turunan BERT yang dirancang khusus untuk memproses teks bahasa Indonesia dengan tetap mempertahankan arsitektur dasar BERT. Model ini dikembangkan menggunakan kosakata bahasa Indonesia dalam skala besar dan dilatih menggunakan korpus Indo4B, yaitu kumpulan data teks berbahasa Indonesia berskala besar yang terdiri dari sekitar 4 miliar kata dari berbagai sumber seperti unggahan media sosial, artikel blog, dan berita daring. Keberagaman sumber data tersebut memungkinkan IndoBERT untuk menangkap variasi ekspresi serta konteks penggunaan bahasa Indonesia secara lebih luas, sehingga model ini memiliki kinerja yang baik dalam berbagai tugas pemrosesan bahasa alami, termasuk analisis sentimen [67]. Dalam

penerapan analisis sentimen, IndoBERT digunakan melalui proses *fine-tuning*, yaitu dengan menambahkan lapisan klasifikasi pada model yang telah melalui tahap pelatihan awal (*pre-training*) [68].

IndoBERT menggunakan arsitektur *Transformer* yang sama dengan BERT asli dan terdapat beberapa varian dengan konfigurasi yang berbeda, di antaranya IndoBERT-*base* yang memiliki 768 *hidden units,* 12 lapisan (*layers),* dan 12 *attention heads,* dengan total sekitar 124 juta parameter, serta IndoBERT-*large* yang memiliki 1024 *hidden units*, 24 lapisan, dan 16 *attention heads*, dengan total sekitar 335 juta parameter [66], [68]. Selain itu, IndoBERT-*base* memiliki dua varian, yaitu IndoBERT-*base*-P1 dan IndoBERT-*base*-P2. Varian IndoBERT-*base*-P1 dilatih menggunakan pendekatan *transfer learning* dengan dataset teks bahasa Indonesia yang besar dan beragam, seperti artikel berita, Wikipedia, dan media sosial sehingga cocok untuk berbagai tugas NLP secara umum, sedangkan varian IndoBERT-*base*-P2 dilatih menggunakan dataset yang lebih kompleks dan beragam sehingga menghasilkan keluaran yang lebih presisi dan dinilai lebih sesuai untuk tugas yang memerlukan pemahaman bahasa yang lebih mendalam, seperti klasifikasi dokumen dan analisis sentimen [69], [70].



Gambar II.3 Tahap Tokenisasi IndoBERT [86]

Proses klasifikasi sentimen menggunakan IndoBERT diawali dengan tahap tokenisasi, sebagaimana ditunjukkan pada Gambar 2.3. Pada tahap ini, teks dipecah menjadi token menggunakan *tokenizer* IndoBERT. Model menambahkan token khusus seperti [CLS] di awal kalimat sebagai representasi keseluruhan teks, [SEP] sebagai penanda batas atau pemisah kalimat,

serta [PAD] untuk menyeragamkan panjang urutan token dalam satu *batch*. Token-token tersebut kemudian dikonversi menjadi indeks numerik berdasarkan kosakata IndoBERT [69]. Selanjutnya, indeks token tersebut diubah menjadi representasi vektor melalui proses *embedding*. Indeks token dipetakan ke dalam bentuk *embedding* yang merupakan gabungan dari token *embedding*, *position embedding*, dan *segment embedding*. Representasi *Embedding* ini kemudian menjadi inputan bagi arsitektur IndoBERT yang berbasis *Transformer encoder* [71]. Secara matematis dirumuskan sebagai berikut (2.1) [72]:

$$E_i = E_{token\,i} + E_{segment\,i} + E_{position\,i} \quad \text{.....(2.1)}$$

dimana $E_{token\,i}$ adalah token *embedding* yang merepresentasikan setiap token input ke dalam vektor berdimensi tinggi berdasarkan kosakata model, $E_{segment\,i}$ adalah *segment embedding* yang menunjukkan asal token dalam suatu pasangan kalimat sehingga model dapat membedakan apakah token berasal dari kalimat pertama atau kedua, $E_{position\,i}$ adalah *positional embedding* yang memberikan informasi mengenai posisi token dalam urutan kalimat [73]. Ketiga komponen tersebut digabungkan untuk menghasilkan E_i, yaitu vektor representasi akhir untuk token ke-i. Kombinasi ketiga komponen ini memungkinkan model memahami makna kata sekaligus posisi antar hubungan dalam satu kalimat [72].

Embedding yang dihasilkan kemudian diproses oleh lapisan *encoder* pada arsitektur *Transformer*. Pada varian IndoBERT-*base*, arsitektur ini terdiri dari 12 lapisan *encoder* dengan 12 *attention heads* dan 768 *hidden units*. Setiap lapisan dilengkapi mekanisme *multi-head self-attention* yang memungkinkan model memahami hubungan antar token secara dua arah (*bidirectional*), sehingga makna suatu kata dapat ditentukan berdasarkan keseluruhan konteks kalimat [71]. Representasi input pada mekanisme *self-attention* dapat dinyatakan dalam bentuk matriks (2.2) [65]:

$$H \in \mathbb{R}^{m\,x\,d} \quad \text{.....(2.2)}$$

dimana m menunjukkan jumlah token dalam suatu teks dan d merupakan dimensi vektor representasi [65]. Untuk menghitung *self-attention*, setiap token akan diproyeksikan ke dalam tiga representasi yang berbeda yaitu *Query* (Q), *Key* (K), dan *Value* (V) melalui tranformasi *linear* sebagai berikut (2.3), (2.4), (2.5) [65]:

$$Q = HW_q \quad \text{.....(2.3)}$$

$$K = HW_k \quad \text{..... (2.4)}$$

$$V = HW_v \quad \text{..... (2.5)}$$

Parameter W_q, W_k, W_v merupakan matriks bobot yang dipelajari selama proses pelatihan model [65]. Selanjutnya nilai *attention* dihitung sebagai berikut (2.6) [72]:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{..... (2.6)}$$

di mana Q, K, dan V masing-masing merepresentasikan *Query, Key*, dan *Value*, serta d_k merupakan dimensi *vektor key* [72]. Rumus tersebut digunakan untuk menghitung tingkat keterkaitan antara satu token dengan token lainnya dalam satu kalimat. Dalam proses ini, *Query* (Q) merupakan vektor yang mewakili token yang sedang di proses oleh model, *Key* (K) merupakan vektor yang merepresentasikan informasi dari token lain yang digunakan sebagai acuan untuk menentukan tingkat keterkaitan, sedangkan *Value* (V) merupakan vektor yang berisi informasi dari token yang akan digunakan dalam proses perhitungan *attention* [74]. Untuk menangkap berbagai jenis hubungan makna secara bersamaan, digunakan mekanisme *multi-head attention* yang dirumuskan sebagai berikut (2.7) [72]:

$$MultiHead(Q,K,V) = Concat\left(head_1, \ldots, head_h\right)W^0 \quad \text{..... (2.7)}$$

Proses tersebut dilakukan secara berulang pada setiap lapisan *encoder* sehingga model dapat memahami hubungan dan konteks kata dalam kalimat dengan lebih baik [72].

Setelah melalui seluruh lapisan *encoder*, keluaran dari token [CLS] digunakan sebagai ringkasan informasi dari seluruh teks [71]. Untuk tugas analisis sentimen, IndoBERT kemudian di-*fine-tune* dengan menambahkan lapisan klasifikasi berupa *fully connected layer* [71]. Pada tahap *fine-tuning*, model *BertForSequenceClassification* yang menambahkan lapisan klasifikasi di atas *encoder* IndoBERT untuk menghasilkan prediksi label sentimen [71].

Representasi dari token [CLS] kemudian diteruskan untuk menghasilkan nilai *logits* sesuai jumlah kelas sentimen yang ditentukan [71]. Nilai *logits* merupakan skor keluaran mentah dari model sebelum dikonversi menjadi probabilitas menggunakan fungsi *softmax*. [72], di mana nilai tersebut merepresentasikan skor mentah untuk setiap kelas sentimen yang dihasilkan dari proses klasifikasi. Dirumuskan sebagai berikut (2.8) [72]:

$$logits = W.a_{attention} + b \quad \text{.....(2.8)}$$

di mana W adalah matriks bobot lapisan *linear* (*weight matrix*), yang merupakan parameter model yang dipelajari selama proses pelatihan untuk menghasilkan skor pada masing-masing kelas. b adalah vektor *bias*, yang juga merupakan parameter yang dioptimasi selama pelatihan untuk menyesuaikan hasil prediksi, dan a *attention* merupakan representasi keluaran dari token [CLS] hasil proses *self-attention* [72]. Nilai tersebut kemudian diubah menjadi probabilitas menggunakan fungsi *softmax*, sehingga model dapat menentukan label sentimen akhir [71]. Rumus fungsi *softmax* adalah sebagai berikut (2.9) [72]:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{.....(2.9)}$$

di mana z adalah vektor keluaran mentah dari jaringan saraf dan K adalah jumlah kelas. Nilai e merupakan konstanta eksponensial dengan nilai sekitar 2,718. Kemudian, elemen ke-i dari vektor hasil fungsi *softmax* merepresentasikan probabilitas bahwa suatu data diprediksi termasuk dalam kelas ke-i [74]. Nilai masukan pada fungsi *softmax* dapat berupa bilangan positif, negatif, nol, maupun nilai yang lebih besar dari satu. Fungsi ini kemudian mengonversi nilai tersebut ke dalam rentang 0 hingga 1 sehingga dapat diinterpretasikan sebagai probabilitas. Semakin besar nilai masukan, maka probabilitas yang dihasilkan akan semakin tinggi, sedangkan nilai masukan yang lebih kecil akan menghasilkan probabilitas yang lebih rendah. Meskipun demikian, total probabilitas dari seluruh kelas akan selalu berjumlah satu [74].

Probalitas yang dihasilkan dari fungsi *softmax* tersebut kemudian digunakan dalam proses pelatihan model untuk menghitung kesalahan prediksi. Pada tahap *fine-tuning,* model dilatih menggunakan dataset berlabel dengan meminimalkan fungsi *cross-entropy loss* [68], yang umum digunakan dalam klasifikasi multi-kelas dan dirumuskan sebagai berikut (2.10) [72]:

$$L = -\sum_{i=1}^{K} yi \log(\hat{y}i) \quad \text{.....(2.10)}$$

di mana $\hat{y}i$ merupakan label sebenarnya dan $\hat{y}i$ merupakan probabilitas prediksi yang dihasilkan oleh model. Semakin akurat prediksi model terhadap label sebenarnya, semakin kecil nilai *loss* yang diperoleh [74]. Oleh karena itu, kombinasi *softmax* dan *Cross-Entropy Loss* umum digunakan dalam pelatihan model klasifikasi teks[72]. Fungsi *Cross entropy* digunakan untuk mengukur perbedaan antara dua distribusi probabilitas, yaitu distribusi probabilitas sebenarnya dan distribusi probabilitas yang dihasilkan oleh model untuk suatu variabel atau kejadian tertentu. Selama proses

pelatihan, fungsi ini digunakan untuk menyesuaikan bobot model dengan tujuan meminimalkan nilai *loss*. Semakin kecil nilai *loss* yang diperoleh, maka semakin baik kinerja model dalam melakukan prediksi [74].

Untuk meminimalkan nilai *loss* tersebut, proses pelatihan model biasanya menggunakan teknik optimasi. Salah satu teknik optimasi model yang biasanya digunakan adalah *AdamW* [68]. Selain itu, pengaturan panjang urutan teks (*maximum sequence length*) juga menjadi faktor penting dalam proses pelatihan. Panjang urutan dapat disesuaikan agar mampu merepresentasikan konteks kalimat secara optimal tanpa meningkatkan kompleksitas komputasi secara berlebihan. Beberapa penelitian menetapkan panjang urutan sebesar 128 token yang dinilai cukup untuk merepresentasikan sebagian besar teks ulasan berbahasa Indonesia tanpa kehilangan informasi penting [75]. Sementara itu, penggunaan panjang urutan hingga 256 token juga dapat diterapkan untuk menjaga konteks kalimat tetap terwakili secara lebih lengkap tanpa memperpanjang waktu pelatihan secara signifikan [76]. Selain teknik optimasi dan pengaturan panjang urutan, proses pelatihan juga dapat menggunakan teknik regularisasi seperti *early stopping*, yaitu teknik yang digunakan untuk membantu mencegah terjadinya *overfitting* [83].

Dengan karakteristik tersebut, IndoBERT memiliki keunggulan dalam memahami makna dan struktur teks berbahasa Indonesia secara lebih efektif dibandingkan metode konvensional. Keunggulan ini menjadikan IndoBERT banyak digunakan dalam berbagai tugas NLP, khususnya analisis sentimen, serta menunjukkan performa yang lebih baik dibandingkan beberapa model pralatih lain dan metode klasifikasi tradisional, seperti *Support Vector Machine*, *Naïve Bayes*, dan *K-Nearest Neighbor*. Oleh karena itu, pemanfaatan IndoBERT dinilai tepat untuk analisis sentimen pada komentar di media sosial yang tidak bersifat tidak terstruktur, beragam, dan memiliki volume data yang besar, sehingga persepsi publik dapat diidentifikasi secara lebih akurat [77].

2.8 Hyperparameter

Hyperparameter tuning merupakan tahap penting dalam pengembangan model *deep learning* untuk memperoleh konfigurasi yang optimal sehingga dapat menghasilkan performa terbaik [78]. Dalam model berbasis *Transformer* seperti IndoBERT, proses ini dilakukan melalui *fine-tuning*, yaitu dengan mengatur *hyperparameter* agar model dapat beradaptasi dengan kebutuhan tugas tertentu. Sebagian besar *hyperparameter* yang digunakan pada tahap *fine-tuning* memiliki kesamaan dengan parameter yang digunakan pada proses pelatihan BERT secara umum [79]. Melalui proses *fine-tuning*, model dapat menyesuaikan *parameter* internalnya sehingga lebih

sesuai dengan karakteristik dataset yang digunakan dalam pada proses pelatihan model. Selama proses pelatihan, *parameter* model akan terus diperbarui secara bertahap hingga mencapai performa yang stabil [80]. Karena IndoBERT merupakan pengembangan dari model BERT, maka proses *fine-tuning* dan pengaturan *hyperparameter* mengikuti prinsip yang sama, dengan penyesuaian pada konteks bahasa Indonesia [81]. Oleh karena itu, penyesuaian *hyperparameter* menjadi langkah penting untuk memastikan model berbasis BERT, termasuk IndoBERT, dapat menghasilkan kinerja yang optimal sesuai dengan karakteristik data yang digunakan [82]. Dalam proses pelatihan, terdapat beberapa *hyperparameter* utama untuk mengoptimalkan kinerja model, yaitu sebagai berikut:

1. *Learning rate* merupakan parameter yang menentukan seberapa besar pembaruan bobot model pada setiap iterasi selama proses pelatihan. Pemilihan nilai *learning rate* perlu diperhatikan dengan baik karena nilai yang terlalu besar dapat menyebabkan model tidak konvergen, sedangkan nilai yang terlalu kecil akan memperlambat proses pelatihan [83]. Pada proses *fine-tuning* BERT, nilai *learning rate* yang umum digunakan berada pada rentang 2e-5 hingga 5e-5 untuk menjaga kestabilan pembelajaran [84]. Selain itu, *Learning rate* juga sering disesuaikan dalam kisaran 1e-5 hingga 5e-5 guna memastikan proses konvergensi berjalan dengan baik [82]. Beberapa penelitian menggunakan variasi nilai seperti 5e-5, 3e-5, dan 2e-5 untuk mendapatkan hasil terbaik, dimana IndoBERT memperoleh akurasi rata-rata sebesar 0,75 dengan akurasi tertinggi 0,78 pada konfigurasi *epoch* 3, *learning rate* 2e-5, dan *batch size* 32 [85]. Penggunaan *learning rate* sebesar 2e-5 juga menunjukkan hasil yang konsisten dengan akurasi mencapai 77% pada berbagai variasi *epoch* [79]. Nilai *learning rate* yang lebih kecil, seperti 5e-6, juga dapat digunakan untuk mengontrol kecepatan pembelajaran model agar lebih stabil [86]. Pemilihan *learning rate* sebesar 2e-5 sering digunakan karena cukup kecil untuk melakukan penyesuaian bobot secara bertahap tanpa merusak representasi bahasa yang telah dipelajari sebelumnya [76]. Selain itu, penggunaan *learning rate* yang kecil dalam proses *fine-tuning* model *Transformer* juga penting untuk mencegah terjadinya penurunan kemampuan model dalam mempertahankan pengetahuan yang telah dipelajari sebelumnya, sehingga pengetahuan awal model tetap terjaga [87].
2. *Batch size* merupakan parameter yang menentukan jumlah data yang diproses dalam satu iterasi pembaruan bobot selama proses pelatihan [86]. Dalam proses *fine-tuning* model, ukuran *batch* yang umum digunakan berada pada rentang 16 hingga 32, dengan

mempertimbangkan keterbatasan memori GPU yang tersedia [84]. Pengaturan *batch size* merupakan bagian penting dari *hyperparameter* yang perlu disesuaikan bersama parameter lain seperti *learning rate* dan jumlah *epoch* untuk memperoleh performa model yang optimal [80]. Beberapa penelitian menunjukkan bahwa penggunaan *batch size* 16 dan 32 dapat menghasilkan performa yang baik pada model BERT, di mana kombinasi *batch size* 32, *learning rate* 2e-5, dan *epoch* 3 mampu menghasilkan akurasi tertinggi sebesar 0,78 pada model IndoBERT [85]. Selain itu, penggunaan *batch size* sebesar 32 juga banyak diterapkan karena memberikan keseimbangan antara efisiensi memori dan stabilitas gradien selama pelatihan [76]. Namun, dalam beberapa kondisi, ukuran *batch* yang lebih kecil seperti 8 untuk data latih dan 16 untuk evaluasi juga digunakan untuk menyesuaikan kapasitas komputasi serta membantu mengurangi risiko *overfitting* [88]. Hasil evaluasi menunjukkan bahwa pemilihan *batch size* memiliki peran penting dalam performa model, sehingga perlu mempertimbangkan keseimbangan antara kecepatan konvergensi, efisiensi komputasi, dan stabilitas proses pelatihan [89].

3. *Epoch* merupakan jumlah perulangan pelatihan model terhadap seluruh dataset, di mana setiap *epoch* merepresentasikan satu siklus lengkap proses pembelajaran [86]. Dalam proses *fine-tuning* model, jumlah *epoch* yang umum digunakan berkisar antara 3 hingga 5 untuk memperoleh performa yang optimal tanpa meningkatkan risiko *overfitting* [84]. Selain itu, beberapa penelitian merekomendasikan penggunaan *epoch* dalam rentang 2 hingga 4 sebagai pilihan yang efektif untuk mencapai konvergensi yang baik sekaligus menjaga stabilitas pelatihan [88]. Hasil eksperimen menunjukkan bahwa performa model IndoBERT cenderung mencapai nilai terbaik pada *epoch* ke-3, khususnya ketika dikombinasikan dengan *hyperparameter* lain seperti *learning rate* 2e-5 dan *batch size* 32, yang menghasilkan akurasi tertinggi sebesar 0,78 [85]. Di sisi lain, penggunaan jumlah *epoch* yang lebih besar, seperti 5, juga dapat dilakukan untuk memastikan model mempelajari keseluruhan data secara menyeluruh dalam beberapa putaran pelatihan [86]. Namun, pemilihan jumlah *epoch* tetap perlu disesuaikan dengan kondisi dataset dan dikombinasikan dengan *hyperparameter* lain agar dapat menghasilkan performa yang optimal serta menghindari *overfitting* [79].
4. Pada tahap kompilasi model, ditentukan komponen penting seperti *optimizer* yang berperan dalam proses pelatihan model [84]. *Optimizer* merupakan algoritma dalam kecerdasan buatan yang berfungsi untuk menyesuaikan *parameter* model, seperti bobot dan *bias*, dengan tujuan

meminimalkan fungsi kerugian (*loss*) sehingga kinerja model dapat meningkat. Oleh karena itu, pemilihan *optimizer* yang tepat memiliki pengaruh signifikan terhadap akurasi dan efisiensi model [90]. Selama proses pelatihan, fungsi *loss* digunakan untuk mengukur kesalahan prediksi, sedangkan *optimizer* bekerja secara iteratif untuk memperbarui bobot model hingga mencapai performa yang optimal [80]. Salah satu *optimizer* yang umum digunakan adalah *AdamW*, yang dikenal stabil dalam proses pembaruan bobot model [82]. *AdamW* merupakan pengembangan dari *optimizer Adam* dengan memperkenalkan teknik *weight decay* secara lebih eksplisit dalam proses optimisasi. *AdamW* memodifikasi algoritma *Adam* dengan memisahkan mekanisme *weight decay* dari proses pembaruan parameter, sehingga mampu mengatasi kelemahan *Adam* dalam mengelola regularisasi [90]. Dengan karakteristik tersebut, *AdamW* dapat meningkatkan efisiensi pembelajaran serta menghasilkan performa prediksi yang lebih baik [82]. Dalam praktik *fine-tuning* model, proses pelatihan umumnya dilakukan selama beberapa *epoch*, misalnya 3 *epoch*, dengan menggunakan *optimizer* seperti *AdamW* untuk mencapai konvergensi yang optimal [88].

2.9 Teknik Regularisasi

Teknik regularisasi merupakan sekumpulan metode krusial yang diterapkan selama proses pelatihan model *deep learning* dengan tujuan utama untuk mencegah terjadinya fenomena *overfitting* [91]. Fenomena *overfitting* ini merujuk pada suatu kondisi spesifik di mana model mampu mencapai tingkat akurasi yang sangat tinggi pada data latih, namun justru gagal melakukan generalisasi ketika dihadapkan pada data pengujian yang baru [92]. Kegagalan generalisasi tersebut umumnya terjadi karena model terlalu fokus menghafal derau (*noise*) serta detail-detail yang tidak relevan dari dataset, alih-alih mempelajari pola semantik utama di dalamnya. Mengingat penelitian ini menggunakan arsitektur *Transformer* berskala besar seperti IndoBERT yang dibekali dengan sekitar 124 juta parameter bawaan, risiko terjadinya *overfitting* menjadi sangat tinggi [93]. Risiko tersebut semakin meningkat secara drastis ketika model melalui tahap *fine-tuning* menggunakan dataset komentar pengguna TikTok terkait program Makan Bergizi Gratis, yang sangat identik dengan gaya bahasa informal, slang, dan keragaman linguistik. Oleh karena itu, penerapan strategi regularisasi menjadi langkah mitigasi yang mutlak diperlukan agar model tidak sekadar menghafal frasa spesifik, melainkan benar-benar memahami konteks sentimen publik terhadap aspek kualitas makanan [94]. Beberapa komponen regularisasi yang dapat diimplementasikan secara terintegrasi untuk memastikan model klasifikasi sentimen dapat

beroperasi dengan tingkat stabilitas dan ketangguhan (*robustness*) yang optimal sebelum akhirnya diuji keakuratannya.

1. *Early stopping* merupakan teknik regularisasi yang digunakan untuk menghentikan proses pelatihan ketika performa pada data validasi tidak lagi menunjukkan peningkatan [78]. Teknik ini bertujuan untuk menjaga keseimbangan antara proses pembelajaran dan kemampuan generalisasi model sehingga dapat mencegah *overfitting* [95]. Secara umum, mekanisme *early stopping* dilakukan dengan memantau nilai validation loss selama proses pelatihan. Pelatihan akan dihentikan secara otomatis apabila tidak terjadi penurunan nilai tersebut dalam beberapa *epoch* berturut-turut, misalnya dengan parameter patience sebesar 3, serta mengembalikan bobot model terbaik yang telah diperoleh sebelumnya [81]. Dengan demikian, ketika performa validasi tidak lagi meningkat, proses pelatihan dihentikan dan model menggunakan bobot terbaik (*best state*) untuk tahap evaluasi selanjutnya [96]. Penggunaan *early stopping* terbukti efektif dalam meningkatkan kinerja model, khususnya dalam klasifikasi teks, karena mampu mencegah *overfitting* sekaligus mempercepat proses pencapaian kondisi optimal selama pelatihan [82].
2. *Gradient clipping* merupakan teknik regularisasi yang digunakan untuk menjaga stabilitas proses pelatihan dengan membatasi nilai gradien agar tidak terlalu besar dalam satu iterasi [97]. Teknik ini bertujuan untuk mencegah terjadinya gradien yang meningkat secara tidak terkendali yang dapat menyebabkan perubahan bobot secara drastis dan mengganggu proses pembelajaran model [75]. Dalam penerapannya, *gradient clipping* biasanya dilakukan dengan menetapkan batas maksimum (*max norm*) tertentu, misalnya 1.0, sehingga proses pembelajaran menjadi lebih stabil dan terkendali [98]. Setelah proses pembatasan gradien dilakukan, pembaruan parameter dilanjutkan melalui langkah *optimizer*, kemudian diikuti oleh *scheduler* sebelum memasuki tahap validasi pada setiap *epoch* untuk memantau performa model [99]. Dengan demikian, penggunaan *gradient clipping* dapat membantu menjaga kestabilan pelatihan serta mendukung proses evaluasi model secara berkelanjutan [96].

2.10 Tahapan Preprocessing Teks

Preprocessing teks merupakan tahapan krusial dalam pemrosesan bahasa alami (NLP) dan analisis sentimen untuk memperbaiki kualitas data teks mentah menjadi bentuk data yang lebih baik dan terstruktur, seperti menghapus data duplikat, pengecekan data tidak konsisten, dan perbaikan kesalahan di data [98], [101], [102]. Tanpa *preprocessing* yang baik, kesalahan dalam

ekstraksi fitur bisa terjadi, sehingga performa dan akurasi model klasifikasi sentimen bisa menurun secara signifikan [100], [102], [103]. Berdasarkan literatur terkait analisis sentimen berbahasa Indonesia, berikut adalah langkah-langkah *preprocessing* yang diterapkan pada penelitian ini:

2.10.1 Case Folding

Case folding berfungsi untuk mengubah semua huruf dalam teks menjadi huruf kecil (*lowercase*) sehingga variasi huruf besar dan kecil tidak memengaruhi analisis. Proses ini mencegah model menganggap kata yang sama sebagai entitas berbeda hanya karena perbedaan huruf kapital, sehingga mengurangi dimensi data yang harus terjadi [100], [101].

1. Cara Kerja: Sistem membaca seluruh urutan karakter (*string*) dalam dataset komentar dan mengonversi setiap huruf kapital (A-Z) menjadi huruf kecil (a-z) menggunakan fungsi bawaan bahasa pemrograman.
2. Contoh: Teks ulasan "Layanan internet SANGAT lambat hari ini, tolong diperbaiki!" diubah menjadi huruf kecil seluruhnya: "layanan internet sangat lambat hari ini, tolong diperbaiki!".

2.10.2 Cleaning

Cleaning adalah proses untuk menghapus bagian-bagian yang tidak diperlukan dalam teks atau dianggap sebagai *noise*, seperti tanda baca, angka, simbol, *link* URL, emoji, dan *username* (*mention*) [98], [104]. Tujuannya agar teks lebih rapi dan model hanya berfokus pada kata-kata yang membawa makna sentimen.

1. Cara Kerja: Menggunakan pola *Regular Expression* (RegEx) untuk mendeteksi karakter selain huruf alfabet. Elemen yang cocok dengan pola tersebut, termasuk tanda baca berlebih, *sticker*, atau karakter tidak bermakna, dihapus atau diganti dengan spasi kosong.
2. Contoh: Teks "Wah, barangnya bagus bgt*!* 😍👍 cek *link* http://olshop.com @tokotermurah" dibersihkan dari emoji, tautan, dan *mention* sehingga menyisakan "wah barangnya bagus bgt cek link".

2.10.3 Normalisasi Data

Pengguna media sosial cenderung menggunakan bahasa tidak baku (*slang*), singkatan, atau memiliki kesalahan ketik (*typo*). Normalisasi berfungsi untuk memetakan kata-kata tidak baku tersebut menjadi kata baku yang sesuai dengan ejaan standar agar fitur teks menjadi lebih konsisten [100].

1. Cara Kerja: Melakukan pencocokan setiap kata dalam teks dengan kamus translasi kata tidak baku (*dictionary*). Jika sebuah kata ditemukan dalam kamus tersebut, sistem akan otomatis menggantinya dengan padanan kata bakunya.
2. Contoh: Kata "brg", "udh", dan "ga" dinormalisasi berturut-turut menjadi "barang", "sudah", dan "tidak". Kalimat "brg udh sampe tp ga sesuai" berubah menjadi "barang sudah sampai tapi tidak sesuai".

2.10.4 Tokenizing

Tokenizing adalah proses memecah teks utuh menjadi unit-unit terkecil (token) berupa kata tunggal, sehingga setiap kata dalam kalimat dapat diproses secara terpisah oleh model [101], [104]. Model berbasis *Transformer* seperti IndoBERT membutuhkan masukan berupa token terpisah sebelum dikonversi menjadi representasi numerik (ID Token).

1. Cara Kerja: Memisahkan teks berdasarkan pembatas karakter tertentu, umumnya menggunakan spasi (*whitespace*), dan memasukkannya ke dalam bentuk struktur data *list* atau *array*.
2. Contoh: Kalimat hasil normalisasi "barang sudah sampai dengan aman" dipecah menjadi himpunan token: ['barang', 'sudah', 'sampai', 'dengan', 'aman'].

2.10.5 Stopword Removal

Stopword removal digunakan untuk menghapus kata-kata hubung yang sangat umum namun tidak memiliki arah sentimen yang spesifik [100], [101]. Menghapus kata-kata ini membantu model untuk lebih fokus pada kata-kata inti yang membawa bobot sentimen.

1. Cara Kerja: Token hasil tahap tokenisasi dicocokkan dengan pustaka daftar *stopword* bahasa Indonesia (misalnya dari pustaka NLTK). Token yang terdapat dalam daftar tersebut akan dieliminasi.
2. Contoh: Dari *list* token ['kualitas', 'dari', 'produk', 'ini', 'sangat', 'bagus'], kata *"dari"* dan *"ini"* termasuk dalam *stopword* sehingga dihapus. Hasil akhirnya hanya menyisakan token inti, yaitu: ['kualitas', 'produk', 'sangat', 'bagus'].

Namun, pada model berbasis transformer seperti BERT dan IndoBERT, *stopword* tidak selalu perlu dihapus. Model ini dapat memahami hubungan antar kata dalam satu kalimat, termasuk kata-kata umum tersebut. Beberapa penelitian menunjukkan bahwa menghapus *stopword* pada teks pendek, seperti komentar di media sosial, justru dapat menghilangkan makna penting dalam

kalimat. IndoBERT diketahui bekerja lebih baik ketika menggunakan teks asli tanpa penghapusan *stopword*. Hasil serupa juga dilaporkan dalam penelitian lain, di mana model tetap akurat meskipun tanpa proses tersebut [105], [106].

2.10.6 Stemming

Stemming adalah proses pengolahan morfologis yang bertujuan mereduksi kata berimbuhan (afiksasi) kembali menjadi bentuk kata dasarnya. Langkah ini mencegah redundansi fitur, di mana kata turunan akan dianggap memiliki representasi makna yang sama dengan kata dasarnya [100], [101], [104].

1. Cara Kerja: Menggunakan pustaka *stemmer* khusus bahasa Indonesia (seperti Sastrawi) yang bekerja dengan memotong awalan (*prefix*), sisipan (*infix*), dan akhiran (*suffix*) berdasarkan aturan tata bahasa Indonesia.
2. Contoh: Kata turunan *"pengiriman"* direduksi awalan (peng-) dan akhirannya (-an) menjadi bentuk dasar *"kirim"*. Demikian pula kata *"mengecewakan"* direduksi menjadi *"kecewa"*.

Meskipun sering digunakan dalam metode lama, *stemming* tidak selalu diperlukan untuk model seperti IndoBERT. Model ini sudah mampu memahami berbagai bentuk kata melalui proses tokenisasi otomatis. Beberapa penelitian menunjukkan bahwa *stemming* tidak memberikan peningkatan hasil yang signifikan pada model berbasis *transformer*. Bahkan, mengubah kata ke bentuk dasar dapat menghilangkan makna tertentu yang sebenarnya penting dalam sebuah kalimat. Selain itu, penggunaan *stemming* bersama *stopword removal* juga tidak terbukti meningkatkan akurasi, terutama pada teks pendek di media sosial. Dengan demikian, penggunaan teks asli tanpa *stemming* dan tanpa penghapusan *stopword* dinilai lebih efektif, karena model dapat memahami konteks kalimat secara lebih lengkap [105], [106].

2.11 Pelabelan Data

Pelabelan data merupakan tahapan penting dalam pengolahan dataset yang bertujuan untuk memberikan kategori atau label tertentu pada setiap data. Proses ini dilakukan agar data memiliki identitas yang jelas sesuai dengan karakteristik yang ingin dianalisis [107]. Dengan adanya pelabelan, data dapat digunakan secara optimal dalam proses analisis maupun pelatihan model [108]. Label yang diberikan umumnya berupa informasi tambahan yang relevan dengan data, seperti kategori, kelas, atau nilai tertentu, yang nantinya digunakan sebagai dasar dalam proses pembelajaran model maupun analisis lanjutan [109].

Dalam praktiknya, pelabelan data dapat dilakukan melalui dua pendekatan, yaitu secara manual dan semi-otomatis. Pendekatan manual dilakukan untuk memastikan ketepatan konteks dan makna dari data, khususnya pada data teks yang memerlukan pemahaman bahasa. Sementara itu, pendekatan semi-otomatis memanfaatkan bantuan algoritma, seperti pencocokan kata kunci (*keyword matching*) dan analisis frekuensi kata, sehingga mampu mempercepat proses pelabelan terutama pada dataset berukuran besar [110].

Untuk menjaga konsistensi hasil pelabelan manual, diperlukan pengukuran tingkat kesepakatan antar anotator menggunakan *Cohen's Kappa*, yaitu metode statistik yang digunakan untuk mengukur tingkat kesepakatan dua penilai terhadap data kategorikal dengan mempertimbangkan kemungkinan kesepakatan yang terjadi secara kebetulan [111]. Penggunaan *Cohen's Kappa* bertujuan untuk memastikan bahwa hasil pelabelan memiliki reliabilitas yang baik dan konsisten antar anotator. Nilai *Cohen's Kappa* dihitung menggunakan persamaan berikut [112], [113] :

$$K = \frac{P_0 - P_e}{1 - P_e} \quad \text{.....(2.11)}$$

Dengan:

$$P_0 = \frac{a + d}{n} \quad \text{.....(2.12)}$$

Sebagai proporsi kesepakatan aktual (*observed agreement*), dan:

$$P_e = \left[\left(\frac{n_1}{n}\right) \times \left(\frac{m_1}{n}\right)\right] + \left[\left(\frac{n_0}{n}\right) \times \left(\frac{m_0}{n}\right)\right] \quad \text{.....(2.13)}$$

Sebagai proporsi kesepakatan yang terjadi secara kebetulan (*expected agreement*). Interpretasi nilai *kappa* dapat dilihat pada Tabel 2.1 [111].

Tabel II.1 Interpretasi nilai *kappa*

Nilai Kappa	Interpretasi
<0.00	*Poor Agreement*
0.00 – 0.20	*Slight Agreement*
0.21 – 0.40	*Fair Agreement*
0.41 – 0.60	*Moderate Agreement*
0.61 – 0.80	*Substantial Agreement*
0.81 – 1.00	*Almost Perfect Agreement*

Semakin tinggi nilai *Cohen's Kappa*, maka semakin tinggi tingkat kesepakatan antar anotator sehingga hasil pelabelan dianggap memiliki reliabilitas yang baik dan dapat digunakan dalam proses pelatihan model klasifikasi sentimen berbasis pembelajaran mesin [111].

Secara umum, proses pelabelan data dikelompokkan ke dalam tiga kategori utama, yaitu positif, negatif, dan netral [114]. Kategori negatif diberikan pada teks yang mengandung penolakan atau evaluasi yang tidak mendukung, kategori netral pada teks yang bersifat ambigu atau tidak menunjukkan kecenderungan tertentu, sedangkan kategori positif diberikan pada teks yang mengandung dukungan atau penilaian yang menguntungkan terhadap suatu isu [115]. Analisis sentimen dapat dipahami sebagai pendekatan yang berfokus pada evaluasi opini, sikap, dan emosi terhadap suatu entitas [116]. Dalam kajian analisis sentimen, istilah sentimen tidak hanya dimaknai sebagai kategori positif, negatif dan netral semata, tetapi juga dapat berupa emosi yang dapat diukur berdasarkan sumber tekstual, termasuk data dari media sosial TikTok yang memungkinkan pengguna bebas berpendapat dan cenderung menggunakan bahasa yang tidak formal [1], [117]. Meskipun pembagian sentimen ke dalam kategori positif, negatif, dan netral masih umum digunakan, pendekatan tersebut dinilai memiliki keterbatasan dalam menggambarkan makna emosional teks secara lebih rinci [118]. Keterbatasan tersebut menunjukkan bahwa pelabelan data

dapat dikembangkan menjadi lebih spesifik sesuai dengan konteks dan karakteristik data penelitian.

Dalam konteks media sosial, komentar pengguna tidak hanya berisi dukungan atau penolakan, tetapi juga dapat memuat kritik maupun saran sebagai bentuk evaluasi terhadap suatu layanan [119], [120]. Komentar pada media sosial dalam konteks layanan publik secara alami mengandung bentuk umpan balik seperti pujian, saran, dan kritik, bukan hanya ekspresi emosi sederhana. Hal ini menunjukkan bahwa struktur opini pengguna bersifat evaluatif dan konstruktif, sehingga tidak sepenuhnya dapat direpresentasikan oleh klasifikasi positif, negatif, dan netral [121]. Komentar pada TikTok tidak hanya merepresentasikan sentimen, tetapi juga berbagai bentuk tindak tutur seperti ekspresif (memuji dan mengkritik) serta direktif (memberi saran), yang menunjukkan adanya perbedaan fungsi komunikasi dalam setiap ujaran. Hal ini menegaskan bahwa komentar pengguna tidak sekadar menyampaikan emosi, tetapi juga melakukan evaluasi dan memberikan arahan terhadap suatu objek, sehingga pendekatan berbasis polaritas tidak mampu menangkap keragaman fungsi tersebut [122]. Penggunaan label pujian, kritik, dan saran didasarkan pada perspektif tindak tutur yang memandang komentar sebagai bentuk evaluasi dan arahan, bukan sekadar ekspresi emosi. Dalam konteks layanan publik, komentar tidak hanya menyatakan sentimen, tetapi juga berfungsi sebagai apresiasi, penilaian, dan rekomendasi perbaikan. Oleh karena itu, klasifikasi positif, negatif, dan netral tidak mampu merepresentasikan perbedaan fungsi tersebut, sehingga digunakan kategori pujian, kritik, dan saran untuk menggambarkan opini pengguna secara lebih akurat [123].

Untuk memperoleh gambaran opini yang lebih spesifik, penelitian ini menggunakan tiga kategori pelabelan, yaitu pujian, kritik, dan saran. Pujian merupakan pernyataan yang mengandung kekaguman, penghargaan atau apresiasi terhadap suatu objek [124]. Sementara itu, kritik merupakan respons atau penilaian seseorang terhadap baik atau buruknya suatu hal yang didasarkan pada nilai atau norma yang diyakininya. Kritik biasanya muncul dari sikap psikologis negatif, seperti ketidaksukaan, ketidaksetujuan, atau ketidakpuasan terhadap suatu kondisi [125]. Selain itu, saran dikategorikan sebagai bentuk umpan balik yang memuat harapan atau usulan perbaikan terhadap suatu proses maupun kualitas tertentu [126].

2.12 Pembagian Dataset

Proses pembagian dataset merupakan salah satu tahap penting dalam pengembangan model yang bertujuan untuk memisahkan data menjadi data pelatihan dan data pengujian agar model

dapat dievaluasi dengan baik [127]. Dalam perkembangannya, pembagian data tidak hanya dilakukan menjadi dua bagian, tetapi juga dapat dibagi menjadi tiga bagian yaitu data latih, data validasi, dan data uji untuk memperoleh evaluasi yang lebih menyeluruh [128]. Pada pembagian ini, data latih digunakan untuk melatih model dalam mengenali pola, sedangkan data validasi berfungsi untuk memantau performa selama proses pelatihan dan mencegah *overfitting* [128]. Sementara itu, data uji digunakan untuk mengevaluasi performa akhir model terhadap data yang belum pernah digunakan sebelumnya [128]. Dengan adanya pembagian tersebut, model dapat diuji menggunakan data baru sehingga hasilnya lebih mencerminkan kondisi nyata. Oleh karena itu, pembagian dataset menjadi langkah penting dalam memastikan kualitas model yang dihasilkan [129].

Dalam proses pembagian dataset, data biasanya diacak terlebih dahulu sebelum dilakukan pemisahan agar distribusi data tetap seimbang pada setiap bagian. Setelah proses pengacakan, pembagian data dilakukan menggunakan rasio tertentu sesuai kebutuhan penelitian [130]. Salah satu pendekatan yang digunakan adalah membagi data menjadi 80% data latih dan 20% data uji sebagai bentuk pembagian dasar, di mana data dipecah menjadi dua bagian dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian [130]. Selain itu, pembagian juga dapat dilakukan menjadi tiga bagian dengan rasio 80:10:10, di mana secara teknis proses pembagian dilakukan dalam dua langkah, yaitu pertama membagi dataset menjadi *training set* (80%) dan *subset* sementara (20%), kemudian *subset* tersebut dibagi kembali menjadi *validation set* (10%) dan *test set* (10%) [131]. Beberapa penelitian menunjukkan bahwa perbedaan rasio seperti 80:20, 70:30, dan 60:40 dapat mempengaruhi performa model yang dihasilkan. Hasil penelitian tersebut menunjukkan bahwa rasio 80:20 memberikan performa yang lebih baik dibandingkan beberapa rasio lainnya [132]. Sementara itu, penelitian lain menunjukkan bahwa rasio 80:10:10 dapat menghasilkan akurasi yang lebih tinggi dibandingkan rasio seperti 75:15:15 dan 60:20:20 [133].

Selain rasio pembagian, metode yang digunakan dalam proses pembagian dataset juga berperan penting dalam menjaga kualitas data [134]. Salah satu metode yang digunakan adalah *stratified sampling* untuk memastikan distribusi kelas tetap seimbang pada setiap *subset* data. Metode ini membantu menghindari ketidakseimbangan data yang dapat mempengaruhi hasil model [135]. Selain itu, proses pembagian data dilakukan secara acak dengan menggunakan parameter *random state* untuk menjaga konsistensi hasil. Beberapa penelitian menggunakan nilai *random state* seperti 10, 15, 20, 25, dan 42 untuk menghasilkan variasi pembagian data. Namun,

nilai 42 sering digunakan karena memberikan hasil yang konsisten dan dapat diulang dengan hasil yang sama [136]. Dengan demikian, penggunaan *random state* atau *random seed* membantu memastikan pembagian data dapat dijalankan kembali dengan hasil yang tetap sama sehingga mendukung proses evaluasi model yang lebih jelas dan teratur [137], [135].

2.13 Teknik Augmentasi

Teknik augmentasi merupakan suatu metode yang digunakan untuk meningkatkan jumlah data pelatihan dengan menghasilkan variasi tambahan dari data teks yang sudah ada tanpa mengubah makna aslinya. Pendekatan ini bertujuan untuk memperkaya jumlah sampel pada setiap kelas sentimen sehingga menghasilkan dataset yang lebih variatif [138]. Dengan demikian, teknik ini dapat membantu mengatasi permasalahan ketidakseimbangan kelas pada analisis sentimen [139]**.** Penerapan teknik augmentasi juga perlu memperhatikan urutan tahapan dalam pengolahan data, karena pelaksanaan augmentasi sebelum pembagian dataset berpotensi menyebabkan kebocoran data (*data leakage*), yang dikenal sebagai *late split* dalam literatur [140].

Dalam penerapannya, terdapat berbagai teknik augmentasi yang umum digunakan dalam pengolahan teks, antara lain *Back Translation*, *Random Swap*, dan *Synonym Replacement*. *Back Translation* merupakan augmentasi data yang menghasilkan teks baru dengan cara menerjemahkan teks asli ke dalam bahasa lain, kemudian menejemahkannya kembali ke bahasa Indonesia [141]. *Random swap* adalah teknik yang menghasilkan variasi teks dengan menukar posisi dua kata secara acak dalam suatu kalimat [142], sedangkan *Synonym Replacement* yaitu teknik yang dilakukan dengan mengganti kata tertentu dalam satu kalimat dengan kata lain yang memiliki makna serupa [141]. Contoh kalimat yang sudah dilakukan augmentasi dapat dilihat pada Tabel 2.2 [143], [144]:

Tabel II.2 contoh hasil *back translation*, *random swap*, dan *synonym replacement*

Teknik Augmentasi	Kalimat Asli	Hasil
Back Translation	***coursea*** *does not allow me to* ***quit*** *the class also i cannot do the homework or watch video at my own pace*	***kursa*** *does not allow me to* ***finish*** *the class also I can not do the homework or watch video at my own pace*
Random Swap	Putusan **mk sudah** jelas sah dan mengikat	Putusan **sudah** **mk** jelas sah mengikat
Synonym Replacement	*coursea does not allow me to* ***quit*** *the class also i cannot do the homework or watch* ***video*** *at my* ***own*** *pace*	*coursea does not allow me to* ***fall*** ***by the wayside*** *the class also i can not do the homework or watch* ***video*** ***recording*** *at my possess pace*

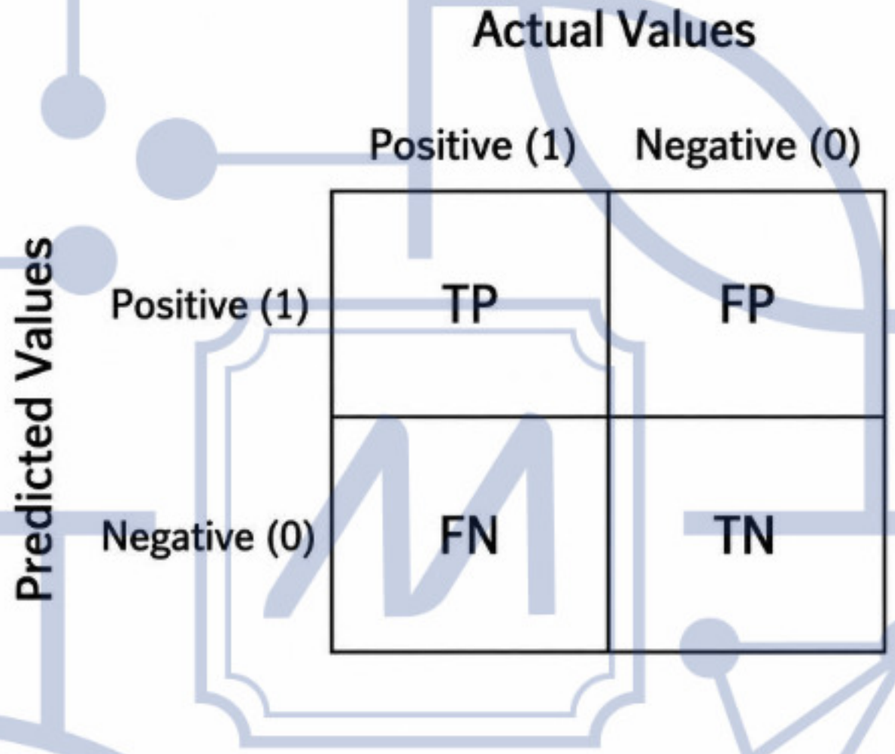
2.14 Evaluasi Kinerja Model

Evaluasi kinerja model merupakan tahapan krusial untuk mengukur sejauh mana kemampuan model IndoBERT dalam mengklasifikasikan sentimen pengguna TikTok terhadap program Makan Bergizi Gratis (MBG). Berdasarkan karakteristik data media sosial yang

digunakan, evaluasi dilakukan menggunakan beberapa metrik standar, yaitu *Confusion Matrix* dan Akurasi, Presisi, *Recall*, dan *F1-Score*.

2.14.1 Confusion Matrix

Confusion Matrix adalah teknik evaluasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang sebenarnya [145]. Matriks ini memberikan representasi tabular dari performa model, di mana baris merepresentasikan kelas aktual (sebenarnya) dan kolom merepresentasikan kelas prediksi [146].



Gambar II.4 Confusion Matrix [147]

Pada Gambar 2.4, *Confusion Matrix* menunjukkan struktur dasar dalam evaluasi klasifikasi. Penggunaan *Confusion Matrix* memungkinkan analisis yang lebih mendalam mengenai kesalahan prediksi model, tidak hanya sekadar mengetahui jumlah prediksi yang benar atau salah, tetapi juga jenis kesalahan yang terjadi antar kelas [148]. Komponen dasar yang terdapat dalam *Confusion Matrix* terdiri dari empat parameter utama, yaitu:

a. *True Positive* (TP): Jumlah data dari suatu kelas tertentu yang diprediksi dengan benar oleh model sebagai kelas tersebut.

b. *True Negative* (TN): Jumlah data yang bukan merupakan kelas tertentu dan diprediksi dengan benar oleh model sebagai bukan kelas tersebut.

c. *False Positive* (FP): Jumlah data yang diprediksi oleh model sebagai kelas tertentu, padahal sebenarnya bukan (kesalahan Tipe I).

d. *False Negative* (FN): Jumlah data yang sebenarnya merupakan kelas tertentu, tetapi diprediksi salah oleh model sebagai kelas lain (kesalahan Tipe II) [145], [149].

Analisis nilai-nilai pada komponen ini menjadi dasar perhitungan untuk metrik evaluasi selanjutnya.

2.14.2 Akurasi

Akurasi merupakan metrik evaluasi yang paling umum digunakan untuk mengukur kinerja model klasifikasi secara keseluruhan. Akurasi didefinisikan sebagai rasio antara jumlah prediksi yang benar (*True Positive* dan *True Negative*) terhadap total keseluruhan data yang diuji [150]. Metrik ini memberikan gambaran intuitif mengenai seberapa sering model IndoBERT membuat prediksi yang tepat [151]. Secara matematis, akurasi dapat dihitung menggunakan persamaan berikut [152]:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{.....(2.14)}$$

Keterangan:

TP: *True Positive*

TN: *True Negative*

FP: *False Positive*

FN: *False Negative*

Meskipun akurasi merupakan indikator kinerja yang penting, penelitian terdahulu menunjukkan bahwa penggunaan akurasi saja bisa menjadi bias, terutama pada dataset yang tidak seimbang (*imbalanced* dataset) seperti data komentar media sosial [153]. Dalam kasus data tidak seimbang, model mungkin memiliki akurasi tinggi hanya dengan memprediksi kelas mayoritas, namun gagal mengenali kelas minoritas dengan baik [154]. Oleh karena itu, pada penelitian ini, nilai akurasi akan dianalisis bersamaan dengan metrik lain seperti Presisi, *Recall*, dan *F1-Score* untuk mendapatkan evaluasi yang lebih komprehensif.

2.14.3 Presisi

Presisi merupakan rasio antara jumlah data positif yang di prediksi dengan benar (*true positive*) terhadap keseluruhan data positif yang sebenarnya. *Precision* suatu algoritma dapat diukur dengan membandingkan jumlah data yang terklasifikasi dengan benar (TP) dengan total

data yang telah diprediksi dengan benar (TP + FP). Dihitung dengan persamaan berikut [155], [156] :

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(2.15)}$$

Precision digunakan untuk mengevaluasi kemampuan model dalam mengidentifikasi data positif secara benar dari seluruh data yang di prediksi sebagai positif. Nilai *precision* yang tinggi menunjukkan jumlah kesalahan positif palsu (*false positive*) yang rendah [157].

2.14.4 Recall

Recall merupakan rasio antara jumlah data positif yang berhasil diprediksi dengan benar (*true positive*) terhadap keseluruhan data positif yang sebenarnya. Dihitung dengan persamaan berikut [155], [156]:

$$Recall = \frac{TP}{TP+FN} \quad \text{.....(2.16)}$$

recall digunakan untuk mengukur kemampuan model dalam mengenali seluruh data positif yang ada, sehingga meminimalkan *false negative* [157].

2.14.5 F1-Score

F1-*Score* merupakan nilai rata-rata gabungan dari *precision* dan *recall* untuk mengindikasi keseimbangan antara keduanya. Dihitung dengan persamaan berikut [155], [156]:

$$F1 - Score = \frac{Precision \times Recall}{Precision + Recall} \times 2 \quad \text{.....(2.17)}$$

F1-*Score* memberikan evaluasi yang seimbang antara *precision* dan *recall*, khususnya pada dataset yang tidak seimbang [157].

2.15 Streamlit

Streamlit merupakan *framework* berbasis *Python* yang digunakan untuk membangun aplikasi *web* interaktif secara cepat dan efisien, khususnya dalam bidang data *science* dan model pembelajaran mesin [158]. *Framework* ini dirancang untuk mempermudah proses penyajian model dan visualisasi data tanpa memerlukan kemampuan pengembangan *front-end* yang kompleks. *Streamlit* menyediakan berbagai komponen antarmuka pengguna seperti input teks, *slider*, *checkbox*, dan fitur unggah *file* yang memungkinkan interaksi langsung antara pengguna dan sistem [159]. Setiap interaksi yang dilakukan oleh pengguna akan menyebabkan *script Python* dijalankan kembali secara otomatis [159]. Selain itu, *Streamlit* juga bersifat gratis dan tidak

memerlukan kemampuan pengembangan *front-end* yang kompleks dalam penggunaannya. *Framework* ini mendukung penggunaan *Python* versi 3.7 ke atas dan dapat dijalankan melalui berbagai editor seperti *PyCharm* dan *Visual Studio Code*. Dengan kemudahan tersebut, *Streamlit* menjadi salah satu *framework* yang banyak digunakan dalam pengembangan aplikasi berbasis data [159].

Dalam penelitian terkait analisis sentimen pada media sosial, *Streamlit* terbukti efektif dalam mendukung integrasi model *Bidirectional Encoder Representations from Transformers* (BERT), yang memungkinkan proses klasifikasi teks dilakukan secara efisien dalam sebuah antarmuka berbasis *web* [160]. *Framework* ini juga sangat andal untuk membangun aplikasi yang melakukan analisis data secara langsung (*live data analysis*), di mana pengguna dapat memasukkan kata kunci tertentu untuk mendapatkan sentimen dari data yang ditarik secara *real-time* melalui API [161]. Selain untuk analisis teks, *Streamlit* juga digunakan secara luas dalam pengembangan sistem prediksi lingkungan, seperti pemantauan Indeks Kualitas Udara (AQI) dan prediksi kelayakan air minum, dengan mengintegrasikan algoritma seperti *XGBoost* dan *CatBoost* untuk memberikan hasil prediksi beserta skor kepercayaan (*confidence scores*) kepada pengguna [162], [163]. Lebih lanjut, dalam bidang pertanian presisi, dimanfaatkan untuk mengimplementasikan sistem rekomendasi tanaman berbasis metode *ensemble learning* tingkat lanjut (seperti *Stacking*), yang mempermudah interaksi pengguna dalam memasukkan data tanah dan memperoleh rekomendasi yang akurat secara interaktif [164].

Dalam pengembangan model pembelajaran mesin, *Streamlit* digunakan untuk mempermudah proses *deployment* model ke dalam bentuk aplikasi *web* yang interaktif [158]. Model yang telah dilatih dapat disimpan dalam format *file* seperti pkl menggunakan modul *pickle* sehingga dapat digunakan kembali tanpa perlu pelatihan ulang [165]. Selain itu, model juga dapat disimpan menggunakan format *safetensors*, yang digunakan untuk menyimpan bobot model secara lebih efisien dan aman dalam proses implementasi [166]. Selain itu, dataset juga dapat disimpan untuk menjaga konsistensi data dalam pengembangan sistem [165]. *Streamlit* tidak memerlukan *template* tambahan karena seluruh tampilan aplikasi dapat ditulis dalam satu *file Python* [167]. Pengguna dapat memasukkan data melalui antarmuka yang tersedia dan memperoleh hasil prediksi secara langsung [168]. Selain itu, *Streamlit* menyediakan antarmuka yang sederhana sehingga memudahkan pengguna dalam berinteraksi dengan sistem [169]. Oleh karena itu, *Streamlit* dapat digunakan untuk menyajikan model ke dalam aplikasi yang mudah diakses oleh pengguna [170].

2.16 Metode Cross Industry Standard Process For Data Mining (CRISP-DM)

CRISP-DM merupakan salah satu kerangka kerja yang banyak digunakan dalam proses data *mining* dan analisis data karena menyediakan alur kerja yang sistematis dan terstruktur. Metode ini digunakan sebagai pedoman dalam mengelola seluruh tahapan analisis data mulai dari pemahaman masalah hingga evaluasi hasil yang diperoleh [16]. *CRISP-DM* memiliki sifat fleksibel dan iteratif sehingga setiap tahapan dapat disesuaikan dengan kebutuhan penelitian [171]. Selain itu, metode ini telah terbukti efektif dalam berbagai penelitian analisis data, termasuk analisis sentimen [172]. Kerangka kerja ini tidak hanya digunakan sebagai panduan teknis, tetapi juga membantu menghasilkan analisis yang lebih relevan dan bernilai bagi pengguna [173]. Oleh karena itu, *CRISP-DM* banyak digunakan dalam berbagai penelitian berbasis data karena kemampuannya dalam menyusun proses analisis secara terarah [173]. Dengan karakteristik tersebut, metode ini dinilai sesuai untuk digunakan dalam penelitian analisis sentimen berbasis data teks [172].

CRISP-DM menekankan pemahaman permasalahan pada tahap awal sebelum masuk ke proses eksplorasi dan pemodelan data, sehingga alur analisis menjadi lebih terarah dan sesuai dengan tujuan yang di tetapkan [174]. Selain itu, *framework* ini merupakan salah satu metode yang paling dominan digunakan dalam data *mining* dengan tingkat penggunaan yang lebih tinggi dibandingkan metode lainnya, serta dikenal memiliki fleksibilitas dalam penerapannya di berbagai bidang penelitian. Hal ini menunjukkan bahwa CRISP-DM telah menjadi standar dalam banyak proyek analisis data baik di lingkungan akademik maupun industri [17].

CRISP-DM terdiri dari enam tahapan utama yang saling berkaitan dalam proses analisis data, yaitu [172]:

1. *Business Understanding*

   Pada *Business Understanding* merupakan tahap awal yang bertujuan untuk memahami permasalahan serta menetapkan tujuan dari proses analisis data yang akan dilakukan. Pada tahap ini, peneliti menentukan apa yang ingin dianalisis serta hasil yang diharapkan dari penelitian. Selain itu, ditentukan juga ruang lingkup penelitian agar proses analisis lebih terarah. Tahap ini sangat penting karena menjadi dasar dalam menentukan langkah selanjutnya [172].

2. *Data Understanding*

Tahap *Data Understanding* dilakukan untuk memahami data yang akan digunakan dalam penelitian. Pada tahap ini, data dikumpulkan dan dianalisis untuk mengetahui karakteristik serta kualitasnya. Peneliti juga melakukan pengecekan awal untuk menemukan kemungkinan masalah pada data, seperti data yang tidak lengkap atau tidak sesuai. Dengan memahami data, proses selanjutnya dapat dilakukan dengan lebih baik [172].

3. *Data Preparation*

Tahap *Data Preparation* merupakan proses menyiapkan data sebelum digunakan dalam pemodelan. Proses ini meliputi pembersihan data, penghapusan data yang tidak diperlukan, serta perbaikan format data. Selain itu, dilakukan juga *preprocessing* seperti tokenisasi dan penghapusan kata yang tidak penting. Tujuan dari tahap ini adalah agar data menjadi lebih bersih dan siap digunakan dalam analisis [172].

4. *Modeling*

Tahap *Modeling* adalah proses membangun model menggunakan metode atau algoritma tertentu. Data yang telah diproses digunakan untuk melatih model agar dapat mengenali pola dalam data. Pemilihan algoritma disesuaikan dengan kebutuhan penelitian. Hasil dari tahap ini berupa model yang dapat digunakan untuk melakukan klasifikasi atau prediksi [172].

5. *Evaluation*

Tahap *Evaluation* dilakukan untuk menilai kinerja model yang telah dibuat. Penilaian dilakukan menggunakan beberapa metrik untuk melihat seberapa baik model dalam melakukan prediksi. Hasil evaluasi ini digunakan untuk mengetahui apakah model sudah sesuai dengan tujuan penelitian. Jika hasilnya belum optimal, maka model dapat diperbaiki kembali [172].

6. *Deployment*

Tahap *Deployment* merupakan tahap akhir dalam proses analisis data. Pada tahap ini, model yang telah dibuat digunakan untuk menganalisis data baru atau menghasilkan informasi. Hasil analisis dapat disajikan dalam bentuk laporan atau visualisasi. Tujuan dari tahap ini adalah agar hasil penelitian dapat digunakan dan memberikan manfaat [172].

2.17 Analisis Kebutuhan Non-Fungsional Menggunakan Kerangka Kerja PIECES

Metode PIECES merupakan kerangka kerja yang digunakan untuk menganalisis dan mengevaluasi sistem informasi berdasarkan enam dimensi utama, yaitu *Performance*, *Information*,

Economics, *Control and Security*, *Efficiency*, dan *Service*, sehingga diperoleh gambaran menyeluruh mengenai kondisi sistem yang dapat dijadikan dasar untuk rekomendasi perbaikan dan pengembangan sistem [175]. Penggunaan metode ini membantu peneliti mengidentifikasi celah teknis serta peluang pengembangan sistem yang lebih reliabel pada tahap analisis kebutuhan [176]. Penjabaran dimensi PIECES dalam konteks analisis kebutuhan non-fungsional untuk sistem analisis sentimen ini adalah sebagai berikut:

1. *Performance* (Kinerja)

   Dimensi ini mengukur seberapa baik sistem menyelesaikan tugas dalam batasan waktu tertentu dengan indikator utama berupa *throughput* dan *response time* [177]. Dalam sistem ini, kinerja dinilai berdasarkan kecepatan algoritma IndoBERT dalam memproses teks komentar TikTok hingga menghasilkan label prediksi sentimen yang tepat bagi pengguna [178].

2. *Information* (Informasi)

   Aspek ini berfokus pada kualitas luaran informasi agar bersifat akurat, relevan, dan mudah dipahami oleh pemangku kepentingan. Kriteria kualitas informasi mencakup keakuratan hasil klasifikasi sentimen serta kelengkapan penyajian data statistik dalam bentuk diagram lingkaran yang informatif pada antarmuka sistem [178].

3. *Economics* (Ekonomi)

   Dimensi ekonomi menilai kelayakan sistem dari perspektif biaya dan manfaat serta pemanfaatan sumber daya komputasi secara efektif. Aspek ini ditinjau dari optimalisasi penggunaan layanan *cloud* untuk meminimalkan biaya operasional saat *prototype* sistem dijalankan untuk memproses dataset dalam jumlah besar secara efisien [179].

4. *Control / Security (Kendali* / Keamanan)

   Aspek ini menitikberatkan pada perlindungan data serta keandalan sistem terhadap kegagalan teknis atau akses yang tidak sah [180]. Kebutuhan pada dimensi ini mengharuskan sistem memiliki tingkat reliabilitas tinggi guna mencegah terjadinya *error* saat memproses *input* teks yang tidak standar atau mengandung karakter khusus dari media sosial [181].

5. *Efficiency* (Efisiensi)

   Efisiensi mengukur pemanfaatan sumber daya mesin secara optimal, seperti penggunaan memori dan kapasitas CPU, selama proses eksekusi model berlangsung. Sebuah sistem dikatakan efisien apabila mampu mencapai hasil analisis sentimen yang diinginkan tanpa memerlukan langkah atau upaya komputasi yang berlebihan dari sisi perangkat keras [182].

6. *Service* (Layanan)

   Dimensi ini berkaitan dengan kualitas layanan dan kemudahan penggunaan (*usability*) antarmuka bagi para penggunanya [180]. Kebutuhan layanan menuntut sistem memiliki navigasi yang intuitif serta tampilan antarmuka *Streamlit* yang ramah pengguna guna memudahkan interaksi dan pembacaan hasil analisis secara nyata [182].

2.18 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai landasan untuk memahami perkembangan metode dan temuan terkait analisis sentimen serta penggunaan model berbasis *transformer* seperti IndoBERT. Ringkasan beberapa peneliti relevan disajikan pada Tabel 2.3 berikut:

Tabel II.3 Ringkasan Penelitian Terdahulu

Penulis	**Tahun**	**Data**	**Metode**	**Hasil**	**Keterbatasan**
Tyas dkk. [10]	2025	Data X terkait program makan bergizi gratis (3.312 data)	IndoBERT + *lexicon - based*	Akurasi 85%; F1-*score* 0,85 (*macro & weighted*); *precision*: positif 0,95, netral 0,78; *recall* netral 0,87; distribusi: netral 39,7%, negatif 36,3%, positif 24,0%	Kelas netral sering *overlap* dengan kelas lain sehingga performa kurang optimal
Aryanti dan Suria [11]	2025	Twitter (36.597 tweet, isu PHK)	IndoBERT, SVM, *Random Forest*, *Decision Tree*	IndoBERT (89,6%), SVM 88%, *Random Forest* 78,04%, *Decision Tree* 70,40%	Kelas netral sulit diklasifikasi, data tidak seimbang, konteks teks pendek menyebabkan ambiguitas

Tabel II.3 Ringkasan Penelitian Terdahulu (sambungan)

Penulis	**Tahun**	**Data**	**Metode**	**Hasil**	**Keterbatasan**
Suryaningtyas dkk. [12]	2025	TikTok (1.975 komentar)	IndoBERTa dan Random Forest	Akurasi 82%, performa baik pada sentimen negatif	Data tidak seimbang, performa rendah pada kelas positif, variasi bahasa komentar mempengaruhi model
Marchelo dkk. [14]	2025	Dataset multi-domain media sosial (900 data uji)	IndoBERT + *fine-tuning* + augmentasi (*Synonym Replacement* & *Back Translation*)	*Baseline* F1 0,3523; *fine-tuning* 0,5994; SR terbaik (akurasi 69,67%, F1 0,6714); BT (akurasi 68,78%, F1 0,6676) lebih seimbang pada kelas netral (F1 0,2593)	Model *baseline* gagal pada netral (F1 0,0206); SR menurunkan performa netral; BT berpotensi *overfitting* akibat *noise* terjemahan
Wang dan Xiang [15]	2024	SST-5 (8.544), SST-2 (6.920), IMDB (20.000)	*Antonym Replacement* & *Random Swap* (BERT, LSTM)	Augmentasi meningkatkan performa; *Random* Swap terbaik (SST-2: 92,25%, IMDB: 92,52%)	Dataset terbatas, *random swap* berpotensi mengubah makna

2.19 Tools dan Library yang Digunakan

Penelitian ini menggunakan beberapa *library* berbasis bahasa pemrograman *Python* untuk mendukung tahapan pengolahan data, pemodelan, hingga evaluasi. *Python* dipilih karena memiliki ekosistem pustaka yang komprehensif untuk mendukung riset sains data dan *machine learning* [183]. Berikut adalah pustaka yang digunakan dalam penelitian ini:

1. Pandas – Digunakan untuk membaca, mengolah, dan memanipulasi dataset dalam bentuk *DataFrame*. Pustaka ini sangat efisien dalam menangani data tabular berskala besar pada tahap pra-pemrosesan [183].
2. NumPy – Digunakan untuk operasi numerik dan manipulasi *array*/matriks multidimensi, yang merupakan struktur data fundamental dalam komputasi *machine learning* [183].
3. NLTK (*Natural Language Toolkit*) – Digunakan untuk proses *preprocessing* teks tingkat dasar, seperti tokenisasi dan penghapusan *stopword* (kata-kata umum yang tidak bermakna spesifik) [184].
4. *Sastrawi* – Pustaka yang dirancang khusus untuk bahasa Indonesia, digunakan untuk proses *stemming*, yaitu mengembalikan kata yang memiliki imbuhan menjadi bentuk kata dasarnya [184], [185].
5. Regular *Expression* (re) – Pustaka bawaan *Python* yang digunakan untuk pembersihan teks (*text cleaning*) tingkat lanjut, seperti menghapus emoji, simbol, tautan, dan karakter khusus melalui pencocokan pola *string* [186].
6. *Matplotlib* – Pustaka dasar untuk visualisasi data di *Python* yang digunakan untuk merepresentasikan data dalam bentuk grafik statis, seperti grafik distribusi kelas sentiment [187].
7. *Seaborn* – Pustaka visualisasi yang dibangun di atas *Matplotlib*. Pustaka ini digunakan untuk menghasilkan grafik statistik yang lebih informatif dan estetis, seperti visualisasi *confusion matrix* [187].
8. WordCloud – Digunakan untuk memvisualisasikan frekuensi kata dalam bentuk awan kata (*word cloud*), sehingga kata yang paling sering muncul dalam komentar pengguna akan ditampilkan lebih besar [188].

9. *Scikit-learn* (*sklearn*) – Pustaka utama untuk *machine learning* tradisional yang digunakan untuk pembagian dataset (*train, validation, test*), *label encoding*, serta perhitungan metrik evaluasi seperti *accuracy, precision, recall*, dan *F1-score* [189].
10. *PyTorch* (*torch*) – *Framework deep learning* berbasis tensor yang digunakan sebagai infrastruktur dasar untuk membangun, melatih, dan mengoptimalkan model *neural network* dengan dukungan komputasi GPU [190].
11. *HuggingFace Transformers* – Pustaka *open-source* mutakhir yang digunakan untuk mengimplementasikan arsitektur model IndoBERT (*indobenchmark/indobert-base-p2*) dalam proses *fine-tuning* klasifikasi teks [191].
12. *HuggingFace* Datasets – Digunakan untuk pengelolaan dan pemuatan dataset secara efisien dalam format yang langsung kompatibel dengan model-model berbasis arsitektur *transformer* [191].
13. *HuggingFace Evaluate* – Pustaka terstandardisasi yang digunakan untuk menghitung dan memvalidasi metrik evaluasi model secara konsisten pada ekosistem *Hugging Face* [192].
14. TQDM – Digunakan untuk menampilkan *progress bar* yang interaktif selama iterasi proses *training* dan *evaluation* model berlangsung, sehingga mempermudah pemantauan estimasi waktu [183].