

BAB III

METODOLOGI PENELITIAN

3.1 Jenis dan Fokus Penelitian

3.1.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan penelitian eksperimental kuantitatif (*experimental quantitative research*). Karakteristik penelitian kuantitatif diterapkan karena seluruh luaran (*output*) dari proses klasifikasi sentimen ini diukur dan disajikan dalam bentuk metrik numerik statistik yang bersifat objektif, yaitu nilai Akurasi (*Accuracy*), Presisi (*Precision*), *Recall*, dan *F1-Score*. Sifat eksperimental dalam penelitian ini diwujudkan melalui serangkaian tahapan pengujian dan rekayasa arsitektur model. Eksperimen dilakukan dengan membangun arsitektur hibrid yang menggabungkan model bahasa berbasis *Transformer* yaitu IndoBERT dengan lapisan jaringan saraf konvolusi satu dimensi (*1D-Convolutional Neural Network*). Selanjutnya, kontribusi ilmiah dari penambahan lapisan konvolusi tersebut idealnya dapat dibuktikan melalui metodologi studi ablasi (*ablation study*), yaitu sebuah eksperimen terstruktur untuk membandingkan kinerja model hibrid usulan secara langsung dengan model *baseline* (IndoBERT murni) di bawah kondisi dataset yang sama. Namun demikian, perbandingan tersebut belum dilakukan pada penelitian ini dan diusulkan sebagai arah pengembangan bagi penelitian selanjutnya (lihat sub-bab 5.2).

3.1.2 Fokus Penelitian

Fokus utama dalam penelitian ini diarahkan pada klasifikasi sentimen tingkat kalimat (*sentence-level sentiment analysis*). Secara spesifik, penelitian ini berfokus pada penyelesaian kendala pemrosesan data teks tidak terstruktur (*unstructured data*) berupa ulasan pengguna (*user reviews*) aplikasi TikTok yang diunduh dari platform Google Play Store. Tantangan linguistik utama yang menjadi pusat perhatian dalam riset ini adalah tingginya tingkat variasi bahasa informal yang digunakan oleh netizen, seperti penggunaan kosakata gaul (*slang words*), penyingkatan kata yang tidak beraturan, *noise* berupa simbol/emoji, hingga struktur kalimat yang tidak memenuhi kaidah tata bahasa Indonesia yang baku. Penelitian ini berfokus pada bagaimana arsitektur model yang dirancang mampu menangkap fitur kontekstual global sekaligus

mengekstraksi fitur pola frasa lokal (*n-gram*) secara optimal dari teks informal tersebut guna menghasilkan klasifikasi polaritas sentimen (Positif, Negatif, dan Netral) yang presisi.

3.2 Sumber Data Penelitian

Data yang digunakan dalam penelitian ini merupakan data sekunder (*secondary data*) berupa teks opini digital. Karakteristik data sekunder melekat pada penelitian ini karena peneliti tidak melakukan pengambilan data secara langsung dari subjek manusia melalui instrumen kuesioner atau wawancara, melainkan memanfaatkan data publik yang telah tersedia dan terdokumentasi secara terbuka pada repositori platform Google Play Store. Proses pengumpulan data dilakukan secara otomatis melalui teknik ekstraksi data web (*web scraping*) menggunakan skrip berbasis bahasa pemrograman Python dengan memanfaatkan pustaka (*library*) pihak ketiga bernama `google-play-scraper`. Pustaka ini digunakan karena mampu melakukan interaksi langsung dengan *Application Programming Interface* (API) internal Google Play Store tanpa memerlukan konfigurasi *web driver* eksternal, sehingga menjamin efisiensi dan integritas data yang ditarik. Untuk memastikan relevansi dan keabsahan dataset terhadap ruang lingkup eksperimen yang dilakukan, ditetapkan tiga ketentuan batasan (*constraints*) pengumpulan data sebagai berikut:

1. Batasan Objek Data Objek data dibatasi khusus pada teks ulasan (*review*) tertulis yang menggunakan bahasa Indonesia oleh pengguna aplikasi TikTok. Batasan geografis dan linguistik ini diterapkan secara sengaja agar selaras dengan kemampuan model bahasa yang digunakan sebagai *backbone* utama sistem, yaitu IndoBERT, yang telah di-*pre-train* menggunakan korpus bahasa Indonesia. Ulasan yang ditulis sepenuhnya dalam bahasa asing atau hanya berupa rangkaian karakter acak (*noise*) akan dieliminasi pada tahapan pembersihan data.
2. Atribut Data yang Diambil Untuk menjaga kerahasiaan informasi dan privasi pengguna (*data privacy*), proses *scraping* tidak merekam identitas pribadi seperti nama akun atau foto profil. Sistem hanya mengekstraksi tiga atribut krusial yang diperlukan untuk kebutuhan analisis, yaitu:
 - a. Teks Ulasan (*Content*): Barisan kalimat opini yang ditulis oleh pengguna aplikasi, yang menjadi objek utama ekstraksi fitur sentimen.

- b. Skor Bintang (*Score*): Nilai numerik skala Likert dari 1 hingga 5 yang diberikan oleh pengguna. Atribut ini berfungsi sebagai parameter *ground truth* atau acuan awal dalam proses pelabelan kelas sentimen.
 - c. Tanggal Ulasan (*Date*): Informasi waktu pengiriman ulasan untuk memastikan pemenuhan syarat linearitas rentang waktu penelitian.
3. Rentang Waktu Penarikan Data Pengambilan data dibatasi dalam jendela waktu yang terukur secara spesifik, yaitu ulasan yang dikirimkan oleh pengguna dalam rentang waktu Januari 2026 sampai dengan Juni 2026. Pembatasan temporal ini diterapkan agar dataset yang diperoleh mencerminkan persepsi pengguna yang paling aktual terhadap performa dan pembaruan fitur (*update*) aplikasi TikTok terkini, sekaligus mengontrol volume dataset agar berada pada kapasitas komputasi yang ideal untuk fase pelatihan *Deep Learning*.

3.3 Teknik Pelabelan Data (*Distant Supervision*)

Mengingat dataset yang digunakan tidak memiliki label sentimen secara eksplisit dari sumbernya, penelitian ini menerapkan teknik **pelabelan otomatis (*Distant Supervision*)**, yaitu metode pemberian label kelas menggunakan sinyal/informasi lain yang berkorelasi kuat dengan sentimen, tanpa memerlukan anotasi manual pada setiap data. Dalam konteks penelitian ini, skor bintang (*rating*) yang diberikan pengguna dijadikan sebagai proksi (*proxy*) untuk menentukan label sentimen, dengan aturan pemetaan sebagai berikut:

1. Bintang 4 dan 5 → **Positif**
2. Bintang 3 → **Netral**
3. Bintang 1 dan 2 → **Negatif**

Pendekatan ini dipilih karena efisien untuk dataset berskala besar (5.000 data) dan telah banyak digunakan pada penelitian sejenis (lihat 1.7). Namun demikian, teknik ini memiliki keterbatasan, yaitu rating bintang tidak selalu secara konsisten merepresentasikan sentimen teks ulasannya secara akurat, khususnya pada rating 3 yang bersifat ambigu.

3.4 Alat dan Bahan Penelitian

Implementasi arsitektur hibrid *Deep Learning* yang melibatkan model bahasa berparameter besar memerlukan dukungan infrastruktur komputasi yang memadai. Oleh karena itu, penelitian ini memanfaatkan kombinasi lingkungan kerja lokal dan komputasi awan (*cloud computing*) dengan spesifikasi sebagai berikut:

3.4.1 Perangkat Keras (*Hardware*)

Komponen perangkat keras dalam penelitian ini dibagi menjadi dua kategori berdasarkan fungsi operasionalnya, yaitu untuk aktivitas pengembangan lokal serta untuk fase komputasi berat (pelatihan model):

1. Perangkat Komputasi Lokal (Laptop): Digunakan untuk penyusunan laporan ilmiah, analisis data awal, visualisasi metrik, serta penulisan draf kode program dasar. Spesifikasi perangkat keras lokal yang digunakan adalah:
 - a. **Model Perangkat:** Lenovo ThinkPad L13 Yoga
 - b. **Prosesor:** Intel® Core™ i5-10310U CPU @ 1.70GHz
 - c. **Memori Utama (RAM):** 16 GB
 - d. **Media Penyimpanan:** *Solid State Drive (SSD)*
2. Perangkat Infrastruktur Awan (*Cloud Hardware Acceleration*): Karena pelatihan model IndoBERT dan 1D-CNN membutuhkan operasi matriks tensor tingkat tinggi yang tidak efisien jika dijalankan pada prosesor (CPU) lokal, fase pelatihan dan pengujian model dialihkan ke infrastruktur Google Colab yang menyediakan akselerator grafis komputasi berupa Nvidia T4 GPU dengan 16 GB VRAM. Akselerator ini bertugas mempercepat waktu komputasi pembaruan bobot (*weight updating*) secara paralel selama proses *epoch* berlangsung.

3.4.2 Perangkat Lunak (*Software*)

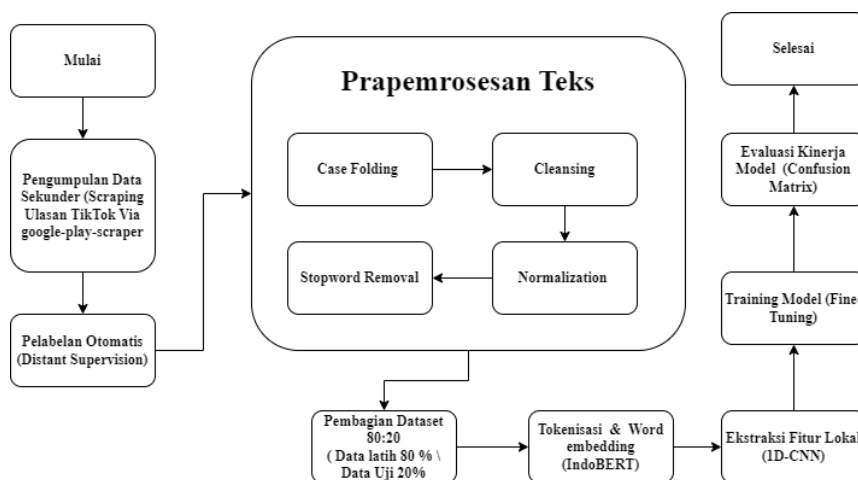
Selain perangkat keras, penelitian ini juga memanfaatkan sejumlah perangkat lunak dan pustaka (*library*) pemrograman sebagai berikut:

1. Bahasa Pemrograman: Python 3, digunakan sebagai bahasa utama untuk seluruh proses pengambilan data, prapemrosesan, pelatihan model, hingga evaluasi.

2. Lingkungan Pengembangan (*IDE*): Google Colaboratory (Google Colab), digunakan sebagai lingkungan eksekusi kode berbasis *notebook* yang terintegrasi dengan akselerator GPU.
3. Pustaka Pengambilan Data: *google-play-scraper*, digunakan untuk melakukan ekstraksi ulasan aplikasi dari Google Play Store.
4. Pustaka Prapemrosesan Teks: *Sastrawi*, digunakan pada tahap *stopword removal* Bahasa Indonesia.
5. Pustaka Deep Learning: *PyTorch* dan *Transformers* (HuggingFace), digunakan untuk memuat model *pre-trained* IndoBERT, membangun arsitektur hibrid IndoBERT + 1D-CNN, serta menjalankan proses pelatihan dan evaluasi model.
6. Pustaka Pendukung Analisis Data: *Pandas*, *NumPy*, dan *Scikit-learn*, digunakan untuk manipulasi data, perhitungan numerik, serta perhitungan metrik evaluasi (*Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix*)

3.5 Desain Sistem Penelitian

Desain sistem menggambarkan arsitektur bagaimana data mengalir, ditransformasikan, hingga menghasilkan keputusan klasifikasi akhir. Blok diagram alur pemrosesan data pada model hibrid ini dirancang secara sistematis untuk menggambarkan tahapan rekayasa data dari kondisi mentah hingga menjadi metrik evaluasi.



Gambar 3.5 Alur Diagram Penelitian

3.5.1 Penjelasan Alur Sistem Penelitian

Untuk memberikan gambaran yang lebih detail mengenai diagram alir di atas, berikut adalah deskripsi teknis dari setiap tahapan yang dijalankan di dalam sistem:

1. Mulai (*Start*): Tahap awal inisiasi penelitian yang meliputi perumusan masalah, studi literatur mengenai arsitektur NLP, serta penentuan parameter dasar eksperimen.
2. Pengumpulan Data Sekunder: Proses penarikan data mentah (*raw data*) ulasan aplikasi TikTok berbahasa Indonesia dari Google Play Store menggunakan skrip Python. Output dari tahap ini berupa dataset berformat tabel data (*data frame*).
3. Pelabelan Otomatis (Distant Supervision): Data mentah hasil scraping diberi label sentimen secara otomatis berdasarkan skor rating (bintang 4-5 = Positif, bintang 3 = Netral, bintang 1-2 = Negatif), sesuai teknik yang dijelaskan pada sub-bab 3.3, sebelum data masuk ke tahap prapemrosesan teks.
4. Prapemrosesan Teks (*Text Preprocessing*): Rangkaian pipa pembersihan teks untuk mengubah ulasan media sosial yang kotor menjadi representasi kata yang standar. Tahap ini meliputi mengubah huruf menjadi kecil (*case folding*), menghapus karakter non-alfabet menggunakan RegEx (*cleansing*), memperbaiki kata gaul/singkatan (*normalization*), serta membuang kata hubung tidak bermakna (*stopword removal*).
5. Pembagian Dataset: Memisahkan dataset bersih menggunakan metode pemisahan *holdout* dengan rasio tetap, yaitu 80% untuk data latih (*training set*) yang digunakan untuk memperbarui bobot neural network, dan 20% untuk data uji (*testing set*) sebagai data independen untuk pengujian generalisasi model.
6. Tokenisasi & *Word Embedding* (IndoBERT): Teks bersih dipecah menjadi unit-unit token (*sub-words*) berbasis algoritma *WordPiece*, lalu dilewatkan ke lapisan *Embedding* IndoBERT untuk ditransformasikan menjadi matriks padat (*dense vector*) berdimensi 768 yang kaya akan konteks semantik bahasa.
7. Ekstraksi Fitur Lokal 1D-CNN: Matriks *embedding* dari IndoBERT dialirkan ke lapisan *1D-Convolutional Neural Network*. Operasi konvolusi dengan ukuran *kernel* tertentu akan mengekstrak fitur frasa lokal (*n-gram*), yang kemudian direduksi dimensinya oleh lapisan *Global Max Pooling* untuk mengambil fitur sentimen paling dominan.

8. Training Model: Model hibrid dilatih menggunakan optimizer AdamW dengan learning rate 2×10^{-5} dan fungsi kerugian Cross-Entropy Loss selama beberapa epoch, sesuai konfigurasi yang dijelaskan pada sub-bab 3.9, guna memperbarui bobot IndoBERT dan CNN secara bersamaan (end-to-end fine-tuning).
9. Evaluasi Kinerja: Menguji model akhir menggunakan data uji 20%. Hasil prediksi model dikonversi menjadi tabel *Confusion Matrix* untuk menghitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* guna mengukur performa model hibrid usulan secara menyeluruh.
10. Selesai (*End*): Seluruh proses eksperimen telah selesai, hasil metrik dicatat, dan sistem siap menyajikan analisis pembahasan untuk penarikan kesimpulan.

3.6 Tahapan Prapemrosesan Teks (*Text Preprocessing Pipeline*)

Tahap prapemrosesan teks merupakan rangkaian pipa pembersihan data (*data cleaning pipeline*) yang bertujuan untuk menyelaraskan karakteristik teks ulasan mentah (*raw text*) menjadi format yang bersih dan standar. Tahapan ini sangat krusial dalam analisis sentimen media sosial untuk mereduksi keanekaragaman bentuk kata (*noise reduction*) sehingga mempercepat konvergensi model saat fase pelatihan. Proses pembersihan data ini diimplementasikan menggunakan bahasa pemrograman Python dan dieksekusi secara berurutan melalui sub-bab berikut:

3.6.1 Case Folding

Case folding adalah tahap paling awal yang berfungsi untuk menyeragamkan seluruh karakter huruf di dalam dataset ulasan menjadi huruf kecil (*lowercase*). Di dalam kode program Python, proses ini dieksekusi menggunakan fungsi bawaan teks yaitu `.lower()`. Tujuan utama dari tahapan ini adalah untuk menghilangkan sifat *case-sensitive* pada komputer. Tanpa adanya proses *case folding*, sistem akan menganggap kata "Bagus", "bagus", dan "BAGUS" sebagai tiga token yang berbeda karena memiliki representasi nilai ASCII yang tidak sama. Dengan mengubahnya menjadi bentuk seragam ("bagus"), beban komputasi representasi kata pada kamus kosakata model IndoBERT dapat diefisiensikan secara signifikan.

3.6.2 Cleansing

Tahap *cleansing* bertujuan untuk mengeliminasi komponen-komponen non-kontekstual yang tidak membawa bobot informasi sentimen di dalam teks ulasan.

Proses ini diimplementasikan secara teknis menggunakan teknik pencocokan pola *Regular Expression* (RegEx) melalui pustaka bawaan Python `re`.

Beberapa pola *noise* yang dihapus dari teks ulasan meliputi:

1. Tautan URL dan Modul Web: Dihapus menggunakan pola RegEx `http\S+|www\S+` karena alamat situs tidak merepresentasikan opini pengguna.
2. Angka dan Tanda Baca: Karakter numerik serta simbol tanda baca (seperti `.`, `,`, `!`, `?`, `*`, `(`, `)`) dihapus menggunakan pola `[^a-zA-Z\s]` untuk menyisakan karakter alfabet saja.
3. Emoji dan Karakter Unik: Rangkaian emoji atau simbol grafis dibersihkan agar tidak menimbulkan kegagalan *encoding* karakter saat dibaca oleh pustaka pemroses teks.

3.6.3 Normalization

Ulasan pengguna aplikasi TikTok pada platform Google Play Store didominasi oleh bahasa informal yang tidak baku, penuh dengan singkatan (misalnya "bgt", "dgn", "jg"), kata serapan gaul ("ngelag", "gabut"), hingga penulisan kata berulang karena ekspresi emosi ("lemotttt"). Tahap *normalization* (normalisasi) dirancang khusus untuk mentransformasikan kata-kata tidak baku tersebut menjadi bentuk kamus bahasa Indonesia yang formal. Secara teknis, proses ini memanfaatkan kamus normalisasi khusus (*slang dictionary*) yang disusun oleh peneliti dalam format pasangan kunci dan nilai (*key-value pair* / tipe data *dictionary* pada Python). Kode program akan memindai setiap kata di dalam kalimat ulasan secara urut (*token-by-token matching*). Jika suatu kata ditemukan cocok dengan salah satu kunci (*key*) di dalam kamus gaul, sistem secara otomatis akan menggantinya (*replace*) menjadi padanan kata bakunya (*value*). Proses ini menjamin agar transfer pengetahuan kontekstual dari model pra-latih IndoBERT dapat berjalan maksimal karena kata yang dimasukkan sudah sesuai dengan basis pengetahuan bahasa formalnya.

3.6.4 Stopword Removal

Stopword removal adalah proses pengeliminasian kata-kata umum yang memiliki frekuensi kemunculan sangat tinggi di dalam struktur tata bahasa namun tidak mengandung nilai informasi atau muatan emosi sentimen yang spesifik. Contoh kata-kata ini meliputi kata depan, kata hubung, dan kata ganti seperti "yang", "di", "dan",

"dari", "ke", "adalah", atau "ini". Di dalam kode program, proses penyaringan ini diimplementasikan dengan memanfaatkan daftar kata hubung bawaan dari pustaka **Sastrawi** (pustaka NLP khusus untuk bahasa Indonesia) yang dikombinasikan dengan beberapa kata tambahan buatan peneliti yang dinilai tidak relevan terhadap konteks ulasan aplikasi. Setiap token kalimat ulasan akan dicocokkan dengan daftar *stopword* tersebut; jika token tersebut terdaftar sebagai *stopword*, maka token tersebut akan dibuang dari kalimat. Hasil akhir dari pipa prapemrosesan ini adalah untai kalimat yang bersih, padat, dan kaya akan kata kunci sentimen inti yang siap dialirkan menuju fase pemisahan data serta pemodelan.

3.7 Skenario Pembagian Data (*Holdout Method*)

Setelah dataset melalui pipa prapemrosesan teks dan berada dalam kondisi bersih, tahapan selanjutnya adalah melakukan pembagian data (*data splitting*). Pembagian data ini bertujuan untuk memisahkan data eksperimen menjadi bagian terisolasi agar evaluasi performa model terhindar dari permasalahan *data leakage* (kebocoran data) dan dapat diukur tingkat generalisasinya secara objektif. Penelitian ini menerapkan skema Metode Holdout (*Holdout Method*), di mana dataset bersih dipecah menjadi dua bagian independen dengan rasio proporsi 80% untuk data latih (*training set*) dan 20% untuk data uji (*testing set*). Fungsi penugasan teknis dari masing-masing bagian data diatur sebagai berikut:

1. Data Latih (*Training Set* - 80%): Digunakan secara penuh untuk melatih arsitektur gabungan model IndoBERT dan 1D-CNN. Pada subset data inilah proses *fine-tuning* terjadi, di mana model mempelajari representasi kontekstual, mengekstraksi fitur pola frasa bersentimen, serta memperbarui nilai bobot (*weight*) internal jaringan saraf tiruan melalui kalkulasi rugi (*loss function*) dan propagasi balik (*backpropagation*).
2. Data Uji (*Testing Set* - 20%): Bertindak sebagai data evaluasi independen yang bersifat *unseen data* (tidak pernah dilihat atau dipelajari oleh model selama fase pelatihan). Subset data ini digunakan murni untuk menguji performa akhir model hibrid setelah terlatih, guna memunculkan nilai prediksi sentimen yang objektif yang nantinya akan dikonversi menjadi tabel metrik evaluasi.

Secara teknis di dalam baris kode program Python, proses pembagian data ini diimplementasikan menggunakan fungsi `train_test_split` yang disediakan oleh

pustaka `Scikit-Learn`. Proses pengacakan data tidak dilakukan secara acak konvensional (*simple random sampling*), melainkan menerapkan teknik Pembagian Acak Terstrata (*Stratified Random Sampling*) dengan mengaktifkan parameter `stratify` yang diarahkan pada kolom label sentimen (Positif, Negatif, dan Netral). Penggunaan teknik pembagian terstrata ini sangat krusial untuk menjamin stabilitas eksperimen. Melalui mekanisme ini, sistem akan mengunci proporsi distribusi kelas sentimen pada dataset awal, lalu mendistribusikannya secara adil dan seimbang ke dalam data latih (80%) maupun data uji (20%). Sebagai contoh, jika pada dataset awal terdapat rasio sentimen negatif sebesar 40%, maka setelah dilakukan *splitting*, baik di dalam data latih maupun data uji akan tetap mempertahankan persentase sentimen negatif yang sama, yaitu mendekati nilai 40%. Hal ini dilakukan secara sengaja untuk mencegah terjadinya bias pelatihan akibat ketimpangan distribusi kelas (*class imbalance*) pada salah satu subset data, yang dapat menurunkan nilai generalisasi akurasi model pada saat diuji.

3.8 Perancangan Arsitektur Model Hibrid (IndoBERT + 1D-CNN)

Inti dari rekayasa komputasi dalam penelitian ini terletak pada perancangan arsitektur model hibrid yang menggabungkan kekuatan model bahasa berbasis *Transformer* (IndoBERT) dengan jaringan saraf tiruan konvolusi satu dimensi (1D-CNN). Struktur model ini dirancang untuk memanfaatkan kemampuan IndoBERT dalam menangkap hubungan semantik kontekstual jarak jauh secara global, yang kemudian dipadukan dengan kemampuan 1D-CNN untuk mengekstraksi fitur-fitur lokal yang padat berupa kombinasi kata (*n-gram*) bersentimen kuat. Alur transformasi data tensor di dalam model PyTorch ini dijabarkan secara rinci pada sub-bab berikut:

3.8.1 Tahap Tokenisasi & Contextual Word Embedding (IndoBERT)

Sebelum data teks ulasan yang telah bersih dimasukkan ke dalam jaringan saraf, teks tersebut harus diubah menjadi representasi numerik yang valid melalui lapisan *Tokenizer* bawaan IndoBERT (`indobenchmark/indobert-base-pl`).

1. Tokenisasi & Pemetaan Indeks: Teks bersih dipecah menjadi unit token sub-kata menggunakan metode *WordPiece*. Setiap token kemudian dikonversi menjadi barisan indeks angka (*Token IDs*) berdasarkan repositori kosakata IndoBERT yang berjumlah 50.443 kosakata. Kalimat juga diseragamkan

panjangnya melalui proses *padding* dan *truncation* ke sebuah panjang maksimum (N)

2. Transformasi Vektor Kontekstual: Barisan *Token IDs* dialirkan masuk ke dalam lapisan *Embedding* arsitektur IndoBERT. Lapisan ini terdiri dari 12 lapisan *Transformer Block* bertingkat yang dilengkapi dengan mekanisme *Self-Attention*. Melalui lapisan ini, setiap token dipetakan menjadi vektor angka riil padat berdimensi 768. Hasil akhir dari tahapan ini adalah sebuah tensor tiga dimensi berukuran $B \times N \times 768$ di mana B mewakili ukuran *batch*, N mewakili panjang barisan token kalimat, dan 768 mewakili dimensi representasi fitur *embedding* kontekstual bawaan IndoBERT.

3.8.2 Lapisan Konvolusi (1D-CNN)

Tensor output dari IndoBERT berdimensi $B \times N \times 768$ dikirimkan sebagai masukan (*input*) bagi lapisan 1D-CNN. Di dalam framework PyTorch, urutan dimensi disesuaikan terlebih dahulu menjadi $B \times 768 \times N$ melalui operasi permutasi tensor agar jumlah dimensi fitur bertindak sebagai saluran input (*in-channels*).

1. Mekanisme Konvolusi: Lapisan 1D-CNN dikonfigurasi secara spesifik dengan parameter `in_channels=768` dan `out_channels=128`. Artinya, terdapat 128 filter konvolusi independen yang akan bekerja secara paralel untuk mengekstraksi variasi fitur bahasa yang berbeda.
2. Ukuran *Kernel* & Fitur Lokal: Penulisan fungsi konvolusi menggunakan ukuran *kernel* sebesar 3 (`kernel_size=3`). Filter berukuran 3 ini akan bergerak menyusuri dimensi panjang barisan kalimat N untuk menangkap hubungan spasial lokal antar kata yang berdekatan. Secara linguistik, pergeseran *kernel* 3 ini setara dengan metode ekstraksi frasa tiga kata berturutan (*tri-gram*), seperti frasa "aplikasi sangat lemot" atau "fitur bagus sekali".
3. Fungsi Aktivasi: Hasil kalkulasi dot-product dari proses konvolusi dilewatkan ke fungsi aktivasi ReLU (*Rectified Linear Unit*) secara elemen per elemen. Fungsi aktivasi nonlinear ini bertugas untuk mengeliminasi nilai-nilai fitur negatif menjadi angka nol, sehingga model dapat mempelajari pola-pola hubungan sentimen serumit apa pun secara lebih cepat tanpa mengalami gejala *vanishing gradient*. Output dari lapisan konvolusi ini adalah matriks fitur berdimensi $B \times 128 \times N$

3.8.3 Lapisan *Pooling & Dropout*

Ekstraksi matriks fitur dari lapisan konvolusi 1D-CNN masih memiliki dimensi yang cukup besar dan mengandung informasi sekunder yang tidak terlalu signifikan. Oleh karena itu, diperlukan proses reduksi dimensi dan regularisasi jaringan melalui lapisan berikut:

1. Lapisan *Global Max Pooling 1D*: Lapisan ini bekerja dengan memindai seluruh panjang barisan fitur hasil konvolusi pada setiap filter, lalu hanya mengambil satu nilai numerik tertinggi (*maximum value*). Logika matematis di balik penggunaan *max pooling* ini adalah untuk menangkap fitur frasa paling dominan yang membawa bobot muatan sentimen paling kuat di dalam satu kalimat ulasan, sekaligus mereduksi dimensi tensor secara drastis dari $B \times 128 \times N$ menjadi vektor padat berdimensi statis $B \times 128$
2. Lapisan *Dropout*: Untuk menjamin objektivitas pelatihan, vektor fitur padat berdimensi 128 tersebut dilewatkan ke lapisan regularisasi *Dropout* dengan nilai koefisien probabilitas sebesar **0.3**. Secara teknis, pada setiap putaran latihan (*epoch*), sistem PyTorch akan mematikan secara acak 30% saraf (*neurons*) di dalam lapisan tersebut. Penerapan *dropout* 0.3 ini sangat krusial untuk mencegah terjadinya fenomena *overfitting* (kondisi di mana model menghafal data latih terlalu kaku sehingga gagal melakukan generalisasi prediksi saat menguji data ulasan baru).

3.8.4 Lapisan Klasifikasi Akhir (*Fully Connected Layer*)

Lapisan klasifikasi akhir atau *Fully Connected Layer* (diimplementasikan menggunakan fungsi `nn.Linear` pada PyTorch) bertindak sebagai gerbang keputusan akhir dari arsitektur hibrid ini. Lapisan linear ini menerima masukan berupa vektor fitur yang telah lolos dari regularisasi *dropout* berdimensi $B \times 128$. Lapisan ini memetakan secara linear ruang fitur 128 dimensi tersebut menjadi 3 dimensi output yang merepresentasikan jumlah kelas target analisis sentimen dalam penelitian ini, yaitu Positif, Negatif, dan Netral. Vektor logit mentah berdimensi 3 yang dihasilkan oleh lapisan ini kemudian akan dievaluasi menggunakan fungsi rugi *Cross-Entropy Loss* saat proses pelatihan untuk dihitung tingkat kesalahannya, dan diubah menjadi nilai probabilitas persentase menggunakan fungsi *Softmax* saat fase inferensi untuk menentukan keputusan label akhir ulasan pengguna TikTok.

3.9 Konfigurasi Eksperimen (*Hyperparameter Tuning*)

Fase pelatihan arsitektur hibrid IndoBERT dan 1D-CNN dikendalikan oleh serangkaian parameter penalaan (*hyperparameters*) tetap. Pengaturan konfigurasi ini ditentukan secara terukur berdasarkan rekomendasi empiris dari pengembangan model berbasis *Transformer* serta karakteristik dataset ulasan TikTok yang digunakan. Seluruh proses komputasi pelatihan pada framework PyTorch di Google Colab diatur dengan spesifikasi parameter sebagai berikut:

3.9.1 Algoritma Pengoptimasi (*Optimizer*): AdamW

Penelitian ini menggunakan algoritma AdamW (*Adam with Decoupled Weight Decay*) sebagai metode pengoptimasi bobot jaringan saraf. AdamW dipilih sebagai pembaruan dari algoritma Adam konvensional karena memisahkan perhitungan penalaan bobot (*weight decay*) dari kalkulasi momentum gradien. Secara teknis, penggunaan AdamW sangat krusial dalam *fine-tuning* model bahasa besar seperti IndoBERT. Algoritma ini mampu mengontrol laju perubahan bobot agar tidak terlalu ekstrem, memberikan regularisasi yang lebih baik, serta secara efektif mencegah terjadinya fenomena *overfitting*. Dengan demikian, model dapat mencapai titik konvergensi (kondisi optimal) secara lebih stabil.

3.9.2 Laju Pembelajaran (*Learning Rate*): 2×10^{-5} (0,00002)

Laju pembelajaran (*learning rate*) dikonfigurasi pada nilai yang cukup kecil dan presisi, yaitu 2×10^{-5} . Pemilihan nilai *learning rate* yang rendah ini merupakan standar metodologi yang wajib diterapkan dalam skenario *fine-tuning* model *Pre-trained Language*. Jika nilai laju pembelajaran diset terlalu besar (misalnya 10^{-3}), hal tersebut berisiko memicu terjadinya *catastrophic forgetting*, di mana bobot pengetahuan bahasa Indonesia asli yang sudah tertanam di dalam model IndoBERT akan rusak dan terhapus secara drastis oleh data baru. Sebaliknya, nilai 2×10^{-5} memastikan bahwa model melakukan penyesuaian bobot secara halus, mempertahankan representasi semantik yang sudah ada, sekaligus beradaptasi secara optimal terhadap karakteristik teks ulasan TikTok.

3.9.3 Fungsi Kerugian (*Loss Function*): Cross-Entropy Loss

Untuk mengukur tingkat kesalahan (*error*) prediksi model selama fase pelatihan, digunakan fungsi kerugian **Cross-Entropy Loss**. Fungsi ini merupakan standar matematis dalam menyelesaikan permasalahan klasifikasi multi-kelas (*multi-*

class classification). Logika kerja dari *Cross-Entropy Loss* adalah dengan menghitung jarak kedekatan antara distribusi nilai probabilitas yang dikeluarkan oleh lapisan klasifikasi akhir (*Softmax output*) dengan distribusi label aktual (*ground truth*) dari data latih. Semakin jauh prediksi model dari label yang sebenarnya, nilai kerugian akan semakin tinggi. Nilai kerugian ini kemudian dialirkan mundur melalui proses *backpropagation* untuk meminimalkan tingkat kesalahan model pada putaran selanjutnya.

3.9.4 Jumlah Putaran Pelatihan (*Epochs*): 5 *Epochs*

Durasi pelatihan model dibatasi sebanyak **5 *Epochs*** (5 kali putaran penuh terhadap seluruh data latih). Pembatasan jumlah *epoch* pada nilai yang relatif kecil ini disengaja karena model dasar yang digunakan (IndoBERT) pada dasarnya telah memiliki basis pengetahuan linguistik yang sangat matang. Eksperimen *fine-tuning* pada model berbasis *Transformer* umumnya hanya membutuhkan 3 hingga 5 *epoch* untuk mencapai performa puncak. Menjalankan pelatihan melebihi 5 *epoch* pada dataset ulasan terstruktur rawan menimbulkan efek jenuh pada jaringan saraf (*overtraining*), di mana nilai kerugian pada data latih terus menurun tetapi nilai kerugian pada data uji justru meningkat (*overfitting*).

3.10 Metode Evaluasi Kinerja Model

Tahap akhir dari metodologi penelitian ini adalah pengukuran performa untuk menguji sejauh mana arsitektur hibrid yang dibangun mampu mengklasifikasikan sentimen secara akurat. Pengujian ini dilakukan dengan mengalirkan data uji (*testing set*) sebesar 20% yang bersifat independen dan belum pernah dipelajari oleh model selama fase pelatihan (*unseen data*). Hasil perbandingan antara label prediksi sistem dengan label aktual (*ground truth*) dari data uji tersebut kemudian dipetakan ke dalam tabel Matrix Kebingungan (*Confusion Matrix*). Tabel ini merupakan instrumen evaluasi standar yang menyajikan metrik frekuensi klasifikasi yang tepat maupun meleset untuk setiap kelas sentimen (Positif, Negatif, Netral). Komponen utama di dalam matriks ini meliputi:

1. True Positive (TP): Jumlah data sentimen tertentu yang diprediksi benar oleh model sesuai dengan kelas aktualnya.
2. True Negative (TN): Jumlah data bukan anggota kelas tertentu yang diprediksi secara benar oleh model sebagai bukan anggota kelas tersebut.

3. False Positive (FP): Jumlah data salah yang diprediksi oleh model sebagai anggota kelas tertentu, padahal secara aktual bukan.
4. False Negative (FN): Jumlah data anggota kelas tertentu yang gagal dideteksi oleh model (diprediksi sebagai kelas lain).

Berdasarkan nilai dari komponen *Confusion Matrix* tersebut, dilakukan perhitungan matematis secara terukur untuk memunculkan empat parameter metrik penilaian utama, yaitu:

3.10.1 Akurasi (*Accuracy*)

Akurasi merupakan metrik representasi kedekatan hasil secara menyeluruh. Nilai akurasi mengukur persentase total prediksi benar (baik positif maupun negatif) yang dihasilkan oleh model dari keseluruhan total data uji yang dievaluasi. Rumus formal perhitungan akurasi dituliskan sebagai berikut:

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN)$$

3.10.2 Presisi (*Precision*)

Presisi mengukur tingkat ketepatan antara informasi yang diminta oleh pengguna dengan hasil prediksi yang diberikan oleh model. Metrik ini menghitung rasio jumlah data yang benar-benar positif dari seluruh data yang diprediksi sebagai kelas positif oleh sistem. Rumus formal perhitungan presisi adalah:

$$\text{Presisi} = TP / (TP + FP)$$

3.10.3 Recall (*Sensitivitas*)

Recall mengukur keberhasilan model dalam menemukan kembali informasi (*information retrieval*). Metrik ini menghitung rasio jumlah data positif yang berhasil diprediksi benar oleh model dibandingkan dengan total keseluruhan data yang secara aktual berlabel positif. Rumus formal perhitungan *recall* dituliskan sebagai berikut:

$$\text{Recall} = TP / (TP + FN)$$

3.10.4 F1-Score

F1-Score merupakan metrik evaluasi yang menghitung rata-rata harmonik (*harmonic mean*) yang menyeimbangkan nilai Presisi dan *Recall*. Metrik ini sangat krusial digunakan sebagai tolok ukur utama jika sebaran jumlah data antar-kelas sentimen (Positif, Negatif, Netral) di dalam data uji tidak berimbang secara sempurna. Rumus formal perhitungan *F1-Score* dituliskan sebagai berikut:

$$\text{F1-Score} = 2 \times \text{Presisi} \times \text{Recall} / (\text{Presisi} + \text{Recall})$$

Seluruh hasil perhitungan metrik di atas dianalisis secara mendalam pada bagian pembahasan, baik untuk mengevaluasi model hibrid usulan (IndoBERT + 1D-CNN) maupun sebagai basis perbandingan ilmiah terhadap model *baseline* pada eksperimen studi ablasi.