EPISTEMEUS
ESSAYS

Volume 1 ; Issue 1 15.05.2026

Perbandingan Kinerja IndoBERT dan TF-IDF pada Analisis Sentimen Ulasan Aplikasi ChatGPT

Lazzu Fadlin Muslim, M. Fathurrochim, M. Raya Bilfikri, Rohim Amrullah

ABSTRAK

Penelitian ini bertujuan untuk membandingkan kinerja model klasik Term Frequency–Inverse Document Frequency (TF-IDF) dengan model Deep Learning berbasis Transformer (IndoBERT) dalam tugas analisis sentimen ulasan aplikasi ChatGPT berbahasa Indonesia dari Google Play Store. Tantangan utama yang diangkat adalah ketidakseimbangan data (*imbalanced data*) yang ekstrem serta sulitnya mendeteksi kelas sentimen *netral* atau *campuran*. Sebanyak 15.000 ulasan mentah diolah melalui strategi Anotasi Hibrida berbasis kamus leksikon dan diseimbangkan menjadi 3.000 sampel per kelas. Hasil pengujian menunjukkan IndoBERT mencapai akurasi ∼89% dengan F1-Score Netral 0,90, sementara TF-IDF berbasis Multinomial Naive Bayes mencapai akurasi 89,50% dengan F1-Score Netral 0,83. IndoBERT terbukti lebih unggul dalam memahami konteks kalimat ambigu, sedangkan TF-IDF tetap kompetitif berkat strategi anotasi hibrida dan oversampling.

Kata Kunci: Analisis Sentimen, IndoBERT, TF-IDF, *Imbalanced Data*, Anotasi Hibrida, ChatGPT.

BAB I Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*) telah mengalami lonjakan pesat dalam satu dekade terakhir, terutama dengan

munculnya *Large Language Models* (LLM). Salah satu produk yang paling fenomenal adalah ChatGPT yang dikembangkan oleh OpenAI. Sejak peluncurannya, aplikasi ChatGPT di platform mobile (Google Play Store) telah diunduh oleh jutaan pengguna di seluruh dunia, termasuk di Indonesia. Ulasan pengguna pada platform distribusi aplikasi menjadi sumber data yang sangat berharga bagi pengembang untuk memahami tingkat kepuasan, mengidentifikasi *bug*, serta merencanakan pengembangan fitur di masa depan.

Namun, menganalisis ribuan ulasan secara manual merupakan tugas yang tidak efisien dan memakan waktu. Oleh karena itu, pendekatan *Natural Language Processing* (NLP) diperlukan untuk mengotomatisasi proses analisis sentimen. Tantangan utama dalam analisis sentimen ulasan aplikasi nyata adalah karakteristik data yang seringkali sangat tidak seimbang (*imbalanced dataset*). Pengguna cenderung memberikan ulasan ketika mereka merasa sangat puas (bintang 5) atau sangat kecewa (bintang 1), sehingga ulasan dengan sentimen "Netral" atau "Campuran" menjadi minoritas dan sulit dideteksi oleh model pembelajaran mesin standar.

Metode klasifikasi teks tradisional seperti TF-IDF seringkali gagal menangkap nuansa konteks pada kalimat netral atau campuran karena hanya bergantung pada frekuensi kemunculan kata dalam dokumen (Manning dkk., 2008). Di sisi lain, metode Deep Learning berbasis arsitektur Transformer, seperti BERT, menawarkan kemampuan pemahaman konteks yang jauh lebih dalam dengan mekanisme *attention* (Vaswani dkk., 2017). Khusus untuk Bahasa Indonesia, terdapat varian IndoBERT yang telah dilatih menggunakan korpus Bahasa Indonesia yang luas (IndoBenchmark, 2020).

Penelitian ini bertujuan untuk menerapkan dan membandingkan kinerja model IndoBERT dengan metode klasik TF-IDF dalam melakukan analisis sentimen terhadap ulasan aplikasi ChatGPT. Penelitian ini juga memfokuskan pada penerapan teknik pra-pemrosesan data lanjutan, seperti anotasi hibrida (*hybrid annotation*) dan penyeimbangan data (*resampling*), untuk mengatasi masalah ketimpangan data serta meningkatkan akurasi deteksi pada kelas sentimen netral.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana cara mengatasi masalah ketidakseimbangan data (*imbalanced data*) pada ulasan aplikasi ChatGPT untuk mencegah bias model terhadap kelas mayoritas (positif)?
2. Apakah penerapan metode anotasi hibrida (*hybrid annotation*)

dan teknik *balancing* (*oversampling* dan *undersampling*) dapat meningkatkan kemampuan model dalam mendeteksi sentimen netral atau campuran?

3. Bagaimana perbandingan performa antara model berbasis Transformer (IndoBERT) dengan model klasik (TF-IDF) dalam mengklasifikasikan sentimen ulasan berbahasa Indonesia?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari pelaksanaan tugas besar ini adalah:

1. Mengimplementasikan model Deep Learning IndoBERT untuk melakukan analisis sentimen pada ulasan aplikasi ChatGPT di Google Play Store.

2. Menerapkan strategi *data engineering* melalui teknik anotasi hibrida berbasis kamus (*lexicon-based*) dan penyeimbangan data (*undersampling* kelas mayoritas dan *oversampling* kelas minoritas) untuk meningkatkan akurasi pada kelas sentimen netral.

3. Mengevaluasi dan membandingkan kinerja model IndoBERT dengan model *baseline* TF-IDF menggunakan metrik F1-Score untuk menentukan metode yang paling efektif dalam menangani data ulasan yang kompleks.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. **Manfaat Teoretis:** Memberikan wawasan mengenai efektivitas model Transformer (IndoBERT) dibandingkan metode klasik dalam menangani kasus klasifikasi teks dengan data yang sangat tidak seimbang (*highly imbalanced*) dalam Bahasa Indonesia.

2. **Manfaat Praktis:** Menghasilkan prototipe sistem analisis sentimen yang dapat digunakan untuk memetakan opini publik secara otomatis, memisahkan ulasan positif, negatif, dan netral dengan akurasi yang lebih baik, yang berguna bagi pengembang aplikasi untuk pengambilan keputusan.

BAB II Landasan Teori

2.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) atau Pemrosesan Bahasa Alami adalah cabang dari kecerdasan buatan yang berfokus pada interaksi antara komputer dan bahasa manusia. Tujuan utama NLP adalah memungkinkan komputer untuk memahami, menafsirkan, dan memanipulasi bahasa manusia dengan cara yang bermanfaat. NLP menggabungkan komputasi linguistik dengan model statistik, *machine learning*, dan *deep learning* untuk memproses data teks yang tidak terstruktur menjadi format yang dapat dianalisis secara komputasional.

2.2 Analisis Sentimen

Analisis sentimen, sering disebut juga sebagai *opinion mining*, adalah proses komputasi untuk mengidentifikasi dan mengkategorikan opini yang diungkapkan dalam sepotong teks, terutama untuk menentukan apakah sikap penulis bersifat positif, negatif, atau netral. Dalam konteks ulasan aplikasi, analisis sentimen bertujuan untuk mengekstrak informasi subjektif dari pengguna. Tantangan utama dalam analisis sentimen adalah menangani ambiguitas bahasa, sarkasme, dan kalimat dengan sentimen campuran (*mixed sentiment*), di mana pengguna dapat memberikan pujian dan kritik dalam satu kalimat yang sama.

2.2.1 Pendekatan Berbasis Leksikon (Lexicon-Based)

Pendekatan ini menggunakan kamus kata-kata yang telah diberi bobot sentimen (positif atau negatif). Metode ini tidak memerlukan data latih, melainkan bergantung pada aturan (*rules*) dan pencocokan kata. Dalam penelitian ini, pendekatan leksikon digunakan sebagai bagian dari strategi anotasi hibrida untuk mendeteksi sentimen campuran yang sering terlewatkan oleh pelabelan otomatis berdasarkan skor rating.

2.3 Imbalanced Data & Teknik Resampling

Masalah ketidakseimbangan data terjadi ketika distribusi kelas dalam dataset tidak merata, di mana satu kelas mayoritas mendominasi kelas minoritas. Pada ulasan produk, ulasan positif seringkali mendominasi ulasan negatif atau netral. Hal ini dapat menyebabkan model menjadi bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Untuk mengatasinya diterapkan teknik *resampling*:

- **Undersampling:** Mengurangi jumlah sampel dari kelas mayoritas secara acak untuk menyeimbangkan distribusi. Teknik ini efektif untuk mengurangi waktu komputasi dan mencegah model bias terhadap kelas dominan.
- **Oversampling:** Menambah jumlah sampel pada kelas minoritas dengan menduplikasi data secara acak. Tujuannya agar model memiliki lebih banyak kesempatan untuk mempelajari pola dari kelas yang jarang muncul.

2.4 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF adalah metode ekstraksi fitur klasik dalam NLP yang mengubah teks menjadi vektor angka. Metode ini bekerja dengan prinsip:

1. **TF (Term Frequency):** Mengukur seberapa sering sebuah kata muncul dalam satu dokumen.
2. **IDF (Inverse Document Frequency):** Mengukur seberapa penting kata tersebut. Kata yang muncul di semua dokumen (seperti "dan", "yang") dianggap tidak penting (bobot rendah), sedangkan kata yang jarang muncul diberi bobot tinggi.

Meskipun efektif dan cepat, TF-IDF memiliki kelemahan yaitu tidak menangkap urutan kata atau konteks semantik antar kata dalam kalimat.

2.5 Deep Learning & Arsitektur Transformer

Berbeda dengan metode klasik, *Deep Learning* menggunakan jaringan saraf tiruan berlapis untuk mempelajari representasi data. Terobosan terbesar dalam NLP modern adalah arsitektur *Transformer* yang diperkenalkan oleh Google (Vaswani dkk., 2017). Transformer menggunakan mekanisme *Self-Attention* yang memungkinkan model untuk menimbang pentingnya setiap kata dalam kalimat terhadap kata lainnya secara bersamaan, bukan berurutan. Hal ini memungkinkan model memahami konteks yang kompleks, ambiguitas, dan hubungan jarak jauh antar kata dengan jauh lebih baik daripada metode sebelumnya seperti RNN atau LSTM.

2.6 IndoBERT (Pre-trained Model)

IndoBERT adalah model bahasa berbasis arsitektur BERT (*Bidirectional Encoder Representations from Transformers*) yang telah dilatih secara spesifik menggunakan korpus data Bahasa Indonesia yang sangat besar (Indo4B),

mencakup lebih dari 4 miliar kata dari berbagai sumber seperti berita, media sosial, dan Wikipedia Indonesia. Karena telah melalui proses *pre-training* pada Bahasa Indonesia, IndoBERT memiliki pemahaman mendalam tentang struktur, tata bahasa, dan semantik Bahasa Indonesia. Dalam penelitian ini, teknik *fine-tuning* diterapkan pada IndoBERT untuk tugas klasifikasi sentimen ulasan aplikasi.

2.7 Evaluasi Model dan Label Smoothing

2.7.1 Metrik Evaluasi

Untuk mengukur kinerja model pada data yang tidak seimbang, akurasi saja tidak cukup. Penelitian ini menggunakan:

1. **F1-Score (Weighted):** Rata-rata harmonis antara *Precision* dan *Recall* yang memperhitungkan proporsi setiap kelas. Ini memberikan gambaran yang lebih adil tentang kinerja model pada kelas minoritas.
2. **Classification Report:** Laporan detail yang menunjukkan metrik performa untuk setiap kelas secara individu.

2.7.2 Label Smoothing

Label Smoothing adalah teknik regularisasi yang digunakan untuk mencegah model menjadi terlalu percaya diri (*overconfident*) atau *overfitting* pada data latih. Teknik ini mengubah target label dari nilai pasti (misal: 1,0 untuk Positif) menjadi nilai yang lebih lunak (misal: 0,9 untuk Positif, dan sisa 0,1 disebar ke kelas lain). Ini membantu model untuk belajar pola yang lebih umum dan tidak sekadar menghafal data, terutama pada kelas dengan jumlah sampel terbatas.

BAB III Metodologi Penelitian

3.1 Alat dan Lingkungan Pengembangan

Penelitian ini dilaksanakan menggunakan lingkungan pengembangan berbasis awan (*cloud computing*) untuk memanfaatkan akselerasi perangkat keras yang diperlukan dalam melatih model Deep Learning. Berikut spesifikasi alat dan pustaka:

1. **Platform Pengembangan:** Google Colab (runtime GPU T4).
2. **Bahasa Pemrograman:** Python 3.10+.
3. **Pustaka Utama:**

- *Hugging Face Transformers* – mengimplementasikan model IndoBERT dan Tokenizer.
- *PyTorch – backend framework* untuk komputasi neural network.
- *Scikit-learn* – pembagian data, *resampling*, dan perhitungan metrik evaluasi.
- *Pandas & NumPy* – manipulasi dan analisis data tabular.
- *Gradio* – antarmuka pengguna berbasis web untuk demonstrasi model.

3.2 Sumber Data (Dataset)

Data yang digunakan adalah dataset sekunder berisi ulasan pengguna aplikasi ChatGPT berbahasa Indonesia yang bersumber dari Google Play Store.

- **Jumlah Data Awal:** Sekitar 15.000 baris data mentah.
- **Atribut Data:**
  - `content`: Teks ulasan yang ditulis pengguna.
  - `score`: Rating bintang (skala 1–5).

3.3 Perancangan Alur Kerja (Pipeline Design)

Penelitian ini mengikuti alur kerja sistematis yang dirancang untuk menangani masalah ketidakseimbangan data dan ambiguitas sentimen. Alur kerja dibagi menjadi dua jalur eksperimen: jalur *Deep Learning* (IndoBERT) dan jalur *Machine Learning* klasik (TF-IDF) sebagai pembanding.

3.4 Tahapan Penelitian Model IndoBERT

Metodologi *end-to-end* dirancang sistematis mulai dari penanganan data mentah hingga penyajian model dalam bentuk aplikasi.

3.4.1 Pra-pemrosesan Data dan Rekayasa Data

Tahap ini bertujuan mengubah data mentah yang tidak terstruktur dan bias menjadi data berkualitas tinggi. Proses terdiri dari tiga langkah:

1. **Pembersihan Data:** Mekanisme pembersihan tangguh (*robust*) untuk menangani kesalahan format pada CSV, normalisasi tipe data kolom skor, serta penghapusan baris yang rusak atau kosong.

2. **Anotasi Hibrida:** Menggabungkan label asli (skor bintang) dengan pendekatan berbasis aturan menggunakan kamus leksikon khusus. Kamus mencakup kata sentimen umum serta istilah teknis spesifik aplikasi AI ("halusinasi", "limit", "berbayar"). Ulasan yang mengandung kata positif dan negatif secara bersamaan akan dilabeli ulang menjadi Netral.

3. **Penyeimbangan Data:** Distribusi data asli sangat timpang dengan dominasi kelas positif hingga 90%. Target ditetapkan 3.000 sampel per kelas dengan kombinasi *undersampling* pada kelas Positif dan *oversampling* pada Negatif dan Netral.

3.4.2 Pembagian Data (Data Splitting)

Dataset bersih dan seimbang (total 9.000 baris) dibagi menjadi dua himpunan yang saling lepas menggunakan teknik *Stratified Sampling*, yang menjamin proporsi kelas sentimen tetap seimbang (1:1:1) pada data latih dan validasi. Dilakukan dua skenario:

- Skenario A (80:20) – eksperimen utama untuk performa maksimal.
- Skenario B (70:30) – uji ketahanan (*robustness test*).

3.4.3 Representasi Teks (Tokenization)

Tokenisasi menggunakan tokenizer bawaan dari model *pre-trained* `indobenchmark/indobert-base-p1`. Tahap ini mencakup *padding* untuk menyeragamkan panjang input ke 128 token, serta *truncation* untuk memotong kalimat yang melebihi batas.

3.4.4 Pembangunan dan Pelatihan Model (Fine-Tuning)

Inti penelitian adalah proses pelatihan model menggunakan metode *Transfer Learning*. Model dasar IndoBERT dimodifikasi dengan menambahkan *classification head* linear untuk 3 kelas sentimen. Konfigurasi pelatihan:

- **Iterasi:** 3 *epoch* penuh.
- **Batch Size:** 16.
- **Label Smoothing (0,1):** Regularisasi untuk melunakkan target probabilitas guna memaksa model mempelajari fitur umum, bukan menghafal sampel.

3.4.5 Evaluasi Model

Evaluasi dilakukan secara ketat menggunakan data validasi yang tidak pernah dilibatkan dalam pelatihan. Fokus pada F1-Score Weighted serta analisis *classification report* per kelas.

3.4.6 Penyimpanan Model (Model Persistence)

Komponen yang disimpan: bobot model (`pytorch_model.bin`) dan konfigurasi tokenizer (`vocab.txt`). Langkah ini memisahkan fase pelatihan dari fase penggunaan.

3.4.7 Pengembangan Antarmuka Pengguna

Tahap akhir mengembangkan prototipe berbasis web menggunakan *Gradio*. Antarmuka memungkinkan pengguna memasukkan teks ulasan dan menerima hasil prediksi label sentimen beserta skor keyakinan (*confidence score*) secara *real-time*, kemudian di-*deploy* dengan fitur *shareable link*.

3.5 Tahapan Penelitian Model TF-IDF (Pembanding)

Pendekatan *feature engineering* berbasis statistik (TF-IDF) yang diperkuat dengan logika *hybrid* (kamus).

3.5.1 Persiapan Data, Pra-Pemrosesan, dan Feature Engineering

- **Pembersihan Data:** Menghapus baris kosong dan memfilter kolom skor agar hanya berisi nilai numerik valid (1–5).
- **Anotasi Hibrida:** Menggabungkan skor rating asli dengan pendekatan *lexicon-based* untuk mendeteksi Sentimen Campuran.
- **Pra-Pemrosesan Teks Penuh:** Meliputi *stopword removal* (dengan mempertahankan negasi seperti "tidak", "belum") dan *stemming* menggunakan Sastrawi.
- **Penyeimbangan Data:** *Undersampling* kelas Positif dan *oversampling* kelas Negatif dan Netral, target 3.000 sampel per kelas.

3.5.2 Pembagian Data

Dua skenario menggunakan *stratified sampling*: Skenario Utama (80:20) dan Skenario Uji Stabilitas (70:30).

3.5.3 Representasi Teks

Teks yang telah melalui *full preprocessing* diubah menjadi format numerik menggunakan `TfidfVectorizer` dengan pembatasan 6.000 fitur dan *n-gram range* 1–2. `TfidfVectorizer` di-*fit* pada data latih dan hanya di-*transform* pada data uji.

3.5.4 Pembangunan dan Pelatihan Model

Model klasifikasi menggunakan algoritma yang cocok untuk *sparse matrix*: *Logistic Regression* (dengan `class_weight='balanced'`) atau *Multinomial Naive Bayes* (dengan *Laplace Smoothing* `alpha=1.0`).

3.5.5 Validasi dan Uji Stabilitas

Model divalidasi menggunakan *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Fokus khusus pada F1-Score kelas Netral.

3.5.6 Implementasi dan Deployment Hibrida di UI

Fungsi prediksi di Gradio UI memprioritaskan logika konflik kamus. Jika konflik (Positif + Negatif) terdeteksi, hasil dipaksa menjadi Netral (*override*). Jika tidak, input melalui *full preprocessing* dan `tfidf.transform()` sebelum prediksi `clf.predict()`.

BAB IV Implementasi Sistem

Implementasi sistem dilakukan menggunakan Python pada lingkungan Google Colab yang mendukung akselerasi GPU.

4.1 Implementasi Model IndoBERT

Implementasi IndoBERT mengikuti *pipeline* 9 tahap utama, mulai dari persiapan lingkungan hingga penyebaran aplikasi.

4.1.1 Persiapan Lingkungan dan Pustaka (Tahap 0)

Sistem menginstalasi paket eksternal yang krusial untuk penelitian: `transformers` (model IndoBERT), `datasets` (manajemen data), `accelerate` (optimalisasi GPU), dan `gradio` (UI). Selain itu, pustaka standar seperti `pandas`, `numpy`, dan `torch` dimuat ke memori aktif.

Komponen utama yang diimpor mencakup `AutoTokenizer`, `AutoModelForSequenceClassification`, `Trainer`, dan `TrainingArguments` dari Hugging Face.

4.1.2 Pemuatan dan Pembersihan Data (Tahap 1)

Sistem menghubungkan lingkungan kerja dengan Google Drive untuk mengakses dataset mentah. Mekanisme pembacaan tangguh diterapkan: percobaan pertama dengan encoding `utf-8`, lalu *fallback* ke `latin1` bila gagal. Kolom skor dikonversi ke format numerik dengan `pd.to_numeric(errors='coerce')` sehingga data teks/tanggal yang salah masuk akan diubah menjadi `NaN`, kemudian dihapus dengan `df.dropna`.

4.1.3 Rekayasa Data: Anotasi Hibrida dan Penyeimbangan (Tahap 2)

Tahap ini merupakan inovasi utama. Kamus leksikon disusun secara manual mencakup kata sentimen umum, istilah teknis ("halusinasi", "limit", "berbayar"), masalah teknis ("error", "lag", "crash"), serta masalah akses ("mahal", "premium").

Fungsi `anotasi_hibrida` memeriksa keberadaan kata positif dan negatif secara bersamaan. Jika ditemukan keduanya, label dipaksa menjadi 1 (Netral). Jika tidak, label ditentukan berdasarkan skor: ≤ 2 Negatif, ≥ 4 Positif, lainnya Netral.

Setelah anotasi, dilakukan *balancing* dengan target 3.000 sampel per kelas: *undersampling* positif (`replace=False`) untuk mempertahankan variasi data unik, serta *oversampling* negatif dan netral (`replace=True`) untuk memperbanyak eksposur kelas minoritas.

4.1.4 Pembagian Data (Tahap 3)

Pembagian data dilakukan dalam dua skenario dengan parameter `test_size`:

- Skenario A (80:20): 7.200 data latih, 1.800 data validasi.
- Skenario B (70:30): 6.300 data latih, 2.700 data validasi.

Parameter `stratify=df_final['label']` menjaga konsistensi distribusi kelas, lalu dikonversi ke format Hugging Face `Dataset` agar kompatibel dengan `Trainer`.

4.1.5 Representasi Teks / Tokenization (Tahap 4)

Sistem memuat tokenizer dari `indobenchmark/indobert-base-p1`. `AutoTokenizer.from_pretrained` memuat kamus kata milik IndoBERT. `padding="max_length"` menyeragamkan panjang input, dengan `truncation=True` dan `max_length=128` membatasi input maksimal demi efisiensi memori. Operasi `.map(..., batched=True)` menerapkan tokenisasi pada ribuan baris secara paralel.

4.1.6 Konfigurasi Model dan Metrik Evaluasi (Tahap 5)

`AutoModelForSequenceClassification` memuat arsitektur IndoBERT dan menambahkan "kepala" klasifikasi baru (`num_labels=3`). Pemetaan label diatur: `{0: "Negatif", 1: "Netral", 2: "Positif"}`.

Fungsi `compute_metrics` menghitung Akurasi dan *F1-Score weighted* setiap kali model dievaluasi. Metrik *weighted* dipilih karena ideal untuk dataset tidak seimbang.

4.1.7 Pelatihan Model / Fine-Tuning (Tahap 6)

`TrainingArguments` mengatur parameter pelatihan: 3 *epoch*, *batch size* 16, `warmup_steps=500`, `weight_decay=0.01`, evaluasi setiap akhir *epoch*, dan `load_best_model_at_end=True`. Parameter kunci adalah `label_smoothing_factor=0.1` sebagai regularisasi untuk melunakkan target label (dari 1,0 menjadi 0,9) guna meningkatkan generalisasi model.

Objek `Trainer` mengelola seluruh proses dengan `trainer.train()`.

4.1.8 Evaluasi Model (Tahap 7)

Setelah pelatihan selesai, `trainer.evaluate()` menghitung skor rata-rata pada data validasi, sementara `classification_report` menampilkan matriks performa terperinci (Precision, Recall, F1-Score) untuk kelas Negatif, Netral, dan Positif.

4.1.9 Penyimpanan Model (Tahap 8)

`trainer.save_model(OUTPUT_MODEL_DIR)` menyimpan bobot model (`pytorch_model.bin`) dan konfigurasi arsitektur, sementara `tokenizer.save_pretrained` menyimpan kamus kata (`vocab.txt`). Hal ini memungkinkan model dimuat ulang kapan saja tanpa perlu pelatihan ulang.

4.1.10 Implementasi Antarmuka Pengguna (Tahap 9)

`pipeline("sentiment-analysis", ...)` memuat model dan tokenizer yang tersimpan untuk inferensi. Fungsi `predict_sentiment` memvalidasi input, meminta prediksi, dan memformat hasil output ke format yang mudah dibaca. `gr.Interface` membangun komponen visual, dan `launch(share=True)` menjalankan server web dengan *public link* sementara untuk demonstrasi.

4.2 Implementasi Model TF-IDF

Implementasi TF-IDF mengikuti alur komprehensif dari *setup* hingga *deployment* model.

4.2.1 Setup dan Impor Library

Library yang diinstal adalah `Sastrawi` untuk pemrosesan Bahasa Indonesia dan `Gradio` untuk antarmuka. `pandas` dan `numpy` digunakan untuk manipulasi data, sementara modul `re` untuk *regex cleaning*. Dari `scikit-learn` diimpor `resample`, `train_test_split`, `TfidfVectorizer`, dan `LogisticRegression`.

4.2.2 Load Data Mentah

Dataset ulasan ChatGPT berbahasa Indonesia dari Play Store dimuat dari Google Drive yang telah di-*mount*. Pembersihan awal menghapus baris dengan nilai kosong pada kolom `content` dan `score`.

4.2.3 Labeling dan Balancing

Kamus sentimen disusun dalam tiga kelompok: positif, negatif, dan netral. Fungsi `clean_text` melakukan pembersihan dasar (*lowercase*, penghapusan simbol, perapian spasi).

Fungsi `hybrid_label` mencocokkan kata dalam ulasan dengan kamus sentimen. Setiap kategori dihitung jumlah kecocokannya. Bila terdapat campuran positif dan negatif, label diarahkan ke netral (1). Jika hanya satu kategori positif: label 2; hanya negatif: label 0; hanya netral atau tidak ada kecocokan: label 1.

Balancing diterapkan dengan target 3.000 sampel per kelas: *undersampling* positif tanpa pengembalian, *oversampling* netral dengan pengembalian, dan kelas negatif disesuaikan dengan metode sampling yang sesuai. Dataset kemudian diacak (`sample(frac=1)`).

4.2.4 Pra-Pemrosesan Teks

Pra-pemrosesan dilakukan secara berlapis:

- **Pembersihan Format Awal:** *lowercase*, penghapusan URL, angka, dan simbol non-alfabetik.
- **Stopword Removal:** Membuang kata umum dengan pengecualian negasi ("tidak", "kurang", "tanpa", "belum").
- **Stemming:** Mengembalikan kata berimbuhan ke bentuk dasar menggunakan `Sastrawi`.
- **Normalisasi Akhir:** Merapikan spasi berlebih.

Hasil disimpan dalam kolom `content_clean`.

4.2.5 Train-Test Split

Data dibagi 80% latih dan 20% uji dengan `stratify=y` dan `random_state=42`.

4.2.6 TF-IDF Vectorization

`TfidfVectorizer(max_features=6000, ngram_range=(1,2))` dipakai untuk mengubah data teks menjadi representasi numerik. Vectorizer di-*fit* pada data latih dan hanya di-*transform* pada data uji.

4.2.7 Training Model dengan Smoothing

Model `MultinomialNB(alpha=1.0)` dilatih pada vektor TF-IDF. Parameter `alpha=1.0` mengaktifkan *Laplace Smoothing* untuk mengatasi probabilitas nol pada kata yang muncul di data uji tetapi tidak ada di data latih.

4.2.8 Evaluasi Model

`classification_report` menyajikan metrik Precision, Recall, dan F1-Score untuk setiap kelas sentimen, sementara `accuracy_score` menunjukkan persentase prediksi yang benar.

4.2.9 Deploy Gradio UI

Fungsi `predict_sentiment` memprioritaskan logika konflik kamus. Jika ulasan mengandung kata positif dan negatif sekaligus, hasil dipaksa menjadi Netral. Jika tidak, input melalui *full preprocessing*

dan `tfidf.transform()` sebelum prediksi oleh `clf_nb`. Antarmuka `gr.Interface` menyediakan `Textbox` input dan output teks dengan `iface.launch()`.

BAB V Hasil dan Pembahasan

5.1 Evaluasi Performa Model IndoBERT

5.1.1 Analisis Proses Pelatihan

Proses *fine-tuning* dijalankan terpisah untuk kedua skenario eksperimen guna mengamati pola pembelajaran pada jumlah data latih berbeda.

A. Skenario Utama (80:20). Model dilatih menggunakan 7.200 data. Model mencapai konvergensi yang sangat baik: nilai *Validation Loss* menurun secara konsisten hingga angka terendah 0,49 pada akhir *epoch* ke-3, dengan akurasi akhir mencapai 89%.

B. Skenario Uji Stabilitas (70:30). Dengan 6.300 data latih, grafik pembelajaran menunjukkan pola identik dengan skenario utama. *Validation Loss* tetap menurun stabil tanpa indikasi *overfitting*, membuktikan strategi *Label Smoothing* bekerja efektif menstabilkan proses belajar.

5.1.2 Evaluasi Metrik dan Uji Stabilitas

A. Hasil Skenario Utama (80:20). Model mencapai performa puncak dengan F1-Score kelas Netral 0,90, membuktikan keberhasilan strategi anotasi hibrida dalam menangkap pola sentimen campuran.

B. Hasil Skenario Uji Stabilitas (70:30). Dengan data latih 70%, F1-Score Netral tetap di angka 0,90 dan akurasi global 89%.

5.1.3 Pembahasan Stabilitas Model (Robustness Analysis)

1. **Stabilitas Tinggi:** Penurunan performa dari 80/20 ke 70/30 sangat minim (selisih F1-Score rata-rata $\sim$ 0,01). Model IndoBERT yang dibangun memiliki *robustness* tinggi: tidak "rapuh" meskipun data belajarnya dikurangi.

2. **Konsistensi Deteksi Netral:** Baik 80/20 maupun 70/30, kemampuan deteksi kelas Netral tetap konsisten tinggi, memvalidasi bahwa kualitas data Netral hasil Anotasi Hibrida memang berkualitas tinggi dan polanya kuat.

5.2 Evaluasi Performa Model TF-IDF

5.2.1 Analisis Proses Pelatihan

Model *baseline* TF-IDF dilatih menggunakan vektor TF-IDF dari ~7.200 sampel hasil *oversampling* dan pra-pemrosesan Sastrawi. Berbeda dengan IndoBERT, proses pelatihan MNB berbasis perhitungan probabilitas:

- **Smoothing:** *Laplace Smoothing* (`alpha=1.0`) mencegah probabilitas nol pada kata yang tidak muncul di data latih.
- **Konvergensi dan Efisiensi:** Tidak ada iterasi konvergensi kompleks. Waktu pelatihan sangat cepat dibandingkan model Transformer.
- **Kualitas Balancing:** Dengan data *balanced* 3.000 sampel per kelas, MNB dapat menghitung probabilitas frekuensi kata yang representatif.

5.2.2 Evaluasi Metrik (Classification Report)

A. Hasil Skenario Utama (80:20). Pengujian pada 1.800 sampel data uji:

- **Akurasi Global:** 89,50%, menunjukkan *baseline* TF-IDF yang dikombinasikan dengan Anotasi Hibrida sangat kompetitif.
- **Keberhasilan Deteksi Netral:** F1-Score Netral 82,65% ($\sim$ 0,83). Meskipun *Recall* relatif rendah (0,72), *Precision* yang sangat tinggi (0,97) menunjukkan bahwa ketika model memprediksi Netral, tingkat keyakinannya sangat akurat.
- **Konsistensi:** *Macro Avg* F1-Score 0,89 menunjukkan performa relatif seimbang di ketiga kelas.
- **Pola Prediksi:** Kelas Negatif memiliki *Recall* sempurna (1,00) tetapi *Precision* lebih rendah (0,81). Model tidak pernah melewatkan ulasan Negatif, tetapi terkadang salah mengklasifikasikan ulasan lain sebagai Negatif.

B. Hasil Skenario Uji Stabilitas (70:30). Hasil hampir sama dengan 80/20, hanya berbeda sedikit pada persentase akurasi dan F1-Score Netral.

5.3 Analisis Perbandingan (IndoBERT vs TF-IDF)

Metrik	TF-IDF (Baseline)	IndoBERT (Deep Learning)
Pendekatan	Frekuensi Kata (*Statistical*)	Pemahaman Konteks (*Contextual*)
Akurasi Total (80/20)	83%	89%
Akurasi Total (70/30)	83%	89%
F1-Score Netral (80/20)	85%	90%
F1-Score Netral (70/30)	85%	90%

Analisis: Model IndoBERT terbukti lebih unggul dibandingkan TF-IDF, terutama dalam mendeteksi sentimen Netral atau Campuran.

- **Kelemahan TF-IDF:** Cenderung kesulitan mengenali arti sebenarnya dari kalimat. Akurasi dan F1-Score Netral lebih rendah karena sulit membedakan ulasan ambigu. Selain itu, TF-IDF harus melewati tahap pra-pemrosesan yang panjang (*stemming*, *stopword removal*) agar bekerja optimal, sebab hanya menghitung frekuensi kata tanpa memahami makna atau urutan.

- **Keunggulan IndoBERT:** Arsitektur Transformer dan *Self-Attention* membuat IndoBERT mengerti konteks bahasa, menghasilkan Akurasi dan F1-Score Netral yang lebih tinggi. IndoBERT memahami makna kata berdasarkan seluruh kalimat (kontekstual), sangat baik dalam memprediksi sentimen Netral. IndoBERT juga tidak memerlukan pra-pemrosesan rumit; cukup proses tokenisasi ringan yang menghemat waktu penyiapan data.

5.4 Pengujian Aplikasi (User Interface Testing)

Pengujian fungsional dilakukan menggunakan antarmuka Gradio. Hasil pengujian menunjukkan model mampu merespons input pengguna secara *real-time* dengan menyertakan skor keyakinan (*confidence score*). Namun, ditemukan keterbatasan *Domain Mismatch*: model bekerja sangat baik pada ulasan terkait aplikasi/teknologi, namun performanya menurun ketika diberikan input di luar domain pelatihannya (misalnya ulasan makanan), karena model memang dikhususkan (*fine-tuned*) untuk domain ulasan aplikasi ChatGPT.

BAB VI Kesimpulan dan Saran

6.1 Kesimpulan

1. **Efektivitas Model IndoBERT:** Model Deep Learning berbasis Transformer berhasil diimplementasikan untuk klasifikasi sentimen Bahasa Indonesia. Akurasi pengujian mencapai sekitar 90% pada dataset yang telah diseimbangkan, menunjukkan kemampuan model dalam memahami pola bahasa ulasan aplikasi yang informal dan tidak baku.
2. **Keberhasilan Strategi Rekayasa Data:**
   - Masalah kesulitan mendeteksi kelas sentimen "Netral" atau "Campuran" berhasil diatasi.
   - *Anotasi Hibrida* terbukti efektif memperkaya kualitas data latih kelas Netral dengan menangkap ulasan bersentimen campuran (contoh: "aplikasi bagus tapi berbayar").
   - Kombinasi *Penyeimbangan Data* (*undersampling* mayoritas, *oversampling* minoritas hingga 3.000 data) dan *Label Smoothing* terbukti mencegah *overfitting*. F1-Score kelas Netral mencapai $\sim$ 0,91, dari yang sebelumnya mendekati 0 pada percobaan awal.
3. **Perbandingan dengan TF-IDF:** IndoBERT lebih unggul dibandingkan metode klasik TF-IDF, khususnya dalam menangani kalimat dengan konteks kompleks. IndoBERT mampu memberikan akurasi dan F1-Score Netral yang lebih tinggi serta tidak memerlukan prapemrosesan manual yang memakan waktu seperti TF-IDF.

6.2 Saran

1. **Perluasan Domain Data (Domain Adaptation):** Model saat ini dilatih spesifik pada domain ulasan aplikasi teknologi. Pengujian menunjukkan penurunan performa signifikan (*domain mismatch*) pada teks dari domain lain. Penelitian lanjutan dapat memperluas data latih dengan berbagai domain agar model lebih *robust* dan *general*.
2. **Analisis Aspek (Aspect-Based Sentiment Analysis):** Saat ini model hanya memberikan sentimen secara umum (positif/negatif/netral). Pengembangan selanjutnya disarankan menerapkan *Aspect-Based Sentiment Analysis* (ABSA) untuk mendeteksi sentimen pada aspek spesifik (misal: "Positif pada aspek Fitur" tetapi "Negatif pada aspek Harga").

3. **Deployment ke Lingkungan Produksi:** Sistem demo masih berjalan di lingkungan pengembangan (Google Colab). Untuk implementasi nyata, disarankan *deployment* ke layanan *cloud* (Google Cloud Run, AWS) menggunakan kontainerisasi (Docker) agar API dapat diakses secara publik dan stabil 24/7.

Lampiran

Link Dataset: `https://www.kaggle.com/datasets/ahmadseloabadi/chatgpt-reviews-from-google-play-store`

Dosen Pengampu: Syahru Rahmayuda, S.Kom., M.Kom.

Tim Penyusun:

- Lazzu Fadlin Muslim (H1101231012)
- M. Fathurrochim (H1101231034)
- M. Raya Bilfikri (H1101231040)
- Rohim Amrullah (H1101231052)

Program Studi Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura, Pontianak, 2025.

Daftar Pustaka

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

IndoBenchmark. (2020). *IndoBERT: Pre-trained Language Model for Indonesian*. Diakses melalui Hugging Face Model Hub: `https://huggingface.co/indobenchmark/indobert-base-p1`.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Rangkuti, F. (2015). *Analisis SWOT: Teknik Membedah Kasus Bisnis*. Gramedia Pustaka Utama.

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Packt Publishing.

Sastrawi. (2017). *Sastrawi: Python Library for Indonesian Language Preprocessing*. Tersedia di: `https://github.com/sastrawi/sastrawi`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems* (NeurIPS).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... & Rush, A. M. (2020). Transformers: State-of-the-art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.