PENERAPAN MODEL INDOBERT UNTUK DETEKSI POTENSI SUMBER STRES DALAM TEKS MEDIA SOSIAL

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang

DISUSUN OLEH :

M.SIROJUDIN MAHDI FAZA
32602100069

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

APPLICATION OF THE INDOBERT MODEL TO DETECTION OF POTENTIAL SOURCES OF STRESS IN SOCIAL MEDIA TEXTS

FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (S1) at Informatics Engineering Departement of Industrial Technology Faculty Sultan Agung Islamic University

ARRANGED BY:

M.SIROJUDIN MAHDI FAZA

32602100069

MAJORING OF INFORMATICS ENGINEERING

INDUSTRIAL TECHNOLOGY FACULTY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

2025

LEMBAR PENGESAHAN

TUGAS AKHIR

PENERAPAN MODEL INDOBERT UNTUK DETEKSI POTENSI SUMBER STRES DALAM TEKS MEDIA SOSIAL

M.SIROJUDIN MAHDI FAZA
NIM 32602100069

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 02 September 2025

TIM PENGUJI UJIAN SARJANA:

Ir.Sri Mulyono, M.Eng 04-09-2025
NIK. 210616049
(Penguji 1)

Andi Riansyah, ST., M.Kom 04-09-2025
NIK. 210616053
(Penguji 2)

Moch Taufik, S.T., M.IT 04-09-2025
NIK. 210604034
(Pembimbing)

Semarang, 04 September 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch Taufik, S.T., M.IT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : M.Sirojudin Mahdi Faza

NIM : 32602100069

Judul Tugas Akhir : PENERAPAN MODEL INDOBERT UNTUK DETEKSI POTENSI SUMBER STRES DALAM TEKS MEDIA SOSIAL

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apbila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 12 September 2025

Yang Menyatakan,

M.Sirojudin Mahdi Faza

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: M.Sirojudin Mahdi Faza
NIM	: 32602100069
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri
Alamat Asal	: Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Penerapan Model Indobert Untuk Deteksi Potensi Sumber Stres Dalam Teks Media Sosial.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 12 September 2025

Yang menyatakan,

M.Sirojudin Mahdi Faza

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Penerapan Model Indobert Untuk Deteksi Potensi Sumber Stres Dalam Teks Media Sosial." ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
3. Dosen pembimbing penulis Bapak Moch.Taufik.,ST,MIT yang telah memberikan arahan, bimbingan, dan saran yang berarti dalam penyelesaian tugas akhir ini.
4. Seluruh dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah memberikan ilmunya kepada penulis.
5. Orang tua penulis, serta kedua adik penulis yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
6. Teman-teman seperjuangan atas kebersamaanya yang telah bekerja keras serta semangat dalam proses penyelesaian tugas akhir ini.
7. Terima kasih juga untuk Shaun the sheep, Adella, dan Mahesa music yang sudah menjadi playlist dalam menemani penulis dalam pembuatan tugas akhir.
8. Penulis juga menyampaikan terima kasih kepada Deankt dan Nastasia Adeline yang melalui live streaming menghibur senantiasa memberikan motivasi dan semangat selama proses penyusunan tugas akhir ini.

Semarang, 24 September 2025

M.Sirojudin Mahdi Faza

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Media sosial, khususnya Twitter, sering digunakan untuk mengekspresikan pengalaman pribadi yang berpotensi memuat ungkapan stres. Penelitian ini bertujuan untuk menerapkan model IndoBERT dalam mengklasifikasikan potensi sumber stres pada teks media sosial berbahasa Indonesia. Data diperoleh melalui proses *scraping* Twitter, kemudian diproses dengan normalisasi, pembersihan, dan tokenisasi sebelum diberi label ke dalam kategori akademik, hubungan, kesehatan, pekerjaan, dan keuangan. Model IndoBERT dilatih dengan pendekatan *supervised learning* melalui *fine-tuning*, dan performanya diukur menggunakan *precision, recall*, serta *F1-score*. Hasil pengujian menunjukkan bahwa IndoBERT mampu melakukan klasifikasi dengan sangat baik, dengan rata-rata *precision* sebesar 0.9842, *recall* sebesar 0.9841, dan *F1-score* sebesar 0.9831. Kesimpulan penelitian ini adalah IndoBERT efektif digunakan untuk mendeteksi potensi sumber stres pada teks media sosial, sekaligus berkontribusi pada pengembangan NLP untuk kesehatan mental digital.

Kata kunci: IndoBERT, klasifikasi teks, media sosial, potensi stres

ABSTRACT

Social media, particularly Twitter, is often used to express personal experiences that may contain signs of stress. This study aims to apply the IndoBERT model to classify potential sources of stress in Indonesian-language social media texts. The dataset was collected through Twitter scraping, followed by preprocessing steps including normalization, cleaning, and tokenization, before being labeled into categories such as academic, relationship, health, work, and finance. IndoBERT was fine-tuned using a supervised learning approach, and its performance was evaluated using precision, recall, and F1-score. The results show that IndoBERT achieved strong performance with an average precision of 0.9748, recall of 0.9742, and F1-score of 0.9731. The findings conclude that IndoBERT is effective for detecting potential stress sources in social media texts and contributes to the advancement of NLP for digital mental health.

Keywords: IndoBERT, text classification, social media, potential stress.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era digital sekarang, media sosial terutama Twitter telah menjadi platform yang signifikan bagi pengguna untuk menyampaikan serta mengekspresikan perasaan, sifat dasar, emosi, dan bahkan kegelisahan yang dapat memengaruhi kesehatan mental atau stres. Stres merupakan fenomena mental atau fisik yang muncul dari penilaian kognitif individu terhadap rangsangan dan hasil interaksinya dengan lingkungan. Seperti penyakit lainnya, stres perlu segera diatasi agar tidak mengganggu kehidupan sehari-hari (Johan & Aurelia Azka, 2023).

Stres dapat dialami oleh siapa saja, termasuk anak-anak, orang dewasa, hingga lansia. Penyebabnya pun bervariasi, seperti tekanan sekolah, beban kerja, situasi keluarga, serta lingkungan sekitarnya. Jika tidak ditangani dan terus berlanjut, stres yang berlebihan dapat menyebabkan timbulnya berbagai penyakit (Ria Wiyani, 2022)

Semakin berkembangnya kecerdasan buatan, terdapat berbagai metode *machine learning* dan *deep learning* yang banyak diterapkan untuk klasifikasi teks dalam bidang NLP (*Natural Language Processing*) salah satunya IndoBERT, IndoBERT adalah hasil modifikasi dari BERT (*Bidirectional Encoder Representations from Transformers*) yang dilatih khusus untuk Bahasa Indonesia, IndoBERT telah terbukti memberikan kinerja yang baik dalam tugas-tugas NLP yang berfokus pada Bahasa Indonesia, seperti klasifikasi teks (Saputra dkk., 2025).

Penerapan model IndoBERT dalam klasifikasi teks untuk mendeteksi potensi sumber stres di media sosial diharapkan dapat membantu mengurangi risiko gangguan kesehatan mental. Hasil deteksi ini berfungsi sebagai alat bantu deteksi dini, sehingga stres yang dialami individu tidak berkembang menjadi gangguan kesehatan mental yang lebih berat. Hal ini penting karena gangguan kesehatan mental yang berkepanjangan dapat menimbulkan dampak serius terhadap kesehatan, bahkan dapat memperburuk kondisi individu. Selain itu, gangguan mental juga berpotensi menimbulkan beban psikologis dan ekonomi bagi keluarga, masyarakat, serta pemerintah (Erzha Tri Setyo Rochman dkk., 2024).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menerapkan model IndoBERT dalam mendeteksi potensi sumber stres pada teks media sosial twitter.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah :

1. Bagaimana penerapan model IndoBERT dalam melakukan klasifikasi teks untuk mendeteksi potensi sumber stres pada media sosial twitter?
2. Sejauh mana model IndoBERT dapat mengidentifikasi potensi sumber stres berdasarkan kategori tertentu (misalnya pekerjaan, akademik, hubungan, keuangan, kesehatan, dan lain-lain)?
3. Kategori potensi sumber stres apa saja yang paling sering muncul pada hasil klasifikasi teks media sosial twitter?
4. Bagaimana performa model yang dihasilkan dalam mendeteksi potensi sumber stres pada teks media sosial twitter?

1.3 Batasan Masalah

Untuk menjaga fokus dan ruang lingkup penelitian, terdapat beberapa batasan yang ditetapkan yaitu:

1. Penelitian ini hanya berfokus pada penerapan model IndoBERT untuk mendeteksi potensi sumber stres dalam teks media sosial Twitter, dengan dataset yang diperoleh melalui proses *scraping*.
2. Dataset yang dianalisis hanya berisi cuitan yang berpotensi menjadi sumber stres dari berbagai pengguna Twitter, dengan rentang usia pengguna dari mahasiswa, pekerja, hingga yang sudah berkeluarga.
3. Penelitian ini memiliki keterbatasan pada jumlah data antar kategori yang tidak seimbang. Misalnya, kategori Kesehatan memiliki jumlah data terbanyak yaitu 1.298 cuitan, sedangkan terendah yaitu kategori Keluarga hanya memiliki 19 cuitan dan kategori Keuangan hanya 17 cuitan.

4. Kategori sumber stres yang digunakan ditentukan berdasarkan kata kunci tertentu, meliputi pekerjaan, akademik, hubungan, keuangan, kesehatan, serta faktor relevan lainnya.
5. Evaluasi performa model dalam penelitian ini hanya menggunakan metrik *precision*, *recall*, dan *F1-score*.

1.4 Tujuan Penelitian

1. Menerapkan model IndoBERT dalam melakukan klasifikasi teks media sosial twitter untuk mendeteksi potensi sumber stres.
2. Mengidentifikasi kategori potensi sumber stres berdasarkan hasil klasifikasi.
3. Mengetahui kategori potensi sumber stres yang paling sering muncul dari hasil klasifikasi teks.
4. Mengevaluasi performa model menggunakan metrik evaluasi.

1.5 Manfaat Penelitian

1. Memberikan kontribusi pada pengembangan NLP bahasa Indonesia melalui penerapan IndoBERT untuk klasifikasi teks.
2. Memberikan gambaran mengenai kategori sumber stres yang paling sering muncul di media sosial.
3. Informasi ini dapat dimanfaatkan oleh pihak terkait. Misalnya, apabila kategori Pekerjaan menjadi yang paling banyak muncul, perusahaan dapat menggunakan informasi tersebut sebagai dasar untuk menyediakan program konseling atau manajemen stres bagi karyawan.

1.6 Sistematika Penulisan

Struktur penulisan yang akan diterapkan dalam pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, Batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penelitian

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang mendukung, termasuk konsep stres, NLP, dan model IndoBERT..

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan tahapan penelitian mulai dari pengumpulan data hingga evaluasi performa.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari pembuatan sistem sampai dengan proses deployment.

BAB V : KESIMPULAN DAN SARAN

Pasa BAB V merangkum keseluruhan proses penelitian dari awal sampai akhir.

BAB II
TINJAUNA PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Model IndoBERT terbukti memiliki efektivitas tinggi dalam mendeteksi potensi depresi pada teks media sosial, dengan capaian akurasi, *precision, recall*, dan *F1-score* sebesar 94,91%. Hal ini menunjukkan bahwa IndoBERT cukup andal dalam menganalisis sentimen dan kondisi mental berdasarkan unggahan pengguna. Meskipun demikian, penelitian ini memiliki keterbatasan, di antaranya ketergantungan pada data media sosial yang belum tentu merepresentasikan kondisi emosional sesungguhnya, karena adanya faktor eksternal seperti privasi dan kecenderungan pengguna untuk tidak sepenuhnya mengekspresikan perasaan di ruang publik. Selain itu, penelitian hanya berfokus pada analisis teks tanpa melibatkan data multimodal seperti gambar atau video yang juga berpotensi memuat informasi emosional penting. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan model yang dapat mengintegrasikan data teks dan multimodal, serta melakukan validasi eksternal melalui kolaborasi dengan tenaga ahli kesehatan mental. Dengan langkah tersebut, sistem deteksi depresi berbasis *deep learning* diharapkan mampu memberikan dukungan yang lebih optimal bagi upaya peningkatan kesehatan mental Masyarakat (Situmorang & Purba, 2024).

Berdasarkan penelitian mengenai penerapan *fine-tuning* model IndoBERT untuk mendeteksi berita hoaks politik, diperoleh hasil bahwa model ini dapat mengklasifikasikan berita politik sebagai fakta atau hoaks dengan performa yang sangat baik. Model ini berhasil mencapai akurasi sebesar 95% pada data uji dengan nilai AUC 0.946. Kelebihan IndoBERT terlihat dari kemampuannya dalam memahami bahasa Indonesia dalam konteks berita politik, yang memungkinkan untuk menghasilkan prediksi yang tepat dan relevan. Meski demikian, terdapat keterbatasan pada penelitian ini, yakni model masih cukup sensitif terhadap bias dataset, terutama ketika distribusi data antara berita fakta dan hoaks tidak seimbang (Jocelynne dkk., 2025).

Hasil penelitian ini membuktikan bahwa model IndoBERT yang di *fine-tuning* dapat dimanfaatkan secara efektif dalam tugas klasifikasi berita hoaks berbahasa Indonesia, meskipun data yang digunakan memiliki distribusi yang tidak seimbang. Penerapan *Focal Loss* terbukti meningkatkan kinerja model, terutama pada kelas minoritas (berita hoaks). Model yang dihasilkan mampu mencapai akurasi keseluruhan sebesar 98,3%, dengan peningkatan signifikan pada nilai *F1-score* dan *Recall* untuk kelas hoaks dibandingkan dengan penggunaan *Cross Entropy Loss*. Pendekatan ini membuat model lebih peka terhadap kesalahan klasifikasi pada kelas yang kurang terwakili, tanpa harus menerapkan teknik manipulasi data seperti oversampling, undersampling, ataupun *re-weighting*. Oleh karena itu, penggunaan *Focal Loss* dalam pelatihan IndoBERT menjadi alternatif yang efisien dan stabil untuk mengatasi permasalahan ketidakseimbangan data pada klasifikasi berita. Penelitian lanjutan disarankan untuk mengombinasikan pendekatan ini dengan strategi augmentasi data atau pemanfaatan model multilingual guna memperluas kemampuan model terhadap data baru (Kunaefi dkk., 2025).

Pada penelitian ini sejumlah platform pihak ketiga memang menyediakan akses ke media sosial populer lain, seperti akun publik di Facebook, Instagram, dan YouTube, serta ke situs berita daring dan forum diskusi. Umumnya, peneliti masih dapat melakukan penyaringan berdasarkan bahasa, rentang waktu, lokasi, atau sumber, meskipun ketersediaannya berbeda pada setiap platform. Namun, akses yang diberikan jarang bersifat tanpa batas, misalnya hanya membolehkan maksimal 50.000 penyebutan per pencarian, pembatasan kuota bulanan hingga tiga juta, atau rentang waktu pencarian yang terbatas satu hingga tiga tahun ke belakang. Keterbatasan ini memengaruhi kualitas data, sehingga menjadikan Twitter sebagai pilihan yang lebih baik karena menyediakan API resmi (*standard, premium, enterprise*, hingga *Academic Research* API) dengan akses data yang lebih terbuka, luas, dan legal untuk penelitian (Chen dkk., 2022).

Penelitian ini menunjukkan bahwa IndoBERT, khususnya model *fine-tuned*, lebih unggul dibandingkan Bi-LSTM dalam klasifikasi emosi teks berbahasa Indonesia dari Twitter. Dengan dataset 7.629 tweet (6 emosi dasar + 1 netral),

IndoBERT *fine-tuned* mencapai akurasi tertinggi 93,8%, sementara Bi-LSTM terbaik hanya 84%. Model mampu mengenali emosi *jijik* dengan baik, namun kesulitan pada emosi netral. Kontribusi utama penelitian ini adalah penyediaan dataset emosi berbahasa Indonesia yang akan dipublikasikan, serta rekomendasi arah lanjutan berupa perluasan dataset, penambahan modalitas (suara/gambar), dan eksplorasi arsitektur *deep learning* lain untuk meningkatkan performa, dengan potensi penerapan pada sistem afektif seperti *chatbot* atau manusia *virtual* (William dkk., 2024).

Pada penelitian ini mengusulkan penggunaan model bahasa berbasis *transformator* yang telah dilatih sebelumnya pada korpus besar berbahasa Indonesia untuk tugas klasifikasi aktivitas ekonomi di Indonesia. Hasil penelitian menunjukkan bahwa model yang diusulkan, yaitu IndoBERTLARGE, secara konsisten memberikan performa lebih baik dibandingkan model dasar. Secara rinci, IndoBERTLARGE berhasil mencapai nilai F1 sebesar 96,82%, *balanced accuracy* (BA) 96,10%, serta *recall* 96,96%. Selain itu, model ini menunjukkan tingkat *false positive rate* (FPR) yang rendah, yaitu 0,217%, dan memperoleh nilai *Imbalance Accuracy Metric* (IAM) sebesar 0,771. Temuan ini juga menegaskan adanya peningkatan kinerja yang signifikan, di mana IndoBERTLARGE mampu meningkatkan skor F1 model dasar hingga 7,37% pada *CatBoost* dan 1,55% pada DistilBERT. Meskipun demikian, karena perbedaan performa antara IndoBERTBASE dan DistilBERT relatif kecil, kedua model tersebut masih dianggap layak dan berpotensi digunakan sebagai *predictor* (Syazali & Yulianti, 2025).

2.2 Dasar Teori

2.2.1 Stres

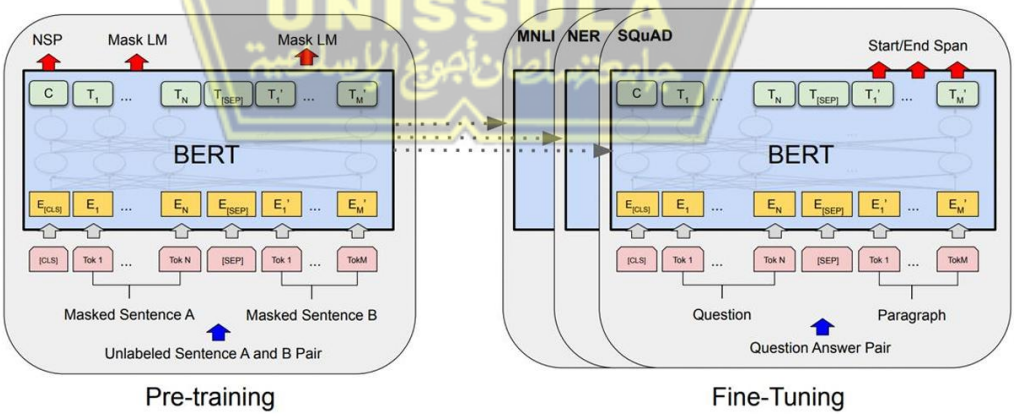
Stres adalah rasa tertekan yang dirasakan seseorang. Orang yang sedang mengalami stres biasanya akan bersikap berbeda dengan orang yang tidak sedang stres. Gejala stres ini bisa terlihat dari kondisi fisik maupun keadaan psikologisnya (Hafidh dkk., 2023), Hal ini menunjukkan bahwa stres tidak hanya memengaruhi

kesehatan tubuh, tetapi juga berdampak pada pola pikir, perilaku, dan kemampuan seseorang dalam menjalani aktivitas sehari-hari.

Stres juga dapat diartikan sebagai kondisi ketegangan emosional yang muncul akibat situasi yang menekan dan sulit dihindari. Situasi ini dapat disebabkan oleh berbagai faktor, seperti tekanan pekerjaan, pertikaian dalam hubungan sosial, masalah finansial, serta pengalaman traumatis (Larasati dkk., 2024).

2.2.2 IndoBERT (*Indonesian Bidirectional Encoder Representations from Transformers*)

IndoBERT merupakan model bahasa yang dikembangkan secara khusus untuk Bahasa Indonesia dengan menggunakan platform *Huggingface*. Model ini berbasis pada arsitektur Bidirectional Encoder Representations from Transformers (BERT) yang kemudian disesuaikan agar lebih optimal dalam memahami teks berbahasa Indonesia (Hakim & Riana, 2024). Sama seperti BERT, IndoBERT menjalani dua fase utama, yaitu pre-training dan fine-tuning. Pada fase pre-training, model dilatih dengan data tak berlabel untuk mengenali pola bahasa. Selanjutnya, dalam tahap fine-tuning, model menggunakan parameter dari pre-training yang telah didapatkan dan disesuaikan kembali dengan data berlabel untuk meningkatkan kemampuannya dalam melakukan tugas klasifikasi (Sayarizki & Nurrahmi, 2024). Berikut Ilustrasi tahap *pre-training* dan *fine tuning* ditunjukkan pada gambar:



Gambar 2. 1 Ilustrasi tahap *pre-training* dan *fine-tuning*

Menurut gambar 2.1, proses *pre-training* dan *fine-tuning* pada model BERT memiliki struktur yang sama, tetapi berbeda pada bagian layer keluarannya. Pada tahap *fine-tuning*, model memanfaatkan parameter yang sudah diperoleh dari pre-

training, lalu disesuaikan kembali menggunakan data berlabel sesuai dengan tugas tertentu. Dalam tahap ini, token khusus seperti [CLS] dan [SEP] ditambahkan pada input untuk membantu model memanfaatkan pengetahuan awal yang sudah dipelajari dan menyesuaikannya dengan kebutuhan tugas spesifik.

Sementara itu, pada tahap *pre-training* yang ditunjukkan di sisi kiri gambar, BERT dilatih melalui dua jenis tugas utama, yaitu *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP). Pada tugas MLM, sejumlah kata dalam kalimat disembunyikan secara acak, dan model diberi tugas untuk menebak kata-kata yang hilang tersebut, sebagaimana ditandai dengan kotak hijau "Mask LM." Sedangkan pada tugas NSP, model dilatih untuk memprediksi apakah sebuah kalimat B benar-benar merupakan kelanjutan dari kalimat A dalam teks asli, yang digambarkan dengan kotak merah "NSP." Dengan demikian, selama pre-training, BERT memproses pasangan kalimat yang tidak berlabel, baik berupa kalimat berurutan maupun tidak, serta kalimat dengan kata yang ditutup sebagian untuk menyelesaikan tugas NSP dan MLM.

2.2.3 *Text Classification*

Text Classification atau Klasifikasi teks, yang merupakan salah satu dasar dalam *Natural Language Processing* (NLP), adalah proses mengelompokkan data tekstual secara otomatis ke dalam kategori yang telah ditentukan sebelumnya. Teknik ini berperan penting dalam membantu pengelolaan serta pengambilan informasi dari jumlah teks yang terus bertambah setiap harinya. Secara umum, klasifikasi teks dapat diartikan sebagai proses penempatan suatu dokumen ke dalam satu atau lebih kategori berdasarkan isi maupun makna yang terkandung di dalamnya (Firizkiansah dkk., 2025).

Masalah klasifikasi teks telah lama menjadi topik penelitian dan diterapkan dalam berbagai bidang nyata selama beberapa dekade terakhir. Dengan adanya kemajuan pesat dalam *Natural Language Processing* (NLP) dan teknik penambangan teks, semakin banyak peneliti yang tertarik untuk merancang serta mengembangkan aplikasi yang berbasis pada metode klasifikasi teks (Kowsari dkk., 2019).

2.2.4 *Deep Learning*

Deep Learning adalah bagian dari *Machine Learning* yang menggunakan jaringan saraf buatan (*Artificial Neural Networks*) dengan banyak lapisan (*deep neural networks*) untuk memproses data yang rumit dan menyelesaikan berbagai tugas seperti pengenalan pola, klasifikasi, dan prediksi. Teknologi ini banyak digunakan dalam bidang pengenalan emosi karena metode tradisional dianggap kurang cepat dan kurang akurat. Selain itu, ukuran data yang besar seringkali membuat metode konvensional menjadi lambat diproses dan membutuhkan biaya yang lebih tinggi (Rahmadani dkk., 2022).

Penerapan *deep learning*, yang merupakan salah satu cabang dari *machine learning* dengan menggunakan jaringan syaraf tiruan (*neural network*) yang berlapis, menawarkan solusi yang lebih maju dan efektif. *Deep learning* merupakan pendekatan pembelajaran mesin berbasis jaringan saraf tiruan yang mampu mempelajari representasi data secara mendalam serta mengenali pola-pola abstrak yang tersembunyi. Metode ini bekerja melalui beberapa tingkat representasi, di mana setiap lapisan sederhana mengubah data dari bentuk awal (input mentah) menjadi representasi yang lebih tinggi dan kompleks pada tahap berikutnya (Prasetyo & Dewayanto, 2024).

2.2.5 Media Sosial

Media sosial adalah platform online yang memungkinkan penggunanya untuk terlibat, membagikan, dan menciptakan berbagai jenis konten, seperti blog, jaringan sosial, wiki, forum, serta dunia maya. Beberapa manfaat positif dari media sosial meliputi memudahkan interaksi dengan banyak orang, memperluas hubungan pertemanan, menghapus batas jarak dan waktu, menyediakan ruang untuk berekspresi, menyebarkan informasi secara cepat, serta menawarkan biaya yang cenderung lebih rendah (Kustiawan dkk., 2022).

Media sosial merupakan hasil perkembangan teknologi komunikasi yang berawal dari kemunculan internet dan komputer sebagai sarana komunikasi digital. Kehadirannya telah membawa perubahan besar dalam cara manusia berinteraksi, bertukar informasi, serta membangun komunitas global. Sejak kemunculannya pada awal tahun 2000-an, media sosial mengalami kemajuan

yang sangat cepat, Dimulai dari platform sederhana seperti Friendster dan MySpace, Kini, media sosial telah menjadi unsur penting dalam kehidupan sehari-hari. Banyak platform terkenal seperti Facebook, Instagram, Twitter, TikTok, dan YouTube dipakai oleh jutaan hingga miliaran orang di seluruh dunia (Qadir & Ramli, 2024).

BAB III
METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian digunakan untuk merencanakan, melaksanakan, dan menganalisis penelitian. Metode penelitian ini dapat membantu dalam merancang prosedur yang tepat untuk menyusun data bermanfaat, ini adalah beberapa metode penelitian yang digunakan:

Gambar 3. 1 Alur Penelitian

Berdasarkan gambar 3.1 langkah -langkah yang akan dilakukan pada tahapan ini yaitu memulai dengan studi literatur, ditahap kedua yaitu pengumpulan dan pengolahan data, kemudian perancangan sistem, setelah itu *preprocessing data* dan dilanjutkan pelatihan model, tahap keenam yaitu pengujian atau evaluasi model, lalu diakhiri dengan tahap deployment model.

3.2 Studi Literatur

Langkah pertama yang dilakukan dalam penelitian ini adalah mengidentifikasi permasalahan yang akan diangkat. Proses ini dilakukan dengan menelusuri berbagai sumber ilmiah, seperti jurnal, prosiding, skripsi, maupun

laporan penelitian yang relevan. Fokus kajian diarahkan pada topik deteksi potensi sumber stres, pemrosesan bahasa alami (NLP), serta pemanfaatan model transformer seperti *IndoBERT*, yang telah terbukti efektif dalam berbagai tugas NLP, termasuk *Text Classification*, karena kemampuannya memahami konteks kata secara lebih mendalam.

3.3 Pengumpulan Data

Pada tahap yang kedua, data yang digunakan dalam penelitian ini diperoleh melalui proses pengumpulan dari media sosial Twitter. Pengumpulan menggunakan pendekatan web scraping untuk mendapatkan data berupa teks yang relevan dengan penelitian. Dalam hal ini, tweet dikumpulkan berdasarkan kata kunci tertentu, seperti "jenuh", dan data dibatasi dalam rentang waktu dari 1 Januari 2020 sampai 31 Desember 2024, data yang dibutuhkan hanya kolom "*full text*".

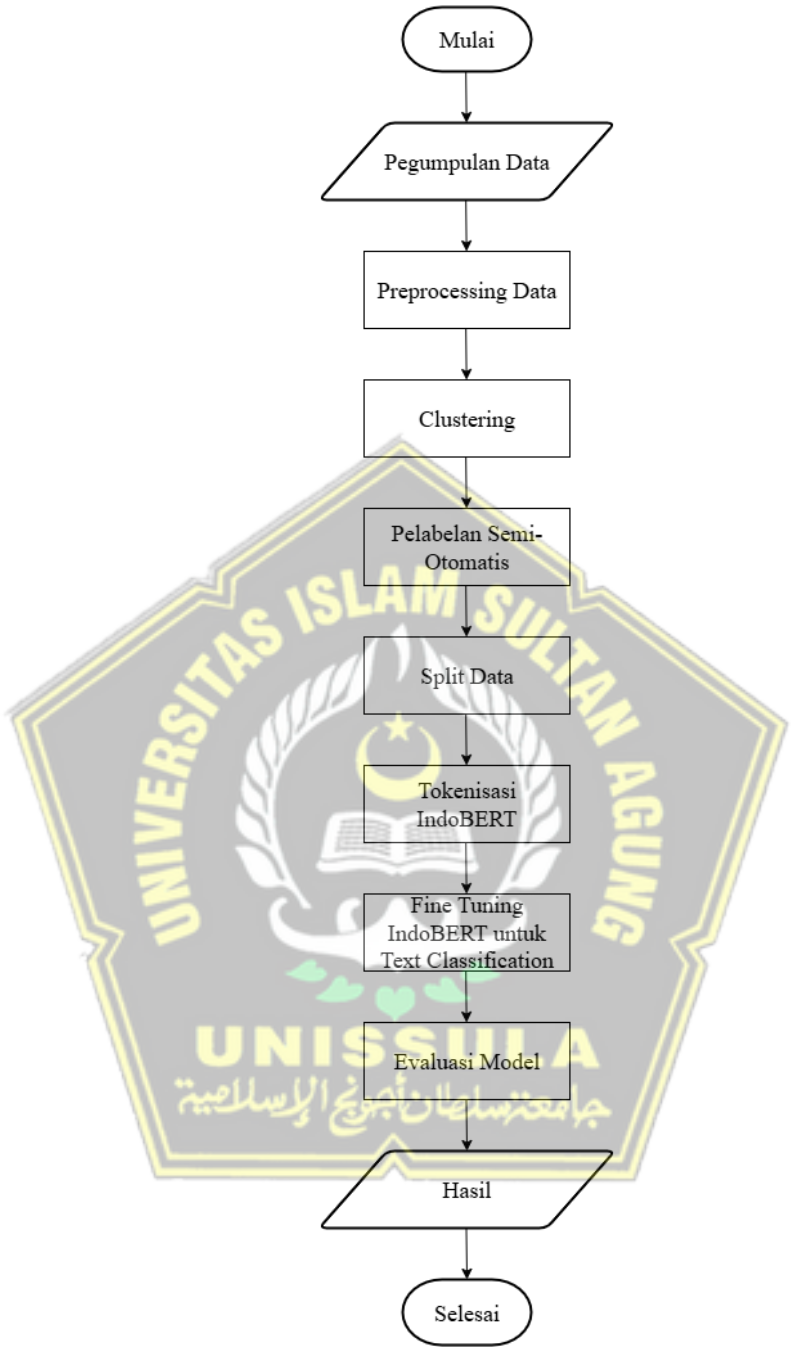
	conversation_id_str	created_at	favorite_count	full_text	id_str
0	1873381167849935027	Sun Dec 29 14:50:50 +0000 2024	4	Lebih mending beli setik daripada sego goreng ...	1873381167849935027
1	1873379755375853900	Sun Dec 29 14:45:14 +0000 2024	67	Work! Gimana cara balikin semangat kerja ya? L...	1873379755375853900
2	1873377115992928720	Sun Dec 29 14:34:44 +0000 2024	0	Monoton Gara2 kerja terus hidupku beneran berp...	1873377115992928720
3	1872704599716778114	Sun Dec 29 14:33:22 +0000 2024	0	@tjerealjoker Kawan awak dah merepek kat letri...	1873376769002308044
4	1873373727024611837	Sun Dec 29 14:31:07 +0000 2024	1	@rlo_dll apa itu? tenggara?	1873376205287911522

Gambar 3. 2 Data hasil scraping dari twitter

Pada Gambar 3.2 ditampilkan hasil scraping data dari Twitter yang berhasil dikumpulkan dalam bentuk tabel. Setiap baris mewakili satu cuitan (tweet) yang diperoleh, dengan beberapa kolom informasi seperti isi teks cuitan, waktu unggah, jumlah suka (likes), serta identitas unik tweet. Namun, dalam penelitian ini hanya digunakan kolom teks cuitan sebagai data utama untuk di*gunakan.*

3.4 Perancangan Sistem

Pada tahap ini akan ditentukan alur kerja sistem berupa *flowchart* yang menggambarkan proses penelitian. Sistem dirancang untuk melakukan deteksi potensi sumber stres dalam teks media sosial menggunakan pendekatan model *Indonesian Bidirectional Encoder Representations from Transformers* (IndoBERT). Model ini berfungsi untuk menganalisis teks secara menyeluruh, kemudian dilakukan metode *text classification* untuk mengelompokkan teks ke dalam kategori sumber stres yang telah ditentukan.



Gambar 3. 3 *Flowchart* Sistem

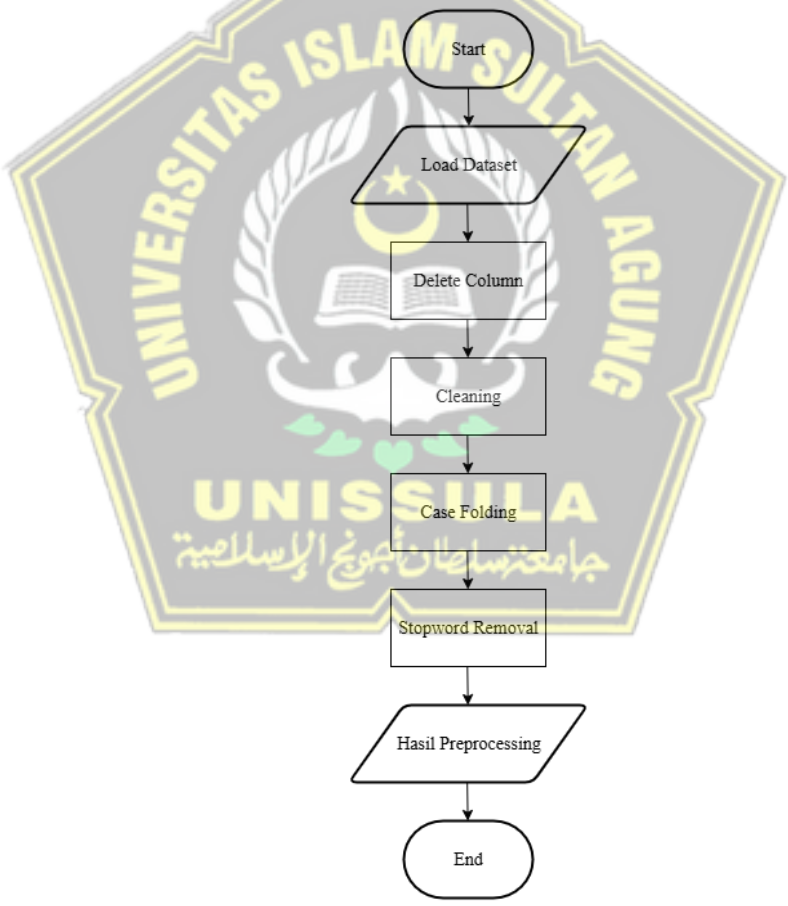
Gambar 3.3 menunjukkan flowchart sistem yang menggambarkan alur kerja proses deteksi sumber stres menggunakan model *IndoBERT* dan metode *text classification*. Flowchart ini memaparkan tahapan-tahapan penting yang dilalui selama penelitian, mulai dari pengumpulan data hingga diperoleh hasil deteksi. Penjelasan dari masing-masing tahapan sebagai berikut:

3.4.1 Pengumpulan Data

Data dikumpulkan melalui proses scraping data dari media sosial twitter dengan menggunakan *library phyton*, data ini berisi cuitan pengguna twitter berbahasa indonesia yang mengandung kata yang berkaitan dengan potensi sumber stres.

3.4.2 *Preprocessing*

Pada tahapan proses *preprocessing data* menjadi langkah penting untuk memastikan kualitas input yang akan diterima oleh model IndoBERT. Dataset yang digunakan berupa teks atau kalimat hasil scraping dari media sosial Twitter yang mengandung kata kata yang berkaitan dengan potensi sumber stres.



Gambar 3. 4 *Flowchart Preprocessing*

Pada Gambar 3.4 diatas adalah alur *Preprocessing* yang diawali dengan tahapan *Delete Column* kemudian lanjut pembersihan data (*cleaning*) yang

mencakup menghapus url, mention, hastag, angka, dan tanda baca, kemudian *Case Folding, Stopword,* dan setelah itu hasil dari *Preprocessing* disimpan.

1. ***Load Dataset***

   Tahap awal adalah memuat dataset (*load dataset*) yang berisi kumpulan teks dari Twitter. Data diperoleh melalui proses *scraping*, yaitu pengambilan data secara otomatis dari platform Twitter dengan menggunakan *tools* atau *library* tertentu. Dataset yang diperoleh masih berupa data mentah, sehingga di dalam teks masih terdapat banyak elemen yang tidak relevan, seperti tautan, mention, hashtag, angka, maupun tanda baca.

2. ***Delete Column***

   Setelah dataset berhasil dimuat, selanjutnya adalah *delete column*. Dataset yang diperoleh dari Twitter biasanya memiliki banyak kolom tambahan seperti identitas pengguna, waktu unggahan, atau jumlah interaksi yang tidak diperlukan dalam penelitian ini. Oleh karena itu, hanya kolom teks yang berisi cuitan serta kolom label yang dipertahankan, sementara kolom-kolom lain dihapus. Proses ini membuat dataset menjadi lebih sederhana, ringkas, dan terfokus pada data yang memang relevan dengan tujuan penelitian.

3. ***Cleaning***

   Tahap selanjutnya adalah cleaning atau pembersihan data. Pada tahap ini, teks dari Twitter yang masih berupa data mentah dibersihkan dari berbagai elemen yang tidak relevan agar lebih siap digunakan pada proses analisis berikutnya. Proses cleaning dilakukan melalui beberapa langkah, yaitu:

a. Menghapus URL

   Teks Twitter sering kali mengandung tautan (*link*) yang tidak memiliki makna penting untuk analisis. Misalnya teks “lagi stress banget, cek ini deh https://bit.ly/abc123”. Bagian URL seperti https://bit.ly/abc123 akan dihapus karena tidak berkontribusi pada isi pesan utama.

b. Menghapus *Mention*

   Twitter banyak menggunakan mention untuk menandai akun lain, contohnya “@teman lagi pusing banget dengan tugas”. *Mention* @teman tidak diperlukan dalam analisis sehingga dihapus agar teks lebih fokus pada isi pernyataan.

c. Menghapus Hastag

Hashtag sering muncul untuk menandai topik tertentu, contohnya "stress gara-gara deadline #kuliah". Simbol #kuliah biasanya dihapus, namun kata "kuliah" tetap bisa dipertahankan jika relevan, karena memiliki arti penting dalam identifikasi kategori stres.

d. Menghapus Angka

Angka yang muncul dalam teks, misalnya "tugas numpuk ada 5", biasanya dihapus karena tidak memberikan makna penting dalam klasifikasi kategori. Angka "5" dihilangkan sehingga fokus tetap pada kata "tugas numpuk".

e. Menghapus Tanda Baca

Tanda baca seperti koma, titik, tanda seru, atau tanda tanya juga dihapus. Misalnya teks "kenapa, sih??" akan menjadi "kenapa sih". Hal ini dilakukan agar kata-kata dapat diproses lebih konsisten.

4. *Case Folding*

Selanjutnya adalah mengubah seluruh teks menjadi huruf kecil. Hal ini dilakukan agar tidak terjadi perbedaan makna hanya karena variasi penggunaan huruf besar dan huruf kecil. Dengan demikian, semua kata akan diperlakukan secara konsisten sehingga analisis data dapat berlangsung dengan lebih akurat.

5. *Stopword Removal*

Pada tahap ini, kata-kata yang dianggap umum dan tidak memberikan makna penting dalam analisis dihapus dari teks. *Stopword Removal* biasanya berupa kata-kata penghubung atau kata yang terlalu sering muncul sehingga tidak berkontribusi dalam proses klasifikasi. Dengan menghapus kata-kata tersebut, data teks menjadi lebih padat makna karena hanya menyisakan kata-kata yang berpotensi memberikan informasi penting.

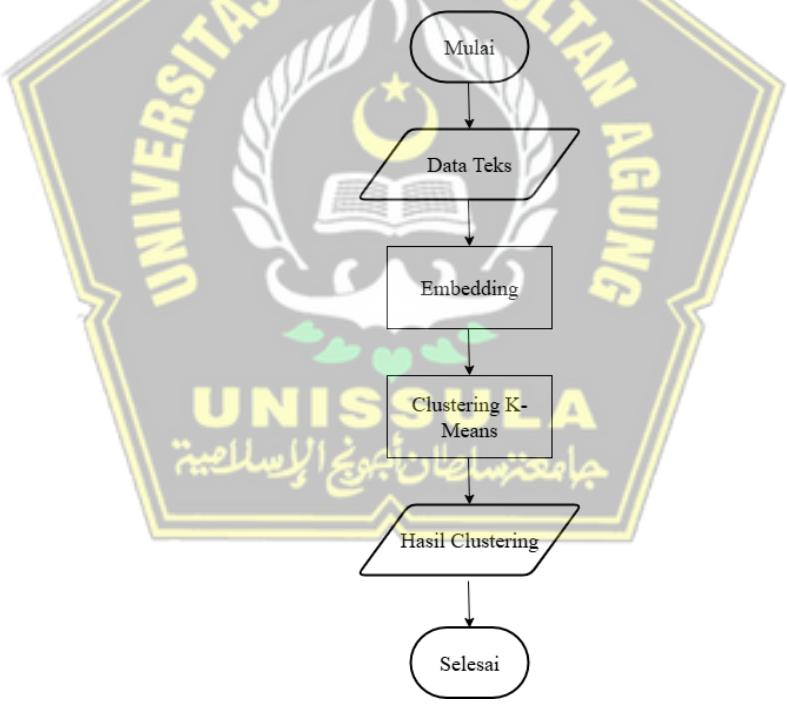
6. Hasil *Preprocessing*

Setelah melalui seluruh tahapan *preprocessing,* diperoleh data teks yang sudah bersih dan konsisten. Dengan demikian, teks hasil preprocessing lebih fokus hanya pada kata-kata penting yang relevan dengan konteks potensi stres. Hasil akhir berupa dataset yang telah terstruktur dalam bentuk *clean text* dengan

hanya menyisakan kolom yang relevan, sehingga dapat digunakan sebagai input pada tahap selanjutnya .

3.4.3 *Clustering*

Pada tahap *clustering*, data teks yang sudah melalui *preprocessing* kemudian dikelompokkan berdasarkan kemiripan makna. Sebelum masuk ke proses *clustering*, teks terlebih dahulu diubah menjadi bentuk numerik menggunakan *embedding*, sehingga setiap kalimat atau dokumen direpresentasikan sebagai vektor angka yang mencerminkan makna dari teks tersebut. Setelah data berbentuk vektor, digunakan algoritma *K-Means* untuk membagi data ke dalam beberapa kelompok (cluster). Algoritma *K-Means* mengelompokkan data teks ke dalam beberapa cluster berdasarkan kemiripan vektor *embedding*, sehingga teks dengan topik serupa akan terkumpul dalam satu kelompok.



Gambar 2. 2 *Flowchart Clustering*

1. Data Teks

Tahap Data Teks merupakan proses di mana input utama berupa kumpulan teks mentah terlebih dahulu melalui serangkaian langkah *preprocessing*. Hasil akhirnya adalah teks bersih yang tersimpan pada kolom *clean_text*. Dengan demikian, teks yang dihasilkan menjadi lebih terstruktur, ringkas, dan siap diproses lebih lanjut untuk tahap *embedding*.

2. *Embedding*

Pada tahap *Embedding*, teks yang sudah bersih pada kolom *clean_text* diubah menjadi representasi numerik menggunakan IndoBERT, menghasilkan vektor berdimensi 768 yang menyimpan informasi semantik dari teks. Karena dimensi ini terlalu besar untuk divisualisasikan, dilakukan reduksi dimensi dengan PCA sehingga menjadi 2 dimensi utama (PCA *Component* 1 dan 2). Hasil reduksi ini kemudian digunakan untuk memudahkan visualisasi pola dan pengelompokan data sebelum masuk ke tahap *clustering K-Means*.

3. *Clustering K-Means*

Tahap *Clustering K-Means* adalah proses pengelompokan data teks yang sudah direpresentasikan dalam bentuk vektor 2 dimensi hasil reduksi PCA. Algoritma ini bekerja dengan menentukan sejumlah pusat cluster (*centroid*), lalu setiap data ditempatkan pada cluster terdekat berdasarkan jarak. Proses ini berlangsung secara iteratif melalui tahap penugasan data ke *centroid* terdekat dan pembaruan posisi centroid hingga stabil. Hasil akhirnya berupa pembagian teks ke dalam beberapa cluster yang menunjukkan kemiripan tema atau konteks, meskipun pada tahap ini masih berupa angka (0, 1, 2, dst) yang belum memiliki label semantik.

4. Hasil *Clustering*

Setelah *clustering* dilakukan, diperoleh hasil berupa pembagian data ke dalam beberapa kelompok (cluster) yang masih ditandai dengan angka (0, 1, 2, dan seterusnya) tanpa makna semantik. Dengan cara ini, *clustering* membantu mempermudah pada proses pelabelan semi-otomatis.

3.4.4 Pelabelan Semi-Otomatis

Setelah data teks berhasil dikelompokkan ke dalam beberapa cluster, tahap berikutnya adalah pelabelan semi-otomatis. Pada tahap ini, setiap cluster yang dihasilkan dari *K-Means* dan *embedding* dianalisis untuk mengetahui tema atau topik yang paling dominan. Proses pelabelan disebut semi-otomatis karena sistem sudah membantu mengelompokkan data berdasarkan kemiripan makna, tetapi peneliti tetap perlu melakukan pengecekan dan pemberian label kategori secara manual agar hasilnya lebih akurat.Dengan cara ini, proses pelabelan data menjadi

lebih cepat dan efisien dibanding melabeli teks satu per satu secara manual, sekaligus tetap menjaga kualitas data.

3.4.5 Split Data

Setelah data teks selesai melalui tahap pelabelan semi-otomatis, langkah selanjutnya adalah melakukan pembagian data (split data) agar dapat digunakan dalam proses pelatihan dan evaluasi model. Pada penelitian ini, data dibagi menjadi tiga bagian dengan proporsi 70% untuk data *training*, 15% untuk data validasi, dan 15% untuk data *testing*. Data *training* digunakan untuk melatih model agar dapat mengenali pola dari teks. Data validasi berfungsi untuk mengevaluasi performa model selama proses pelatihan, sehingga dapat membantu mencegah terjadinya overfitting. Sementara itu, data *testing* digunakan setelah model selesai dilatih, dengan tujuan untuk mengukur performa model secara objektif pada data baru yang belum pernah dilihat sebelumnya.

3.4.6 Tokenisasi IndoBERT

Sebelum data teks dimasukkan ke dalam model IndoBERT, data harus melalui proses tokenisasi. Tokenisasi adalah tahap pemecahan teks menjadi potongan-potongan kecil yang disebut token, agar dapat dipahami oleh model. IndoBERT memiliki tokenizer khusus yang dirancang sesuai dengan struktur bahasa Indonesia. Tokenizer ini akan mengubah teks menjadi representasi numerik berupa ID token yang sesuai dengan kosakata yang dimiliki IndoBERT. Selain itu, tokenizer IndoBERT juga menambahkan token khusus, seperti [CLS] di awal kalimat dan [SEP] di akhir kalimat, yang berguna dalam tugas klasifikasi teks. Tahap ini sangat penting karena model hanya bisa memahami data dalam bentuk token numerik, bukan teks mentah. Dengan tokenisasi, data siap diproses lebih lanjut pada tahap fine-tuning IndoBERT.

3.4.7 *Fine-Tuning* IndoBERT untuk *Text Classification*

Setelah data teks berhasil melalui tahap tokenisasi dengan tokenizer IndoBERT, langkah selanjutnya adalah melakukan *fine-tuning* IndoBERT untuk tugas klasifikasi teks. IndoBERT merupakan model bahasa pra-latih (*pre-trained language model*) yang sudah memahami struktur dan kosakata bahasa Indonesia, namun perlu disesuaikan kembali (*fine-tuning*) agar dapat bekerja optimal pada data

dan kategori yang spesifik di penelitian ini. Pada tahap ini, representasi token dari IndoBERT diteruskan ke lapisan tambahan berupa *classifier* (*fully connected layer*) yang berfungsi untuk memetakan hasil representasi teks ke dalam label kategori tertentu, seperti Pekerjaan, Akademik, Lingkungan, dan lain lain. Proses *fine-tuning* dilakukan dengan memanfaatkan data *training*, sementara data validasi digunakan untuk memantau performa model dan menghindari *overfitting*. Dengan *fine-tuning* ini, IndoBERT tidak hanya memahami bahasa Indonesia secara umum, tetapi juga menjadi lebih terlatih untuk melakukan klasifikasi teks sesuai kebutuhan penelitian.

3.4.8 Evaluasi Model

Setelah proses *fine-tuning* IndoBERT selesai, tahap selanjutnya adalah evaluasi model. Evaluasi model dalam penelitian ini bertujuan untuk mengukur performa sistem dalam mengklasifikasikan teks ke dalam kategori potensi sumber stres menggunakan pendekatan *fine-tuning* model IndoBERT. Proses evaluasi ini dilakukan dengan menggunakan data uji (*testing set*) yang telah dipisahkan sebelumnya saat tahap pembagian data. Data uji dipilih karena tidak pernah digunakan selama proses pelatihan maupun validasi, sehingga hasil evaluasi dapat mencerminkan kinerja model secara lebih objektif.

Metrik yang digunakan pada evaluasi model meliputi *precision, recall,* dan *F1-score*, yang merupakan metrik umum dalam tugas klasifikasi teks.

$$\textbf{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positive}$$

Precision digunakan untuk mengukur ketepatan model dalam memberikan prediksi yang benar pada setiap kategori,

$$\textbf{Recall} = \frac{True\ Positives}{True\ Positives + False\ Positive}$$

Recall digunakan untuk menilai sejauh mana model berhasil menemukan seluruh data yang relevan pada suatu kategori,

$$\textbf{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Sedangkan *F1-score* digunakan sebagai ukuran keseimbangan antara precision dan recall.

Berdasarkan hasil evaluasi, jika nilai *precision* tinggi tetapi *recall* rendah, maka model dapat dikatakan cukup hati-hati dan akurat dalam memberikan

prediksi, namun masih banyak teks yang seharusnya masuk ke dalam suatu kategori justru tidak terdeteksi. Sebaliknya, apabila *recall* tinggi tetapi *precision* rendah, maka model cenderung memasukkan terlalu banyak teks ke dalam suatu kategori, termasuk teks yang sebenarnya tidak relevan. Nilai *F1-score* yang tinggi dan seimbang menunjukkan bahwa model mampu mengklasifikasikan teks ke dalam kategori sumber stres dengan baik, yaitu secara akurat sekaligus menyeluruh.

3.4.9 Hasil

Tahap terakhir dari penelitian ini adalah penyajian hasil evaluasi model. Hasil yang ditampilkan berupa performa model berdasarkan nilai rata-rata *precision, recall,* dan *F1-score*. Dari hasil tersebut dapat diketahui seberapa baik IndoBERT yang telah melalui proses *fine-tuning* mampu mengklasifikasikan teks sesuai label yang ditentukan. Selain itu, interpretasi nilai metrik juga digunakan untuk melihat kelebihan dan kelemahan model dalam klasifikasi teks. Informasi ini menjadi dasar penarikan kesimpulan penelitian, sekaligus menandai bahwa sistem telah selesai dilaksanakan.

3.5 *Deployment Model*

Tahap akhir dari penelitian ini adalah *deployment model*, yaitu penerapan model IndoBERT yang telah dilatih agar dapat digunakan secara praktis di luar lingkungan penelitian. *Deployment* dilakukan dengan cara menyimpan bobot model terbaik hasil pelatihan, kemudian mengintegrasikannya ke dalam sebuah aplikasi sederhana. Dalam penelitian ini, aplikasi dapat dibangun menggunakan *framework* seperti Streamlit, sehingga model dapat menerima masukan berupa teks dari media sosial, memprosesnya, lalu menampilkan hasil klasifikasi ke dalam kategori sumber stres yang sesuai, Dengan adanya *deployment*, model yang sebelumnya hanya diuji pada dataset penelitian dapat dimanfaatkan secara langsung oleh pengguna. Hal ini membuat hasil penelitian lebih bermanfaat karena model dapat digunakan untuk membantu mengidentifikasi potensi sumber stres secara otomatis.

3.6 Bahasa pemrograman yang digunakan

Dalam penelitian ini bahasa pemrograman yang digunakan adalah :

1. *Python* 3.11.0

   Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpretatif, interaktif, dan berorientasi objek. Bahasa ini menjadi pilihan utama dalam penelitian ini karena ekosistemnya yang kaya akan pustaka (*library*) untuk komputasi ilmiah dan analisis data.

   Dalam penelitian ini, *Python* versi 3.11.0 digunakan sebagai bahasa utama untuk seluruh proses pengolahan data hingga pelatihan model. Pemilihan versi ini bertujuan untuk menjaga kompatibilitas antar pustaka yang digunakan serta memanfaatkan peningkatan performa yang tersedia. Beberapa pustaka penting yang digunakan dalam penelitian ini antara lain *Pandas* dan *NumPy* untuk pengolahan data, *Scikit-learn* untuk pembagian data dan evaluasi model, serta *Transformers* dari *Hugging Face* untuk pemanggilan dan *fine-tuning* model IndoBERT. Selain itu, digunakan juga pustaka *PyTorch* sebagai *backend* dalam proses pelatihan model.

3.7 *Software* yang digunakan

Dalam penelitian ini menggunakan beberapa *software* yang akan digunakan untuk mengolah data, pembuatan model serta evaluasi hasil. Berikut daftar *software* yang akan digunakan :

1. *Google Colaboratory*

Gambar 3. 5 Google Colab

Dalam penelitian ini, *Google Colab* dipilih sebagai platform berbasis *cloud* untuk mengembangkan dan melatih model. Colab menyediakan lingkungan *notebook Jupyter* yang dapat diakses langsung melalui peramban web, sehingga tidak memerlukan instalasi perangkat lunak secara lokal.

Keunggulan utama dari Colab adalah ketersediaan GPU (*Graphics Processing Unit*) yang dapat dimanfaatkan untuk mempercepat proses pelatihan model *deep learning* berbasis IndoBERT, yang membutuhkan sumber daya komputasi cukup besar. Selain itu, Colab juga mempermudah pengelolaan pustaka *Python* serta memungkinkan integrasi langsung dengan *Google Drive*, sehingga memudahkan penyimpanan, pemanggilan, dan pengelolaan dataset maupun model. Dengan berbagai keunggulan tersebut, *Google Colab* menjadi platform yang efektif dan efisien dalam mendukung pelaksanaan penelitian ini.

2. *Visual Studio Code*

Gambar 3. 6 *Visual Studio Code*

Visual Studio Code merupakan sebuah *software* kode ediror modern yang *dikembangkan* oleh Microsoft. Perangkat lunak ini bersifat gratis (*open-source)*, dan dapat dijalankan di berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Dalam penelitian ini, VS Code digunakan sebagai lingkungan pengembangan utama untuk proses pengkodean. Penggunaan VS Code dipilih karena fleksibilitasnya, dukungan ekstensi yang lengkap untuk pengembangan *Python*, dan integrasi terminal yang mempercepat alur kerja dari penulisan kode hingga proses *deployment* aplikasi.

3. Streamlit

Streamlit adalah *framework Python open-source* yang dalam penelitian ini digunakan sebagai alat utama pada tahap *deployment*, yaitu proses mengubah model IndoBERT yang telah dilatih menjadi sebuah aplikasi web interaktif.

Dengan Streamlit, antarmuka pengguna (UI) dapat dibangun secara sederhana namun fungsional, sehingga memungkinkan interaksi langsung antara pengguna dengan model. Skrip Streamlit bertugas menerima input berupa teks dari pengguna, kemudian memanggil model IndoBERT yang telah disimpan, menjalankan proses klasifikasi, dan menampilkan hasil prediksi kategori potensi sumber stres. Selain itu, aplikasi ini juga dapat menampilkan hasil evaluasi model seperti nilai *precision, recall,* dan *F1-score*, maupun visualisasi distribusi prediksi dalam bentuk grafik. Dengan demikian, Streamlit tidak hanya menyajikan hasil penelitian dalam bentuk kode, tetapi juga menghadirkannya sebagai aplikasi praktis yang mudah diakses dan dipahami.

3.8 *Library* yang digunakan

Dalam penelitian ini menggunakan beberapa *library python* yang akan digunakan. Berikut daftar *library python* yang akan digunakan :

1. re

   Library ini digunakan untuk *regular expressions*, yang berfungsi membersihkan teks dari karakter yang tidak relevan. Dalam penelitian ini, re digunakan untuk menghapus simbol, angka, atau tanda baca tertentu agar data teks menjadi lebih bersih sebelum dilakukan *tokenisasi* dan *embedding*.

2. *String*

   Digunakan untuk memanipulasi string, seperti mengakses semua huruf, angka, dan tanda baca. Dalam penelitian ini, string membantu proses *preprocessing*, misalnya menghapus tanda baca dari teks yang bisa mengganggu pemodelan.

3. Os

   Digunakan untuk operasi sistem file, seperti membuat direktori, membaca path, atau mengakses file data. Dalam penelitian, os memudahkan pengaturan file dataset dan model yang disimpan setelah pelatihan.

4. *Glob*

Library ini memudahkan pencarian file dengan pola tertentu di direktori. Dalam penelitian ini, *glob* digunakan untuk mengambil semua file teks atau CSV dalam folder dataset secara otomatis.

5. *google.colab.files*

   Digunakan khusus di Google Colab untuk mengunggah file dataset dan mengunduh hasil model. Hal ini mempermudah interaksi data antara lokal dan lingkungan Colab.

6. *Pandas*

   Library utama untuk manipulasi data tabular. Dalam penelitian ini, *pandas* digunakan untuk menyimpan data teks dalam *dataframe*, mengatur kolom teks dan label, serta mempermudah *preprocessing* dan analisis statistik awal.

7. *Numpy*

   Digunakan untuk komputasi numerik. Dalam penelitian, *numpy* berperan penting dalam operasi vektor, misalnya menyimpan *embedding* teks, menghitung jarak antar-vektor untuk *clustering*, atau normalisasi data.

8. nltk

   Toolkit pemrosesan bahasa alami. Dalam penelitian ini, nltk digunakan untuk *tokenisasi* teks, penghilangan kata-kata yang tidak relevan, dan persiapan teks agar bisa diubah menjadi vektor numerik.

9. *nltk.corpus.stopwords*

   Menyediakan daftar kata umum (*stopwords*). Dalam penelitian, *stopwords* bahasa Indonesia dihapus dari teks agar model fokus pada kata-kata yang relevan untuk klasifikasi atau pengelompokan.

10. *sklearn.preprocessing.LabelEncoder*

    Digunakan untuk mengubah label kategori menjadi angka. Dalam penelitian, *LabelEncoder* membantu mengubah kategori teks menjadi format numerik yang bisa dipahami oleh model klasifikasi

11. *sklearn.model_selection.train_test_split*

    Digunakan untuk membagi dataset menjadi data latih dan uji. Dalam penelitian, *library* ini menjamin evaluasi model dilakukan secara objektif dan data uji tidak tercampur dengan data latih.

12. *sklearn.cluster.KMeans*

    Digunakan untuk melakukan *clustering* atau pengelompokan teks. Dalam penelitian, *KMeans* membantu mengelompokkan dokumen berdasarkan kemiripan *embedding*, sehingga pola atau topik teks dapat dianalisis.

13. *Torch*

    Framework deep learning yang menjadi fondasi model BERT. Dalam penelitian ini, *torch* menjalankan model *neural network* untuk ekstraksi fitur dan klasifikasi teks, baik di CPU maupun GPU.

14. *Transformers*

    Library dari *Hugging Face* yang menyediakan model *pretrained* NLP. Dalam penelitian, *transformers* memungkinkan penggunaan model BERT untuk *feature extraction* dan klasifikasi teks tanpa harus melatih model dari awal.

15. *AutoTokenizer / BertTokenizer*

    Digunakan untuk tokenisasi teks menjadi token numerik. Dalam penelitian, *tokenizer* ini mengubah teks mentah menjadi input yang bisa diproses oleh model BERT.

16. *AutoModel/BertForSequenceClassification/AutoModelForSequenceClassification*

    Model *pretrained* untuk ekstraksi fitur dan klasifikasi teks. Dalam penelitian, model-model ini digunakan untuk mengubah teks menjadi representasi vektor dan memprediksi kategori teks.

17. *Trainer & TrainingArguments*

    API untuk mempermudah pelatihan model. Dalam penelitian, *library* ini digunakan untuk mengatur *batch size, learning rate,* jumlah *epoch*, dan optimisasi sehingga proses pelatihan model menjadi efisien dan terstruktur.

18. *datasets.Dataset*

    Struktur dataset *Hugging Face*. Dalam penelitian, *library* ini memudahkan penyimpanan dan pengelolaan data teks agar dapat langsung digunakan oleh *Trainer* tanpa perlu konversi manual.

19. *Evaluate*

Digunakan untuk menghitung metrik evaluasi model. Dalam penelitian ini, *evaluate* digunakan untuk menghitung *precision, recall,* dan *F1-score*, sehingga performa model dapat diukur secara objektif.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Penelitian

4.1.1 Pengumpulan data

Proses pengumpulan data dilakukan dengan memanfaatkan media sosial Twitter sebagai sumber utama Proses pengumpulan data dilakukan dengan pendekatan web scraping, yaitu teknik ekstraksi informasi dari halaman web secara otomatis menggunakan bantuan *library* atau *tools* pemrograman.

Pengumpulan data difokuskan pada tweet yang mengandung kata kunci tertentu yang berhubungan dengan gejala atau ekspresi stres, salah satunya adalah kata kunci "jenuh", Selain kata kunci utama, dimungkinkan pula ditambahkan kata kunci lain yang masih berada dalam cakupan tema penelitian agar data yang diperoleh lebih bervariasi dan representatif.

Rentang waktu pengambilan tweet dibatasi dari tanggal "1 Januari 2020 sampai 31 Desember 2024". Rentang waktu yang cukup panjang ini bertujuan agar data yang diperoleh lebih beragam dan mampu merepresentasikan perubahan pola ekspresi stres masyarakat di media sosial selama beberapa tahun terakhir.

Dari hasil scraping, data yang diperoleh mencakup 15 atribut bawaan Twitter, seperti username, tanggal unggah, jumlah retweet, jumlah like, isi tweet, dan lain sebagainya. Namun, pada penelitian ini hanya digunakan atribut *"full_text"*, yaitu teks utuh dari setiap tweet.

Berikut adalah sebagian atribut bawaan dari hasil scraping pada twitter, termasuk atribut paling penting "*full text*", seperti pada tabel 4.1

Tabel 4. 1 Dataset Hasil Scraping

conversation_id_str	created_at	favorite_count	full_text	id_str
1873811678849935027	Sun Dec 29 14:50:50 +0000 2024	4	Lebih mending beli setik daripada sego goreng…	1873811678849935027
1873797955375853900	Sun Dec 29	67	Work! Gimana cara balikin	1873797955375853900

	14:45:14 +0000 2024		semangat kerja ya? L...	
1873377115992928720	Sun Dec 29 14:34:44 +0000 2024	0	Monoton Gara2 kerja terus hidupku beneran berp…	1873377115992928720
1872704599716778114	Sun Dec 29 14:33:22 +0000 2024	0	@tjerealjoker Kawan awak dah merepek kat lotri…	1872704599716778114

Tampilan Sebagian atribut dari 15 atribut yang diambil meliputi:

- *Conversation_id_str* : Merupakan ID percakapan dalam bentuk *string* yang digunakan untuk mengelompokkan tweet ke dalam suatu percakapan atau thread.
- *Created_at* : Menunjukkan waktu dan tanggal ketika tweet dipublikasikan.
- *Favorite_count* : Menunjukkan jumlah "*likes*" yang diterima oleh suatu tweet.
- *Full_text* : Merupakan isi teks lengkap dari tweet.
- *Id_str* **:** ID unik untuk setiap tweet dalam bentuk string.

4.1.2 Perancangan Sistem

Perancangan sistem dalam penelitian ini bertujuan untuk memberikan gambaran menyeluruh mengenai alur kerja yang dilakukan, mulai dari tahap pengumpulan data hingga diperoleh hasil akhir berupa deteksi sumber stres pada teks. Sistem dirancang dengan beberapa tahapan utama sebagai berikut:

4.1.2.1 Pengumpulan Data

Tahap awal adalah pengumpulan data dari media sosial Twitter menggunakan metode web scraping. Tweet dikumpulkan dengan kata kunci tertentu yang berhubungan dengan ekspresi stres, salah satunya "jenuh", dalam rentang waktu 1 Januari 2020 hingga 31 Desember 2024. Hasil scraping berupa dataset dengan berbagai atribut bawaan Twitter, namun penelitian ini hanya memanfaatkan kolom "*full_text*" sebagai data utama.

4.1.2.2 *Preprocessing Data*

Data mentah hasil scraping masih mengandung banyak elemen yang tidak relevan, sehingga perlu dilakukan *preprocessing* untuk membersihkan dan mempersiapkan data. Tujuannya adalah untuk meningkatkan kualitas data sehingga model dapat belajar secara efektif.

1. *Delete Column*

Setelah data diperoleh dari proses web scraping, dataset yang dihasilkan secara *default* memuat berbagai atribut atau kolom bawaan dari Twitter. Beberapa di antaranya antara lain *conversation_id_str*, *created_at, favorite_count, id_str, full_text* , dan lain sebagainya. Meskipun atribut-atribut tersebut dapat memberikan informasi tambahan mengenai tweet, namun dalam konteks penelitian ini tidak semua atribut dianggap relevan dengan tujuan analisis. Fokus utama penelitian adalah pada konten teks dari setiap tweet, sehingga hanya kolom *"full_text"* yang digunakan sebagai data utama.

```
# Hapus kolom tertentu
kolom_dihapus = [
    'conversation_id_str', 'favorite_count', 'id_str',
    'image_url', 'location',
    'quote_count', 'retweet_count',
    'user_id_str', 'created_at',
    'username','reply_count','lang','in_reply_to_screen_name','tweet_url'
    ]

# Hapus kolom-kolom tersebut dari DataFrame
df.drop(columns=kolom_dihapus, inplace=True, errors='ignore')
```

Gambar 4. 1 *Code* Untuk *Delete Column*

Gambar 4.1 Potongan kode program untuk menghapus kolom-kolom yang tidak relevan pada data mentah Twitter dengan menggunakan fungsi *drop* dari *library Pandas*. Langkah ini dilakukan agar data yang digunakan dalam penelitian hanya berfokus pada kolom penting, khususnya teks cuitan (*full_text*).

Tabel 4. 2 *Delete Column*

Kolom Bawaan	Setelah Delete Column
conversation_id_str, created_at, favorite_count, id_str, full_text, image_url, location, quote_count, retweet_count, user_id_str, username,	full_text

reply_count, lang, in_reply_to_screen_name, tweet_url	

Tabel 4.2 di atas menunjukkan perbedaan antara dataset sebelum dan sesudah dilakukan proses *delete column*. Pada dataset bawaan, terdapat banyak kolom seperti *conversation_id_str, created_at, favorite_count, id_str, full_text, location,* dan lain-lain. Namun, sebagian besar kolom tersebut tidak relevan dengan tujuan penelitian sehingga dihapus. Setelah proses *delete column*, hanya kolom *full_text* yang dipertahankan karena berisi teks utama yang menjadi fokus analisis sentimen dan akan diproses lebih lanjut pada tahap *preprocessing*.

2. *Cleaning*

Tahap *cleaning* dataset merupakan proses awal dalam *preprocessing* yang bertujuan untuk membersihkan data teks dari berbagai elemen yang tidak relevan. Data mentah hasil scraping dari Twitter umumnya masih mengandung banyak komponen tambahan selain teks utama, seperti tautan, simbol, maupun karakter khusus. Oleh karena itu, diperlukan serangkaian langkah pembersihan agar data lebih terfokus pada isi teks yang akan dianalisis. Adapun proses *cleaning* dilakukan melalui beberapa tahapan berikut.

a. Menghapus URL

Tweet sering kali mengandung tautan ke situs web eksternal. Tautan ini tidak memiliki makna semantik terhadap analisis teks, sehingga perlu dihapus

```
# Preprocessing
def clean_text(text):
    text = re.sub(r"http\S+|www\S+|https\S+", '', text)  # Hapus URL
```

Gambar 4. 2 *Code* Menghapus URL

Gambar 4.2 Menampilkan awal dari tahapan *cleaning* data, yaitu proses penghapusan URL pada teks cuitan dengan memanfaatkan ekspresi reguler (*regular expression*) dari *library re*. Hasil pembersihan ini nantinya digunakan untuk mentransformasi data dari kolom *full_text* menjadi kolom *clean_text*.

Tabel 4. 3 Menghapus URL

Full text	Clean text
Aku merasa jenuh banget hari ini https://t.co/xyz123	Aku merasa jenuh banget hari ini

Lagi stres banget hari ini... https://t.co/abcd1234	Lagi stres banget hari ini
Kelar juga sumpah capek banget https://t.co/FzLfCkh5NN	Kelar juga sumpah capek banget

Tabel 4.3 di atas memperlihatkan contoh hasil *preprocessing* pada tahap *cleaning text*. Kolom *full text* menunjukkan teks asli dari Twitter yang masih mengandung tautan (*URL*), sedangkan kolom *clean text* merupakan teks yang sudah dibersihkan dengan menghapus URL. Proses pembersihan ini bertujuan agar data yang digunakan lebih bersih dan fokus pada konten teks, sehingga memudahkan tahap analisis lebih lanjut.

b. Menghapus *Mention*

Mention digunakan untuk menandai akun lain di Twitter. Elemen ini tidak relevan dalam konteks analisis stres sehingga harus dihapus.

```
text = re.sub(r'@\w+|#\w+', '', text)  # Hapus mention & hashtag
```

Gambar 4. 3 *Code* Menghapus *Mention* dan Hastag

Gambar 4.3 Potongan kode *preprocessing* untuk menghapus *mention* (kata yang diawali dengan @) pada teks cuitan dengan memanfaatkan ekspresi reguler (*regular expression*) dari library re. Sedangkan penghapusan *hashtag* (kata yang diawali dengan #) akan dijelaskan pada tahap berikutnya.

Tabel 4. 4 Menghapus *Mention*

Full text	**Clean text**
@Ceciliapermataa Udah capek banget ya ini	Udah capek banget ya ini
@khatolix Sekarang ini pasti udah capek banget	Sekarang ini pasti udah capek banget
@yhaterus Capek banget kerja tuh	Capek banget kerja tuh

Tabel 4.4 memperlihatkan hasil *preprocessing* pada tahap penghapusan *mention*. *Mention* (*@username*) yang biasanya terdapat pada awal atau isi tweet dihapus karena tidak memiliki kontribusi terhadap analisis makna teks. Dengan dihilangkannya *mention*, data menjadi lebih bersih.

c. Menghapus Hastag

Hastag biasanya digunakan untuk menandai topik tertentu. Namun, dalam penelitian ini yang dibutuhkan adalah teks inti, sehingga hashtag dihapus.

```
text = re.sub(r'@\w+|#\w+', '', text)  # Hapus mention & hashtag
```

Gambar 4. 4 *Code* Menghapus *Mention* dan *Hastag*

Gambar 4.4 Potongan kode *preprocessing* untuk menghapus *hashtag* (kata yang diawali dengan #) pada teks cuitan. Proses ini dilakukan dengan memanfaatkan ekspresi reguler (*regular expression*) dari *library re*, sehingga teks menjadi lebih bersih dan terfokus pada isi utama cuitan tanpa adanya simbol atau tanda khusus.

Tabel 4. 5 Menghapus Hastag

Full text	**Clean text**
Jenuh banget dikantor #jenuh	Jenuh banget dikantor
Lembur hari ini capek banget #lembur#capek	Lembur hari ini capek banget
Insomnia terus menerus tuh bikin stres lama2 #stres #insomnia	Insomnia terus menerus tuh bikin stres lama2

Tabel 4.5 ini menunjukkan proses pembersihan teks dari tanda pagar (#hashtag) yang umumnya digunakan pada media sosial. Hashtag dihapus karena tidak memberikan makna semantik langsung terhadap isi kalimat, melainkan hanya berfungsi sebagai penanda topik atau tren. Dengan menghilangkan hashtag, teks menjadi lebih bersih dan fokus pada konten utama, sehingga memudahkan model untuk memahami konteks kalimat secara lebih akurat pada tahap analisis selanjutnya.

d. Menghapus Angka

Angka sering muncul pada tweet, tetapi dalam penelitian ini tidak memberikan informasi penting untuk deteksi stres, sehingga dihapus.

```
text = re.sub(r"[0-9]+", "", text)  # Hapus angka
```

Gambar 4. 5 *Code* Menghapus Angka

Gambar 4.5 Potongan kode *preprocessing* untuk menghapus angka pada teks cuitan dengan memanfaatkan ekspresi reguler (*regular expression*) dari *library re.*

Tabel 4. 6 Menghapus Angka

Full text	Clean text
Ngerjain tugas skripsi udah 2 malem begadang terus	Ngerjain tugas skripsi udah malem begadang terus
Capek banget nunggu dosen udah 2 jam	Capek banget nunggu dosen udah jam
Pusing 7 hari lagi deadline skripsi	Pusing hari lagi deadline skripsi

Tabel 4.6 menunjukkan perbandingan antara teks asli *(full text)* dengan teks yang telah dibersihkan *(clean text)*. Pada tahap pembersihan, dilakukan penghapusan elemen yang tidak relevan seperti angka.

e. Menghapus Tanda Baca

Tweet sering kali menggunakan tanda baca berlebihan seperti titik-titik (...), tanda seru (!!!), atau tanda tanya (???). Semua tanda baca ini dihapus agar teks lebih bersih.

```
text = text.translate(str.maketrans('', '', string.punctuation))  # Hapus tanda baca
```

Gambar 4. 6 *Code* Menghapus Tanda Baca

Gambar 4.6 Potongan kode *preprocessing* untuk menghapus tanda baca pada teks cuitan dengan menggunakan fungsi *translate()* dan *str.maketrans()* dari *Python*, serta memanfaatkan *string.punctuation* dari *library* standar *Python*.

Tabel 4. 7 Menghapus Tanda Baca

Full text	Clean text
Kesel? sedih? kecewa? jangan ditanya	Kesel sedih kecewa jangan ditanya
Aku bener-bener jenuh banget!!!	Aku bener-bener jenuh banget
Kenapa ya tugas akhir itu menguras energi banget?	Kenapa ya tugas akhir itu menguras energi banget

Pada Tabel 4.7 menunjukkan perbedaan antara teks asli (*Full Text*) dengan teks hasil *preprocessing* (*Clean Text*). Pada tahap *preprocessing* dilakukan pembersihan data dengan cara menghapus tanda baca sehingga teks menjadi lebih sederhana dan siap digunakan pada proses analisis selanjutnya.

2.3 *Case folding*

Tahap *case folding* bertujuan untuk menyeragamkan penulisan huruf pada teks. Pada data mentah hasil scraping Twitter, kata-kata dapat ditulis dengan variasi

huruf besar (kapital) dan huruf kecil, misalnya “Jenuh”, “JENUH”, atau “jenuh”. Jika tidak dilakukan penyamaan bentuk huruf, maka model akan memperlakukan kata-kata tersebut sebagai token yang berbeda meskipun memiliki makna yang sama. Hal ini dapat menimbulkan redundansi dalam representasi kata dan berpotensi mengurangi performa model.

```
text = text.lower() # Case folding
```

Gambar 4. 7 *Code Case Folding*

Gambar 4.7 Potongan kode *preprocessing* untuk melakukan *case folding*, yaitu mengubah seluruh huruf pada teks cuitan menjadi huruf kecil menggunakan fungsi .lower(). Fungsi ini merupakan fungsi bawaan *Python* sehingga tidak memerlukan *library* tambahan. Tujuannya adalah untuk menyeragamkan bentuk teks agar tidak terjadi perbedaan makna antara huruf besar dan huruf kecil pada saat analisis.

Hasil dari case folding disajikan pada tabel 4.8

Tabel 4. 8 Contoh Hasil *Case Folding*

Clean text	Case folding
Pasangan Paling strategis ini ketemu…	pasangan paling strategis ketemu pas…
Kok Rumah Tangga Rasanya Berat….	kok rumah tangga rasanya berat…
Capek Banget dengan Segala Drama…	capek banget dengan segala drama…
Rasanya kayak mau mati kaga Tidur…	Rasanya kayak mau mati kaga tidur…

Berdasarkan Tabel 4.8, kolom *clean text* berisi hasil teks yang sudah melalui proses *cleaning*. Selain itu, pada kolom *case folding* teks juga telah diubah seluruhnya menjadi huruf kecil (*lowercase*) melalui tahap *case folding*.

2.4 *Stopword Removal*

Tahapan *stopword Removal* merupakan salah satu langkah penting dalam *preprocessing* teks, yaitu menghapus kata-kata umum yang sering muncul dalam bahasa, namun tidak memberikan kontribusi signifikan terhadap makna atau analisis. Dalam konteks penelitian ini, *stopword* yang digunakan mengacu pada daftar kata umum dalam bahasa Indonesia yang tidak memiliki bobot semantik kuat, seperti “dan”, “yang”, “di”, “ke”, “dengan”, “atau”, “sebagai”, dan lain

sebagainya. Proses *stopword* dilakukan setelah teks dibersihkan dan diseragamkan melalui *case folding*.

```
    text = " ".join([word for word in text.split() if word not in stop_words])  # Hapus stopword
    return text

df['clean_text'] = df['full_text'].apply(clean_text)
```

Gambar 4. 8 *Code Stopword Removal*

Gambar 4.8 Potongan kode penutup pada tahap *preprocessing*. Baris pertama berfungsi untuk menghapus *stopword* pada teks cuitan, yaitu kata-kata umum yang dianggap tidak memiliki makna penting dalam analisis (misalnya: *dan, yang, di, ke*). Proses ini dilakukan dengan membandingkan setiap kata pada teks dengan daftar *stopword* yang telah didefinisikan, kemudian hanya menyimpan kata yang bukan termasuk *stopword*. Selanjutnya, hasil *preprocessing* dari kolom *full_text* disimpan ke dalam kolom baru bernama *clean_text* dengan memanfaatkan fungsi .apply() dari library Pandas.

Tabel 4. 9 Contoh Hasil *Stopword Removal*

Clean text	Stopword Removal
Aku merasa di kantor ini sangat stres karena kerjaan numpuk	Aku merasa kantor ini stres karena kerjaan numpuk
Kenapa tugas akhir itu begitu berat dan bikin cemas	Kenapa tugas akhir itu berat bikin cemas
Aku sedang jenuh karena hubungan dengan pasangan tidak sehat.	Aku jenuh karena hubungan pasangan tidak sehat.

Tabel 4.9 menunjukkan perbedaan teks sebelum dan sesudah dilakukan proses *stopword removal*. *Stopword* adalah kata-kata umum yang sering muncul dalam teks, tetapi tidak memiliki kontribusi besar terhadap makna, seperti di, ini, sangat, sedang, dengan, dan. Setelah *stopword* dihapus, kalimat menjadi lebih ringkas namun tetap mempertahankan inti makna yang relevan untuk proses analisis selanjutnya.

2.5 Hasil *Preprocessing*

Pada tahap preprocessing, data mentah yang diperoleh dari Twitter masih mengandung banyak elemen yang tidak relevan, seperti tautan (*URL*), tanda baca berlebihan, serta *mention* pengguna lain. Oleh karena itu, dilakukan beberapa

langkah pembersihan data, di antaranya *delete column, cleaning*, *case folding*, dan *stopword removal*. Melalui proses ini, teks diubah menjadi lebih sederhana, konsisten, dan hanya berisi kata-kata yang relevan dengan analisis. Hasil akhir dari *preprocessing* berupa kolom *Clean Text* yang menjadi representasi teks bersih dan siap digunakan pada tahap berikutnya.

Tabel 4. 10 Contoh Hasil *Preprocessing*

No	Clean text
1	saya merasa sangat stres karena lembur
2	bingung Tabungan habis mau gimana lagi
3	susah tidur membuat saya cemas

Tabel 4.10 menampilkan contoh hasil akhir *preprocessing* data teks dari Twitter. Pada tahap ini, teks sudah dibersihkan dari karakter yang tidak relevan. Hasilnya terlihat pada kolom *Clean Text* yang berisi kalimat singkat dan konsisten. Dengan demikian, data teks yang sudah bersih ini siap digunakan pada tahap berikutnya.

4.1.2.3 *Clustering*

Setelah *preprocessing*, data teks dikelompokkan menggunakan metode *clustering*. *Clustering* bertujuan untuk menemukan pola alami dalam data tanpa label awal, dengan mengelompokkan tweet yang memiliki kesamaan makna ke dalam satu cluster. Proses *clustering* dilakukan dengan bantuan *embedding* teks (representasi vektor dari kalimat) dan algoritma pengelompokan seperti *K-Means*. Hasil *clustering* menghasilkan beberapa kelompok tweet yang merepresentasikan tema tertentu.

1. Data Teks

Data teks yang digunakan pada tahap *clustering* berasal dari kolom *clean_text*, yaitu hasil dari proses *preprocessing* sebelumnya. Kolom ini berisi teks yang telah dibersihkan dari berbagai elemen yang tidak relevan. Dengan demikian, data pada kolom *clean_text* sudah dalam bentuk yang lebih terstruktur dan siap untuk diolah lebih lanjut pada tahap *clustering*.

Tabel 4. 11 Tabel Data Teks

No	Clean text

1	rumah tangga berat banget ngerasa kayak gini
2	capek banget kerja tuh
3	kayak mati kaga tidur jam segini ngantor

Tabel di atas menampilkan contoh data hasil *preprocessing* yang tersimpan pada kolom *clean_text*. Setiap baris berisi teks yang telah melalui proses pembersihan dari elemen-elemen yang tidak relevan, Data teks pada kolom ini selanjutnya akan digunakan sebagai masukan untuk masuk ke tahapan *embedding*.

2. *Embedding*

Embedding teks yang dihasilkan oleh model IndoBERT memiliki dimensi 768. Untuk mempermudah visualisasi dan interpretasi, dimensi embedding tersebut direduksi menjadi 2 dimensi menggunakan metode PCA/t-SNE. Hasil reduksi ini kemudian ditampilkan dalam bentuk *scatter plot*, sehingga distribusi dan kedekatan antar teks dapat diamati secara visual. Meskipun demikian, proses *clustering* tetap menggunakan *embedding* berdimensi penuh agar hasil pengelompokan lebih optimal."

Tabel 4. 12 Contoh Representasi *Embedding*

Clean text	Embedding 2D (x,y)
rumah tangga berat banget ngerasa kayak gini	(0.85,-1.22)
capek banget kerja tuh	(0.63,-0.95)
kayak mati kaga tidur jam segini ngantor	(1.10,-140)

Tabel di atas hanya memberikan contoh representasi *embedding* hasil reduksi dimensi ke dalam dua koordinat (x, y) untuk tiga data teks. Contoh ini bukan merupakan data asli, melainkan ilustrasi untuk mempermudah pemahaman mengenai bentuk representasi numerik dari teks setelah melalui tahapan *embedding* dan reduksi dimensi.

3. *Clustering K-Means*

Setelah data teks direpresentasikan dalam bentuk *embedding* dan direduksi menjadi dua dimensi, langkah selanjutnya adalah melakukan proses *clustering* menggunakan algoritma *K-Means*. *K-Means* bekerja dengan cara menentukan

sejumlah pusat cluster (*centroid*) kemudian mengelompokkan setiap data ke dalam cluster yang memiliki jarak terdekat dengan *centroid* tersebut. Proses ini dilakukan secara iteratif hingga posisi *centroid* menjadi stabil dan tidak banyak berubah.

Tabel 4. 13 Contoh *Clustering K-Means*

Clean text	Cluster
rumah tangga berat banget ngerasa kayak gini	1
capek banget kerja tuh	2
kayak mati kaga tidur jam segini ngantor	2

Tabel 4.13 Contoh hasil *clustering* menggunakan algoritma *K-Means.* Pada tabel ini, teks cuitan yang sudah melalui tahapan *preprocessing* dan *embedding* dipetakan ke dalam klaster tertentu berdasarkan kesamaan makna.

4. Hasil *Clustering*

Proses clustering dilakukan untuk mengelompokkan teks berdasarkan kemiripan konteks, sehingga setiap cluster mewakili tema dominan dalam dataset. Metode KMeans digunakan pada *embedding* teks yang sudah dibersihkan dan diproses, menghasilkan beberapa kelompok dengan karakteristik berbeda.

Tabel 4. 14 Contoh Hasil *Clustering*

Clean text	Embedding 2D (x,y)	Cluster
rumah tangga berat banget ngerasa kayak gini	(0.85,-1.22)	1
capek banget kerja tuh	(0.63,-0.95)	2
kayak mati kaga tidur jam segini ngantor	(1.10,-140)	2

Hasil analisis menunjukkan bahwa teks telah diproses melalui embedding dengan 768 dimensi yang kemudian direduksi menjadi 2 dimensi untuk visualisasi. Setelah itu dilakukan clustering menggunakan algoritma K-Means, dan diperoleh dua kluster utama: kluster pertama menggambarkan beban rumah tangga, sedangkan kluster kedua berisi keluhan terkait kelelahan akibat pekerjaan dan kurangnya waktu istirahat.”

4.1.2.4 Pelabelan Semi-Otomatis

Hasil *clustering* kemudian dipetakan ke dalam kategori sumber stres dengan pendekatan semi-otomatis. Artinya, proses pelabelan tidak sepenuhnya manual, namun juga tidak sepenuhnya otomatis. Pertama, setiap cluster yang dihasilkan dianalisis dengan melihat kata-kata dominan dan contoh tweet dalam cluster tersebut. Selanjutnya, cluster diberikan label sesuai dengan kategori stres yang sesuai, Tahap ini tetap melibatkan peran peneliti untuk memastikan label yang diberikan benar-benar mencerminkan isi cluster.

Tabel 4. 15 Kategori dan Keyword Pelabelan Semi Otomatis

No	Kategori dan Keyword
1	Pekerjaan (kerja,bos,resign,lembur,PHK,gaji kecil,interview kerja)
2	Akademik (ujian,skripsi,tugas akhir,nilai jelek,dosen,tugas numpuk,revisi,tidak lulus)
3	Hubungan (diselingkuhi,ditinggal nikah,ghosting,hubungan toxic,pasangan selingkuh)
4	Keuangan (utang,bangkrut,Tabungan habis,bayar cicilan)
5	Kesehatan (sakit,kelelahan,jenuh,cemas,insomnia,operasi,overthinking, penyakit,trauma)
6	Keluarga (masalah keluarga,orangtua,ortu pisah)
7	Lingkungan (dibully,dikucilkan,kantor,rumah)

Pada Tabel 4.15 di atas menunjukkan pembagian kategori sumber stres beserta kata kunci (*keyword*) yang digunakan dalam proses pelabelan data secara semi otomatis. Setiap kategori dirancang untuk merepresentasikan aspek kehidupan yang berpotensi menimbulkan stres pada pengguna media sosial. Dengan adanya daftar kata kunci ini, proses pelabelan semi otomatis dapat dilakukan lebih terarah dan konsisten dalam mengklasifikasikan sumber stres dari teks media sosial.

Tabel 4. 16 Contoh Pelabelan Semi Otomatis

Clean text	Cluster	Label
rumah tangga berat banget ngerasa kayak gini	4	Lingkungan
capek banget kerja tuh	2	Pekerjaan

kayak mati kaga tidur jam segini ngantor	1	Kesehatan

Tabel 4.16 ini merupakan kelanjutan dari hasil *clustering* pada Tabel sebelumnya. Setelah dilakukan pengelompokan otomatis, setiap cluster dianalisis dan diberi label kategori stres menggunakan pendekatan semi otomatis berbasis *keyword*. Kolom Cluster mengacu pada nomor kelompok dari proses *K-Means*, sementara kolom Label menunjukkan kategori stres yang sesuai. Dengan demikian, tabel ini menampilkan hasil yang lebih bermakna, karena setiap teks sudah dikaitkan dengan kategori sumber stres tertentu.

4.1.2.5 Split Data

Setelah proses pelabelan semi-otomatis selesai dilakukan, dataset yang telah berlabel kemudian dibagi menjadi tiga bagian, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*). Pembagian ini bertujuan untuk memastikan bahwa model dapat dilatih secara optimal, dievaluasi selama proses pelatihan, serta diuji menggunakan data yang belum pernah dilihat sebelumnya.

Tabel 4. 17 Split Data

Training	Validation	Testing
2346 data	503 data	503 data
69.99%	15.01%	15.01%

Tabel 4.17 diatas dataset yang telah diperoleh dibagi ke dalam tiga bagian, yaitu data latih (training), data validasi (validation), dan data uji (testing). Pembagian dilakukan dengan proporsi 70% untuk data latih, 15% untuk data validasi, dan 15% untuk data uji. Dari total data yang ada, sebanyak 2.346 data digunakan sebagai data latih, 503 data sebagai data validasi, dan 503 data sebagai data uji.

4.1.2.6 Tokenisasi IndoBERT

Setelah dataset dibagi menjadi data latih, validasi, dan uji, tahap berikutnya adalah melakukan tokenisasi menggunakan *tokenizer* IndoBERT. Tokenisasi mengubah teks mentah menjadi representasi yang dapat diproses model, di mana setiap kata bisa dipecah menjadi *subword* menggunakan metode *WordPiece Tokenization*. *Subword* lanjutan ditandai dengan prefix ##, sedangkan token khusus

[CLS] dan [SEP] ditambahkan di awal dan akhir kalimat untuk klasifikasi. Hasil tokenisasi berupa token *WordPiece* kemudian dikonversi menjadi Token IDs (*input_ids*) sesuai *vocabulary* IndoBERT, dilengkapi *attention mask* untuk menandai token valid. Pasangan *input_ids* dan *attention_mask* inilah yang menjadi input siap pakai dalam tahap *fine-tuning* IndoBERT untuk klasifikasi potensi sumber stres pada teks media sosial.

Tabel 4. 18 Contoh Hasil Tokenisasi Teks Dengan IndoBERT

Clean text	**Token Wordpiece**	**Token IDs**	**Attention Mask**
saya merasa sangat stress karena pekerjaan menumpuk	[CLS], saya, merasa, sangat, stress, karena, pekerjaan, men, ##umpuk, [SEP]	[101, 2072, 6996, 18476, 8263, 2298, 2820, 6020, 3156, 102]	[1,1, 1, 1, 1, 1, 1, 1, 1, 1]
pacar saya tidak membalas chat, jadi saya cemas	[CLS], pacar, saya, tidak, membalas, chat, ,, jadi, saya, cemas, [CLS]	[101, 11824, 2072, 7966, 5163, 6169, 117, 3705, 2072, 8363, 102]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
cuaca panas membuat saya mudah marah dan tidak fokus	[CLS], cuaca, panas, membuat, saya, mudah, marah, dan, tidak, focus, [SEP]	[101, 17107, 5577, 7072, 2072, 11926, 6967, 1106, 7966, 102]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Tabel di atas menampilkan contoh hasil tokenisasi teks (*clean text*) menggunakan *tokenizer* IndoBERT. Kolom *Token WordPiece* menunjukkan hasil pemecahan kalimat menjadi token sesuai metode *WordPiece*. Kolom *Token IDs* merupakan representasi numerik dari setiap token berdasarkan *vocabulary* IndoBERT. Kolom *Attention Mask* menandai token yang relevan dengan nilai 1, sedangkan token yang berupa *padding* (jika ada) akan diberi nilai 0.

Secara khusus, token [CLS] dengan ID 101 selalu ditambahkan di awal kalimat untuk mewakili representasi keseluruhan kalimat dalam tugas klasifikasi.

Token [SEP] dengan ID 102 ditambahkan di akhir kalimat sebagai penanda akhir kalimat atau pemisah antar segmen. Nilai 1 pada *Attention Mask* berarti token tersebut diproses oleh model, sedangkan nilai 0 menunjukkan token *padding* yang diabaikan oleh model.

4.1.2.7 *Fine Tuning* IndoBERT untuk *Text Classification*

Tahap selanjutnya adalah melakukan *fine-tuning* terhadap model IndoBERT agar dapat digunakan untuk tugas klasifikasi teks dalam mendeteksi potensi sumber stres. Dataset berlabel digunakan untuk melakukan *fine-tuning*, sehingga model dapat belajar mengenali pola bahasa yang berhubungan dengan ekspresi stres pada teks Twitter.

Tabel 4. 19 Hasil *Fine Tuning* IndoBERT Untuk *Text Classification*

Epoch	Training Loss	Validation Loss	Precission	Recall	F1-Score
1	0.233700	0.092509	0.969247	0.974155	0.971132
2	0.070200	0.037709	0.988428	0.988072	0.987588
3	0.004400	0.022609	0.994093	0.994036	0.993477

Tabel 4.19 menunjukkan hasil pelatihan IndoBERT selama tiga *epoch*. Pada epoch pertama, nilai *training loss* sebesar 0.2337 dan *validation loss* 0.0925, dengan *precision* 0.9692, *recall* 0.9741, dan *F1-score* 0.9711. Pada *epoch* kedua, *training loss* menurun menjadi 0.0702 dan *validation loss* 0.0377, dengan peningkatan metrik evaluasi yang mencapai *precision* 0.9884, *recall* 0.9880, dan *F1-score* 0.9876. *Epoch* ketiga menunjukkan hasil terbaik dengan *training loss* 0.0044 dan *validation loss* 0.0226, serta *precision* 0.9940, *recall* 0.9940, dan *F1-score* 0.9935. Tren penurunan *loss* yang konsisten disertai kenaikan metrik evaluasi di atas 0.98 pada setiap *epoch* menandakan model belajar optimal tanpa *overfitting*, serta mampu melakukan klasifikasi dengan performa yang sangat tinggi dalam mendeteksi potensi sumber stres.

4.1.2.8 Evaluasi Model

Model yang telah dilatih kemudian dievaluasi untuk mengukur performanya. Evaluasi dilakukan menggunakan data uji dengan beberapa metrik, seperti presisi,

recall, dan *F1-score*, untuk mengetahui sejauh mana model mampu mendeteksi sumber stres dengan baik.

Tabel 4. 20 Hasil Evaluasi Model

Precision	Recall	F1-Score
0.9842	0.9841	0.9831

Tabel 4.20 menunjukkan hasil evaluasi model IndoBERT setelah proses pelatihan. Nilai *precission* sebesar 0.9842 mengindikasikan bahwa sebagian besar prediksi model benar. Nilai *recall* sebesar 0.9841 menunjukkan kemampuan model mengenali hampir seluruh data relevan dengan sangat baik. Sementara itu, nilai *F1-score* sebesar 0.9831 menggambarkan keseimbangan antara *precision* dan *recall*. Secara keseluruhan, metrik ini menegaskan bahwa performa model dalam klasifikasi teks sangat baik dengan akurasi mendekati sempurna.

4.1.2.9 Hasil

Hasil dari penelitian ini berupa sistem yang mampu melakukan klasifikasi teks untuk mendeteksi potensi sumber stres berdasarkan cuitan Twitter. Setelah melalui proses pengumpulan data, *preprocessing*, tokenisasi dengan IndoBERT, serta *fine-tuning*, model menghasilkan output berupa teks (tweet) yang dilengkapi dengan label kategori sumber stres.

Tabel 4. 21 Contoh Hasil

Clean text	**Prediksi label**
saya merasa sangat stress karena lembur terus	Pekerjaan
bingung tabungan habis mau gimana lagi	Keuangan
setiap ga bisa tidur pasti rasanya cemas dan gelisah	Kesehatan

Tabel 4.21 menampilkan contoh hasil prediksi model IndoBERT pada data uji. Setiap teks (*clean text*) berhasil dipetakan ke dalam kategori yang sesuai, yaitu *Pekerjaan*, *Keuangan*, dan *Kesehatan*. Hal ini menunjukkan bahwa model mampu memahami konteks kalimat dan mengklasifikasikannya ke label yang tepat sesuai dengan potensi sumber stres yang terkandung dalam teks. Dengan demikian, dapat

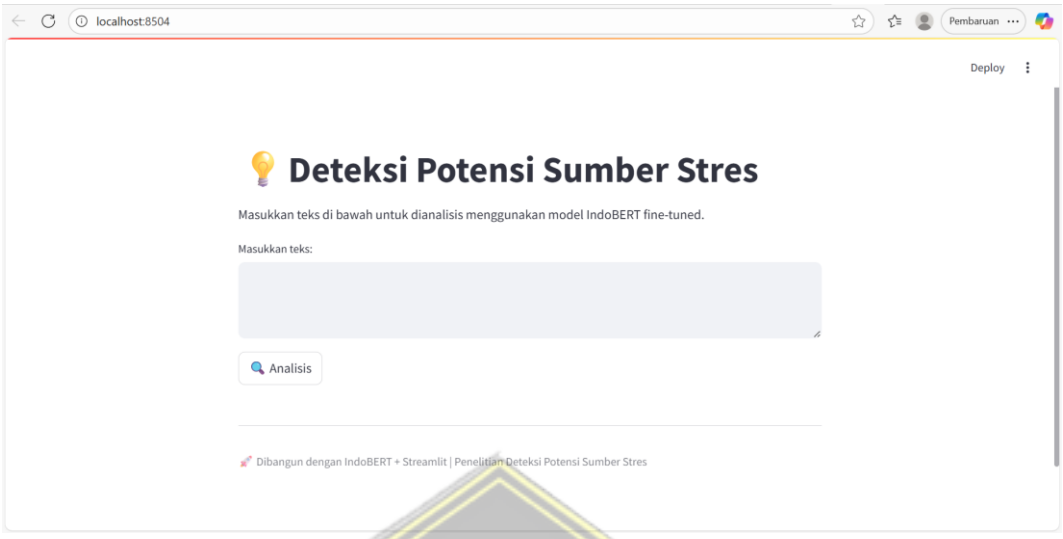
disimpulkan bahwa model bekerja dengan baik dalam melakukan klasifikasi pada data nyata.

4.2 *Deployment Model*

Sebagai tahap akhir implementasi, model IndoBERT yang telah melalui proses *fine-tuning* dan evaluasi di-*deploy* menjadi sebuah aplikasi web interaktif berbasis *text classification*. Tujuan tahap ini adalah mengubah prototipe model dari bentuk kode penelitian menjadi sistem fungsional yang mudah diakses dan digunakan untuk memprediksi kategori potensi sumber stres pada teks (misalnya tweet). Proses *deployment* dilakukan menggunakan Streamlit, sebuah pustaka *Python* yang memungkinkan perancangan antarmuka analitik secara cepat dan efisien.

Proses ini diawali dengan menyimpan model IndoBERT hasil *fine-tuning* beserta *tokenizer* ke dalam file terpisah agar dapat dimuat kembali saat dibutuhkan tanpa perlu melakukan pelatihan ulang. Selanjutnya, sebuah skrip aplikasi utama dengan nama *app.py* dikembangkan untuk merancang antarmuka pengguna. Logika sistem ini mencakup fungsi untuk memuat model IndoBERT terlatih, melakukan proses tokenisasi pada teks input, dan menjalankan prediksi klasifikasi potensi sumber stres. Aplikasi dijalankan secara lokal melalui perintah terminal *streamlit run app.py*, sehingga menghasilkan sebuah Aplikasi Klasifikasi Potensi Sumber Stres berbasis web yang mampu memproses teks baru secara otomatis dan memberikan hasil prediksi kategori stres yang mudah dipahami oleh pengguna.

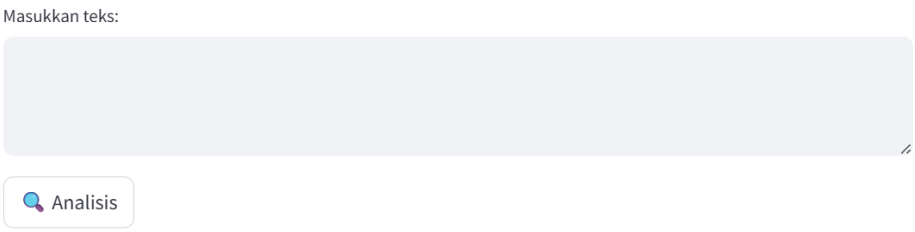
1. Halaman Awal



Gambar 4. 9 Halaman Awal

Gambar 4.9 di atas menampilkan antarmuka aplikasi berbasis web yang dikembangkan menggunakan Streamlit untuk penelitian deteksi potensi sumber stres. Pada halaman utama, terdapat judul "Deteksi Potensi Sumber Stres" yang disertai deskripsi singkat mengenai fungsi aplikasi, yaitu menganalisis teks menggunakan model IndoBERT yang telah di-*fine-tune*. Pengguna dapat memasukkan teks ke dalam kolom input yang tersedia, kemudian menekan tombol "Analisis" untuk memproses teks tersebut dan memperoleh hasil deteksi. Aplikasi ini dirancang dengan tampilan sederhana agar mudah digunakan, serta dilengkapi keterangan pada bagian bawah yang menjelaskan bahwa sistem dibangun menggunakan IndoBERT dan Streamlit sebagai bagian dari penelitian deteksi potensi sumber stres.

2. Bagian untuk memasukkan teks



Gambar 4. 10 Bagian Input Untuk Analisis

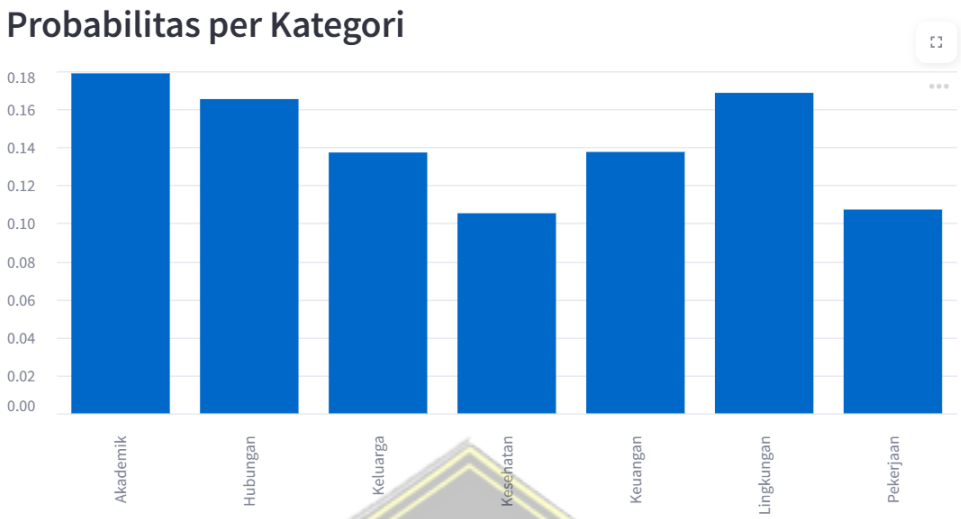
Gambar tersebut menunjukkan tampilan komponen input pada aplikasi deteksi potensi sumber stres. Pada bagian ini terdapat kolom teks yang digunakan untuk memasukkan data berupa kalimat atau pernyataan dari pengguna. Setelah

teks dimasukkan, pengguna dapat menekan tombol "Analisis" yang disertai ikon kaca pembesar untuk memulai proses analisis menggunakan model IndoBERT yang telah dilatih sebelumnya. Fitur ini menjadi inti interaksi pengguna dengan sistem, karena melalui kolom input dan tombol analisis inilah proses klasifikasi teks terhadap potensi sumber stres dijalankan.

3. Hasil analisis teks

Gambar 4. 11 Hasil Input Teks

Gambar 11 memperlihatkan hasil prediksi sistem terhadap input teks “hari ini resign kerja bingung mau kerja dimana lagi”. Sistem menetapkan kategori stres sebagai Akademik. Namun, secara konteks kalimat seharusnya kategori yang lebih tepat adalah Pekerjaan. Hal ini menunjukkan bahwa model masih berpotensi melakukan kesalahan dalam memahami makna sebenarnya dari teks input. Meskipun demikian, kesalahan ini tidak terlalu memengaruhi tujuan penelitian karena fokus utama adalah penerapan model, sementara akurasi keseluruhan sistem tetap tergolong tinggi.



Gambar 4. 12 Probabilitas Hasil Input Teks

Grafik ini menunjukkan distribusi probabilitas dari prediksi pada Gambar 4.11. Kategori Akademik memperoleh probabilitas tertinggi sebesar 18%, sehingga dipilih sebagai hasil akhir, meskipun sebenarnya konteks teks lebih sesuai dengan kategori Pekerjaan yang hanya memperoleh 11%. Sementara itu, kategori Hubungan dan Lingkungan masing-masing mencapai 17%, sedangkan Keluarga dan Keuangan sama-sama berada pada angka 14%, dan kategori Kesehatan sebesar 11%.

Distribusi ini memperlihatkan bahwa meskipun Akademik menjadi kategori dominan, perbedaan probabilitas dengan kategori lain relatif tipis. Dengan kata lain, hasil klasifikasi menunjukkan kecenderungan yang hampir seimbang antar kategori, sehingga model tampak mengalami keraguan dalam menentukan label akhir. Ketidakakuratan ini dapat disebabkan oleh keterbatasan variasi data latih, ketidakseimbangan jumlah data antar kategori, serta penggunaan bahasa tidak baku pada teks media sosial. Oleh karena itu, kelemahan utama penelitian lebih disebabkan oleh keterbatasan dataset dibanding penerapan model IndoBERT, yang secara keseluruhan tetap mampu menunjukkan performa klasifikasi yang baik.

BAB V
KESIMPULAN

5.1 Kesimpulan

Dari hasil penelitian ini dapat memberikan beberapa kesimpulan yaitu :

1. Model IndoBERT berhasil diimplementasikan dan mampu melakukan klasifikasi teks media sosial untuk mendeteksi potensi sumber stres ke dalam beberapa kategori.
2. Berdasarkan hasil evaluasi menggunakan metrik *precision, recall,* dan *F1-score*, model IndoBERT menunjukkan performa tinggi dengan rata-rata *precision* sebesar 0.9842, *recall* sebesar 0.9841, dan *F1-score* sebesar 0.9831. Namun, performa tersebut sebagian besar dipengaruhi oleh kategori mayoritas, sementara kategori minoritas menunjukkan tingkat akurasi yang lebih rendah akibat distribusi data yang tidak seimbang.
3. Hasil evaluasi ini menunjukkan bahwa model IndoBERT memiliki kemampuan yang baik dalam menganalisis teks berbahasa Indonesia untuk mendeteksi potensi stres. Meskipun demikian, ketidakseimbangan kategori masih menjadi kendala yang mempengaruhi akurasi prediksi secara keseluruhan, sehingga perbaikan lebih lanjut, seperti penambahan data pada kategori minoritas atau penyesuaian bobot kelas, diperlukan untuk meningkatkan kinerja model secara menyeluruh.

5.2 Saran

Berdasarkan keterbatasan yang ada dalam penelitian ini, maka beberapa saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambah jumlah dan variasi data, khususnya pada kategori minoritas, agar distribusi data lebih seimbang dan model lebih mampu menangkap keragaman bahasa pada teks media sosial.
2. Mengembangkan metode prapemrosesan yang lebih komprehensif, seperti penanganan kata tidak baku, singkatan, bahasa gaul, dan campuran bahasa Indonesia–Inggris, agar model dapat memahami konteks secara lebih baik.

3. pengembangan dapat diarahkan pada implementasi aplikasi yang lebih interaktif dengan fitur visualisasi dan integrasi real-time dengan media sosial, sehingga dapat dimanfaatkan sebagai alat bantu deteksi dini potensi stres di masyarakat secara lebih praktis.
4. Menyertakan nilai probabilitas (*confidence score*) dalam hasil klasifikasi agar pengguna dapat mengetahui tingkat keyakinan sistem terhadap prediksi yang dihasilkan.

DAFTAR PUSTAKA

Chen, K., Duan, Z., & Yang, S. (2022). Twitter as research data. *Politics and the Life Sciences*, *41*(1), 114–130. https://doi.org/10.1017/pls.2021.19

Erzha Tri Setyo Rochman, Septiana Mukti, Nazwa Mahabatul Thigah Yanani, Fita Sinta Dewi, & Liss Dyah Dewi A. (2024). Pengaruh Media Sosial terhadap Kesehatan Mental pada Anak Muda di Indonesia. *Student Research Journal*, *2*(3), 12–27. https://doi.org/10.55606/srjyappi.v2i3.1219

Firizkiansah, A., Muhammad, A., & Maulana, I. R. (2025). *Optimasi Klasifikasi Data Teks Menggunakan Algoritma Logistic Regression dengan TF-IDF dan SMOTE* (Vol. 2, Nomor 1).

Hafidh, M., Maulana, W., & Widasari, E. R. (2023). *Sistem Deteksi Stres berdasarkan Detak Jantung dan Kelenjar Keringat menggunakan Metode K-Nearest Neighbours* (Vol. 7, Nomor 3). http://j-ptiik.ub.ac.id

Hakim, V. F., & Riana, D. (2024). Analysis of User Complaints for Telecommunication Brands on X (Twitter) using IndoBERT and Deep Learning. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, *13*(2), 270–279. https://doi.org/10.23887/janapati.v13i2.76497

Jocelynne, C., Tobing, L., Lanang Wijayakusuma, I., Putu, L., & Harini, I. (2025). Detection of Political Hoax News Using Fine-Tuning IndoBERT. Dalam *Journal of Applied Informatics and Computing (JAIC)* (Vol. 9, Nomor 2). http://jurnal.polibatam.ac.id/index.php/JAIC

Johan, M., & Aurelia Azka, S. (2023). Prediction of Alleged Stress Symptoms based on Indonesian Sentiment Lexicon using Multilayer Perceptron. *G-Tech: Jurnal Teknologi Terapan*, *7*(3), 958–966. https://doi.org/10.33379/gtech.v7i3.2611

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Dalam *Information (Switzerland)* (Vol. 10, Nomor 4). MDPI AG. https://doi.org/10.3390/info10040150

Kunaefi, A., Abidin, Z., & Kusumawati, R. (2025). KLASIFIKASI BERITA HOAKS BAHASA INDONESIA MENGGUNAKAN INDOBERT FINE-

TUNING DENGAN PENDEKA-TAN FOCAL LOSS PADA DATA TIDAK SEIMBANG. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, *10*(2), 1706–1714. https://doi.org/10.29100/jipi.v10i2.7811

Kustiawan, W., Nurlita, A., Siregar, A., Aini Siregar, S., Ardianti, I., Rahma Hasibuan, M., & Agustina, S. (2022). *Media Sosial Dan Jejaring Sosial*.

Larasati, S. S. A., Dewi, E. N. K., Farhansyah, B. H., Bachtiar, F. A., & Pradana, F. (2024). Penerapan Decision Tree dan Random Forest dalam Deteksi Tingkat Stres Manusia Berdasarkan Kondisi Tidur. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *11*(5), 1043–1050. https://doi.org/10.25126/jtiik.2024117993

Prasetyo, S., & Dewayanto, T. (2024). PENERAPAN MACHINE LEARNING, DEEP LEARNING, DAN DATA MINING DALAM DETEKSI KECURANGAN LAPORAN KEUANGAN-A SYSTEMATIC LITERATURE REVIEW. *DIPONEGORO JOURNAL OF ACCOUNTING*, *13*(3), 1–12. http://ejournal-s1.undip.ac.id/index.php/accounting

Qadir, A., & Ramli, M. (2024). MEDIA SOSIAL (DEFINISI, SEJARAH DAN JENIS-JENISNYA) . *Al-Furqan : Jurnal Agama, Sosial, dan Budaya* , *Vol. 3 No. 6*, 2713–2724.

Rahmadani, S., Rahayu, C. S., Salim, A., & Cahyo, K. N. (2022). DETEKSI EMOSI BERDASARKAN WICARA MENGGUNAKAN DEEP LEARNING MODEL. Dalam *JINTEKS* (Vol. 4, Nomor 3).

Ria Wiyani, J. (2022cf). *Klasifikasi Stres berdasarkan Unggahan pada Media Sosial Twitter menggunakan Metode Support Vector Machine dan Seleksi Fitur Information Gain* (Vol. 6, Nomor 12). http://j-ptiik.ub.ac.id

Saputra, A. C., Saragih, A. S., Ronaldo, D., Raya, U. P., Upr, K., Nyaho, T., Yos Sudarso, J., Palangka, K., Provinsi, R., & Tengah, K. (2025). *PREDIKSI EMOSI DALAM TEKS BAHASA INDONESIA MENGGUNAKAN MODEL INDOBERT*. https://doi.org/10.47111/JTI

Sayarizki, P., & Nurrahmi, H. (2024). Implementation of IndoBERT for Sentiment Analysis of Indonesian Presidential Candidates. *Journal on Computing*, *9*(2), 61–72. https://doi.org/10.34818/indojc.2024.9.2.934

Situmorang, G. F., & Purba, R. (2024). Deteksi Potensi Depresi dari Unggahan Media Sosial X Menggunakan IndoBERT. *Building of Informatics, Technology and Science (BITS)*, *6*(2), 649–661. https://doi.org/10.47065/bits.v6i2.5496

Syazali, M. R., & Yulianti, E. (2025). Classification of Economic Activities in Indonesia Using IndoBERT Language Model. *Jurnal Ilmu Komputer dan Informasi*, *18*(2), 155–165. https://doi.org/10.21609/jiki.v18i2.1446

William, S., Kenny, & Chowanda, A. (2024). EMOTION RECOGNITION INDONESIAN LANGUAGE FROM TWITTER USING INDOBERT AND BI-LSTM. *Communications in Mathematical Biology and Neuroscience*, *2024*. https://doi.org/10.28919/cmbn/7858