TEXT MINING UNTUK KLASIFIKASI *TWEET* BERPOTENSI DEPRESI VIA MEDIA SOSIAL X MENGGUNAKAN MODEL INDOBERT

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1) dalam Ilmu Teknologi Informasi

Diajukan oleh :

SALSABILA SAFIRANA WIBISONO

NIM : 2108096024

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

2025

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama	:	Salsabila Safirana Wibisono
NIM	:	2108096024
Jurusan	:	Teknologi Informasi

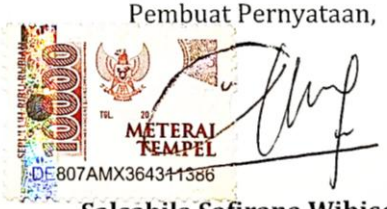
Menyatakan bahwa tugas akhir yang berjudul:

***TEXT MINING* UNTUK KLASIFIKASI *TWEET* BERPOTENSI DEPRESI VIA MEDIA SOSIAL X MENGGUNAKAN MODEL INDOBERT**

Secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 23 Juni 2025

Pembuat Pernyataan,



Salsabila Safirana Wibisono

NIM : 2108096024

LEMBAR PENGESAHAN

KEMENTERIAN AGAMA
UNIVESITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Km. 1 Ngaliyan Telp./Fax – Semarang 50185

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul	:	Text Mining untuk Klasifikasi Tweet Berpotensi Depresi via Media Sosial X Menggunakan Model IndoBERT
Penulis	:	Salsabila Safirana Wibisono
NIM	:	2108096024
Jurusan	:	Teknologi Informasi

Telah diujikan dalam sidang tugas akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana bidang ilmu Teknologi Informasi.

Semarang, 2 Juli 2025

DEWAN PENGUJI

Ketua Sidang / Penguji	**Sekretaris Sidang / Penguji**
Nur Cahyo Hendro Wibowo, S.T., M.Kom.	Hery Mustofa, M.Kom.
NIP. 197312222006041001	NIP. 198703172019031007
Penguji Utama I	**Penguji Utama II**
Moh. Hadi Subowo., M.T.	Mokhamad Iklil Mustofa, M.Kom
NIP. 198703312019031003	NIP. 198808072019031010
Pembimbing I	**Pembimbing II**
Dr. Masy Ari Ulinuha, M.T.	Siti Nur'aini, M.Kom.
NIP. 198108122011011007	NIP. 198401312018012001

NOTA PEMBIMBING I

Semarang, 31 Mei 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	: TEXT MINING UNTUK KLASIFIKASI TWEET BERPOTENSI DEPRESI VIA SOSIAL MEDIA X MENGGUNAKAN MODEL INDOBERT
Nama	: **Salsabila Safirana Wibisono**
NIM	: 2108096024
Jurusan	: Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqasyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,

Dr. Masy Ari Ulinuha, S.T., M.T
NIP. 198108122011011007

NOTA PEMBIMBING II

Semarang, 14 Mei 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	: TEXT MINING UNTUK KLASIFIKASI TWEET BERPOTENSI DEPRESI VIA SOSIAL MEDIA X MENGGUNAKAN MODEL INDOBERT
Nama	: **Salsabila Safirana Wibisono**
NIM	: 2108096024
Jurusan	: Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqsyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,

Siti Nur'aini, M.Kom
NIP. 198401312018012001

MOTTO

"Jangan sampai kehilangan bulan hanya karena sibuk mengejar bintang"

"Don't even try to give up, let me show you the beautiful side of this world"
-AFS-

ABSTRAK

Depresi merupakan gangguan kesehatan mental yang menjadi masalah serius di masyarakat Indonesia, terutama pada kelompok usia muda. Banyak penderita depresi tidak mencari bantuan profesional karena stigma sosial. Media sosial seperti X (Twitter) menjadi solusi alternatif yang digunakan untuk mengekspresikan perasaan dan emosi seseorang secara anonim. Dengan memanfaatkan X, penelitian ini bertujuan untuk mendeteksi potensi depresi melalui analisis teks menggunakan *text mining* berbasis model IndoBERT.

Penelitian ini menggunakan dataset sebanyak 5.000 *tweet* dalam Bahasa Indonesia yang dikumpulkan sejak Oktober 2024 hingga Januari 2025 menggunakan *keyword* penelusuran tertentu. Data dilabeli secara manual oleh peneliti didampingi psikiater menjadi dua kelas, yaitu berpotensi depresi dan normal. Lalu *preprocessing* dilakukan dengan tahapan berupa *case folding, cleaning, normalization, dan stopword removal*. Setelah itu, data dibagi menjadi data latih (80%) dan data uji (20%). Model *pre-trained* IndoBERT yang telah dilatih dilakukan *pre-fine-tuning* dan *fine-tuning* dengan *learning rate* 2e-05, *batch size* 8, dan *epoch* 2 untuk menyesuaikan model ke tugas klasifikasi potensi depresi ini.

Hasil evaluasi menunjukkan bahwa model IndoBERT memberikan performa yang baik dengan akurasi sebesar 87%, *precision* 87%, *recall* 87%, dan *f1-score* 87%. Namun, model mengalami ketimpangan data (*class imbalance*), sehingga cenderung lebih baik dalam memprediksi label mayoritas (normal) daripada minoritas (depresi). Hasil akhir model kemudian diterapkan menjadi aplikasi berbasis *website* dengan Streamlit dan dapat bekerja dengan baik.

Kata Kunci: *text mining*, klasifikasi, depresi, media sosial X, IndoBERT

KATA PENGANTAR

Alhamdulillah, puji syukur ke hadirat Allah Swt. karena atas karunia-Nya penulis dapat menyelesaikan tugas akhir berjudul **"*Text Mining* untuk Klasifikasi *Tweet* Berpotensi Depresi via Sosial Media X Menggunakan Model IndoBERT"** dengan sebaik mungkin. Sholawat serta salam semoga tetap tercurah kepada uswatun hasanah kita, Nabi Muhammad Saw.

Selama menyusun tugas akhir ini, penulis tak lepas dari dukungan dan bimbingan banyak pihak. Oleh karena itu ucapan terimakasih diberikan kepada:

1. Ayah, Bunda, kedua adik, serta keluarga besar penulis yang selalu memberikan dukungan dan doa.
2. Bapak Prof. Dr. H. Nizar Ali, M.Ag, selaku Rektor Universitas Islam Negeri Walisongo Semarang.
3. Bapak Prof. Dr. Musahadi, M.Ag, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
4. Bapak Khotibul Umam, S.T., M.Kom, selaku ketua program studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
5. Bapak Dr. Masy Ari Ulinuha, S.T., M.T dan Ibu Siti Nur'aini, M.Kom selaku dosen pembimbing yang telah memberikan

waktu, ilmu, dan perhatiannya dalam membimbing penulis selama proses pembuatan proposal hingga tugas akhir.

6. Ibu dr. Lidya Santalia, SP.Kj selaku pakar (psikiater) yang telah membantu penulis dan memberi semangat dalam proses pembuatan skripsi.
7. Seluruh dosen Teknologi Informasi yang kaya wawasan dan teman-teman seperjuangan, terutama kelas A angkatan 2021.
8. Teman-teman Asrama Darul Arqam dan KKN Posko 117 yang telah membersamai penulis di masa perkuliahan.
9. Semua pihak yang tidak dapat disebutkan satu persatu yang terlibat dalam pembuatan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Pun masih banyak terdapat kesalahan, baik dalam materi pembahasan maupun dalam segi penulisan sehingga diharapkan kritik dan saran yang membangun dari para pembaca guna memperbaiki kesalahan sebagaimana mestinya. Akhir kata, semoga tugas akhir ini dapat bermanfaat, menjadi referensi bagi para pembaca, dan dapat dikembangkan agar jauh lebih baik lagi.

Semarang, Mei 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

A. Latar Belakang

Gangguan mental ialah kondisi kelainan seseorang di mana adanya perbedaan pola perilaku, pikiran, dan emosi negatif secara terus-menerus sehingga memengaruhi kehidupan sehari-hari. (Febriani & Sulistiani, 2021). Menurut data WHO, pada 2019, sebanyak 970 juta orang di seluruh dunia hidup dengan gangguan mental, dengan kecemasan dan depresi sebagai yang paling umum. Angka tersebut telah naik sebanyak kurang lebih 27% dari angka penderita depresi pada tahun 2016, yaitu sekitar 35 juta orang (WHO, 2022). Data dari Riset Data Kesehatan Dasar (Riskesdas) 2018 menyebutkan lebih dari 12 juta penduduk Indonesia berusia 15 tahun ke atas mengalami depresi, sedangkan tenaga profesional untuk pelayanan kesehatan jiwa hanya 1.053 orang (Kemenkes RI, 2024).

Gangguan kesehatan mental disebabkan oleh multifaktor, beberapa di antaranya yaitu masalah finansial, keluarga, pekerjaan, organisasi, dan akademik (Setyanto, 2023). Depresi pula dapat muncul

dikarenakan adanya faktor lingkungan, psikososisal, maupun kognitif yang ketiganya memiliki gambaran klinis berupa perubahan penderita mulai dari perubahan fisik, perasaan, pikiran, serta pada kebiasaan sehari-hari (Nurtanti & Handayani, 2021). Adapun kasus bunuh diri karena depresi paling banyak pada 2024 adalah karena permasalahan ekonomi (Suara Merdeka, 2024).

Meskipun gangguan mental dianggap parah karena menyerang fisik maupun mental seseorang, masih banyak stigma sosial yang mengatakan bahwa gangguan mental tidak dapat disembuhkan, hal ini mengakibatkan terjadinya pengabaian terhadap individu yang menderita gangguan mental (Ahmad et al., 2023). Data dari beberapa studi menunjukkan bahwa lebih dari 80% penderita gangguan mental tidak pernah mencari bantuan profesional sehingga akhirnya tidak mendapatkan pengobatan dan berujung pada perburukan kondisi kesehatan mental mereka (Maritska et al., 2023).

Salah satu faktor utama yang mendasari tidak sampainya bantuan profesional kepada penderita gangguan mental adalah potensi stigma sosial yang mungkin dialami penderita setelah mengungkapkan kondisinya. Hal ini menjadikan banyak sekali orang

yang menderita gangguan mental menggunakan media sosial sebagai solusi di mana mereka dapat berekspresi dan berkeluh kesah secara anonim tanpa diketahui siapa pun. Melalui *platform* media sosial, banyak penderita yang dapat terhubung dengan sesama penderita lainnya dengan permasalahan serupa yang kurang lebih mampu memahami apa yang mereka rasakan, pikirkan, atau alami (Maritska et al., 2023).

Terganggunya kesehatan mental dapat berdampak pada berbagai aspek kehidupan, jika kondisi ini dibiarkan bertahan lama dan dengan intensitas sedang atau berat maka dapat menjadi kondisi kesehatan yang serius (Aloysius & Salvia, 2021). Penderita akan merasa sangat menderita dan hal terburuknya adalah dapat menyebabkan bunuh diri, dan depresi termasuk pemicu utamanya, di mana tercata lebih dari 700.000 orang meninggal akibat bunuh diri setiap tahun (Komnas Perempuan, 2024; WHO, 2023). Data dari Kepolisian Republik Indonesia (POLRI) menunjukkan bahwa angka kematian akibat bunuh diri di Indonesia pada 2023 meningkat menjadi 1.350 kasus, dari 826 kasus pada tahun sebelumnya (Kemenkes RI, 2024) dan telah menyentuh angka 1.023 kasus sepanjang Januari-Oktober 2024 (GoodStats Data, 2024).

Kasus gangguan kesehatan mental yang berakhir

pada bunuh diri tertinggi di Indonesia tercatat pada kelompok usia muda. Perkumpulan Keluarga Berencana Indonesia (PKBI) menyebutkan, metode bunuh diri yang paling umm adalah gantung diri dan keracunan diri. Adapun provinsi dengan tingkat bunuh diri paling tinggi adalah Bali, Kepulauan Riau, Daerah Istimewa Yogyakarta, Jawa Tengah, dan Kalimantan Tengah (Onie et al., 2024; PKBI Jawa Timur, 2024).

Data di atas menunjukkan bahwa depresi memiliki banyak dampak negatif yang jika tidak segera ditangani dengan baik, di antaranya dapat meningkatkan risiko bunuh diri dan menurunnya kesejahteraan mental seseorang. Banyak cara yang dapat dilakukan untuk mendeteksi atau mengetahui adanya potensi gangguan mental, salah satunya dengan teknik *text mining*. *Text mining* merupakan bagian dari *data mining* yang berfokus pada pengolahan data berbentuk teks. Tujuan dari representasi teks dalam *text mining* adalah untuk menunjukkan dokumen teks tidak terstruktur untuk membuatnya dapat dikomputasi secara matematis dengan mempertahankan makna semantik dari kata-kata dalam teks sehingga menghasilkan informasi dan pengetahuan berguna dari suatu data teks. (Zeberga et al., 2023).

Ada berbagai macam metode yang dapat dilakukan

untuk melakukan *text mining*, salah satunya yaitu model IndoBERT. IndoBERT adalah salah satu model turunan BERT yang menggunakan transformer dalam arsitekturnya di mana menyediakan mekanisme yang dapat mencari tahu makna kontekstual dalam suatu teks (Zeberga et al., 2023). IndoBERT ialah adaptasi BERT yang telah dilatih pada teks berbahasa Indonesia untuk mengidentifikasi potensi depresi dari sebuah teks (Situmorang & Purba, 2024).

Dari uraian di atas untuk mengetahui prevelensi depresi di Indonesia, deteksi dini depresi menjadi penting untuk dilakukan. Sebagaimana tertulis dalam surah Fussilat ayat 49 sebagai berikut.

لَّا يَسْـَٔمُ ٱلْإِنسَـٰنُ مِن دُعَآءِ ٱلْخَيْرِ وَإِن مَّسَّهُ ٱلشَّرُّ فَيَـُٔوسٌ قَنُوطٌ ﴿٤٩﴾

Artinya: “Manusia tidak jemu memohon kebaikan, dan jika ditimpa malapetaka, mereka berputus asa dan hilang harapannya”.

Ayat tersebut menunjukkan bahwa tabiat manusia adalah tidak memiliki kesabaran dan ketangguhan dalam melakukan kebaikan maupun mencegah diri dari keburukan. Manusia tidak bosan berdoa kepada Allah untuk meminta kekayaan dan hal-hal duniawi lainnya. Sedangkan jika ditimpa keburukan atau sesuatu yang tidak disukai seperti penyakit, manusia merasa cepat

berputus asa atas rahmat Allah dan tidak mencari tindakan pencegahan maupun pengobatan.

Berdasarkan uraian latar belakang tersebut, maka penulis melakukan penelitian yang bertujuan untuk mengklasifikasikan *tweet* berpotensi depresi yang diperoleh dari media sosial X menggunakan model IndoBERT. Penulis juga ingin mengukur performa yang diberikan oleh model IndoBERT yang diaplikasikan dalam mengklasifikasikan *tweet* berpotensi depresi.

Oleh karena situasi dan kondisi depresi sangat memprihatinkan sesuai data di atas, model IndoBERT yang diterapkan untuk deteksi dini potensi depresi pada penelitian ini diharapkan dapat digunakan untuk mengetahui kondisi kesehatan mental pengguna dan dapat segera mencari penanganan yang diperlukan.

B. Rumusan Masalah

Berdasarkan latar belakang yang disampaikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan model IndoBERT untuk *text mining* dalam mengklasifikasikan *tweet* berpotensi depresi?

2. Bagaimana performa yang diberikan oleh model IndoBERT yang diaplikasikan untuk *text mining* dalam mengklasifikasikan *tweet* berpotensi depresi?

C. Batasan Masalah

Agar penelitian dapat dilakukan secara jelas, maka penulis peneliti perlu menetapkan batasan masalah pada penelitian ini. Adapun batasan masalah pada penelitian ini adalah:

1. Data yang digunakan adalah data *tweet* pengguna media sosial X, diunduh pada tanggal 1 Oktober 2024 – 31 Januari 2025.
2. Model klasifikasi yang dilakukan dalam penelitian ini adalah IndoBERT.
3. Klasifikasi dilakukan menjadi dua yaitu berpotensi depresi dan normal.
4. Menggunakan bahasa pemrograman *python* dan *software Google Colab* (Jupyter notebook versi google).
5. Jumlah data-data yang digunakan adalah 5.000 data *tweet*.

D. Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengimplementasikan model IndoBERT untuk *text mining* dalam membantu mengklasifikasikan *tweet* berpotensi depresi.
2. Mengukur performa yang diberikan oleh model IndoBERT yang diaplikasikan untuk *text mining* dalam mengklasifikasikan *tweet* berpotensi depresi.

E. Manfaat Penelitian

Manfaat dari hasil penelitian ini dibagi menjadi manfaat teoritis dan manfaat praktis, di antaranya:

1. Manfaat Teoritis
   a. Sebagai bahan referensi dan pengembangan pada penelitian relevan selanjutnya yang berhubungan dengan *text mining*.
   b. Memberikan kontribusi pada pengembangan lebih lanjut model bahasa IndoBERT, khususnya dalam konteks klasifikasi *tweet* berpotensi depresi.
   c. Memperkaya landasan teori tentang penggunaan *text mining* dan model IndoBERT

dalam bidang kesehatan mental, khususnya deteksi dini depresi.

2. Manfaat Praktis
    a. Membantu untuk mengetahui *tweet* berpotensi depresi berdasarkan data *tweet* pengguna media sosial X.
    b. Memudahkan pakar (psikolog dan psikiater) dalam menangani kasus gangguan kesehatan mental karena telah diketahui sejak awal sebelumnya.

BAB II

KAJIAN PUSTAKA

A. Kajian Penelitian yang Relevan

Depresi didefinisikan sebagai perasaan tertekan dan mengganggu aktivitas seseorang untuk berfungsi normal, di mana tidak hanya terjadi pada orang dewasa, melainkan juga anak-anak dan remaja (Nurtanti & Handayani, 2021). Depresi merupakan gangguan medis yang memengaruhi perasaan dan pikiran berupa perasaan bersalah yang terus-menerus dan hilangnya minat sebelum melakukan suatu aktivitas. Agung dalam Nurtanti & Handayani (2021) juga mengungkapkan bahwa depresi menjadi penyumbang beban ekonomi yang sangat besar bagi negara, khususnya negara berkembang seperti Indonesia. Beban kesehatan yang memiliki dampak serius tersebut menjadikan depresi isu kesehatan masyarakat yang penting untuk diupayakan pencegahan, pengobatannya, dan pengedukasiannya karena dapat menghalangi pertumbuhan negara.

Penelitian terkait yang mengacu pada deteksi depresi melalui komentar pengguna media sosial pernah dilakukan oleh Situmorang & Purba (2024) dalam deteksi potensi depresi dari unggahan media sosial X sebanyak lebih dari 35.000 data *tweet* menggunakan teknik NLP dan model

IndoBERT menunjukkan bahwa banyak penderita depresi yang tidak mencari bantuan profesional karena stigma sosial yang buruk terhadap penderita depresi. Stigma ini mengakibatkan banyak kasus depresi tidak terdeteksi dan tidak ditangani dengan baik, sehingga memperburuk kondisi kesehatan individu. Hal ini menunjukkan bahwa deteksi dini depresi penting dilakukan guna memungkinkan intervensi lebih awal dan mengurangi risiko gangguan mental lebih lanjut. Penelitian dengan model IndoBERT tersebut menghasilkan nilai akurasi sebesar 94,91% (Situmorang & Purba, 2024).

Penelitian serupa dilakukan oleh Zeberga et al., 2022 mengenai pendekatan text mining untuk memprediksi *mental health* menggunakan model Bi-LSTM dan BERT. Data pada penelitian ini diperoleh dari Twitter sebanyak 100.000 *tweet* dan Reddit sebanyak 95.000 post. Peneliti berhasil mengembangkan sistem yang mendeteksi depresi dan *anxiety* secara *real-time*. Model Bi-LSTM dan BERT kemudian dibandingkan dengan RandomForest, LG, SVM, CNN, AdaBoost, MLP, dan LSTM ditambah menggunakan teknik embedding berupa TF-IDF, word2vec, dan fastText. Hasil penelitian menunjukkan tingkat akurasi mencapai 98% menggunakan model Bi-LSTM dan BERT sedangkan menggunakan algoritma lain rata-rata akurasinya sebesar 83% (Zeberga et al., 2023).

Penelitian relevan lainnya yaitu tentang klasifikasi penyakit mental dari teks di sosial media menggunakan deep learning dan transfer learning yang dilakukan oleh Ameer et al. (2020). Penelitian ini menggunakan data teks dari Reddit dan berhasil mengklasifikasikan penyakit mental menjadi ADHD, anxiety, bipolar, depresi, dan PTSD dengan dataset total sebanyak lebih dari 16.000 post. Peneliti membandingkan performa kinerja masing-masing algoritma dari classic machine learning berupa LinearSVC, LR, NB, dan RF, deep learning berupa GRU, Bi-GRU, CNN, LSTM, dan Bi-LSTM, hingga transfer learning berupa BERT, XLNet, dan RoBERTa. Hasilnya, akurasi tertinggi didapat oleh model RoBERTa dengan nilai akurasi 83%, disusul BERT 78%, kemudian LSTM yaitu 76%. (Ameer et al., 2022).

Penelitian berikutnya ialah deteksi penyakit mental via media sosial menggunakan deep learning LSTM. Data diambil dari media sosial Twitter. Tujuannya adalah untuk menghitung seberapa banyak frekuensi di mana kata tersebut muncul di sebuah kamus. Hasilnya menunjukkan kata yang sering muncul adalah kata-kata negatif seperti “sedih”, “takut”, “tragis”, “mati”, dan “bunuh diri”. Penelitian ini menghasilkan data yang telah diklasifikasikan menjadi depresi level rendah dan depresi level tinggi dengan akurasi 70,89% (Kholifah et al., 2020).

Dengan data yang telah penulis kumpulkan mengenai data mining untuk klasifikasi dari referensi sebelumnya, terlihat pada tabel kajian pustaka untuk melihat perbandingan penelitian-penelitian terkait.

Tabel 2. 1 Kajian Pustaka

No	Penulis	Judul	Hasil
1	(Situmo-rang dan Purba, 2024)	Deteksi Potensi Depresi dari Unggahan Media Sosial X Menggunakan Teknik NLP dan Model IndoBERT	Penelitian ini menggunakan model IndoBERT dengan data yang berasal dari komentar X. Nilai akurasi 94,91%.
2	(Zeberga, et al., 2022)	A Novel Text Mining Approach for Mental Health Prediction Using Bi-LSTM and BERT Model	Penelitian ini menggunakan model Bi-LSTM dan BERT dengan data yang berasal dari komentar Twitter dan Reddit. Nilai akurasi Bi-LSTM 96% dan BERT 98%.
3	(Ameer, et al., 2022)	Mental Illness Clasification on Social Media Texts using Deep Learning and Transfer Learning	Penelitian ini menggunakan beberapa model deep learning dan transfer learning dengan data dari komentar Reddit. Ketiga model dengan nilai akurasi tertinggi yaitu RoBERTa 83%, BERT 78%, dan LSTM 76%.

4	(Kholifah, et al., 2020)	Mental Disorder via Social Media Mining Using Deep Learning	Penelitian ini menggunakan model deep learning LSTM dengan data dari komentar Twitter. Nilai akurasi LSTM deep learning 70,89%.
5	(Ahmad, et al., 2020)	Hybrid Recommender System for Mental Illness Detection in Social Media Using Deep Learning Techniques	Penelitian ini menggunakan beberapa model deep learning dengan data dari komentar Twitter. Nilai akurasi Bi-LSTM dibanding model lainnya paling tinggi yaitu 93%.

Dari Tabel 2.1 dapat diketahui bahwa objek penelitian atau data yang digunakan adalah kebanyakan bersumber dari media sosial. Pada penelitian ini, penulis menggunakan data *tweet* terbaru dari media sosial X sebagai objek penelitiannya. Hasil dari penelitian ini memiliiki relevansi klinis yaitu implikasi langsung dalam praktik konseling di UIN Walisongo dan pengembangan intervensi. Perbedaan penelitian ini dari penelitian sebelumnya adalah penghapusan tahap *stemming* pada *data preprocessing* walaupun sama-sama menggunakan model IndoBERT. Di mana *stemming* adalah proses pengubahan kata berimbuhan menjadi kata dasar, sehingga penghapusan *stemming* diharapkan tidak

mengubah makna kalimat dan dapat lebih memaksimalkan kinerja model IndoBERT untuk memahami kalimat secara kontekstual.

Hal ini dilakukan karena IndoBERT sendiri adalah model berbasis transformer yang memiliki keunggulan yaitu mampu menangani variasi morfologis kata melalui mekanisme *self-attention* secara inheren, sehingga model dapat memahami hubungan kontekstual dan menangkap nuansa sentimen tanpa reduksi kata. Sedangkan, reduksi kata yang pada tahap *stemming* dilakukan secara agresif, sehingga dapat merusak bentuk kata dan mengurangi informasi penting (Akbik et al., 2018). Misalnya, kata "belajar" dan "pembelajaran" akan sama-sama diubah menjadi kata dasarnya yaitu "belajar", padahal keduanya memiliki arti yang berbeda. *Stemming* juga dinilai mengganggu kemampuan model untuk mempelajari representasi kontekstual yang kaya, karena model transformer menghasilkan representasi kata yang sangat bergantung pada konteks sekitarnya (Devlin et al., 2018; Zhang et al., 2022).

B. Depresi

Kementerian Kesehatan RI menyebutkan bahwa gangguan kesehatan mental (*mental illness* atau *mental disorder*) adalah kondisi kesehatan yang memengaruhi

pemikiran, perasaan, perilaku, suasana hati, atau kombinasi di antaranya di mana kondisi ini dapat sesekali terjadi atau berlangsung lama (kronis) (Kemenkes RI, 2021). Gangguan kesehatan mental yang berkepanjangan dapat memengaruhi seseorang dalam menjalani kehidupan sehari-harinya. Termasuk melakukan kegiatan sosial, akademik, pekerjaan, hingga hubungan dengan keluarga.

Gangguan kesehatan mental memiliki banyak jenis, beberapa di antaranya yaitu depresi, *anxiety*, dan ADHD. Namun, dalam penelitian ini, fokus penelitian adalah untuk mengetahui potensi depresi. Depresi didefinisikan sebagai perasaan tertekan dan mengganggu aktivitas seseorang untuk berfungsi normal, di mana tidak hanya terjadi pada orang dewasa, melainkan juga anak-anak dan remaja (Nurtanti & Handayani, 2021).

Gejala yang ditunjukkan oleh penderita depresi di antaranya kehilangan minat dalam melakukan aktivitas, perubahan nafsu makan, pola tidur, dan kognisi yang sering diekspresikan dengan perasaan sedih, putus asa, dan pesimis oleh penderitanya (Beo et al., 2022). Penderita depresi tidak ditentukan oleh usia, depresi dapat diderita oleh dari anak-anak hingga orang dewasa (Nurtanti & Handayani, 2021). Namun, data mengatakan bahwa prevalensi depresi tertinggi dari 2018-2023 terjadi pada

kelompok umur 15-24 tahun dibandingkan kelompok usia lain (Beo et al., 2022; Komnas Perempuan, 2024).

Dalam penelitian ini, dikumpukan kata kunci yang sudah dikonsultasikan dengan psikiater sebagai kata-kata yang sering digunakan oleh orang yang berpotensi mengidap depresi. Selain itu, dikumpulkan juga kata kunci yang berlawanan dengan depresi untuk menentukan kelas normal. Kata kunci ini nantinya digunakan dalam proses pencarian *tweet* (Situmorang & Purba, 2024).

Tabel 2. 2 Kata Kunci Penelusuran Tweet

Berpotensi depresi	Bersalah, Cemas, Depresi, Diabaikan, Dikucilkan, Frustasi, Gagal, Galau, Gelisah, Hilang kendali, Hampa, Kecewa, Kesepian, Lelah, Menyerah, Menyesal, Pasrah, Putus asa, Sedih, Sendirian, Stress, Tak berharga, Terasing, Terisolasi, Terluka, Terpuruk, Tertekan, Tersisih, Tidak berarti, Tidak berdaya, Tidak berguna, Tidak didengar, Tidak mampu
Normal	Antusias, Bahagia, Berharga, Berhasil, Bersama, Ceria, Dekat, Dihargai, Dipercaya, Diterima, Gembira, Mampu, Optimis, Positif, Riang, Semangat, Senang, Sukses, Tenang

C. Text Mining

Text mining termasuk bagian dari *data mining*, yaitu disiplin ilmu yang mempelajari metode untuk mendapatkan *value* berupa pengetahuan atau pola dari data yang banyak atau juga dikenal dengan nama KDD

(*Knowledge Discovery in Database*) (Febriani & Sulistiani, 2021).

Melalui identifikasi dan eksplorasi data yang sumbernya berupa teks, kutipan, dokumen, dan sebagainya, *text mining* bertujuan untuk menunjukkan dokumen teks tidak terstruktur untuk membuatnya dapat dikomputasi secara matematis dengan mempertahankan makna semantik dari kata-kata dalam teks sehingga menghasilkan informasi dan pengetahuan berguna dari suatu data teks. (Zeberga et al., 2023).

Data teks berbeda dengan data numerik, gambar, data teks membutuhkan teknik *Natural Language Processing* (NLP) untuk diproses dengan hati-hati (Liu et al., 2023). Selain NLP, metode lain yang digunakan dalam *text mining* adalah *machine learning, data mining,* manajemen pengetahuan, serta pencarian informasi (Ridwansyah, 2022).

Text mining meliputi pengolahan atau analisis data teks yang biasanya digunakan dalam klasifikasi dokumen tekstual sesuai topik yang diangkat dalam suatu dokumen (Darwis et al., 2021). Metode *text mining* meliputi *case folding, text cleaning, tokenizing, stopwords removal,* dan *stemming*. Pada umumnya, data yang digunakan dalam implementasi *text mining* diambil dari sekumpulan bahasa

alami yang tidak terstruktur, seperti data *tweet* yang digunakan dalam penelitian ini.

D. Klasifikasi

Klasifikasi merupakan salah satu teknik dalam *data mining* yang mengelompokkan data ke dalam kelas-kelas tertentu berdasarkan penemuan atribut-atribut yang sama pada himpunan objek dalam sebuah *database* (Ente et al., 2020). Klasifikasi adalah salah satu tugas yang dapat dilakukan oleh machine learning, algoritma machine learning memungkinkan pengenalan otomatis kata-kata emosional dalam teks sehingga proses klasifikasi dapat dilakukan dengan cepat (Liu et al., 2023).

Klasifikasi bertujuan untuk menemukan model dari data latih yang akan membedakan atribut ke dalam kategori atau kelas yang sesuai dan telah ditetapkan sebelumnya oleh model. Dalam penelitian ini, kelas yang dibuat termasuk ke dalam *binary classification* dari data *tweet* di media sosial X berupa kelas berpotensi depresi dan normal.

E. Media Sosial X

Media sosial merupakan platform sebagai sarana berbagi dan berkomunikasi yang dapat diakses semua oleh antar penggunanya secara *online* secara mudah dan cepat (Maritska et al., 2023).

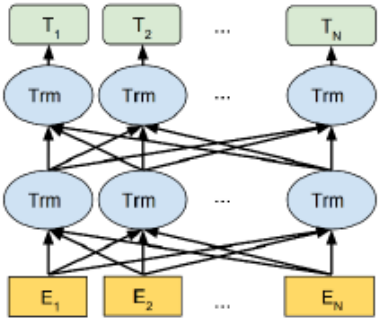
Stigma sosial menjadi salah satu alasan utama yang menyebabkan penderita gangguan kesehatan mental enggan mencari bantuan profesional. Namun dengan adanya media sosial, banyak orang dapat mengekspresikan perasaannya, bercerita mengenai diri sendiri, serta mengomentasi *tweet* orang lain tanpa batas (Aferreyra, 2019; Nabiilah et al., 2024a). Tak terkecuali para penderita depresi, mereka dapat memanfaatkan media sosial sebagai tempat berkeluh kesah dengan sesama penderitanya.

Salah satu platform media sosial yang populer digunakan adalah X atau dahulu disebut Twitter. *Tweet* dalam X dapat dishare, diposting, dikomentari, serta dilihat oleh pengguna X lain. Kelebihan media sosial X adalah salah satunya menyediakan layanan API (*Application Programming Interface*) yang sangat baik, sehingga memudahkan setiap orang untuk mengambil data dari X menggunakan token autentikasi API (Twitter, 2021).

F. IndoBERT

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model *deep learning* yang dikembangkan oleh Google dan dirilis pada tahun 2018 untuk merepresentasikan kata-kata dalam konteks sebagai bagian dari tahap persiapan NLP (Situmorang & Purba,

2024). BERT terdiri dari banyak lapisan pemrosesan untuk menangkap pola kompleks dalam data sehingga dapat memperhatikan makna kontekstual sebuah kalimat (Zeberga et al., 2023). Model BERT menggunakan beberapa lapisan *Transformer* yang saling terhubung untuk memproses input teks di mana setiap lapisan *Transformer* terdiri dari mekanisme perhatian (*attention mechanism*) yang memungkinkan model untuk mempertimbangkan konteks kata dalam kalimat dan fokus pada kata-kata yang relevan (Luo et al., 2020).



Gambar 2. 1 Arsitektur Dasar Model BERT

Gambar 2.1 menunjukkan 3 komponen utama untuk pemrosesan teks menggunakan BERT. Bagian bawah (E_1, E_2, ..., E_n) adalah *embedding layer*, tempat awal token diubah menjadi vektor angka yang juga merupakan langkah pertama dalam pemrosesan teks. Bagian tengah (Trm) adalah vektor dari *embedding* yang masuk ke dalam lapisan-lapisan transformer. Ini adalah langkah kedua di

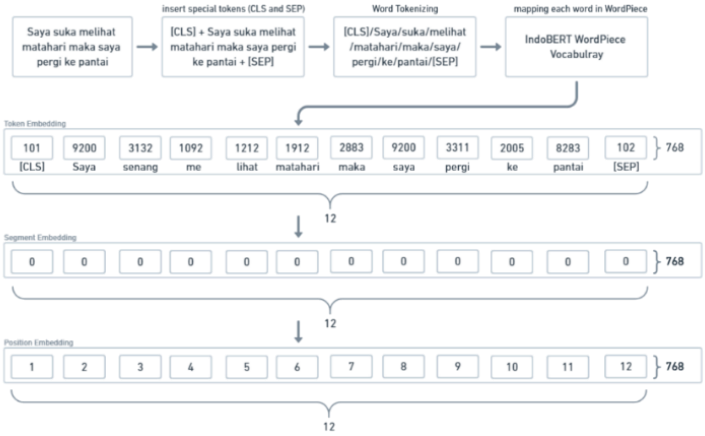
mana terjadi proses analisis konteks antar kata. Bagian atas (T_1, T_2, ..., T_n) mewakili output akhir dari proses, yaitu hasil akhir berupa token yang telah diproses dan dipahami konteksnya. Setelah melalui semua lapisan *transformer*, model menghasilkan output berupa representasi numerik yang kaya akan informasi. Representasi ini sudah "mengerti" makna keseluruhan dari kalimat tersebut. (Situmorang & Purba, 2024).

Tokenisasi adalah proses yang penting dalam tahap *preprocessing* data teks sebelum memasuki tahapan inti pada model *machine learning* (Yefferson et al., 2024). Wolf, et al. dalam Situmorang dan Purba (2024) menjelaskan bahwa teks yang diproses menggunakan BERT akan dipecah menjadi kata atau token berdasarkan frekuensi kemunculannya dalam korpus pelatihan, proses ini disebut tokenisasi. Sehingga kata-kata yang jarang muncul dapat diuraikan menjadi unit yang lebih kecil dan lebih umum.

Proses tokenisasi khusus yang dimiliki oleh BERT dikenal sebagai WordPiece Tokenization (Situmorang & Purba, 2024). Dengan adanya tokenisasi khusus tersebut, tokenisasi biasa tidak ditambahkan dalam tahap *preprocessing* pada penelitian ini, karena nantinya data tetap akan melalui proses tokenisasi dari IndoBERT. Tokenisasi ini memungkinkan BERT secara otomatis mempelajari pecahan kata yang paling sering muncul

dalam korpus pelatihan dan menggunakannya untuk membuat kamus token.

Proses ekstraksi fitur dalam IndoBERT terjadi dengan sangat kompleks. Teks yang diproses akan memasuki *embedding layer*, di mana terjadi pengubahan teks menjadi ID numerik yang sesuai dengan kosakata model. Tokenisasi juga termasuk dalam proses *embedding*. Langkah pertama dalam *embedding* adalah menambahkan token khusus di awal berupa token [CLS] dan akhir kalimat berupa token [SEP]. Token khusus ini digunakan untuk memisahkan kalimat perfiks (imbuhan awal) dan sufiks (imbuhan akhir) (Nabiilah et al., 2024a). Kemudian kata yang telah ditokenisasi akan dipetakan dengan kosakata yang terkandung dalam korpus.



Gambar 2. 2 Proses Embedding IndoBERT

Gambar 2.2 berisi contoh alur proses *embedding* menggunakan IndoBERT. Kalimat input diubah menjadi representasi 12 token, masing-masing terdiri dari 768 *token embedding*. *Segment embedding* dan proses *position embedding* dilakukan untuk membuat model memahami makna secara kontekstual. Dalam *segment embedding*, representasi vektor yang terjadi hanya indeks 0 dan indeks 1. Jika semua input terdiri dari hanya satu kalimat, maka segmen yang disematkan adalah indeks nol. Selain itu, penyematan posisi menerapkan tabel pencarian ukuran (n, 768) di mana n mewakili jumlah panjang kalimat.

Kemudian data memasuki transformers layer yang akan memproses embedding dengan mempertimbangkan konteks kata lain dalam urutan melalui self-attention. Proses ini memungkinkan model untuk memahami hubungan antar kata dan menangkap makna kontekstual. Lapisan terakhir dalam IndoBERT adalah pooling, yaitu representasi fitur tingkat tinggi dari input teks yang telah mempertimbangkan konteks seluruh urutan.

IndoBERT merupakan adaptasi dari BERT yang dirancang khusus untuk menangani teks berbahasa Indonesia dengan model *pre-trained* untuk bahasa Indonesia (Koto et al., 2020; Nabiilah et al., 2024). IndoBERT dikembangkan untuk menjadi solusi atas kebutuhan NLP dalam konteks bahasa Indonesia dengan

sumber-sumber berbahasa Indonesia. IndoBERT dilatih menggunakan korpus teks yang luas dan beragam dari berbagai sumber seperti berita, media sosial, dan dokumen berbahasa Indonesia (Imaduddin et al., 2023). Dalam penelitian ini, model yang digunakan adalah IndoBERT Base karena kompleksitas dan kinerjanya yang seimbang. IndoBERT Base memiliki 124.5 juta parameter, yang cukup besar untuk menangkap berbagai pola dalam data teks tanpa memerlukan sumber daya komputasi yang sangat besar (Situmorang & Purba, 2024).

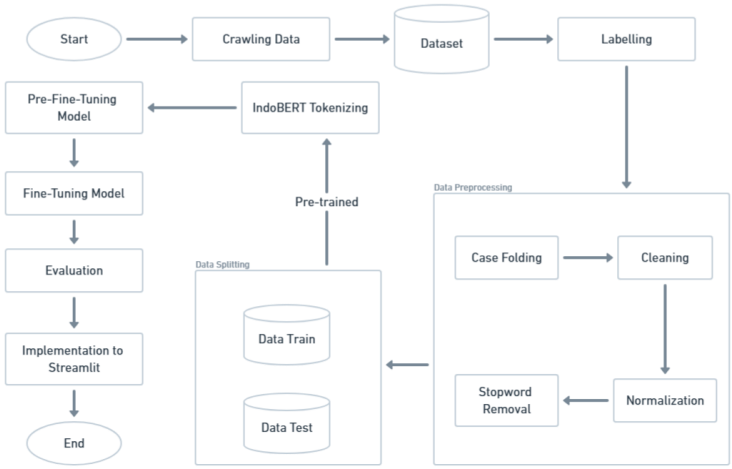
Dalam implementasinya untuk mengklasifikasikan *tweet* berpotensi depresi, model final IndoBERT yang dikembangkan oleh peneliti akan memakai salah satu model *pre-trained* yang telah dilatih untuk mengklasifikasikan beberapa emosi, yaitu prediksi-emosi-indobert dari *platform* Huggingface. Di mana nantinya data akan dilabeli secara manual, setelah itu dilakukan *training model* menggunakan *pre-trained* model tersebut dengan harapan keakuratannya akan lebih terjamin.

Pembagian data dalam penelitian dengan model IndoBERT ini dibagi menjadi dua proporsi yaitu *data train* untuk pelatihan dan *fine-tuning* model dan *data testing* untuk mengevaluasi kinerja akhir model (Hidayat & Nastiti, 2024).

BAB III

METODE PENELITIAN

Prosedur penelitian melibatkan beberapa langkah yang disusun secara sistematis untuk mencapai tujuan yang telah ditetapkan. Tahapan-tahapan yang digunakan haruslah berjalan berurutan dan berkesinambungan untuk memberikan hasil yang optimal. Dengan metode ini, diharapkan penelitian ini dapat menghasilkan model yang dapat diandalkan untuk klasifikasi potensi depresi. Alur metode ialah sebagai berikut.



Gambar 3. 1 Alur Metode Penelitian

Gambar 3.1 menunjukkan alur penelitian yang dimulai dari pengumpulan data menjadi dataset, kemudian dilakukan *data preprocessing* yang meliputi *case folding, cleaning,*

normalization, dan *stopword removal*. Setelah itu, data akan dibagi menjadi dua bagian, yaitu *data train* dan *data test*. Data yang telah dibagi akan ditokenisasi menggunakan tokenisasi khusus IndoBERT (WordPiece). Selanjutnya, model *pre-trained* dilatih lebih lanjut dalam proses *pre-fine-tuning model* dan melalui proses *fine-tuning*, yang diharapkan menghasilkan model yang lebih baik. Barulah model akan diterapkan menjadi aplikasi berbasis *web* menggunakan Streamlit.

A. *Crawling Data*

Data yang diambil dalam penelitian ini adalah data *tweet* pengguna media sosial X berjumlah 5.000 *tweets* dari 1 Oktober 2024 hingga 31 Januari 2025 menggunakan *Tweet Harvest*. Data yang dibutuhkan adalah *tweet* yang di dalamnya banyak terkandung kata kunci bersifat negatif, hal ini menandakan pengguna sedang mengekspresikan emosi negatifnya. Setelah proses pengambilan data (*crawling data*), didapatkan *file* .csv yang kurang lebih berisi sebagai berikut.

Tabel 3. 1 Hasil Crawling Data

full_text
aku sedih pdhl lg pgn di support
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah

Kadang pengen gw ledakin tapi takut ada yg nyiksa diri. Ntar gw gw juga yg merasa bersalah
Rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. Tapi sebab sedar diri pendek kan malu pula nak jengket.
.....
Hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. Hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. Marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.
nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja
@diddekarisma Udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak
udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? I'm not deserve this kind of life oh God please forgive me...
@iteramfs Gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama

B. *Labelling*

Proses *labelling* adalah proses memberikan label atau klasifikasi pada data yang akan dianalisis untuk memudahkan model dalam mengklasifikasikan teks. Dalam proses pelabelan data penelitian ini memberikan *labelling* secara manual sebanyak 5.000 data yang dilakukan oleh peneliti didampingi psikiater untuk menentukan label setiap *tweet* dalam dataset dengan label depresi (berpotensi depresi) dan normal.

Pelabelan dilakukan oleh peneliti didampingi psikiater dengan membaca dan memaknai *tweet* secara seksama dan kontekstual, Hal ini dilakukan karena pelabelan potensi

depresi tidak dapat dilakukan hanya dengan mengandalkan kata kunci tertentu dalam sebuah kalimat seperti “depresi”, “bunuh diri”, “sedih”, atau kata kunci lain yang telah disebutkan sebelumnya dalam Tabel 2.2. Diperlukan analisis kontekstual yang lebih mendalam dan kompleks dalam mendeteksi depresi melalui teks, bukan hanya kata-kata spesifik (Milintsevich et al., 2023; Qasim et al., 2025).

Seluruh data dalam penelitian ini dilabeli secara manual karena beberapa alasan krusial yang berkaitan dengan kompleksitas bahasa, konteks, dan emosi manusia, yang sulit ditangkap secara akurat oleh metode otomatis. Model otomatis seringkali bergantung pada kata kunci tertentu yang terkait dengan depresi (Titla-Tlatelpa et al., 2021). Sedangkan, tidak semua orang yang berpotensi depresi menggunakan kata-kata frontal yang mencerminkan kondisi mereka. Sebaliknya, beberapa orang mungkin menggunakan bahasa yang lebih halus atau tidak langsung, sehingga diperlukan pelabelan manual yang lebih memahami makna dan implikasi emosional dari sebuah teks (Alhamed et al., 2024).

Selain itu, kesalahan dalam mengidentifikasi potensi depresi memiliki konsekuensi terhadap hasil akhir model. Proses pelabelan ini dapat berpengaruh pada tinggi atau rendahnya nilai dua buah variabel dalam matrik evaluasi

model. Pertama, FP (*false positive*) yaitu di mana model melabeli *tweet* normal sebagai berpotensi depresi dapat menyebabkan deteksi palsu dan potensi stigmatisasi. Kedua, kebalikan dari FP yaitu FN (*false negative*), di mana model mengidentifikasi *tweet* berpotensi depresi sebagai *tweet* normal dapat menyebabkan tidak terindikasinya seseorang yang sedang depresi. Oleh karena itu, pelabelan manual dengan pemahaman psikologis sangat diperlukan untuk meminimalisir risiko ini. Berikut adalah contoh penerapan *labelling.*

Tabel 3. 2 Penerapan Labelling

Tweet	**Label**
aku sedih pdhl lg pgn di support	Depresi
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	Normal

C. *Data Preprocessing*

Sebelum pengolahan dataset, dilakukan *data preprocessing* yang melibatkan beberapa tahap, langkah ini bertujuan untuk menyiapkan dan menyempurnakan data tekstual untuk analisis atau penggunaan lebih lanjut. Tahapan-tahapan ini biasanya mencakup konversi huruf besar-kecil (*case folding*), penghapusan simbol atau tanda baca yang mengganggu pada teks, konversi kata *slang* (*normalization*), dan penghapusan kata umum *(stopword removal).*

a. *Case Folding*

Case Folding merujuk pada proses pengolahan teks yang menjadikan keseluruhan huruf menjadi huruf kecil untuk konsistensi analis. Tujuannya agar tiap kata dalam teks menjadi identik. Berikut adalah contoh tahap *case folding*.

Tabel 3. 3 Penerapan Case Folding

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih pdhl lg pgn di support
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	@muh_basra keknya beban depresi pada cash for healing ga si kak (?) wkwkwk

b. *Cleaning*

Cleaning dilakukan untuk menghapus karakter ataupun kata yang dianggap mengganggu pada data teks. Gangguan yang maksud dapat berupa tanda baca, *mention*, *URL,* dan *emoji*. Berikut adalah contoh tahap *cleaning*.

Tabel 3. 4 Penerapan Cleaning

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih pdhl lg pgn di support
@muh_basra keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	keknya beban depresi pada cash for healing ga si kak wkwkwk

c. *Normalization*

Normalization adalah langkah penting dalam meningkatkan aksesibilitas dan analisis data tekstual

dari berbagai bahasa. *Normalization* bertujuan mengubah kata-kata tidak baku seperti singkatan dan *slang* menjadi kata baku sehingga dapat mengurangi keragaman kata dan meningkatkan akurasi. Berikut adalah contoh tahap *normalization*.

Tabel 3. 5 Penerapan Normalization

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih padahal lagi pengin di support
keknya beban depresi pada cash for healing ga si kak wkwkwk	kayaknya beban depresi pada cash for healing enggak si kak wkwkwk

d. Stopword Removal

Stopword removal dilakukan untuk menghilangkan kata-kata umum yang sering muncul, dianggap tidak relevan untuk analisis, dan tidak memberikan kontribusi penting terhadap makna. *Stopword removal* bertujuan untuk mengurangi kumpulan data dan meningkatkan efisiensi komputasi meliputi "adalah", "di", "tapi", "berikut" dan "kalau". Berikut adalah contoh tahap *stopword removal.*

Tabel 3. 6 Penerapan Stopword Removal

Input	***Output***
aku sedih padahal lagi pengin di support	aku sedih padahal pengin support
kayaknya beban depresi pada cash for healing enggak si kak wkwkwk	kayaknya beban depresi cash for healing enggak kak wkwkwk

D. *Data Splitting*

Data splitting atau pembagian dataset adalah proses penting untuk kinerja yang optimal pada sebuah model untuk mencegah *overfitting* dan memvalidasi kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya (Hidayat & Nastiti, 2024). Dalam menggunakan IndoBERT, data dibagi menjadi dua proporsi yaitu data *train* untuk pelatihan dan *fine-tuning model* dan data testing untuk mengevaluasi kinerja akhir model. Rasio data dibagi menjadi 80% untuk *data training* dan 20% untuk *data testing*.

E. IndoBERT *Tokenization*

Terdapat teknik tokenisasi khusus yang dimiliki oleh IndoBERT untuk membangun model dengan menggunakan *pre-trained* IndoBERT yang dirancang khusus untuk pemrosesan bahasa Indonesia. Setelah proses pembagian data, langkah berikutnya adalah proses *tokenizing* yang termasuk fase input, penambahan token khusus [CLS], [SEP] dan proses *encoding*. IndoBERT sendiri memiliki token khusus yaitu [CLS] untuk menentukan kalimat yang menjadi masukan sebagai kategori klasifikasi, [SEP] untuk menandai akhir dari sebuah kalimat, dan [PAD] untuk menambahkan *padding* pada teks yang lebih

pendek sehingga semua *input* memiliki panjang yang sama (Hidayat & Nastiti, 2024).

F. *Pre-Fine-Tuning Model*

Tahapan in merupakan inti dari pemrosesan teks menggunakan model IndoBERT, dilakukan dengan membuat model menggunakan model *pre-trained* (yang telah dilatih sebelumnya) sebagai acuan. Prosesnya melibatkan ideentifikasi fitur-fitur penting dalam data dan kolerasinya dengan data yang ingin diprediksi, dalam hal ini klasifikasi data normal dan berpotensi depresi. Di dalamnya juga termasuk penentuan *hyperparameter* menggunakan Optuna, yaitu *library python* yang dapat menemukan kombinasi terbaik di antara banyak *hyperparameter* untuk *fine-tuning* sebuah model.

G. *Fine-Tuning*

Proses selanjutnya adalah *fine-tuning*, sebagai proses penyesuaian model lebih lanjut dari model yang sudah dilatih, bertujuan untuk meningkatkan performa model sehingga model mampu memberikan hasil prediksi yang lebih akurat (Nugroho et al., 2020). Pada tahap ini dilakukan pula penyesuaian nilai *hyperparameter* untuk mengoptimalkan performa model yang mencakup *learning rate, batch size, dan epoch*. *Learning rate* adalah parameter yang menentukan kecepatan model dalam memahami data

pelatihan, *batch size* adalah jumlah sampel yang digunakan dalam setiap penyesuaian bobot, *epoch* adalah jumlah keseluruhan dataset yang dilihat oleh jaringan, yang mengindikasikan berapa kali seluruh contoh telah diproses oleh jaringan secara lengkap.

Dalam penelitian ini, digunakan *hyperparameter* dengan range yaitu jumlah learning rate antara 1e-5 hingga 3e-5, batch size antara 8 atau 16, dan epoch antara 2 hingga 3. Kombinasi tersebut dinilai efektif untuk dalam mencapai kinerja optimal pada model BERT untuk tugas-tugas NLP (Koto et al., 2020; Situmorang & Purba, 2024).

H. Evaluasi Model

Evaluasi model dilakukan untuk mengetahui atau mengukur kinerja suatu model. Evaluasi model dilakukan dengan cara melihat tingkat performa metode melalui *confusion matrix* 2x2 yang mencakup beberapa metrik seperti akurasi, presisi, *recall* dan *F1-score.*

Akurasi merupakan ketepatan suatu sistem melakukan pengklasifikasian yang benar. Cara perhitungannya dengan cara membagi jumlah yang diprediksi sistem yang sesuai (TP+TN) dengan jumlah keseluruhan data uji (TP+TN+FP+FN). Penghtitungan nilai akurasi dapat dilihat pada Persamaan 1

$$Accuration = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

Precision merupakan seberapa besar tingkat ketepatan antara jumlah data yang terklasifikasi dengan benar (TP) dengan total data yang telah diprediksi dengan benar oleh sistem (TP+FP). Penghitungan nilai *precision* dapat dilihat pada Persamaan 2.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

Recall merupakan perbandingan antara seberapa besar tingkat keberhasilan sistem dalam mengklasifikasikan data dengan benar (TP) dengan total jumlah data yang sebenarnya positif (TP+FN). Untuk menghitung nilai *recall* dapat dilihat pada Persamaan 3.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

F1 Score merupakan pengukuran yang menggabungkan *recall* dan *precision*. Penghitungan nilai *f1 score* dapat dilihat pada Persamaan 4.

$$f1\ score = 2 \times \frac{Precision\ .\ Recall}{Precison + Recall} \times 100\% \quad (4)$$

Berikut pada Tabel 3 menampilkan pengklasifikasian ke dalam dua kelas pada tabel *confusion matrix.*

Tabel 3.7 Confusion Matrix 2x2

		True Class	
		Positive	*Negative*
Predicted Class	*Positive*	TP	FP
	Negative	FN	TN

Dalam *Confusion Matrix* tersebut, ada empat nilai yang dijadikan acuan dalam perhitungan, yaitu :

1. TP (*true positive*), yaitu data yang diprediksi positif dan faktanya data itu positif (Sesuai).
2. TN (*true negative*), yaitu data yang diprediksi negatif dan faktanya data itu negatif (Sesuai).
3. FP (*false positive*), yaitu data yang diprediksi positif dan faktanya data itu negatif (Tidak Sesuai).
4. FN (*false negative*), yaitu data yang diprediksi negatif dan faktanya data itu postif (Tidak Sesuai).

I. Implementasi Model dengan Streamlit

Tahap paling akhir adalah dengan melakukan implementasi model menggunakan Streamlit sebagai aplikasi berbasis *website* untuk mempermudah akses pengguna dalam menggunakan klasifikasi potensi depresi.

Streamlit adalah *toolkit* milik *python* untuk pengembangan aplikasi *web* yang mudah digunakan.

Keunggulan dari Streamlit adalah tersedia secara gratis dan dapat digunakan untuk mengembangkan *dashbord* interaktif dan *real-time* dari model *machine learning* (Revathy et al., 2025). Streamlit dipilih karena mendukung banyak pustaka *python* utama seperti matplotlib, plotly, pandas, matplotlib, seaborn, plotly, Keras, dan PyTorch.

Pada penerapannya, Streamlit digunakan untuk membentuk antarmuka atau *interface website* di mana dalam *website* tersebut, pengguna dapat memasukkan teks atau kalimat kemudia dideteksi potensi depresinya.

BAB IV

HASIL DAN PEMBAHASAN

A. *Crawling Data*

Pengambilan data atau proses *crawling data tweet* pada penelitian ini dilakukan menggunakan *tweet-harvest* dibantu dengan *library pandas* dari *python* dan dilakukan di *Google Colab*. *Google Colab* adalah versi *online* dari *jupyter notebook* secara gratis yang berperan sebagai pendistribusi *python*, termasuk menulis, mengedit, menyimpan, dan membagikan program.

Library pandas merupakan *tools* (alat) yang kerap digunakan untuk analisis data. Di dalamnya terdapat DataFrame yang memungkinkan pengguna untuk mengorganisir, merapikan, membersihkan, dan memanipulasi data dalam format tabel seperti *spreadsheet*.

Tweet-harvest ialah alat yang digunakan untuk mengumpulkan data dari X secara otomatis menggunakan API autentikasi token. *Tweet* yang dikumpulkan memiliki beberapa kriteria, di antaranya yaitu memuat beberapa kata kunci *tweet* berpotensi depresi seperti yang telah disebutkan dalam Tabel 2.2, limit sebanyak 5.000 data, dan rentang waktu pada Oktober 2024 hingga Januari 2025.

Tweet-harvest termasuk alat pihak ketiga yang mengambil data *tweet* menggunakan bahasa pemrograman *JavaScript*, oleh karena itu diperlukan penyesuaian lingkungan runtime dengan instalasi *Node.js* untuk menjalankan kode *JavaScript*. Kode untuk tahap tersebut terdapat dalam kode berikut.

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o

!NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesourc

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

Gambar 4. 1 Penyesuaian Runtime untuk Tweet Harvest

Dengan menginstal *Node.js* dan memastikan versinya (melalui baris terakhir), *Google Colab* akan siap mengeksekusi kode atau skrip pengambilan data dari X. Setelah instalasi selesai, dilanjutkan dengan menginstal *tweet-harvest* dan menggunakannya untuk mengakses, memfilter, dan mengumpulkan *tweet* secara otomatis menggunakan kata kunci tertentu seperti kode berikut.

```
filename = 'tweet_depresi2.csv'
search_keyword = 'depresi OR bersalah OR nyerah OR stres OR gagal lang:id until:2024-11-12
limit = 2500

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit}
```

Gambar 4. 2 Crawling Data dengan Tweet Harvest

Setelah proses tersebut selesai, selanjutnya dataset disimpan dalam bentuk csv. Dataset yang diperoleh memiliki 15 atribut, yaitu *conversation_id_str, created_at, image_url, favorite_count, id_str, location, image_url, in_reply_to_screen_name, full_text, lang, quote_count, reply_count, retweet_count, tweet_url, user_id_str,* dan *username*. Pada Tabel 4.1 berikut ditunjukkan beberapa data dari dataset yang masih menyimpan data kotor sebagai hasil dari proses *crawling data*.

Tabel 4. 1 Data Kotor

full_text
aku sedih pdhl lg pgn di support
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah
Kadang pengen gw ledakin tapi takut ada yg nyiksa diri. Ntar gw gw juga yg merasa bersalah
Rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. Tapi sebab sedar diri pendek kan malu pula nak jengket.
.....
Hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. Hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. Marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.
nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja
@diddekarisma Udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak

udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? I'm not deserve this kind of life oh God please forgive me...
@iteramfs Gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama

Pencarian keyword didasarkan pada 33 kata kunci penelusuran yang telah diverifikasi oleh psikiater dan telah dibahas pada bab sebelumnya. Berikut adalah *source code* untuk mengetahui persebaran *keyword* yang muncul dalam keseluruhan dataset.

```
df['full_text'] = df['full_text'].astype(str).str.lower()

# Daftar 33 keyword
keywords = [
    "bersalah", "cemas", "depresi", "diabaikan", "dikucilkan", "frustasi", "gagal", "galau",
    "gelisah", "hilang kendali", "hampa", "kecewa", "kesepian", "lelah", "menyerah", "menyesal",
    "pasrah", "putus asa", "sedih", "sendirian", "stress", "tak berharga", "terasing",
    "terisolasi", "terluka", "terpuruk", "tertekan", "tersisih", "tidak berarti", "tidak berdaya",
    "tidak berguna", "tidak didengar", "tidak mampu"
]
summary = pd.DataFrame({
    "Keyword": keywords,
    "Total Kemunculan": [keyword_counts[k] for k in keywords],
    "Jumlah Tweet Mengandung Keyword": [tweet_counts[k] for k in keywords]
}).sort_values(by="Total Kemunculan", ascending=False)

print(summary)
```

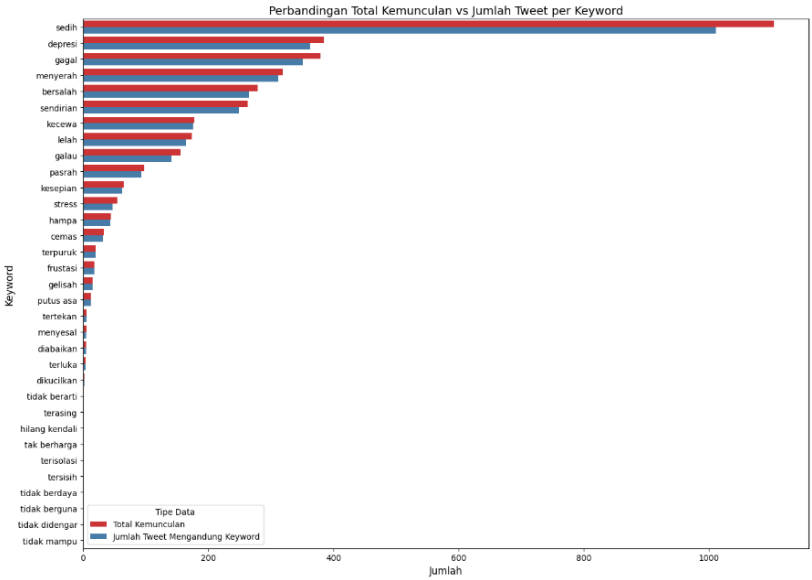
Gambar 4. 3 Kode untuk Mengetahui Persebaran Keyword

Dari kode pada Gambar 4.3 di atas, didapatkan hasil berupa total kemunculan *keyword* serta jumlah *tweet* yang mengandung *keyword* tersebut. Jumlah yang lebih jelas terdapat pada tabel berikut.

Tabel 4. 2 Persebaran Keyword

Keyword	**Total Kemunculan**	**Jumlah *Tweet* per *Keyword***
Bersalah	279	265
Cemas	33	32
Depresi	385	362
Diabaikan	5	5
Dikucilkan	2	2
Frustasi	18	18
Gagal	379	351
Galau	156	141
Gelisah	15	15
Hilang kendali	0	0
Hampa	44	43
Kecewa	178	176
Kesepian	65	62
Lelah	174	164
Menyerah	319	312
Menyesal	6	5
Pasrah	98	93
Putus asa	12	12
Sedih	1.104	1.011
Sendirian	263	249
Stress	55	47
Tak berharga	0	0
Terasing	1	1
Terisolasi	0	0
Terluka	4	4
Terpuruk	20	20
Tertekan	6	6
Tersisih	0	0
Tidak berarti	1	1
Tidak berdaya	0	0
Tidak berguna	0	0
Tidak didengar	0	0
Tidak mampu	0	0
Total	3.622	3.397

Adapun grafik perbandingan persebaran kemunculan *keyword* dengan jumlah *tweet* per *keyword* ialah sebagai berikut.



Gambar 4. 4 Total Kemunculan Keyword dan Jumlah Tweet per Keyword

Dari Tabel 4.2 dan Gambar 4.4, dapat diketahui bahwa dari 33 *keyword*, kata "sedih" memiliki total kemunculan paling banyak yaitu 1.104 kali muncul dengan jumlah *tweet* yang mengandung kata tersebut sebanyak 1.011, sedangkan kata yang tidak muncul sama sekali yaitu "hilang kendali", "tak berharga", "terisolasi", "tersisih", "tidak berdaya", "tidak berguna", "tidak didengar", dan "tidak mampu". Namun perlu diketahui, bahwa kata kunci

tersebut hanya dilakukan untuk pencarian *tweet* saja dan tidak berpengaruh pada proses *labelling* secara signifikan.

B. *Labelling*

Proses *labelling* atau pelabelan data dilakukan dengan menambahkan atribut label ke dalam dataset yang berisi label dari *tweet* yang nantinya akan memudahkan model untuk mengklasifikasikan teks. Label dibagi menjadi dua kategori yaitu depresi (berpotensi depresi) dan normal. Banyaknya *tweet* yang perlu dilabeli menjadikan proses pelabelan sebagai salah satu tantangan dalam penelitian ini. Ditambah, pelabelan setiap *tweet* juga cukup memakan waktu dan hanya dikerjakan oleh sedikit orang, terutama jumlah data yang harus dilabeli sangatlah banyak. Berikut adalah hasil data yang sudah menjalani proses *labelling* dengan rincian 3.287 label normal dan 1.713 label depresi.

Tabel 4. 3 Pelabelan Data

full_text	**label**
aku sedih pdhl lg pgn di support	depresi
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	normal
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah	depresi
Kadang pengen gw ledakin tapi takut ada yg nyiksa diri. Ntar gw gw juga yg merasa bersalah	depresi

Rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. Tapi sebab sedar diri pendek kan malu pula nak jengket.	normal
.....	
Hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. Hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. Marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.	depresi
nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja	depresi
@diddekarisma Udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak	normal
udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? I'm not deserve this kind of life oh God please forgive me...	depresi
@iteramfs Gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama	normal

C. *Data Preprocessing*

Tahap ini terdiri dari beberapa proses untuk membersihkan data teks yang notabene merupakan data tidak terstuktur dan masih memuat banyak *noise*. Melalui tahap ini, data mentah tersebut diolah menjadi data bersih agar siap diproses untuk tahap selanjutnya. Data preprocessing pada penelitian ini dilakukan menerapkan empat langkah secara berurutan, yaitu:

1. *Case Folding*

Langkah ini bertujuan untuk menjadikan semua huruf dalam dataset menjadi huruf kecil agar setiap teks menjadi identik. Implementasi *source code* untuk proses *case folding* ada pada Gambar 4.5.

```
df['full_text'] = df['full_text'].str.lower()
df
```

Gambar 4. 5 Kode Tahap Case Folding

Berikut adalah hasil data teks setelah dilakukan *case folding.*

Tabel 4. 4 Data Teks Setelah Case Folding

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih pdhl lg pgn di support
@muh_basra Keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	@muh_basra keknya beban depresi pada cash for healing ga si kak (?) wkwkwk
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah	gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah
Kadang pengen gw ledakin tapi takut ada yg nyiksa diri. Ntar gw gw juga yg merasa bersalah	kadang pengen gw ledakin tapi takut ada yg nyiksa diri. ntar gw gw juga yg merasa bersalah
Rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. Tapi	rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. tapi

sebab sedar diri pendek kan malu pula nak jengket.	sebab sedar diri pendek kan malu pula nak jengket.
.....	
Hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. Hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. Marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.	hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.
nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja	nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja
@diddekarisma Udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak	@diddekarisma udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak
udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? I'm not deserve this kind of life oh God please forgive me...	udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? i'm not deserve this kind of life oh god please forgive me...
@iteramfs Gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama	@iteramfs gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama

2. *Cleaning*

Cleaning dilakukan untuk membersihkan *tweet* dari karakter yang tidak diperlukan, seperti tanda baca, mention, URL, emoji, dan sebagainya. Namun,

sebelum menjalankan proses ini, dilakukan penghapusan kolom yang tidak diperlukan. Artinya, kolom selain "*label*" dan "*full_text*" akan dihilangkan karena tidak akan dipakai dalam tahap selanjutnya. Kode programnya ditunjukkan pada Gambar 4.6 berikut.

```
columns_to_drop = ['conversation_id_str','image_url','favorite_count','id_str',
                   'location','image_url','in_reply_to_screen_name','lang',
                   'quote_count','reply_count','retweet_count','tweet_url',
                   'user_id_str','created_at','username']
df = df_combined.drop(columns=columns_to_drop, errors='ignore')
```

Gambar 4. 6 Penghapusan Kolom dari Dataset

Selanjutnya, dilakukan penghapusan data duplikat. Hal ini bertujuan agar tidak ada data ganda yang bisa memengaruhi hasil analisis atau pelatihan model. Berikut adalah implementasi kode programnya.

```
df = df.drop_duplicates(subset=['full_text'])
df.duplicated().sum()
```

Gambar 4. 7 Penghapusan Data Duplikat

Selain itu, *library* emoji dan re juga perlu disiapkan. *Library* emoji bertujuan untuk mendeteksi dan menghapus emoji dari *tweet*, sedangkan *library* re (Regular Expression) digunakan untuk mencari dan menghapus pola teks tertentu seperti simbol, link, dan angka. Lalu, dilakukan pemanggilan atau impor *library*

ke dalam kode *python* agar dapat digunakan. Kode programnya ditunjukkan pada Gambar 4.8.

```
!pip install emoji
import emoji
import re
```

Gambar 4. 8 Instalasi dan Impor Library

Setelah itu, barulah dilakukan tahap inti dari *cleaning data*. *Source code* tahap ini ditunjukkan oleh gambar berikut.

```
def clean_text(text):
    # Menghapus mention
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)
    # Menghapus hashtag
    text = re.sub(r'#', '', text)
    # Menghapus link
    text = re.sub(r'http[s]?://\S+', '', text)
    # Menghapus karakter non-alfanumerik
    text = re.sub(r'[^A-Za-z0-9\s]', '', text)
    # Menghapus spasi berlebihan
    text = re.sub(r'\s+', ' ', text).strip()
    # mengubah emoji menjadi text
    text = emoji.demojize(text)
    # mengubah tanda baca berulang menjadi tunggal
    text = re.sub(r'([^\w\s])\1+', r'\1', text)

    return text

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

Gambar 4. 9 Kode Tahap Cleaning

Berikut adalah hasil dataset yang telah dilakukan proses *cleaning.*

Tabel 4. 5 Data Teks Setelah Cleaning

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih pdhl lg pgn di support
@muh_basra keknya beban depresi pada cash for healing ga si kak (?) wkwkwk	keknya beban depresi pada cash for healing ga si kak wkwkwk
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini. iya gue harmonis tapi lebih banyaknya ga terasa. kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah	gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini iya gue harmonis tapi lebih banyaknya ga terasa kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah
kadang pengen gw ledakin tapi takut ada yg nyiksa diri. ntar gw gw juga yg merasa bersalah	kadang pengen gw ledakin tapi takut ada yg nyiksa diri ntar gw gw juga yg merasa bersalah
rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka. tapi sebab sedar diri pendek kan malu pula nak jengket.	rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka tapi sebab sedar diri pendek kan malu pula nak jengket
.....	
hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik. hari demi hari makin di bikin yakin untuk nyerah sama kehidupan. marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin.	hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik hari demi hari makin di bikin yakin untuk nyerah sama kehidupan marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin

nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang' sekitar orang' di tmpt kerja	nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang sekitar orang di tmpt kerja
@diddekarisma udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak	udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak
udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini? i'm not deserve this kind of life oh god please forgive me...	udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini im not deserve this kind of life oh god please forgive me
@iteramfs gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama	gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama

3. *Normalization*

Normalization bertujuan mengonversi kata tidak baku seperti singkatan dan *slang* (kata gaul) menjadi kata baku. Hal ini dilakukan untuk mengurangi keragaman kata. Pada tahap ini, diperlukan beberapa *library* pendukung seperti *library* csv, string, pandas, dan re. Berikut adalah kode programnya.

```
import csv
import string
import pandas as pd
import re
```

Gambar 4. 10 Impor Library Pendukung

Kemudian dilakukan pula pemanggilan *library* requests untuk mengambil *fetch* (konten) dari sebuah

URL di mana URL tersebut mengarah pada file csv dari colloquial-indonesian-lexicon yang berada di repositori GitHub dan dapat diakses secara bebas. File tersebut berisi kumpulan kata-kata tidak baku dan baku dalam bahasa Indonesia yang sangat berguna dalam tahap ini. Hal ini dilakukan karena *python* tidak menyediakan data atau konten terkait, sehingga diperlukan data pendukung agar bisa melakukan tahap ini. Gambar 4.11 menunjukkan kode untuk tahap ini.

```
import requests

url = "https://github.com/nasalsabila/kamus-alay/blob/master/colloquial-indonesian-lexicon

response = requests.get(url)
if response.status_code == 200:
    print("File berhasil diakses!")
    print(response.text[:500])  # Cek sebagian isi file
else:
    print("Gagal mengakses file. Error:", response.status_code)
```

Gambar 4. 11 Pengambilan Fetch dari URL

Setelah pemanggilan konten berhasil, *file raw* (mentah) akan dibaca dan kolom pertamanya akan dijadikan referensi. Artinya, kolom pertama yang sering kali berisi kata *slang* akan dipakai untuk nantinya menghapus kata-kata tersebut dari teks. Caranya adalah dengan memecah kalimat menjadi kata, kemudian mengganti kata *slang* dengan kata baku, lalu menggabungkannya kembali menjadi satu kalimat utuh. Proses inilah yang nantinya akan

diterapkan pada setiap kalimat dalam kolom *"full_text"*. Penerapan ini ditunjukkan pada Gambar 4.12.

```
#Buat dictionary slang jadi formal
norm_dict = dict(zip(slang_df.iloc[:, 0], slang_df.iloc[:, 1]))

#Fungsi normalisasi menggunakan regex
def normalisasi(str_text):
    words = str_text.split()  # Pecah kalimat jadi list kata
    words = [norm_dict.get(word, word) for word in words]
    return " ".join(words)  # Gabungkan kembali jadi kalimat
# Ambil kolom pertama sebagai daftar slang-formal
slang_list = slang_df.iloc[:, 0].tolist()
df['full_text'] = df['full_text'].astype(str).apply(normalisasi)
```

Gambar 4. 12 Penerapan Fetch untuk Normalization

Data teks yang telah melewati tahap *normalization* ditunjukkan pada Tabel 4.6.

Tabel 4. 6 Data Teks Setelah Normalization

Input	***Output***
aku sedih pdhl lg pgn di support	aku sedih padahal lagi pengin di support
keknya beban depresi pada cash for healing ga si kak wkwkwk	kayaknya beban depresi pada cash for healing enggak sih kak wkwkwk
gue udah di titik cape banget nutup nutupin image keluarga yang harmonis selama ini iya gue harmonis tapi lebih banyaknya ga terasa kontrol emosional nyokap yg ga pernah bisa dijaga peran support yang ga pernah gue dapet untuk gue ngelakuin apapun di kuliah	gue sudah di titik capek banget nutup menutupi image keluarga yang harmonis selama ini iya gue harmonis tapi lebih banyaknya enggak terasa kontrol emosional nyokap yang enggak pernah bisa dijaga peran support yang enggak pernah gue dapat untuk gue melakukan apapun di kuliah
kadang pengen gw ledakin tapi takut ada yg nyiksa diri	kadang pengin gue ledakin tapi takut ada yang nyiksa diri

ntar gw gw juga yg merasa bersalah	entar gue gue juga yang merasa bersalah
rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka tapi sebab sedar diri pendek kan malu pula nak jengket	rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka tapi sebab sedar diri pendek kan malu pula nak jengket
.....	
hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik hari demi hari makin di bikin yakin untuk nyerah sama kehidupan marah bgt sama diri sndri di masa lampau karena udah setuju untuk di lahirin	hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik hari demi hari makin di bikin yakin untuk nyerah sama kehidupan marah banget sama diri sendiri di masa lampau karena sudah setuju untuk di lahirin
nyerah aja kali ya beneran capek sm kehidupan capek sm diri sndri orang sekitar orang di tmpt kerja	nyerah saja kali ya benaran capek sama kehidupan capek sama diri sendiri orang sekitar orang di tempat kerja
udah nyerah sama hidup tuh makanya nyemplungin diri ke minyak	sudah nyerah sama hidup tuh makanya nyemplungin diri ke minyak
udah di tahap nyerah sm diri sendiri nyoba buat jalanin aja hidup sampai di panggil tapi mau sampai kapan aku hidup ga ada nyawa gini im not deserve this kind of life oh god please forgive me	sudah di tahap nyerah sama diri sendiri mencoba buat menjalani saja hidup sampai di panggil tapi mau sampai kapan aku hidup enggak ada nyawa begini im not deserve this kind of life oh god please forgive me
gk usah kkn nder nanti rame2 bilang yg salah panitnya ngapain ngewajibin tapi lama	enggak usah kkn nder nanti rame-rame bilang yang salah panitnya mengapai ngewajibin tapi lama

4. *Stopword Removal*

Tahap ini bertujuan untuk menghapus kata-kata umum yang sering muncul, namun tidak memiliki arti penting. Dalam penerapannya, peneliti menggunakan *library* sastrawi milik *python* yang memiliki fokus tugas pada *stopword removal* dan *stemming*. Setelah instalasi berhasil, *library* sastrawi dipanggil dengan kelas StopWordRemoverFactory, StopwordRemover, dan ArrayDictionay. Kombinasi ketiga kelas tersebut digunakan untuk menghapus *stopword* berdasarkan kamus bawaan dan kamus tambahan (*custom*). Berikut adalah kode program untuk impor *library* dan pembuatan kamus tambahan.

```
!pip install Sastrawi

from Sastrawi.StopWordRemover.StopWordRemoverFactory import (
    StopWordRemoverFactory, StopWordRemover, ArrayDictionary
)

# Stopword tambahan
more_stopword = ['tuh','lah','kok','sok','cs','sih','loh','nih']
```

Gambar 4. 13 Instalasi dan Impor Library Sastrawi

Kamus tersebut nantinya dijadikan satu dengan kamus bawaan milik sastrawi, kemudian diterapkan untuk seluruh data teks seperti pada Gambar 4.14.

```
def remove_stopwords(text):
    return stopwords_remover.remove(text)

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

Gambar 4. 14 Sastrawi untuk Stopword Removal

Hasil data teks setelah melalui tahap *stopword removal* ditunjukkan pada Tabel 4.7.

Tabel 4. 7 Data Teks Setelah Stopword Removal

Input	***Output***
aku sedih padahal lagi pengin di support	aku sedih padahal pengin support
kayaknya beban depresi pada cash for healing enggak sih kak wkwkwk	kayaknya beban depresi cash for healing enggak kak wkwkwk
gue sudah di titik capek banget nutup menutupi image keluarga yang harmonis selama ini iya gue harmonis tapi lebih banyaknya enggak terasa kontrol emosional nyokap yang enggak pernah bisa dijaga peran support yang enggak pernah gue dapat untuk gue melakukan apapun di kuliah	gue titik capek banget nutup menutupi image keluarga harmonis selama iya gue harmonis lebih banyaknya enggak terasa kontrol emosional nyokap enggak pernah dijaga peran support enggak pernah gue untuk gue melakukan apapun di kuliah
kadang pengin gue ledakin tapi takut ada yang nyiksa diri entar gue gue juga yang merasa bersalah	kadang pengin gue ledakin takut yang nyiksa diri entar gue gue yang merasa bersalah
rasa bersalah juga kalau jumpa mashayikh tak cuba kucup kepala mereka tapi sebab sedar diri pendek kan malu pula nak jengket	rasa bersalah kalau jumpa mashayikh tak cuba kucup kepala tapi sedar diri pendek kan malu nak jengket
.....	

hari demi hari makin di bikin takut buat ngelanjutin hidup kedepanya makin di bikin takut untuk berharap hidup akan sedikit lebih baik hari demi hari makin di bikin yakin untuk nyerah sama kehidupan marah banget sama diri sendiri di masa lampau karena sudah setuju untuk di lahirin	hari hari makin bikin takut buat ngelanjutin hidup kedepanya makin bikin takut berharap hidup sedikit lebih baik hari hari makin bikin yakin nyerah sama kehidupan marah banget sama diri sendiri masa lampau sudah setuju di lahirin
nyerah saja kali ya benaran capek sama kehidupan capek sama diri sendiri orang sekitar orang di tempat kerja	nyerah kali benaran capek sama kehidupan capek sama diri sendiri orang orang tempat kerja
sudah nyerah sama hidup tuh makanya nyemplungin diri ke minyak	nyerah sama hidup makanya nyemplungin diri minyak
sudah di tahap nyerah sama diri sendiri mencoba buat menjalani saja hidup sampai di panggil tapi mau sampai kapan aku hidup enggak ada nyawa begini im not deserve this kind of life oh god please forgive me	di tahap nyerah sama diri sendiri mencoba buat menjalani hidup di panggil mau kapan aku hidup enggak nyawa begini im not deserve this kind of life god please forgive me
enggak usah kkn nder nanti rame-rame bilang yang salah panitnya mengapai ngewajibin tapi lama	enggak usah kkn nder rame-rame bilang salah panitnya mengapai ngewajibin lama

Setelah melalui keempat tahapan *preprocessing* di atas, didapatkan dataset yang sebelumnya berisi 5.000 data kotor dengan 16 atribut berubah menjadi sebanyak 4.987 data bersih dengan 2 atribut (*label* dan *full_text*) dengan rincian 3.281 data berlabel normal dan 1.706 data berlabel depresi (berpotensi depresi). Gambar 4.15 berikut adalah

kode dan jumlah rinci setiap label pada data *tweet* setelah *preprocessing*.

```
print(df.shape)
```

```
(4987, 2)
```

```
jumlah_per_label = df['label'].value_counts()

print("Jumlah data per label:")
print(jumlah_per_label)
```

```
Jumlah data per label:
label
normal     3281
depresi    1706
```

Gambar 4. 15 Jumlah Data Bersih

Untuk memudahkan pemrosesan data selanjutnya, label akan diubah menjadi karakter numerik. Label depresi diubah menjadi nilai 1, sedangkan label normal diubah menjadi nilai 0. Kode program untuk konversi pelabelan ditunjukkan pada Gambar 4.16.

```
print(df['label'].unique())
```

```
['depresi' 'normal']
```

Gambar 4. 16 Kode Proses Convert Value

Hasil pengubahan nilai label dari *string* (teks) menjadi numerik pada data teks ditunjukkan pada tabel berikut.

Tabel 4. 8 Data Teks Setelah Convert Value

full_text	**label**
aku sedih padahal pengin support	1

kayaknya beban depresi cash for healing enggak kak wkwkwk	0
gue titik capek banget nutup menutupi image keluarga harmonis selama iya gue harmonis lebih banyaknya enggak terasa kontrol emosional nyokap enggak pernah dijaga peran support enggak pernah gue untuk gue melakukan apapun di kuliah	1
kadang pengin gue ledakin takut yang nyiksa diri entar gue gue yang merasa bersalah	1
rasa bersalah kalau jumpa mashayikh tak cuba kucup kepala tapi sedar diri pendek kan malu nak jengket	0
.....	
hari hari makin bikin takut buat ngelanjutin hidup kedepanya makin bikin takut berharap hidup sedikit lebih baik hari hari makin bikin yakin nyerah sama kehidupan marah banget sama diri sendiri masa lampau sudah setuju di lahirin	1
nyerah kali benaran capek sama kehidupan capek sama diri sendiri orang orang tempat kerja	1
nyerah sama hidup makanya nyemplungin diri minyak	0
di tahap nyerah sama diri sendiri mencoba buat menjalani hidup di panggil mau kapan aku hidup enggak nyawa begini im not deserve this kind of life god please forgive me	1
enggak usah kkn nder rame-rame bilang salah panitnya mengapai ngewajibin lama	0

D. *Data Splitting*

Proses selanjutnya adalah pembagian data (*data splitting*) di mana data akan digunakan untuk melatih dan mengukur kemampuan model, dalam hal ini terkait dengan data teks berpotensi depresi.

Seluruh data nantinya akan dibagi menjadi dua kelompok yaitu data latih (*data training*) sebanyak 80% dan data uji (*data testing*) sebanyak 20%. Proses tersebut dapat dilihat pada *source code* berikut.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

Gambar 4. 17 Kode Tahap Data Splitting

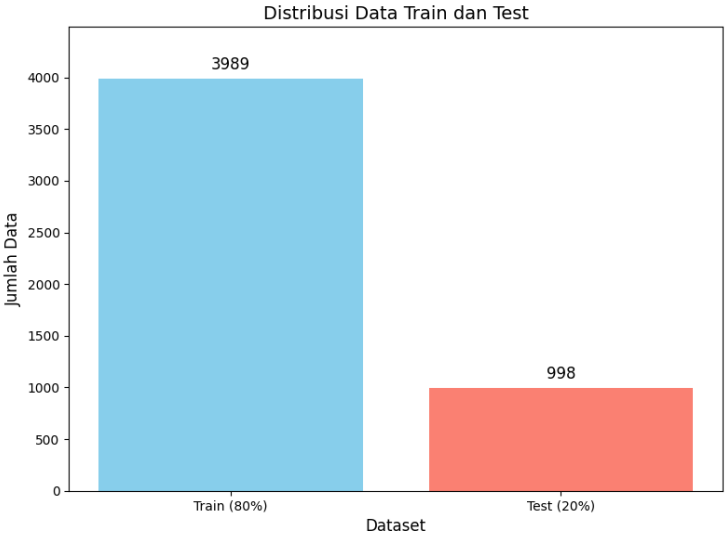
Rincian data yang digunakan dalam proses pelatihan ini adalah 3.989 data untuk training dan 998 data untuk testing. Berikut adalah kode program yang menunjukkan rincian jumlah hasil *data splitting*.

```
# Cek jumlah data
print(f"Jumlah data train: {len(df_train)}")
print(f"Jumlah data test : {len(df_test)}")
```

```
Jumlah data train: 3989
Jumlah data test : 998
```

Gambar 4. 18 Jumlah Data Train dan Test

Lebih jelasnya, berikut pada Gambar 4.19 adalah grafik yang menampilkan distribusi *data train* (batang biru) dan *data test* (batang merah) pada proses ini.



Gambar 4. 19 Distribusi Data Train dan Data Test

E. *IndoBERT Tokenizatoin*

Langkah awal proses tokenisasi IndoBERT adalah dengan mengakses model *pre-trained* IndoBERT dari platform Hugging Face, yaitu azizp128/prediksi-emosi-indobert yang telah digunakan untuk mengklasifikasikan enam kelas emosi berupa marah, sedih, jijik, takut, senang, dan cinta. Model *pre-trained* ini direkomendasikan karena menghasilkan output klasifikasi emosi yang lebih tepat dibandingkan model pre-trained lainnya (Situmorang & Purba, 2024). Model ini juga menggunakan indobert-base-p1 yang telah terbukti cocok untuk tugas NLP (Krisna et al., 2024; Purnomo et al., 2024).

Dalam proses ini, diperlukan *library* transformers untuk bekerja dengan model *pre-trained* yaitu AutoModelForSequenceClassification dan AutoTokenizer. Adapun torch digunakan untuk memanipulasi *layer* klasifikasi pada model (input dan output model). Di tahap ini, peneliti tidak menambahkan argumen untuk mengatur jumlah kelas. Kemudian barulah torch digunakan untuk memodifikasi *layer* klasifikasi model, agar model awal yang telah dilatih untuk memprediksi enam kelas dapat diigunakan untuk menghasilkan dua kelas (berpotensi depresi dan normal). Gambar 4.20 berikut menunjukkan *source code* untuk proses tersebut.

```
from transformers import (
    AutoModelForSequenceClassification, AutoTokenizer
)
import torch

# Path ke model yang sudah diunduh
model_name = "azizp128/prediksi-emosi-indobert"

# Load tokenizer dan model IndoBERT dari lokal
tokenizer = AutoTokenizer.from_pretrained(model_name)
#tanpa jumlah label
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Modifikasi klasifikasi menjadi 2 kelas output
model.classifier = torch.nn.Linear(model.classifier.in_features, 2)
```

Gambar 4. 20 Load dan Modifikasi Tokeneizer

Tahap selanjutnya adalah instalasi datasets transformers. *Library* datasets adalah *library python* yang dikembangkan oleh Hugging Face yang menyediakan akses

secara mudah dan efisien ke berbagai macam dataset yang umum digunakan untuk NLP. Gambar 4.21 berikut adalah kode programnya.

```
!pip install datasets transformers
from datasets import Dataset
```

Gambar 4. 21 Instalasi Library datasets

Proses tokenisasi ini menghasilkan input numerik yang dapat dipahami oleh model, serta *attention masks* yang membantu model memahami struktur urutan. Proses ini juga menangani *padding* (penambahan token agar teks cukup panjang) dan *truncation* (pemotongan teks yang terlalu panjang) untuk memastikan semua input memiliki panjang yang sama dan sesuai. Gambar 4.22 berikut adalah kode programnya.

```
# Fungsi tokenisasi
def tokenize_function(examples):
    return tokenizer(examples["full_text"], padding="max_length", truncation=True)

# Tokenisasi dataset
tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

Gambar 4. 22 Inisialisasi Fungsi Tokenisasi

Dataset yang ada kemudian dibaca dan diubah ke format Hugging Face agar kompatibel untuk proses tokenisasi, khususnya pada kolom *full_text*. Pengaturan ulang indeks DataFrame menjadi indeks integer berurutan seperti pada Gambar 4.23 juga dilakukan untuk konsistensi indeks, menghindari potensi masalah, dan memudahkan

penggunaan fitur-fitur *library* datasets yang bergantung pada indeks, sehingga mempermudah pemrosesan data.

Sebelum memulai keseluruhan proses tokenisasi, terlebih dahulu dilakukan penghapusan indeks yang lama dan membuat ulang indeks agar bersih. Selain itu, ditambahkan pula batas panjang input teks yaitu 128. Angka 128 dipilih karena telah banyak penelitian yang membuktikan bahwa angka ini berhasil memberikan kinerja yang baik untuk tugas NLP tanpa memerlukan sumber daya yang berlebihan (Oswari et al., 2024; Zeberga et al., 2023).

```
# Ubah DataFrame ke Hugging Face Dataset
train_dataset = Dataset.from_pandas(df_train.reset_index(drop=True))
test_dataset = Dataset.from_pandas(df_test.reset_index(drop=True))

# Fungsi tokenisasi yang mendukung batched input
def tokenize_function(examples):
    # kata tokenized dijadikan satu kalimat utuh sebelum diproses
    texts = [' '.join(text) for text in examples["full_text"]]
    return tokenizer(
        texts, # kalimat utuh dimasukkan ke tokenizer
        padding="max_length",
        truncation=True,
        max_length=128
    )
```

Gambar 4. 23 Inisialisasi Fungsi Tokenisasi Sesuai Batched Input

Tokenisasi kemudian diterapkan ke seluruh isi dataset menggunakan fungsi map(), yang mana dataset akan memiliki kolom baru yaitu *input_ads* dan *attention_mask*, namun tetap menyimpan label. Kolom-kolom tersebut nantinya diubah ke dalam format PyTorch tensor agar bisa

langsung digunakan oleh Hugging Face Trainer. Gambar 4.21 adalah kode program yang menunjukkan proses tersebut.

```
# Tokenisasi dataset
train_dataset = train_dataset.map(tokenize_function, batched=True)
test_dataset = test_dataset.map(tokenize_function, batched=True)

# Format dataset untuk digunakan oleh Trainer
train_dataset.set_format(type="torch", columns=["input_ids", "attention_mask", "label"])
test_dataset.set_format(type="torch", columns=["input_ids", "attention_mask", "label"])
```

Gambar 4. 24 Penerapan Map untuk Tokenisasi

Tahapan tokenisasi dengan IndoBERT dapat dilihat pada tabel berikut.

Tabel 4. 9 Tahapan IndoBERT Tokenizing

Teks	aku sedih padahal pengin support	kayaknya beban depresi cash for healing enggak kak wkwkwk
Input	['[CLS]', 'aku', 'sedih', 'pd', '##hl', 'lg', 'pgn', 'di', 'support', '[SEP]']	['[CLS]', 'kek', '##nya', 'beban', 'depresi', 'pada', 'cash', 'for', 'he', '##aling', 'ga', 'si', 'kak', '(', '?', ')', 'wkwkwk', '[SEP]']
Token ID	[2, 304, 5990, 2775, 2056, 4713, 25430, 26, 7195, 3]	[2, 945, 57, 4227, 9138, 126, 7650, 1548, 1991, 9678, 525, 356, 1844, 30464, 30477, 30465, 19720, 3]

Tabel 4.9 menunjukkan konversi teks menjadi fitur numerik seperti “aku” menjadi 304 dan “pgn” menjadi 25430. Semua angka-angka tersebut diperoleh dari *vocabulary* IndoBERT, yaitu kamus bawaan model yang berisi token, subtoken, dan ID numerik yang tidak berubah.

F. *Pre-Fine-Tuning Model*

Dataset kemudian menjalani *pre-fine-tuning model* untuk pemrosesan lebih lanjut dari model *pre-trained*. Diperlukan Optuna dan *library* lain dari transformers pada tahap ini. Berikut adalah kode programnya.

```
!pip install optuna
import optuna
from sklearn.metrics import f1_score
from transformers import (
    AutoTokenizer,
    AutoModelForSequenceClassification,
    Trainer,
    TrainingArguments
)
```

Gambar 4. 25 Instalasi Optuna dan Library Pendukung

Lalu dilakukan inisialisi model *pre-trained* dengan mengubah klasifikasi model menjadi dua kelas seperti pada Gambar 4.26.

```
#Model initialization
def model_init():
    return AutoModelForSequenceClassification.from_pretrained(
        model_name,
        num_labels=2,  # ubah jadi 2 label
        ignore_mismatched_sizes=True  # penting agar model tidak error
    )
```

Gambar 4. 26 Inisialisasi Model dengan Pre-Trained Model

Optuna digunakan untuk *hyperparameter tuning*. Ruang pencariannya melibatkan *learning rate* antara 1e-5 hingga 3e-5, *batch size* antara 8 atau 16, dan *epoch* antara 2 hingga 3 seperti pada Gambar 4.27. Ruang pencarian tersebut telah mengalami sedikit pengurangan dikarenakan sumberdaya komputasi yang terbatas.

Optuna secara otomatis memilih ukuran terbaik untuk masing-masing parameter. Optuna dipilih karena dapat mengidentifikasi kombinasi *hyperparameter* yang optimal untuk meningkatkan efektivitas model (Lai et al., 2024).

```
# Objective function untuk Optuna
def model_objective(trial):
    learning_rate = trial.suggest_float("learning_rate", 1e-5, 3e-5, log=True)
    batch_size = trial.suggest_categorical("batch_size", [8, 16])
    num_epochs = trial.suggest_int("num_train_epochs", 2, 3)
```

Gambar 4. 27 Pengaturan untuk Pemilihan Hyperparameter

Model menggunakan Trainer dari transformers dengan pengaturan seperti strategi evaluasi setiap *epoch*, menyimpan model per *epoch*, dan *seed* tetap (42) untuk reprodusibilitas. Reprodusabilitas ialah kemampuan untuk mengulang kembali suatu eksperimen atau proses, di mana 42 merupakan titik awal yang akan membuat hasil acak menjadi terkendali. Kode tahap ini ada pada Gambar 4.28.

```
training_args = TrainingArguments(
    output_dir="./results",
    eval_strategy="epoch",
    save_strategy="epoch",
    logging_strategy="no",  # Tidak log ke file
    learning_rate=learning_rate,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=num_epochs,
    weight_decay=0.01,
    seed=42,
    load_best_model_at_end=True,
    metric_for_best_model="eval_loss",
    greater_is_better=False,
    report_to="none",
    save_total_limit=1
)
```

Gambar 4. 28 Kode Training Argument

Setelah pengaturan *(training argument)* ditambahkan, kemudian dilakukan proses inti *pre-fine-tuning model* berdasarkan parameter yang sudah diatur. Model kemudian diuji pada data test_dataset. Hasil prediksi nantinya diubah menjadi label prediksi (y_pred) dan dibandingkan dengan label asli (y_true) yang digunakan untuk evaluasi. Evaluasi performa model pada tahap ini menggunakan *weighted f1 score*.

Metrik *f1 score* dipilih karena memiliki kelebihan, yaitu memberikan gambaran yang lebih seimbang tentang kinerja model pada kedua kelas, terutama jika ada ketidakseimbangan kelas (Güldal, 2021). Selain itu, jika menggunakan metrik seperti akurasi sederhana, model yang lebih banyak memprediksi kelas mayoritas (kelas normal) bisa mendapatkan skor akurasi tinggi meskipun sebenarnya tidak memiliki performa yang baik pada kelas minoritas (kelas depresi) yang sebenarnya lebih penting untuk diiidentifikasi. Berikut adalah kode programnya.

```
trainer.train()
preds = trainer.predict(test_dataset_small)
y_true = preds.label_ids
y_pred = np.argmax(preds.predictions, axis=1)

return f1_score(y_true, y_pred, average="weighted")
```

Gambar 4. 29 Pre-Fine-Tuning dan Evaluasi Model

Pengurangan ruang pencarian *hyperparameter* pada penelitian ini, adalah strategi untuk menjaga efisiensi komputasi walaupun dengan sumber daya terbatas. Peneliti menggunakan 20% dari masing-masing dataset dan menambahkan *timeout* maksimal dua jam pada tahap ini. Pendekatan ini didasarkan pada sifat *transfer learning model pre-trained* IndoBERT itu sendiri, di mana model telah memperoleh pemahaman bahasa yang mendalam selama fase *pre-training*.

```
train_dataset.shuffle(seed=42).select(range(int(0.2 * len(train_dataset))))
test_dataset.shuffle(seed=42).select(range(int(0.2 * len(test_dataset))))
trainer.train()
preds = trainer.predict(test_dataset_small)
y_true = preds.label_ids
y_pred = np.argmax(preds.predictions, axis=1)

return f1_score(y_true, y_pred, average="weighted")
study = optuna.create_study(direction="maximize")
study.optimize(model_objective, n_trials=5, timeout=7200)
```

Gambar 4. 30 Pre-Fine-Tuning dengan 20% Dataset dan Timeout

Hasil terbaik dari pencarian *hyperparameter* kemudian akan ditampilkan dalam bentuk parameter dan nilai *f1 score* yang dicapai. Semua percobaan *tuning hyperparameter* ini juga akan disimpan untuk analisis lanjutan. Gambar 4.31 berikut menampilkan seluruh hasil percobaan pemilihan *hyperparameter* yang telah dilakukan oleh Optuna. Terdapat berbagai pilihan *learning rate* antara 1e-05 hingga 2e-05, *batch size* antara 8 hingga 16, dan *epoch* antara 2 atau 3 dari setiap *trial* yang dilakukan.

```
Trial #0
  Status     : 1
  Value      : 0.83
  Hyperparameters:
    learning_rate: 1e-05
    batch_size: 16
    num_train_epochs: 3
------------------------------
Trial #1
  Status     : 1
  Value      : 0.84
  Hyperparameters:
    learning_rate: 1e-05
    batch_size: 8
    num_train_epochs: 3
------------------------------
Trial #2
  Status     : 1
  Value      : 0.83
  Hyperparameters:
    learning_rate: 2e-05
    batch_size: 16
    num_train_epochs: 3
Trial #3
  Status     : 1
  Value      : 0.83
  Hyperparameters:
    learning_rate: 2e-05
    batch_size: 8
    num_train_epochs: 2
------------------------------
Trial #4
  Status     : 1
  Value      : 0.84
  Hyperparameters:
    learning_rate: 2e-05
    batch_size: 8
    num_train_epochs: 2
```

Gambar 4. 31 Hasil Percobaan Pemilihan Hyperparameter

Gambar 4.31 menunjukkan kinerja Optuna tetap efektif menemukan kombinasi *hyperpatameter* terbaik, meskipun dataset dan waktu yang digunakan pada tahap ini dibatasi. Hal ini dikarenakan Optuna menggunakan algoritma optimasi *Bayesian*, bukan *brute force*, sehingga dapat mengevaluasi setiap kombinasi dan menghitung probabilitas hasil terbaik dari kombinasi berikutnya. Selain itu, fitur *pruning* yang dimiliki Optuna memungkinkannya untuk menghentikan percobaan yang tidak menjanjikan, sehingga waktu yang terbatas tetap digunakan secara efisien.

Setelah itu, hasil kombinasi terbaik dari proses *tuning hyperparameter* akan ditampilkan untuk menunjukkan hasil optimasi model dan memberikan informasi tentang konfigurasi terbaik untuk *fine-tuning model*.

```
# Melihat kombinasi terbaik
best_trial = study.best_trial
print(f"Best F1 score: {best_trial.value:.2f}")
print("Best hyperparameters:")
for key, value in best_trial.params.items():
    if key == "learning_rate":
        print(f"    {key}: {value:.0e}")
    elif isinstance(value, float):
        print(f"    {key}: {value:.2f}")
    else:
        print(f"    {key}: {value}")
```

```
Best F1 score: 0.84
Best hyperparameters:
    learning_rate: 2e-05
    batch_size: 8
    num_train_epochs: 2
```

Gambar 4. 32 Melihat Kombinasi Hyperparameter Terbaik

Gambar 4.32 menunjukkan kombinasi *hyperparameter* terbaik. Didapatkan *f1 score* terbaik yaitu 84% dengan *learning rate* 2e-05, *batch size* 8, dan *epoch* sebanyak 2. Kombinasi inilah yang nanitnya akan digunakan untuk *fine-tuning model*.

G. *Fine-Tuning*

Pada tahap ini, hyperparameter terbaik yang telah disimpan dalam variabel best_params akan digunakan kembali dalam tahap *fine-tuning*. Pada fase *fine-tuning*

digunakan keseluruhan dataset agar dapat mencapai performa keseluruhan model dan meningkatkan kemampuan generalisasi. Penggunaan 20% dataset hanya digunakan untuk eksplorasi yang cepat dan penghematan sumber daya pada tahap awal. Sedangkan, penggunaan keseluruhan dataset akan menghasilkan performa model yang paling baik untuk tugas spesifik, dalam hal ini mengklasifikasikan teks berpotensi depresi dan normal.

Sama seperti tahap sebelumnya, pada tahap *fine-tuning* ini juga digunakan *early stopping* yang berguna untuk segera menghentikan keseluruhan proses jika kinerja model pada data uji tidak menunjukkan peningkatan selama 1 epoch berturut-turut (setelah kinerja terbaik terakhir). Kode tahap ini ditunjukkan pada Gambar 4.33 berikut.

```
final_trainer = Trainer(
    model=model_init(),
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    tokenizer=tokenizer,
    data_collator=DataCollatorWithPadding(tokenizer),
    callbacks=[EarlyStoppingCallback(early_stopping_patience=1)]
)
```

Gambar 4. 33 Kode Tahap Fine-Tuning

Dalam proses ini, diketahui pula nilai *training loss* dan *validation loss*. *Training loss* digunakan untuk mengukur seberapa baik model mempelajari data latih. *Validation loss*

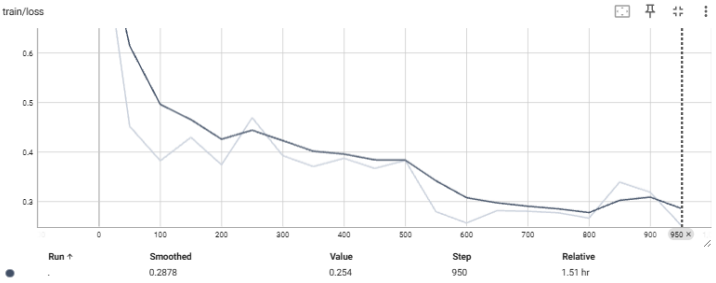
digunakan untuk mengukur seberapa baik model bekerja pada data yang belum pernah dilihat. Gambar 4.34 berikut memperlihatkan nilai *training loss* dan *validation loss* selama 2 epoch (proses pelaithan) dalam *fine-tuning*.

Epoch	Training Loss	Validation Loss
1	0.367900	0.379701
2	0.254000	0.440212

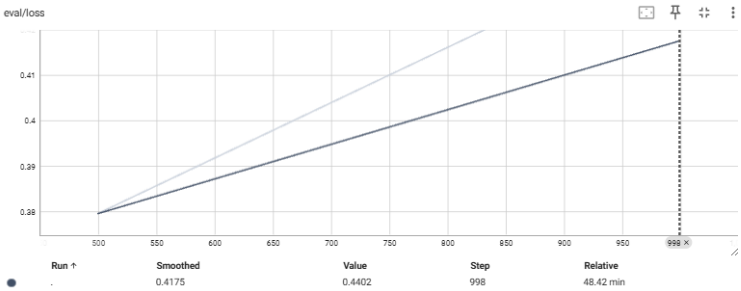
```
Final F1 Score (Full Dataset): 0.8684
```

Gambar 4. 34 Hasil Fine-Tuning Model

Adapun Gambar 4.35 dan 4.36 di bawah ini menunjukkan grafik penurunan *training loss* dan peningkatan *validation loss*. Artinya, model semakin baik dalam memprediksi hasil pada data latih, namun kerjanya cukup memburuk pada data baru (data uji).



Gambar 4. 35 Penurunan Training Loss



Gambar 4. 36 Kenaikan Validation Loss

Sumbu X pada grafik *training loss* dan *validation loss* di atas menunjukkan jumlah langkah pelatihan (*step*), yaitu 500 *step,* berarti model sudah belajar dari 500 batch data. Sedangkan sumbu Y menunjukkan nilai *loss*, seberapa besar kesalahan prediksi model terhadap data latih dan data uji.

Kurva *loss* pada grafik *training loss* menurun seiring waktu, dari 0,37 ke 0,25. Hal ini menunjukkan bahwa model belajar dengan baik dari data pelatihan, semakin lama dilatih maka semakin kecil kesalahannya terhadap data latih. Penurunan *loss* ini normal terjadi dan diharapkan selama proses ini. Sedangkan kurva *loss* pada grafik *validation loss* meningkat dari 0,38 menjadi 0,44. Hal ini menunjukkan bahwa model semakin buruk dalam memprediksi data validasi meskipun *training loss* turun. Model akhir ini kemudian disimpan dalam Google Drive

untuk evaluasi dan tugas klasifikasi berikutnya, kode programnya ditunjukkan pada Gambar 4.37.

```
# Ambil model dari trainer
final_model = final_trainer.model

# Path ke Google Drive (pastikan folder ada)
model_path = "/content/drive/MyDrive/SKRIPSI/my_indobert_model"

# Simpan model dan tokenizer
final_model.save_pretrained(model_path)
tokenizer.save_pretrained(model_path)
```

Gambar 4. 37 Model Disimpan ke Google Drive

H. *Evaluasi Model*

Performa model secara keseluruhan kemudian diukur menggunakan *confusion matrix* 2x2, serta nilai akurasi, presisi, *recall*, dan *f1 score*. Sebelum melakukan proses evaluasi, diperlukan *import* beberapa *library* pendukung dari sklearn.metrics dan matplotlib. Kode program untuk instalasi dan *import* ditunjukkan pada Gambar 4.38.

```
from sklearn.metrics import (
    classification_report,
    confusion_matrix,
    ConfusionMatrixDisplay
)
import matplotlib.pyplot as plt
import numpy as np
```

Gambar 4. 38 Instalasi Library untuk Evaluasi

Confusion matrix 2x2 dipilih karena dalam penelitian ini, kelas hanya terbagi menjadi dua yaitu depresi (berpotensi depresi) dan normal. *Confusion matrix*

merangkum kinerja model klasifikasi dengan membandingkan prediksi model dengan label sebenarnya. Berikut adalah tabel *confusion matrix* 2x2.

Tabel 4. 10 Confusion Matrix Klasifikasi Potensi Depresi

		True Class	
		Berpotensi Depresi	*Normal*
Predicted Class	*Berpotensi Depresi*	TP	FP
	Normal	FN	TN

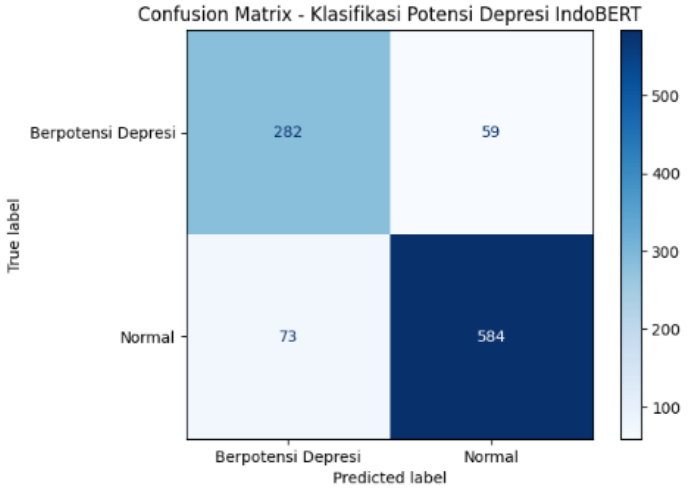
Kode untuk penghitungan *confusion matrix* tersebut beserta visualisasinya terdapat pada Gambar 4.39 berikut.

```
# Hitung confusion matrix
cm = confusion_matrix(y_true, y_pred, labels=[1, 0])

# Tampilkan visualisasi
disp = ConfusionMatrixDisplay(
    confusion_matrix=cm,
    display_labels=["Berpotensi Depresi", "Normal"]
)
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix - Klasifikasi Potensi Depresi IndoBERT")
plt.show()

# Hitung TP, TN, FP, FN berdasarkan urutan: [Normal=0, Depresi=1]
TP = cm[0, 0]
FN = cm[1, 0]
FP = cm[0, 1]
TN = cm[1, 1]
```

Gambar 4. 39 Penghitungan Confusion Matrix 2x2




```
TP (Depresi terdeteksi benar): 282
TN (Normal terdeteksi benar): 584
FP (Normal tapi diprediksi Depresi): 59
FN (Depresi tapi diprediksi Normal): 73
```

Gambar 4. 40 Hasil Penghitungan Confusion Matrix 2x2

Gambar 4.40 menunjukkan hasil nilai *confusion matrix* dengan rincian yaitu TP berjumlah 282, TN berjumlah 584, FP berjumlah 59, dan FN berjumlah 73. Hasil tersebut digunakan untuk mengukur akurasi, presisi, *recall*, dan *f1 score.* Penghitungan keempat metrik tersebut adalah sebagai berikut.

$$Accuration = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (5)$$

$$= \frac{282+584}{282+584+59+73} \times 100\%$$

$$= \frac{866}{998} \times 100\%$$

$$= 0{,}837 \times 100\%$$

$$= 83{,}7\% \approx 84\%$$

Persamaan 5 menunjukkan penghitungan nilai akurasi model, diperoleh nilai akurasi atau ketepatan model dalam melakukan klasifikasi potensi depresi dan normal adalah 84%, sehingga ada 16% data yang tidak terdeteksi secara akurat. Jumlah data yang terdeteksi dengan benar adalah hasil dari penjumlahan TP dan TN yaitu 866 data, sedangkan data yang tidak terdeteksi dengan benar hasil dari penjumlahan FP dan FN yaitu adalah 133 data.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (6)$$

$$= \frac{282}{282+59} \times 100\%$$

$$= \frac{282}{341} \times 100\%$$

$$= 0{,}827 \times 100\%$$

$$= 82{,}7\% \approx 83\%$$

Persamaan 6 menunjukkan penghitungan nilai presisi model, diperoleh nilai presisi atau ketepatan model dalam memprediksi teks berpotensi depresi dan normal adalah 83%. Dengan kata lain, semakin nilai presisi model mendekati 100%, maka semakin akurat pula prediksi yang dibuatnya.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

$$= \frac{282}{282+73} \times 100\%$$

$$= \frac{282}{355} \times 100\%$$

$$= 0{,}794 \times 100\%$$

$$= 79{,}4\% \approx 79\%$$

Persamaan 7 menunjukkan penghitungan nilai *recall* model, diperoleh nilai *recall* atau ketepatan model dalam menemukan kembali teks berpotensi depresi normal dan dari semua data masing-masing kelas adalah 79%.

$$f1\ score = 2 \times \frac{Precision\ .\ Recall}{Precison + Recall} \times 100\% \quad (8)$$

$$= 2 \times \frac{83\%\ .\ 79\%}{83\% + 79\%} \times 100\%$$

$$= 2 \times\ 0{,}441 \times 100\%$$

$$= 88{,}2\% \approx 88\%$$

Persamaan 8 menunjukkan penghitungan nilai *f1 score* model, diperoleh nilai *f1 score* atau keseimbangan antara presisi dan sensitivitas *(recall)* model adalah 88%. Selain penghitungan manual, dilakukan pula penghitungan keempat metrik evaluasi secara otomatis seperti yang terlampir pada gambar di bawah ini.

```
preds = final_trainer.predict(test_dataset)
y_true = preds.label_ids
y_pred = np.argmax(preds.predictions, axis=1)

print("=== Classification Report ===")
print(classification_report(y_true, y_pred, digits=2))
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.91      0.89      0.90       657
           1       0.79      0.83      0.81       341

    accuracy                           0.87       998
   macro avg       0.85      0.86      0.85       998
weighted avg       0.87      0.87      0.87       998
```

Gambar 4. 41 Classification Report

Dari Gambar 4.41 di atas, dapat disimpulkan bahwa tingkat keberhasilan model dalam mencari ketepatan label teks yang diminta *(precision)* untuk kelas normal sebesar 91% dan kelas depresi sebesar 79%, artinya proporsi label yang diprediksi anatara kelas normal dan depresi cukup timpang. Model cenderung terlatih dengan data berlabel normal dibanding depresi karena *class imbalance*.

Adapun tingkat keberhasilan model dalam mengidentifikasi dan menemukan kembali sebuah informasi atau label *(recall)* untuk kelas normal sebesar 89% dan kelas depresi sebesar 83%. Artinya keberhasilan model dalam menemukan kembali teks berlabel depresi lebih rendah dibandingkan dengan menemukan kembali teks berlabel normal. Sedangkan rata-rata harmonik dari *precision* dan *recall (f1 score)* untuk kelas normal sebesar 90% dan kelas depresi sebesar 81%. Hal ini menunjukkan

model cukup efektif dalam mengklasifikasikan kelas normal debandingkan dengan kelas depresi karena presisi dan *recall* kelas depresi juga relatif lebih rendah, sehingga ada kemungkinan bahwa seseorang yang mengalami depresi disalah artikan sebagai normal.

Dari keseluruhan pengukuran, diperoleh nilai *precission* sebesar 87%, *recall* sebesar 87%, *f1 score* sebesar 87%, dan akurasi keseluruhan model sebesar 87%. Meskipun akurasi model hanya 87%, model ini telah melalui berbagai modifikasi terutama pada *layer* output (klasifikasi), di mana model pre-trained sebelumnya memiliki enam klasifikasi emosi, sedangkan model ini berfokus pada dua klasifikasi yaitu berpotesi depresi dan normal. Oleh karena itu, secara keseluruhan, model ini dinilai cukup baik untuk mengklasifikasikan teks berpotensi depresi dan normal (Zhou & Mohd, 2025).

I. Implementasi Model dengan Streamlit

Pada tahap ini, terlebih dahulu dilakukan instalasi *library* streamlit pyngrok agar peneliti dapat menjalankan streamlit sekaligus membuat *tunnel* dari localhost ke internet publik, sehingga aplikasi streamlit tersebut dapat diakses dari browser di perangkat lain, bahkan di luar

Google Colab atau sistem lokal. Kode tersebut ditunjukkan pada Gambar 4.42 berikut.

```
!pip install streamlit pyngrok transformers
```

Gambar 4. 42 Instalasi Streamlit

Kode untuk antarmuka *website* ditulis dan disimpan dalam *file* bernama app.py. Di dalam *file* tersebut terdapat penambahan fitur-fitur utama dari aplikasi ini, antara lain input teks oleh pengguna, penghitungan jumlah kata (maksimal 100 kata), tombol prediksi, hasil prediksi ditampilkan beserta *confidence score*, dan peringatan jika teks kosong atau melebihi batas. Kode untuk *interface website* tersebut terdapat dalam Gambar 4.43.

```
# UI Streamlit
st.title("🧠 Deteksi Potensi Depresi dengan IndoBERT")
st.write("Masukkan kalimat, dan model akan memprediksi apakah kalimat tersebut mengandung potensi depresi.")

user_input = st.text_area("✍️ Masukkan teks (maksimal 100 kata):", height=150)

# Hitung jumlah kata
word_count = len(user_input.strip().split())
st.write(f"Jumlah kata: {word_count}/100")

# Tampilkan warning jika melebihi 100 kata
if word_count > 100:
    st.warning("⚠️ Jumlah kata melebihi batas maksimal (100 kata)")

if st.button("🔍 Prediksi"):
    if user_input.strip() == "":
        st.warning("Masukkan teks terlebih dahulu.")
    else:
        prediction, confidence = predict_sentiment(user_input)

        if prediction == 1:
            st.markdown(f"🔴 Kalimat ini terdeteksi **berpotensi depresi** (confidence: `{confidence:.2f}`)")
        else:
            st.markdown(f"🟢 Kalimat ini terdeteksi **normal** (confidence: `{confidence:.2f}`)")
"""
```

Gambar 4. 43 Kode untuk UI Streamlit

Setelah keseluruhan kode disimpan dalan *file* app.py, peneliti menggunakan *authentication token* khusus dari

Ngrok untuk mendapatkan URL publik. File tersebut kemudian dijalankan dan URL publik akan diberikan oleh Ngrok secara otomatis. Gambar 4.44 menunjukkan kode untuk tahap tersebut.

```
import os
from pyngrok import ngrok

# Jalankan Streamlit
os.system("nohup streamlit run app.py &")

# Buka tunnel
public_url = ngrok.connect(8501)
print("✅ Akses aplikasi Streamlit Anda di:")
print(public_url)
```

```
✅ Akses aplikasi Streamlit Anda di:
NgrokTunnel: "https://7c04-34-83-227-134.ngrok-free.app" -> "http://localhost:8501"
```

Gambar 4. 44 Menjalankan File Kode dan URL

Setelah kode di atas berhasil dijalankan, *website* Streamlit untuk deteksi potensi depresi akan muncul di halaman (*tab*) berbeda. Cara menggunakan *website* ini adalah pengguna diharuskan menginputkan teks dengan batas maksimal 100 kata. Jika kolom teks kosong atau teks melebihi batas, maka otomatis akan muncul peringatan. Setelah pengguna memasukkan teks, pengguna harus mengeklik tombol "prediksi". Kemudian, akan muncul hasil klasifikasi berupa "normal" atau "berpotensi depresi" beserta *confidence score*-nya. *Confidence score* adalah skor yang menunjukkan seberapa yakin model terhadap prediksi yang dibuatnya, di mana skor berkisar antara 0% hingga 100%. Berikut adalah hasil implementasi model

menggunakan *framework* Streamlit untuk aplikasi deteksi potensi depresi berbasis *website.*

Gambar 4. 45 UI Streamlit Menampilkan Hasil Normal

Gambar 4. 46 UI Streamlit Menampilkan Hasil Berpotensi Depresi

Gambar 4.45 dan 4.46 memperlihatkan dua *input* teks dari dua pengguna berbeda dan memiliki hasil prediksi yang berbeda pula. Berikut pada Gambar 4.11 ditunjukkan *input* dan hasil prediksi yang lebih jelas.

Tabel 4. 11 Input dan Output dari Aplikasi Streamlit

Input	**Hasil Prediksi**	**Confidence Score**
oiiii aku abis semprooooo dg jadwal pengumuman H-1 dan belajar pun sama wkwkkwkw. untungnya sempro online dannn pengujinya hamdalah mantap mania uhuy. tp tp tp aku keknya gajadi riset smt ini deh, haha.. soalnya waktunya ga nyukup coy	Normal	93%
heh mau ceritaaaaa sebenernya aku udah riset tapi ga bilang dosen, eh tapi ditanya dosen jadi aku jujur udah riset pas sebelum sempro eh dosen nya kek marah gituuuu terus kalo disuruh riset semester depan aku males huhuu karna bayar ukt lagi	Berpotensi Depresi	86%

Dari Tabel 4.11 diketahui lebih jelas bahwa model IndoBERT menilai teks pertama sebagai kalimat normal dengan tingkat keyakinan 93%. Adapun teks kedua yang dimasukkan oleh *user* dinilai sebagai kalimat berpotensi depresi oleh model dengan tingkat keyakinan 86%. Hal ini menunjukkan aplikasi prediksi depresi berbasis *website* dengan streamlit berhasil dijalankan dengan baik.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa:

1. Model IndoBERT yang dikembangkan dalam penelitian ini bekerja dengan baik untuk *text mining* dalam mengklasifikasikan *tweet* berpotensi depresi. Rasio data yang digunakan dalam penelitian ini ialah sebesar 80% untuk data latih dan 20% untuk data uji. Proses yang dilakukan untuk melakukan klasifikasi data *tweet* pada penelitian ini mulanya melakukan tahapan *preprocessing*, kemudian Indobert *tokenizing*, *pre-fine-tuning* menggunakan model *pre-trained*, dan *fine-tuning* dengan *learning rate* 2e-05, *batch size* 8, dan *epoch* sebanyak 2. Hasil klasifikasi yang diberikan dapat berupa kelas normal dan berpotensi depresi.
2. Performa model IndoBERT dalam penelitian ini untuk tugas mengklasifikasikan *tweet* berpotensi depresi menghasilkan nilai akurasi sebesar 87%, *precission* sebesar 87%, *recall* sebesar 87% serta *f1 score* sebesar 87%. Namun dataset memiliki ketimpangan yang besar antara data berlabel depresi dan normal, sehingga ada

kemungkinan model salah mengklasifikasikan teks yang seharusnya depresi sebagai teks normal.

B. Saran

Berdasarkan penelitian yang dilakukan, penulis berharap kepada peneliti selanjutnya untuk dapat dikembangkan dan terdapat beberapa saran, yaitu:

1. Menambah koleksi kamus bahasa gaul (*slang*), karena pada media sosial banyak komentar yang berisikan bahasa yang kurang baku.

2. Menambah jumlah data dan fokus pada keseimbangan label normal dan berpotensi depresi yang didapat dari media sosial (sumber) lainnya seperti Instagram, YouTube, Facebook, dan Tik Tok.

3. Menambahkan *data validation* dalam proses *data splitting* secara eksplisit untuk membantu mengurangi potensi bias yang mungkin timbul karena tidak menggunakan *data test* sebagai *data validation*.

DAFTAR PUSTAKA

Aferreyra. (2019). *Establishing the Truth: Vaccines, Social Media, and the Spread of Misinformation*. https://hsph.harvard.edu/exec-ed/news/establishing-the-truth-vaccines-social-media-and-the-spread-of-misinformation/

Ahmad, S. S., Rani, R., Wattar, I., Sharma, M., Sharma, S., Nair, R., & Tiwari, B. (2023). *Hybrid Recommender System for Mental Illness Detection in Social Media Using Deep Learning Techniques*. *2023*.

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. *COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, 1638–1649.

Alhamed, F., Bendayan, R., Ive, J., & Specia, L. (2024). Monitoring Depression Severity and Symptoms in User-Generated Content: An Annotation Scheme and Guidelines. *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, 227–233. https://aclanthology.org/2024.wassa-1.18

Aloysius, S., & Salvia, N. (2021). Analisis Kesehatan Mental Mahasiswa Perguruan Tinggi X Pada Awal Terjangkitnya Covid-19 di Indonesia. *Jurnal Citizenship Virtues*, *1*(2), 83–97. https://doi.org/10.37640/jcv.v1i2.962

Ameer, I., Arif, M., Sidorov, G., Gòmez-Adorno, H., & Gelbukh, A. (2022). Mental Illness Classification on Social Media Texts using Deep Learning and Transfer Learning. *ArXiv Preprint ArXiv:2207.01012*.

Beo, Y. A., Zahra, Z., Dharma, I. D. G. C., Alfianto, A. G.,

Kusumawaty, I., Yunike, Eka, A. R., Endriyani, S., Permatasari, L. I., Iwa, K. R., Widniah, A. Z., Dewi, C. F., Nuryati, E., Faidah, N., Suniyadewi, N. W., Martini, S., & Sinthania, D. (2022). *Ilmu Keperawatan Jiwa dan Komunitas*. PENERBIT MEDIA SAINS INDONESIA.

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, *15*(1), 131. https://doi.org/10.33365/jtk.v15i1.744

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Naacl-Hlt 2019*, *Mlm*, 4171–4186. https://aclanthology.org/N19-1423.pdf

Ente, D. R., Thamrin, S. A., Kuswanto, H., Arifin, S., & Andreza. (2020). Klasifikasi faktor-faktor penyebab penyakit diabetes melitus di rumah sakit unhas menggunakan algoritma c4.5. *Indonesian Journal of Statistics and Its Applications*, *4*(1), 80–88.

Febriani, S., & Sulistiani, H. (2021). Analisis Data Hasil Diagnosa Untuk Klasifikasi Gangguan Kepribadian Menggunakan Algoritma C4.5. *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, *2*(4), 89–95.

GoodStats Data. (2024). *Angka Kasus Bunuh Diri di Indonesia Meningkat 60% dalam 5 Tahun Terakhir*. https://data.goodstats.id/statistic/angka-kasus-bunuh-diri-di-indonesia-meningkat-60-dalam-5-tahun-terakhir-2FzH6

Güldal, S. (2021). Balanced DATA by DBSCAN and Weighted Arithmetic Mean to Improve Performance of Machine Learning Algorithms. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *10*(4), 1563–1574. https://doi.org/10.17798/bitlisfen.985519

Hidayat, W. A., & Nastiti, V. R. S. (2024). PERBANDINGAN KINERJA PRE-TRAINED INDOBERT-BASE DAN INDOBERT-LITE PADA KLASIFIKASI SENTIMEN ULASAN TIKTOK TOKOPEDIA SELLER CENTER DENGAN MODEL INDOBERT. *Jurnal Sistem Informasi*, *11*(2), 13–20. https://doi.org/10.30656/jsii.v11i2.9168

Imaduddin, H., A, F. Y., & Nugroho, Y. S. (2023). *Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach*. *14*(8), 113–117.

Kemenkes RI. (2021). *Kemenkes Beberkan Masalah Permasalahan Kesehatan Jiwa di Indonesia*. Kemenkes RI; Sehat Negeriku. Kemeterian Kesehatan RI. Online. https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20211007/1338675/kemenkes-beberkan-masalah-permasalahan-kesehatan-jiwa-di-indonesia/

Kemenkes RI. (2024). *Cegah Bunuh Diri, Kemenkes Ajak Remaja Bicara Soal Kesehatan Mental*. Sehat Negeriku. Kementerian Kesehatan RI. Online. https://sehatnegeriku.kemkes.go.id/baca/umum/20240917/2446492/cegah-bunuh-diri-kemenkes-ajak-remaja-bicara-soal-kesehatan-mental/

Kholifah, B., Syarif, I., & Badriyah, T. (2020). Mental disorder detection via social media mining using deep learning. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *5*(4), 3019–3316. https://doi.org/https://doi.org/10.22219/kinetik.v5i4.1120

Komnas Perempuan. (2024). *Siaran Pers Komnas Perempuan tentang Hari Pencegahan Bunuh Diri*. https://komnasperempuan.go.id/siaran-pers-detail/siaran-pers-komnas-perempuan-tentang-hari-pencegahan-bunuh-diri

Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 757–770. https://doi.org/10.18653/v1/2020.coling-main.66

Krisna, J. I. T., Luthfiarta, A., Cahya, L. D., Winarno, S., & Nugraha, A. (2024). Comparing Optimizer Strategies For Enhancing Emotion Classification In IndoBERT Models. *Advance Sustainable Science, Engineering and Technology*, *6*(2), 1–8. https://doi.org/10.26877/asset.v6i2.18228

Lai, L. H., Lin, Y. L., Liu, Y. H., Lai, J. P., Yang, W. C., Hou, H. P., & Pai, P. F. (2024). The Use of Machine Learning Models with Optuna in Disease Prediction. *Electronics (Switzerland)*, *13*(23), 1–20. https://doi.org/10.3390/electronics13234775

Liu, X., Shi, T., Zhou, G., Liu, M., Yin, Z., Yin, L., & Zheng, W. (2023). Emotion classification for short texts: an improved multi-label method. *Humanities and Social Sciences Communications*, *10*(1), 1–9. https://doi.org/10.1057/s41599-023-01816-6

Luo, X., Ding, H., Tang, M., Gandhi, P., Zhang, Z., & He, Z. (2020). Attention Mechanism with BERT for Content Annotation and Categorization of Pregnancy-Related Questions on a Community QA Site. *Proceedings - 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020*, 1077–1081. https://doi.org/10.1109/BIBM49941.2020.9313379

Maritska, Z., Prananjaya, A. B., Nabila, S. P., & Parisa, N. (2023). Promosi Kesehatan Jiwa Berbasis Media Sosial (Instagram Live) bagi Masyarakat di Masa Pandemi COVID-19. *Wal'afiat Hospital Journal*, *04*(01), 13–22. https://whj.umi.ac.id/index.php/whj/issue/archive

Milintsevich, K., Sirts, K., & Dias, G. (2023). Towards automatic text-based estimation of depression through symptom prediction. *Brain Informatics*, *10*(1). https://doi.org/10.1186/s40708-023-00185-9

Nabiilah, G. Z., Alam, I. N., Purwanto, E. S., & Hidayat, M. F. (2024a). Indonesian multilabel classification using IndoBERT embedding and MBERT classification. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(1), 1071–1078. https://doi.org/10.11591/ijece.v14i1.pp1071-1078

Nabiilah, G. Z., Alam, I. N., Purwanto, E. S., & Hidayat, M. F. (2024b). Indonesian multilabel classification using IndoBERT embedding and MBERT classification. *International Journal of Electrical and Computer Engineering*, *14*(1), 1071–1078. https://doi.org/10.11591/ijece.v14i1.pp1071-1078

Nugroho, K. S., Sukmadewa, A. Y., Bachtiar, F. A., & Yudistira, N. (2020). *BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews*. 1–10.

Nurtanti, S., & Handayani, S. (2021). Peningkatan Pengetahuan Siswa Tentang Deteksi Dini dan Pencegahan Depresi di SMK Muhammadiyah Baturetno. *Warta LPM*, *24*(1), 134–144.

Onie, S., Usman, Y., Widyastuti, R., Lusiana, M., Angkasawati, T. J., Musadad, D. A., Nilam, J., Vina, A., Kamsurya, R., Batterham, P., Arya, V., Pirkis, J., & Larsen, M. (2024). Indonesia's first suicide statistics profile: an analysis of suicide and attempt rates, underreporting, geographic distribution, gender, method, and rurality. *The Lancet Regional Health - Southeast Asia*, *22*, 100368. https://doi.org/10.1016/j.lansea.2024.100368

Oswari, T., Murniyati, M., Yusnitasari, T., Nurasiah, N., & Wijay, S. (2024). Sentiment Analysis of Indonesian Youtube

Reviews About Lesbian, Guy, Bisexual and Transgender (LGBT) using IndoBERT Fine Tuning. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *15*(1), 26. https://doi.org/10.24843//lkjiti.2024.v15.i01.p03

PKBI Jawa Timur. (2024). *Hari Pencegahan Bunuh Diri: Ironi Meningkatnya Kasus Bunuh Diri dan Pentingnya Kesadaran Mental*. https://pkbi-jatim.or.id/hari-pencegahan-bunuh-diri-ironi-meningkatnya-kasus-bunuh-diri-dan-pentingnya-kesadaran-mental/

Purnomo, T. D., Sutopo, J., & History, A. (2024). *COMPARISON OF PRE-TRAINED BERT-BASED TRANSFORMER MODELS FOR REGIONAL*. *3*(3), 11–21.

Qasim, A., Mehak, G., Hussain, N., Gelbukh, A., & Sidorov, G. (2025). Detection of Depression Severity in Social Media Text Using Transformer-Based Models. *Information (Switzerland)*, *16*(2). https://doi.org/10.3390/info16020114

Revathy S.P., Sindhuja M., & Jayashree R.. (2025). *Streamlit-based Web Application for Parkinson ' s Detection using Machine Learning*. *January*. https://doi.org/10.36548/jaicn.2024.4.006

Ridwansyah, T. (2022). Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *2*(5), 178–185. https://doi.org/10.30865/klik.v2i5.362

Setyanto, A. T. (2023). Deteksi Dini Prevalensi Gangguan Kesehatan Mental Mahasiswa di Perguruan Tinggi. *Wacana*, *15*(1), 66. https://doi.org/10.20961/wacana.v15i1.69548

Situmorang, G. F., & Purba. (2024). Deteksi Potensi Depresi dari

Unggahan Media Sosial X Menggunakan Teknik NLP dan Model IndoBERT. *Building of Informatics, Technology and Science (BITS)*, *6*(2), 649–661. https://doi.org/10.47065/bits.v6i2.5496

Suara Merdeka. (2024). *Jumlah Kasus Bunuh Diri di Indonesia Sepanjang 2024 Meningkat, Paling Banyak di Usia Ini*. https://www.suaramerdeka.com/semarang-raya/0413672766/jumlah-kasus-bunuh-diri-di-indonesia-sepanjang-2024-meningkat-paling-banyak-di-usia-ini

Titla-Tlatelpa, J. de J., Ortega-Mendoza, R. M., Montes-y-Gómez, M., & Villaseñor-Pineda, L. (2021). A profile-based sentiment-aware approach for depression detection in social media. *EPJ Data Science*, *10*(1). https://doi.org/10.1140/epjds/s13688-021-00309-3

Twitter. (2021). *Overview of the different authentication methods*. https://developer.x.com/en/docs/tutorials/authenticating-with-twitter-api-for-enterprise/authentication-method-overview

WHO. (2022). *Mental Disorders*. World Health Organization (WHO). Online. https://doi.org/10.5840/ijpp2023916

WHO. (2023). Depressive Disorder (Depression). In *International Immunopharmacology* (Vol. 67, Issue August 2018). World Health Organization (WHO). Online. https://www.who.int/en/news-room/fact-sheets/detail/depression%0Ahttps://www.who.int/news-room/fact-sheets/detail/depression

Zeberga, K., Attique, M., Shah, B., Ali, F., Jembre, Y. Z., & Chung, T.-S. (2023). A Novel Text Mining Approach for Mental Health Prediction Using Bi-LSTM and BERT Model. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/7893775

Zhang, T., Schoene, A. M., Ji, S., & Ananiadou, S. (2022). Natural language processing applied to mental illness detection : a narrative review. *Digital Medicine*, *December*, 1–13. https://doi.org/10.1038/s41746-022-00589-7

Zhou, S., & Mohd, M. (2025). Mental Health Safety and Depression Detection in Social Media Text Data: A Classification Approach Based on a Deep Learning Model. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2025.3559170

Lampiran I. Lembar Pengesahan Proposal Skripsi

KEMENTERIAN AGAMA
UINVESITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Hamka Ngaliyan Semarang
Telp. 024-7601295 Fax. 7615387

LEMBAR PENGESAHAN

Naskah proposal skripsi berikut ini:

Judul	:	Text Mining untuk Klasifikasi Tweet Berpotensi Depresi via Media Sosial X Menggunakan Model IndoBERT
Penulis	:	Salsabila Safirana Wibisono
NIM	:	2108096024
Jurusan	:	Teknologi Informasi

Telah diujikan dalam seminar proposal skripsi oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana bidang ilmu Teknologi Informasi.

Semarang, 18 Desember 2024

DEWAN PENGUJI

Penguji I

Hery Mustofa, M.Kom
NIP. 198703172019031007

Penguji II

Siti Nur'aini, M.Kom
NIP. 198401312018012001

Penguji III

Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

Penguji IV

Wenty Dwi Yuniarti, M.Kom
NIP. 197706222006042005

Lampiran II. Nota Pembimbing I Proposal Skripsi

NOTA PEMBIMBING

Semarang, 7 November 2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi proposal skripsi dengan:

Judul	:	Text Mining untuk Deteksi Dini Depresi Via Media Sosial X Menggunakan Model IndoBERT
Nama	:	**Salsabila Safirana Wibisono**
NIM	:	2108096024
Jurusan	:	Teknologi Informasi

Saya memandang bahwa proposal skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Proposal.

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,

Dr. Masy Ari Ulinuha S.T., M.T
NIP. 198108122011011007

Lampiran III. Nota Pembimbing II Proposal Skripsi

NOTA PEMBIMBING

Semarang, 21 November 2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi proposal skripsi dengan:

Judul	:	Text Mining untuk Deteksi Dini Depresi Via Media Sosial X Menggunakan Model IndoBERT
Nama	:	**Salsabila Safirana Wibisono**
NIM	:	2108096024
Jurusan	:	Teknologi Informasi

Saya memandang bahwa proposal skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Proposal.

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,

Siti Nur'aini, M.Kom
NIP. 198401312018012001

Lampiran IV. Riwayat Hidup

RIWAYAT HIDUP

A. Identitas Diri

1.	Nama Lengkap	:	Salsabila Safirana Wibisono
2.	Tempat & Tanggal Lahir	:	Madiun, 19 Juli 2004
3.	Alamat Rumah	:	Jl. Mawar RT 3 RW 1 Desa Bancong Kec. Wonoasri Kab. Madiun
4.	No. HP	:	082245746416
5.	E-mail	:	salsaswibisono@gmail.com

B. Riwayat Pendidikan

1. Pendidikan Formal:
   a. Madrasah Ibtidaiyyah Muhammadiyah Caruban
   b. Sekolah Menengah Pertama Negeri 1 Mejayan
   c. Sekolah Menengah Atas Negeri 1 Mejayan
2. Pendidikan Non-Formal:
   a. Pondok Pesantren Tahfidz Al-Qur'an Ahmad Dahlan Caruban

C. Karya Ilmiah

a. Perancangan Game Edukasi Pengenalan Buah untuk Anak Usia Dini Berbasis Android Menggunakan Metode R&D
b. Pembuatan Ecobrick sebagai Upaya Pengelolaan Limbah Plastik dan Pelestarian Lingkungan di Desa Rejosari Kecamatan Kangkung Kabupaten Kendal