

**PERBANDINGAN MODEL INDOBERT DAN MODEL HYBRID
PADA ANALISIS SENTIMEN OPINI MASYARAKAT
TERHADAP JUDI ONLINE**

TUGAS AKHIR

Diajukan Oleh:

GHUFRAN AFHAM ASNAWI

NIM. 200705019

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**PRODI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2025 M / 1446 H**

LEMBAR PERSETUJUAN
PERBANDINGAN MODEL INDOBERT DAN MODEL HYBRID
PADA ANALISIS SENTIMEN OPINI MASYARAKAT
TERHADAP JUDI ONLINE

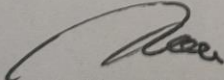
TUGAS AKHIR

Diajukan Kepada Fakultas sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana
pada Prodi Teknologi Informasi

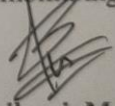
Oleh:
Ghufran Afham Asnawi
NIM. 200705019
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

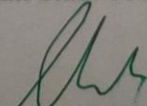
Pembimbing I


Hendri Ahmadian, S.Si., M.I.M
NIP. 198301042014031002

Pembimbing II


Baihaqi, M.T
NIP. 198802212022031001

Mengetahui.
Ketua Program Studi Teknologi Informasi


Malahayati, M.T
NIP. 198301272015032003

LEMBAR PENGESAHAN

**PERBANDINGAN MODEL INDOBERT DAN MODEL HYBRID PADA
ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP JUDI ONLINE**

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Pada Prodi Teknologi Informasi

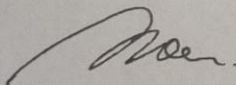
Pada Hari/Tanggal: Rabu, 15 Januari 2025

15 Rajab 1446 H

Di Darussalam, Banda Aceh

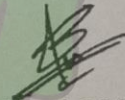
Panitia Ujian Munaqasah Tugas Akhir:

Ketua,



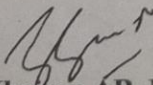
Hendri Ahmadian, S.Si., M.I.M
NIP. 198301042014031002

Sekretaris,



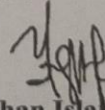
Baihaqi, M.T
NIP. 198802212022031001

Penguji I,



Khairan AR, M.Kom
NIP. 198607042014031001

Penguji II,



Raihan Islamadina, M.T
NIP. 198901312020122011

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,



Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., I.P.U
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Ghufran Afham Asnawi
NIM : 200705019
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul Tugas Akhir : Perbandingan Model IndoBERT dan Model *Hybrid* pada Analisis Sentimen Opini Masyarakat Terhadap Judi Online


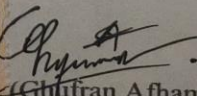
Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 15 Januari 2025
Yang Menyatakan


DAMX129822159

(Ghufran Afham Asnawi)

ABSTRAK

Nama : Ghufran Afham Asnawi
NIM : 200705019
Program Studi : Teknologi Informasi
Judul : Perbandingan Model IndoBERT dan Model *Hybrid* pada Analisis Sentimen Opini Masyarakat Terhadap Judi Online
Tanggal Sidang : 15 Januari 2025
Jumlah Halaman : 67 Halaman
Pembimbing I : Hendri Ahmadian, M.I.M
Pembimbing II : Baihaqi, M.T

Di era digital, media sosial telah menjadi platform penting untuk berbagi informasi, membentuk opini publik, dan menjadi cerminan dinamika sosial. Salah satu isu yang hangat diperbincangkan adalah berita mengenai dugaan keterlibatan sejumlah pihak dalam aktivitas judi online, yang memicu beragam opini di media sosial, termasuk di Instagram. Analisis terhadap komentar masyarakat pada berita-berita ini menjadi penting untuk memahami persepsi publik dan respon publik terhadap fenomena tersebut. Penelitian ini bertujuan untuk menganalisis opini masyarakat terhadap fenomena judi online yang berkembang di media sosial Instagram dengan memanfaatkan model IndoBERT serta pendekatan model hybrid. Dataset yang digunakan mencakup 1.811 komentar dari akun berita nasional, yang telah diklasifikasikan secara manual ke dalam tiga kategori sentimen: positif, negatif, dan netral.

Model yang diimplementasikan dalam penelitian ini mencakup IndoBERT, kombinasi IndoBERT dengan CNN, serta kombinasi IndoBERT dengan RNN. Berdasarkan hasil pengujian, model hybrid IndoBERT-CNN menghasilkan akurasi tertinggi sebesar 75%, diikuti oleh model IndoBERT-RNN dengan akurasi 72%, sedangkan model dasar IndoBERT memperoleh akurasi 60%. Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score. Temuan ini menunjukkan bahwa integrasi IndoBERT dengan CNN lebih efektif dalam mengidentifikasi sentimen publik dibandingkan model lainnya. Studi ini memberikan kontribusi dalam pengembangan teknik analisis sentimen berbasis pemrosesan bahasa alami (NLP) yang dirancang khusus untuk bahasa Indonesia.

Kata Kunci: Analisis Sentimen, IndoBERT, Model Hybrid, CNN, RNN, Judi Online, Instagram.

ABSTRAK

Name : Ghufran Afham Asnawi
Student Number : 200705019
Department : Information Technology
Title : Comparison of IndoBERT Model and Hybrid Model on
Sentiment Analysis of Public Opinion towards Online
Gambling
Bate : 15 January 2025
Number of Pages : 67 Pages
Supervisor I : Hendri Ahmadian, M.I.M
Supervisor II : Baihaqi, M.T

In the digital era, social media has become an important platform for sharing information, shaping public opinion, and reflecting social dynamics. One of the most discussed issues is the news about the alleged involvement of a number of parties in online gambling activities, which triggered various opinions on social media, including on Instagram. Analysis of public comments on these news stories is important to understand public perception and public response to the phenomenon. This study aims to analyze public opinion on the phenomenon of online gambling that has developed on Instagram social media by utilizing the IndoBERT model and a hybrid model approach. The dataset used includes 1,811 comments from national news accounts, which have been manually classified into three sentiment categories: positive, negative and neutral.

The models implemented in this research include IndoBERT, a combination of IndoBERT with CNN, as well as a combination of IndoBERT with RNN. Based on the test results, the IndoBERT CNN hybrid model produced the highest accuracy of 75%, followed by the IndoBERT-RNN model with 72% accuracy, while the basic IndoBERT model obtained 60% accuracy. Performance evaluation was conducted using accuracy, precision, recall, and F1-score metrics. The findings show that the integration of IndoBERT with CNN is more effective in identifying public sentiment than other models. This study contributes to the development of natural language processing (NLP) based sentiment analysis techniques specifically designed for the Indonesian language.

Keywords: Sentiment Analysis, IndoBERT, Hybrid Model, CNN, RNN, Online Gambling, Instagram.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah segala puja dan puji hanya milik Allah SWT, karena dengan rahmat dan kasih sayang-Nya yang telah banyak memberikan karunia-Nya berupa kekuatan, kesatuan, serta kesempatan kepada penulis untuk menyelesaikan penulisan skripsi ini dengan baik dan tepat waktu. Shalawat dan salam juga penulis sanjungkan ke hadirat Nabi Muhammad SAW, yang telah membawa umat manusia dari alam kebodohan kepada alam yang penuh ilmu pengetahuan. Adapun judul tugas akhir ini adalah “Perbandingan Model IndoBERT dan Model *Hybrid* pada Analisis Sentimen Opini Masyarakat Terhadap Judi Online”. Penulisan tugas akhir ini bertujuan untuk memenuhi beban studi guna memperoleh gelar sarjana pada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Tidak banyak yang akan saya lakukan dengan selesainya penulisan skripsi ini, melainkan hanya sekedar ucapan terima kasih kepada semua pihak, baik secara individu maupun kelompok yang telah terlibat dan mendukung saya mulai dari awal hingga selesainya penulisan skripsi ini. Dalam hal ini saya ingin mengucapkan ucapan terimakasih kepada:

1. Kedua orangtua yang senantiasa telah memberikan dukungan dan doa kepada saya dalam menyusun tugas akhir ini.
2. Ketua dan Sekretaris Program Studi Teknologi Informasi Ibu Malahayti, M.T dan bapak Khairan AR, M.Kom yang telah memberikan bimbingan dan arahan dalam penyusunan tugas akhir ini.
3. Bapak Hendri Ahmadian, S.Si., M.I.M selaku Pembimbing I dan bapak Baihaqi, M.T selaku Pembimbing II yang senantiasa memberikan bimbingan, kritik, dan saran, dan juga yang telah mengikhhlaskan waktu, tenaga dan ilmu untuk membimbing penulis dalam menyelesaikan laporan tugas akhir ini.
4. Ibu Cut Ida Rahmadiana, S.Si selaku *staff* Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
5. Bapak Dr. Ir. M. Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.

6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu kepada penulis dalam bidang Teknologi Informasi.
7. Teman-teman seperjuangan Prodi Teknologi Informasi UIN Ar-Raniry tercinta.
8. Segenap pihak yang telah membantu yang tidak dapat penulis ucapkan namanya satu-persatu

Semoga atas partisipasi dan motivasi yang telah diberikan menjadi amal kebaikan dan mendapat pahala yang setimpal disisi Allah SWT. dengan segala kerendahan hati penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari kesempurnaan karena keterbatasan kemampuan ilmu peneliti. Untuk itu, penulis sangat mengharapkan saran dan kritikan yang bersifat membangun demi kesempurnaan skripsi ini dimasa yang akan datang.

Banda Aceh, 05 Januari 2025

Penulis

Ghufran Afham Asnawi



DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
<i>ABSTRAK</i>	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II	4
TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terdahulu	4
2.2 Instagram.....	6
2.3 Analisis Sentimen	6
2.4 Bidirectional Encoder Representation From Transformers (BERT).....	7
2.4.1 Encoder	7
2.4.2 Decoder	8
2.5 IndoBERT.....	8
2.6 Python	9
2.7 Google Colaboratory.....	10
2.7.1 Numpy (Numerical Python)	10
2.7.2 Pandas	10
2.7.3 Scikit-Learn.....	11

2.7.4	TextBlob.....	11
2.8	<i>Confusion Matrix</i>	11
2.9	Data Scraping.....	13
2.10	<i>Convulational Neural Network (CNN)</i>	14
2.11	Model Hybrid.....	14
2.12	<i>Recurrent Neural Network (RNN)</i>	15
BAB III.....		17
METODOLOGI PENELITIAN.....		17
3.1	Tahapan Penelitian.....	17
3.2	Pengumpulan Data.....	17
3.3	Pelabelan Data.....	17
3.4	<i>Pre-processing</i>	18
3.5	Implementasi Model Arsitektur.....	19
3.5.1	IndoBERT.....	19
3.5.2	Model Hybrid.....	20
3.6	Evaluasi Model.....	21
BAB IV.....		22
HASIL DAN PEMBAHASAN.....		22
4.1	Pengumpulan Data.....	22
4.2	Pelabelan Data.....	23
4.3	<i>Pre-Processing Data</i>	23
4.3.1	Case Folding.....	23
4.3.2	Filtering.....	24
4.3.3	Normalisasi Teks.....	24
4.4	Split Data.....	26
4.5	Perancangan Model IndoBERT.....	27
4.6	Perancangan Model IndoBERT dengan CNN.....	27
4.7	Perancangan Model IndoBERT dengan RNN.....	29
4.8	<i>Hyperparameter</i>	31
4.9	Hasil Pelatihan.....	31
4.9.1	Hasil Train validation Accuracy dan Loss Model IndoBERT.....	32
4.9.2	Hasil Train-validation Accuracy dan Loss Model IndoBERT dengan Model CNN.....	33

4.9.3 Hasil Train-validation Accuracy dan Loss Model IndoBERT dengan Model RNN.....	34
4.10 Hasil Confusion Matrix dan Hasil Evaluasi Model	35
4.10.1 <i>Confusion Matrix</i> model IndoBERT.....	36
4.10.2 <i>Confusion Matrix</i> Model IndoBERT dengan Model CNN	37
4.10.3 <i>Confusion Matrix</i> Model IndoBERT dengan Model RNN	38
4.10.4 Hasil Pengujian Model IndoBERT.....	39
4.10.5 Hasil Pengujian Model IndoBERT dengan Model CNN.....	40
4.10.6 Hasil Pengujian Model IndoBERT dengan Model RNN.....	42
BAB V	45
KESIMPULAN DAN SARAN	45
5.1 Kesimpulan.....	45
5.2 Saran	45
DAFTAR PUSTAKA.....	47
LAMPIRAN.....	50



DAFTAR GAMBAR

Gambar 3. 1 Metode Penelitian	17
Gambar 3. 2 IndoBERT Model	19
Gambar 3. 3 Model Hybrid IndoBERT dengan CNN/RNN	20
Gambar 4. 1 Komentar Netizen Terhadap Judi Online	22
Gambar 4. 2 Hasil Pengumpulan Data	23
Gambar 4. 3 Kode Normalisasi Teks	25
Gambar 4. 4 Kode Case Folding dan Filtering.....	25
Gambar 4. 5 Teks setelah melewati Pre-Processing.....	26
Gambar 4. 6 Kode Penerapan IndoBERT base p1	27
Gambar 4. 7 Penerapan model IndoBERT + CNN	28
Gambar 4. 8 Model Hybrid IndoBERT + RNN	30
Gambar 4. 9 Grafik Akurasi dan loss	32
Gambar 4. 10 Gambar Grafik Akurasi dan Loss.....	33
Gambar 4. 11 Gambar Grafik Akurasi dan Loss.....	34
Gambar 4. 12 Confusion Matrix Model IndoBERT.....	36
Gambar 4. 13 Confusion Matrix Model IndoBERT dengan Model CNN	37
Gambar 4. 14 Confusion Matrix Model IndoBERT dengan Model RNN	38
Gambar 4. 15 Performa Model Terhadap Dataset Judol	39
Gambar 4. 16 Performa Model Hybrid Terhadap Dataset Judol.....	40
Gambar 4. 17 Performa Model Hybrid Terhadap Dataset Judol.....	42



DAFTAR TABEL

Table 2. 1 Penelitian Terdahulu	4
Table 2. 2 Confusion Matrix	12
Table 3. 1 Contoh Pelabelan Data.....	18
Table 4. 1 jumlah Split Dataset	26
Table 4. 2 Hyperparamater.....	31
Table 4. 3 Hasil perbandingan Akurasi setiap Dataset.....	44



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam era digital yang terus berkembang, teknologi informasi (TI) memainkan peran penting dalam mengelola dan menganalisis data yang dihasilkan oleh masyarakat melalui berbagai macam platform, contohnya media sosial. Analisis sentimen, sebagai cabang dari kecerdasan buatan (AI) dan pemrosesan bahasa alami (NLP), mengizinkan kita untuk mengklasifikasikan, mengekstrak, dan mengevaluasi opini yang tersebar luas di platform media sosial. Dengan memanfaatkan teknologi ini, kita dapat lebih baik dalam memahami pandangan, pendapat, dan respon masyarakat terhadap berbagai isu sosial, politik, dan ekonomi yang sedang berkembang di dunia.

Analisis sentimen merupakan salah satu teknik penelitian yang paling populer digunakan untuk memahami pendapat dan perasaan orang berdasarkan data teks (Leelawat et al., 2022). Analisis sentimen juga disebut dengan opinion mining adalah bidang studi yang menganalisis opini, sentimen, evaluasi, penilaian, sikap, dan emosi orang terhadap entitas (Liu, 2012).

Analisis sentimen merupakan pengolahan bahasa alami, komputasi linguistik, dan text mining yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian, serta emosi seseorang terkait pembicara atau penulis berkenaan dengan suatu topik, produk layanan, organisasi, individu, ataupun kegiatan tertentu (A. Kaon and S. R. Poteet, Natural Language Processing and Text Mining, 1 ed., London: Springer London, 2007.).

Dalam penelitian ini, peneliti memilih menggunakan media sosial. Media sosial telah menjadi platform yang populer bagi masyarakat untuk mengekspresikan opini dan sentimen mereka terhadap berbagai topik. Instagram sebagai salah satu media sosial yang banyak digunakan oleh masyarakat Indonesia. Menurut laporan Napoleon Cat, ada 90,51 juta pengguna Instagram di Indonesia pada April 2024. Peneliti ingin melakukan penelitian terhadap opini yang dikeluarkan oleh masyarakat terkait berita

yang dikeluarkan oleh beberapa akun berita nasional yang ada di Instagram tentang judi online.

Peneliti dalam penelitian ini menggunakan model IndoBERT dan model hybrid dalam melakukan penelitian, model hybrid sendiri disini yaitu model yang menggabungkan model BERT dengan model jaringan saraf, penggabungan kedua model ini telah menghasilkan peningkatan akurasi dibandingkan dengan menggunakan keduanya secara terpisah. Peneliti menggunakan model Machine Learning Convolutional Neural Network (CNN) dan Recurrent Neural Network (RNN) yang akan digabungkan dengan model BERT dalam penelitian ini, yaitu model IndoBERT.

Indobert, sebagai model bahasa berbasis teknologi BERT (Bidirectional Encoder Representation from Transformer) khusus untuk bahasa Indonesia, menawarkan kemampuan pemahaman konteks dan analisis teks yang mendalam. Dengan semakin meningkatnya penggunaan media sosial dan platform daring di Indonesia, memahami sentimen publik terhadap berbagai isu menjadi semakin penting. Dengan menggunakan model IndoBERT dan model hybrid, peneliti bertujuan untuk membandingkan hasil analisis yang dihasilkan antara model IndoBERT dengan model hybrid, yaitu model IndoBERT dengan CNN dan IndoBERT dengan RNN.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan beberapa permasalahan antara lain :

1. Bagaimana penerapan model IndoBERT dalam analisis sentimen pada komentar berita di media sosial Instagram?
2. Bagaimana penerapan model IndoBERT dengan CNN dan IndoBERT dengan RNN dalam analisis sentimen pada komentar berita di media sosial Instagram?
3. Bagaimana perbandingan kinerja model IndoBERT, IndoBERT dengan CNN dan IndoBERT dengan RNN dalam analisis sentimen pada komentar berita di media sosial Instagram?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk :

1. Menerapkan model IndoBERT dalam analisis sentimen pada komentar berita di media sosial Instagram.
2. Menerapkan model IndoBERT dengan CNN dan IndoBERT dengan RNN dalam analisis sentimen pada komentar berita di media sosial Instagram.
3. Membandingkan hasil kinerja antara model IndoBERT, IndoBERT dengan CNN dan IndoBERT dengan RNN dalam analisis sentimen pada komentar berita di media sosial Instagram.

1.4 Batasan Penelitian

Batasan penelitian ini dibuat untuk memfokuskan inti dari permasalahan yang telah dibuat, supaya tidak melenceng/meluas dari inti permasalahan diluar dari penelitian ini. Batasan masalahnya yaitu sebagai berikut :

1. Menggunakan bahasa pemrograman python dan dieksekusi menggunakan Google Collab.
2. Data yang digunakan dalam penelitian ini yaitu berupa data hasil scraping pada jejaring sosial Instagram.
3. Pengklasifikasian yang digunakan untuk analisis sentimen hanya positif, negatif, netral.

1.5 Manfaat Penelitian

Manfaat penelitian yang ingin di capai dalam penelitian ini adalah :

1. Bagi penulis sendiri dapat mengetahui proses analisis sentimen menggunakan model IndoBERT, IndoBERT dengan CNN, dan IndoBERT dengan RNN.
2. Hasil penelitian ini diharapkan dapat di manfaatkan sebagai referensi dan bahan acuan serta perbandingan dalam melakukan analisis sentimen untuk digunakan dalam penelitian-penelitian selanjutnya.

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai acuan bagi peneliti dalam meneliti dan menghindari adanya kesamaan hasil dari penelitian. Selain itu, penelitian terdahulu juga digunakan sebagai sumber inspirasi bagi peneliti lainnya dalam menghasilkan penemuan yang baru, atau diperbarui dengan berbagai metode dan cara ilmiah lainnya.

Table 2. 1 Penelitian Terdahulu

No	Peneliti	Judul	Hasil penelitian
1	(Viana, 2024)	Klasifikasi emosi opini Twitter menggunakan model BenchMark IndoNLU	Berdasarkan hasil pengujian dari implementasi model BenchMark IndoNLU dalam melakukan klasifikasi emosi, maka diperoleh kesimpulan sebagai berikut: Klasifikasi emosi menggunakan model IndoBERT base p1 menghasilkan akurasi sebesar 71.47%. IndoBERT base p2 memperoleh hasil akurasi 71.19%. IndoBERT large p1 memperoleh akurasi sebesar 73.59%, dan IndoBERT large p2 memperoleh nilai akurasi sebesar 72.60%.
2	(Aulia & Amelia, 2020)	Analisis sentimen Twitter pada isu Mental Health dengan Algoritma klasifikasi Naïve Bayes	Hasil klasifikasi yang diperoleh, 50.000 Tweets tidak diketahui jenis emosinya, sentimen positif berada dengan perolehan data lebih dari 40.000 Tweets, sedangkan negatif berada pada angka lebih dari 25.000 Tweets, dan netral lebih dari 10.000 Tweets.
3	(Lambang et al., 2024)	Analisis sentimen masyarakat media sosial Twitter terhadap kinerja Penjabat Gubernur DKI Jakarta menggunakan model IndoBERT	Hasil dari penelitian ini didapati model yang siap digunakan untuk melakukan sentimen analisis, model ini memperoleh akurasi sebesar 90.5% pada data uji dan

No	Peneliti	Judul	Hasil penelitian
			96.63% pada data latih. Peneliti juga menampilkan metrik evaluasi model lainnya yaitu presisi, recall dan skor f1 yang masing-masing mendapatkan nilai sebesar 96.64%, 96.63% dan 96.62% pada data latih. Kemudian pada data uji didapat 90.6%, 90.5% dan 90.49%.
4	(Merdiansah & Ali Ridha, 2024)	Analisis sentimen pengguna X Indonesia terkait kendaraan listrik menggunakan IndoBERT.	Dalam penelitian ini menunjukkan model yang tidak di latih menggunakan data IndoNLU cenderung memberikan label positif, sementara model yang dilatih menggunakan data IndoNLU lebih sering memberikan label netral pada tweet. Hasil penelitian menunjukkan bahwa model yang dilatih dengan data IndoNLU memberikan kinerja yang lebih baik dalam memprediksi sentimen pada teks tweet, model yang dilatih dengan data IndoNLU menunjukkan kemampuan yang lebih baik dalam memahami konteks teks dengan tingkat kepercayaan yang lebih tinggi yaitu 98%.
5	(Khairani et al., 2024)	Pengaruh Tahapan Preprocessing Terhadap Model IndoBERT dan IndoBEERTweet untuk Mendeteksi Emosi pada Komentar Akun Berita Instagram	Berdasarkan hasil pengujian, dataset yang melalui tahapan <i>remove stopwords</i> dan <i>stemming</i> menghasilkan akurasi sebesar 88,28% untuk model IndoBERTtweet dan 85,35% untuk model IndoBERT. sedangkan dataset yang tidak melalui kedua tahapan tersebut menghasilkan akurasi sebesar 92,54% untuk model

No	Peneliti	Judul	Hasil penelitian
			IndoBERTweet dan 88,51% untuk model IndoBERT.

2.2 Instagram

Instagram adalah aplikasi berbagi foto dan video yang memungkinkan pengguna untuk membagikan momen-momen penting dalam kehidupan mereka. Pengguna dapat membagikan foto dan video dengan menggunakan berbagai filter dan efek yang tersedia di dalam aplikasi tersebut. Instagram juga memungkinkan bagi pengguna untuk dapat berinteraksi dengan pengguna lain melalui komentar, like, dan direct message.

Instagram sendiri merupakan salah satu media sosial yang paling populer saat ini. Aplikasi ini diluncurkan pada tahun 2010, dan saat ini menjadi platform yang sangat penting bagi masyarakat modern. Berdasarkan data yang dirilis oleh Meta, yaitu perusahaan induk Instagram pada tahun 2023, Instagram memiliki jumlah pengguna lebih dari 2 miliar pengguna aktif. Platform media sosial ini dapat digunakan sebagai platform yang efektif dalam berbagai bidang, seperti pemasaran, komunikasi, dan edukasi.

2.3 Analisis Sentimen

Analisis sentimen merupakan salah satu bidang penelitian yang mempelajari hubungan antara informasi emosional dan bahasa tertulis, serta terkait dengan pemrosesan komputasionalnya. Analisis sentimen berfokus pada teks, yang merupakan media penting untuk mengekstraksi karena user interface masih berbasis teks. Analisis sentimen berasal dari penambangan teks dan komputasi linguistik, dan lebih memperhatikan suasana hati, emosi, sikap, opini, dan evaluasi yang dikandungnya. Sentimen sering dipahami sebagai positif, negatif, dan netral, tetapi kita juga dapat memperhitungkan emosi yang jauh lebih mendasar seperti kegembiraan, kesedihan, kebencian, kegembiraan, dan ketakutan. Analisis sentimen terkadang disebut opinion mining, ekstraksi sentimen, atau deteksi sentimen (Hauthal et al., 2020). Tujuan analisis sentimen adalah untuk mengidentifikasi alat otomatis yang mampu mengekstraksi informasi subjektif dari teks bahasa alami, seperti opini dan emosi, untuk menghasilkan pengetahuan yang terstruktur dan dapat ditindak lanjuti

untuk sistem pendukung (pendukung keputusan atau pembuat keputusan pengguna). (Pozzi et al., 2017).

2.4 Bidirectional Encoder Representation From Transformers (BERT)

Bidirectional Encoder Representaion From Transformers (BERT) adalah sebuah model pembelajaran mesin yang dikembangkan oleh google yang dirancang untuk memahami konteks kata dalam teks dengan cara yang lebih mendalam. Model ini menggunakan arsitektur Transformer, yang mengandalkan mekanisme perhatian untuk memproses informasi dalam teks. BERT membaca teks secara bidirectional. Ini berarti BERT mempertimbangkan konteks kata dari kedua arah, yaitu sebelum dan sesudah kata yang sedang dianalisis, untuk memahami makna secara lebih komprehensif.

Pendekatan bidirectional ini memungkinkan BERT untuk menangkap informasi konteks yang lebih luas dan mendalam dari pada model-model sebelumnya yang hanya berbasis satu arah. Model ini dilatih dalam dua tahap: Pre-training dan fine-tuning. Pada tahap pre-training, BERT dilatih untuk memprediksi kata yang hilang dalam kalimat dan untuk memahami hubungan antar kalimat. Pada tahap fine-tuning, BERT disesuaikan dengan tugas spesifik, seperti klasifikasi teks atau analisis sentimen, dengan menggunakan data yang relevan untuk tugas tersebut. (Devlin et al., 2018).

BERT bergantung pada transformer, yaitu sebuah mekanisme yang digunakan untuk memahami konteks kalimat dengan mempelajari hubungan kontekstual antar kata dalam teks. Transformer memiliki kemampuan untuk mempelajari dan menyesuaikan pemahaman melalui mekanisme *self-attention*. *Self-attention* merupakan metode yang digunakan oleh Transformer untuk memodifikasi kata-kata terkait dalam konteks tertentu. Secara keseluruhan, Transformer terdiri dari dua komponen utama, yaitu encoder dan decoder.

2.4.1 Encoder

Mekanisme ini digunakan untuk memproses data input berupa teks, yang terdiri dari tumpukan $N = 6$ lapisan identik, dimana setiap lapisan memiliki dua sublapisan:

self-attention dan jaringan saraf *feed-forward*. Mekanisme encoder dalam lapisan *self-attention* memungkinkan node untuk tidak hanya terfokus pada kata yang sedang diproses, tetapi juga untuk memahami konteks yang lebih luas dari kata tersebut. Selain itu, lapisan *self-attention* ini juga dapat memproses semua posisi kata dari lapisan-lapisan sebelumnya dalam encoder.

2.4.2 Decoder

Mekanisme ini digunakan untuk menghasilkan urutan keluaran yang diprediksi, terdiri dari tumpukan $N = 6$ lapisan yang memiliki dua sublapisan yang sama seperti pada lapisan encoder. Selain itu, terdapat lapisan *attention* tambahan di antara keduanya, yang memungkinkan node untuk mengakses informasi utama dengan menggunakan *multi-head attention* pada output dari encoder. Sama seperti pada encoder, lapisan *self-attention* di dalamnya memungkinkan untuk memproses semua posisi dari lapisan sebelumnya dan saat ini di dalam decoder.

2.5 IndoBERT

IndoBERT adalah model pemrosesan bahasa alami yang khusus dikembangkan untuk bahasa Indonesia, didasarkan pada arsitektur BERT. Model ini dirancang untuk mengatasi berbagai tantangan yang dihadapi dalam pemrosesan bahasa Indonesia, seperti perbedaan struktur kalimat, morfologi, dan variasi dialek. IndoBERT dilatih menggunakan data teks dalam bahasa Indonesia yang besar dan beragam, sehingga mampu menangkap nuansa dan konteks bahasa secara lebih akurat.

IndoBERT terdiri dari empat varian model utama, yaitu IndoBERT-Base, IndoBERT-Large, IndoBERT-liteBase, dan IndoBERT-liteLarge. IndoBERT-Base adalah versi standar dengan 12 lapisan encoder, 768 dimensi, dan 12 attention heads. Model ini dirancang untuk menangani berbagai tugas pemrosesan bahasa alami dalam bahasa Indonesia dengan kinerja yang efisien. IndoBERT-Large adalah versi yang lebih besar dengan 24 lapisan encoder, 1024 dimensi, dan 16 attention heads. Varian ini menawarkan kemampuan yang lebih kuat dalam memahami konteks yang lebih kompleks, meskipun memerlukan lebih banyak sumber daya komputasi. Selanjutnya IndoBERT-liteBase merupakan versi ringan dari IndoBERT-Base yang dirancang

untuk aplikasi yang memerlukan kecepatan dan efisiensi, terutama pada perangkat dengan keterbatasan sumber daya. Meskipun ukurannya lebih kecil, model ini tetap memberikan kinerja yang baik untuk tugas-tugas pemrosesan bahasa alami. Sedangkan IndoBERT-liteLarge adalah versi ringan dari IndoBERT-Large, yang mempertahankan banyak keunggulan dari model besar tersebut tetapi dalam ukuran yang lebih kecil dan lebih efisien, sehingga cocok untuk digunakan pada aplikasi dengan kebutuhan komputasi yang lebih rendah namun tetap memerlukan pemahaman konteks yang lebih baik.

Penelitian untuk model ini pernah dilakukan sebelumnya seperti yang dilakukan oleh (Viana, 2024), dengan judul Klasifikasi emosi opini Twitter menggunakan model BenchMark IndoNLU. Menurut penelitian yang dilakukan terhadap model IndoBERT tersebut, model mendapatkan total akurasi sebesar 71.47%. Sudah menunjukkan kemampuan yang cukup andal dalam menganalisis teks berbahasa Indonesia.

2.6 Python

Python adalah bahasa pemrograman yang populer dan mudah digunakan, terutama bagi pemula. Dengan sintaks yang sederhana dan mudah dibaca, Python sering digunakan dalam berbagai aplikasi, mulai dari pengembangan web hingga analisis data. Kelebihan Python antara lain mudah dipelajari, memiliki komunitas yang kuat, banyak paket (library) yang dapat digunakan, dan cross-platform, sehingga dapat berjalan pada berbagai sistem operasi seperti Windows, macOS, dan Linux.

Python dapat digunakan untuk mengolah dan menganalisis data dengan efektif. Berbagai library seperti Pandas dan NumPy memungkinkan pengguna untuk dengan mudah mengolah data. Python juga dilengkapi dengan library machine learning seperti Scikit-learn dan TensorFlow, yang dapat digunakan untuk menganalisis data menggunakan algoritma klasifikasi, regresi, dan clustering. Dengan menggunakan Python dapat mengotomasi tugas-tugas yang berulang seperti mengimpor data dari berbagai sumber, melakukan perhitungan, dan menghasilkan laporan, sehingga memungkinkan untuk fokus pada analisis dan interpretasi data.

2.7 Google Colaboratory

Google Colaboratory, atau sering disebut sebagai Google Colab, adalah sebuah platform berbasis cloud yang memungkinkan pengguna menjalankan kode python langsung dari browser tanpa perlu melakukan instalasi perangkat lunak tambahan. Google Colab menyediakan lingkungan komputasi yang mendukung pemrograman interaktif, terutama dalam analisis data, pembelajaran mesin, dan kecerdasan buatan.

Penggunaan Google Colab dalam analisis data sangatlah fleksibel dan praktis. Dengan kemampuannya yang mendukung integrasi dengan berbagai pustaka python seperti Pandas, NumPy, Matplotlib, dan Scikit-learn, Google Colab memudahkan peneliti dalam melakukan pengolahan data, visualisasi, hingga penerapan model pembelajaran mesin. Oleh karena itu, Google Colab menjadi pilihan populer bagi banyak akademisi, peneliti, dan praktisi data yang membutuhkan solusi komputasi yang efisien dan mudah diakses.

Berikut beberapa pustaka python yang biasa digunakan dalam analisis data:

2.7.1 Numpy (Numerical Python)

NumPy adalah pustaka python yang sangat penting dalam analisis data karena menyediakan dukungan untuk operasi matematika dan numerik yang efisien pada array multidimensi. Dengan NumPy, pengguna dapat dengan cepat melakukan perhitungan matematis kompleks, transformasi data, dan operasi aljabar linear. Ini sangat berguna untuk analisis data yang melibatkan perhitungan matematis intensif dan manipulasi array besar.

2.7.2 Pandas

Pandas adalah pustaka python yang menyederhanakan manipulasi dan analisis data tabular, seperti data dalam format spreadsheet atau tabel database. Dengan Pandas, pengguna dapat dengan mudah melakukan operasi seperti pembersihan data, penggabungan, dan agregasi. Struktur data utama seperti DataFrame memungkinkan

analisis data yang terorganisir dengan baik, serta memudahkan proses eksplorasi dan visualisasi data.

2.7.3 Scikit-Learn

Scikit-learn adalah pustaka python yang menyediakan alat untuk pembelajaran mesin dan analisis data statistik. Dengan berbagai algoritma untuk klasifikasi, regresi, clustering, dan pengurangan dimensi, scikit-learn memungkinkan pengguna untuk membangun, melatih, dan mengevaluasi model pembelajaran mesin secara efisien. Ini sangat berguna dalam analisis data untuk mengidentifikasi pola, membuat prediksi, dan mengimplementasikan model analitik.

2.7.4 TextBlob

TextBlob adalah pustaka python yang digunakan untuk analisis teks dan pemrosesan bahasa alami (NLP). TextBlob mempermudah ekstraksi informasi dari teks, seperti analisis sentimen, tokenisasi, dan pengenalan entitas. Dalam konteks analisis data, TextBlob memungkinkan pengguna untuk mengolah dan menganalisis data berbasis teks dengan lebih efektif, mengidentifikasi tren, dan mendapatkan wawasan dari data yang tidak terstruktur.

2.8 Confusion Matrix

Confusion matrix adalah alat evaluasi yang sangat berguna dalam mengukur kinerja model klasifikasi dengan cara membandingkan prediksi model dengan hasil yang sebenarnya terjadi. Matriks ini menyajikan data dalam bentuk tabel yang memetakan hasil prediksi ke dalam beberapa kategori berdasarkan kebenaran atau kesalahan klasifikasi yang dilakukan oleh model. Dengan menampilkan jumlah prediksi yang benar dan salah secara terperinci untuk setiap kelas, confusion matrix memungkinkan kita untuk melihat dengan jelas seberapa efektif model tersebut dalam mengklasifikasikan data ke dalam kategori yang tepat. Matriks ini terdiri dari 4(empat) komponen utama, yaitu:

- 1) *True Positives* (TP), keadaan dimana model dengan benar mengidentifikasi data positif sebagai positif.

- 2) *True Negatives* (TN), keadaan dimana model dengan benar mengidentifikasi data negatif sebagai negatif.
- 3) *False Positives* (FP), keadaan dimana model salah mengklasifikasikan data negatif sebagai positif.
- 4) *False Negatives* (FN), keadaan dimana model salah mengklasifikasikan data positif sebagai negatif.

Table 2. 2 Confusion Matrix

		<i>Actual Values</i>	
		Positive	Negative
<i>Predicted values</i>	Positive	<i>TP</i>	<i>FP</i>
	Negative	<i>FN</i>	<i>TN</i>

Dari *confusion matrix* evaluasi model lainnya bisa menghitung akurasi, presisi, recall, dan f1-score untuk setiap kelas. Metrik ini membantu menilai performa model dalam analisis sentimen.

1. Akurasi, mengukur persentase prediksi yang benar dari total prediksi yang dibuat oleh model. Ini memberikan gambaran umum tentang seberapa sering model membuat prediksi yang benar.

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

2. Presisi, mengukur ketepatan model dalam memprediksi kelas positif, yaitu proporsi prediksi positif yang benar. Ini menunjukkan seberapa tepat model dalam memprediksi kelas positif.

$$precision = \frac{TP}{(TP + FP)}$$

3. Recall, adalah rasio data positif yang berhasil diidentifikasi dengan benar oleh model dibandingkan dengan total data positif yang ada. Ini mengukur kemampuan model untuk mendeteksi semua instance positif.

$$recall = \frac{TP}{(TP + FN)}$$

4. F1-score, adalah metrik yang menyediakan ukuran yang seimbang antara presisi dan recall, sehingga cocok untuk data yang tidak seimbang.

$$f1\ score = \frac{2(precision \times recall)}{(precision + recall)}$$

2.9 Data Scraping

Data scraping adalah proses otomatis untuk mengekstraksi informasi dari situs web secara terstruktur. Metode ini melibatkan penggunaan skrip atau alat khusus yang mengakses halaman web dan mengambil data tertentu yang sesuai dengan kriteria yang telah ditentukan. Data yang diperoleh melalui scraping biasanya disimpan dalam format yang lebih mudah dianalisis, seperti CSV, JSON, atau database, yang kemudian dapat digunakan untuk berbagai tujuan, termasuk analisis data, riset pasar, atau pemantauan kompetitor.

Dalam analisis data, data scraping sering digunakan untuk mengumpulkan informasi dari berbagai sumber online, termasuk harga produk, ulasan pelanggan, dan konten berita. Teknik ini memungkinkan pengumpulan data secara lebih efisien dibandingkan metode manual, dan dapat dilakukan secara berkala untuk memastikan data selalu terbaru. Selain itu, data scraping memungkinkan pengguna untuk mengakses informasi yang mungkin tidak tersedia melalui API atau layanan lainnya, sehingga memberikan fleksibilitas yang lebih besar dalam memperoleh data.

Menurut penelitian yang diterbitkan dalam *Journal of Web Engineering*, data scraping telah menjadi teknik yang sangat penting dalam pengumpulan data berbasis web, terutama dalam konteks big data dan analisis prediktif. Penelitian tersebut menunjukkan bahwa data scraping memungkinkan pengumpulan informasi dalam jumlah besar dengan cepat, yang kemudian dapat digunakan untuk mendukung pengambilan keputusan dan strategi bisnis berbasis data.

2.10 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN), merupakan jenis jaringan saraf tiruan yang sangat efektif untuk tugas-tugas yang melibatkan data visual, seperti pengenalan gambar, pengenalan objek, dan analisis video. Meskipun awalnya dirancang untuk pengolahan data visual, telah terbukti efektif juga dalam analisis sentimen teks. Dalam penerapannya, CNN memanfaatkan representasi teks sebagai matriks numerik, di mana setiap kata diubah menjadi vektor melalui teknik embedding. Setelah teks dikonversi menjadi matriks, CNN menerapkan filter yang bergerak melintasi sekuens kata untuk menangkap pola-pola lokal yang signifikan, seperti frasa yang menunjukkan sentimen tertentu. Hasil dari lapisan convolutional ini kemudian diperkecil melalui pooling, yang mengekstrak fitur paling menonjol dari teks. Setelah itu, fitur-fitur ini diteruskan ke lapisan fully connected untuk klasifikasi akhir, misalnya untuk menentukan apakah teks tersebut bernada positif, negatif, atau netral. Dengan pendekatan ini, CNN mampu menangkap nuansa dan pola dalam teks yang mengindikasikan sentimen, menjadikannya alat yang kuat dalam analisis sentimen.

Menurut penelitian yang berjudul “Deteksi Aspek Review E-Commerce Menggunakan IndoBERT Embedding dan CNN” oleh (Imron, S., Setiawan, E. I., & Santoso, J. 2023) Penggabungan Model IndoBERT dengan Model CNN telah diterapkan untuk meningkatkan kemampuan model dalam mendeteksi aspek pada ulasan e-commerce. Kombinasi ini mencapai akurasi sebesar 94,86%, menunjukkan peningkatan performa dalam tugas deteksi aspek.

2.11 **Model Hybrid**

Model Hybrid dalam analisis sentimen merujuk pada pendekatan yang menggabungkan berbagai teknik atau algoritma untuk meningkatkan akurasi dan kinerja prediksi sentimen. Pendekatan ini seringkali memanfaatkan kekuatan dari model yang berbeda, seperti menggabungkan metode berbasis aturan dengan model machine learning atau mengintegrasikan model deep learning seperti CNN. Dengan cara ini, model hybrid dapat menangkap berbagai aspek dari data teks, baik itu pola lokal yang mendalam maupun hubungan jangka panjang antar kata.

Model hybrid yang menggabungkan CNN dan IndoBERT dalam analisis sentimen merupakan pendekatan yang memanfaatkan kekuatan masing-masing model untuk menghasilkan prediksi sentimen yang lebih akurat. IndoBERT, dengan kemampuannya memahami konteks dan makna kata dalam teks berbahasa Indonesia, dapat menangkap nuansa yang mendalam dalam kata teks. Di sisi lain, CNN unggul dalam mendeteksi pola-pola lokal, seperti frasa penting yang mungkin menunjukkan sentimen tertentu. Dalam model hybrid ini, IndoBERT pertama-tama digunakan untuk menghasilkan representasi vektor dari teks, yang kemudian diproses oleh CNN untuk mengenali pola-pola yang relevan. Kombinasi ini memungkinkan model untuk menangani berbagai aspek dari analisis sentimen, mulai dari pemahaman konteks yang kompleks hingga identifikasi pola lokal, sehingga memberikan hasil yang lebih robust dan akurat dibandingkan dengan menggunakan satu model saja.

2.12 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah salah satu jenis arsitektur jaringan saraf yang sering digunakan dalam analisis sentimen, terutama karena kemampuannya memproses data berurutan seperti teks. RNN dirancang untuk menangkap hubungan temporal dalam data dengan memanfaatkan mekanisme *feedback loop*, di mana output dari langkah sebelumnya digunakan sebagai input tambahan untuk langkah berikutnya. Dalam konteks analisis sentimen, RNN memungkinkan model untuk memahami konteks kata-kata dalam kalimat, yang sangat penting karena sentimen sering kali bergantung pada urutan dan hubungan antar kata. Misalnya, frasa seperti "tidak buruk" memiliki arti yang berbeda dibandingkan "tidak baik," meskipun kata-kata penyusunnya serupa. Dengan mengolah teks sebagai rangkaian elemen berurutan, RNN dapat mengenali pola dan konteks yang lebih kompleks dibandingkan metode tradisional. Namun, RNN juga memiliki keterbatasan, seperti kesulitan menangani urutan panjang karena masalah *vanishing gradient*. Untuk mengatasi kendala ini, varian RNN seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) sering digunakan karena mampu mempertahankan informasi penting lebih lama, sehingga lebih efektif untuk menganalisis teks yang kompleks dalam tugas sentimen.

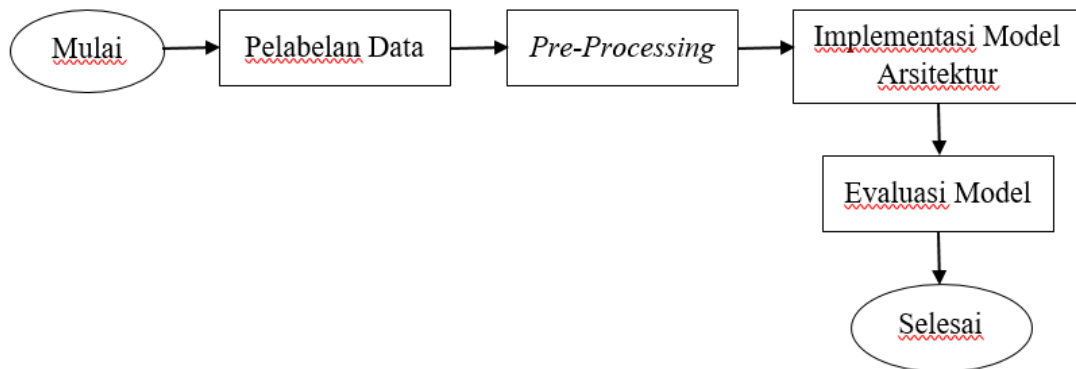
Penelitian serupa pernah dilakukan dengan judul “Hybrid model: IndoBERT and long short-term memory for detecting Indonesian hoax news” oleh Yefferson, D. Y., Lawijaya, V., & Girsang, A. S. (2024). Yang dimana penelitian tersebut memakai model IndoBERT dan RNN untuk mendeteksi hoax dari informasi yang tersebar di media sosial. Hasil penelitian menunjukkan bahwa model tersebut mendapatkan hasil mencapai 93,2%.



BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Alur penelitian dapat dilihat pada gambar 3.1 di bawah ini. Alur penelitian digunakan sebagai prosedur yang menjelaskan proses berjalannya sebuah penelitian.



Gambar 3. 1 Metode Penelitian

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini menggunakan data yang berasal dari sosial media Instagram. Data yang digunakan adalah komentar yang berasal dari akun berita nasional dengan topik Judi Online. Data diambil menggunakan teknik *scraping data* dengan menggunakan bahasa pemrograman *Python* dengan library *instaloader* yang biasa digunakan untuk mengambil data dari *Instagram*. Dataset yang terkumpul disimpan ke dalam format CSV, dengan total data komentar sebanyak 1.811 data.

3.3 Pelabelan Data

Pelabelan data adalah proses memberi tanda atau kategori tertentu pada data mentah untuk tujuan analisis lebih lanjut. Dalam konteks pembelajaran mesin, pelabelan data biasanya dilakukan dengan memberikan label pada data sehingga model dapat dilatih untuk mengenali pola dan membuat prediksi. Pelabelan data dalam penelitian ini menggunakan label positif, negatif dan netral. Pelabelan data dilakukan secara manual. Contoh kalimat yang sudah di lebel dapat dilihat pada tabel di bawah ini.

Table 3. 1 Contoh Pelabelan Data

No	Contoh data	Label
1	Emang negara ini sudah harus direformasi total dari atas sampe kebawah...	<i>Positive</i>
2	Biar ada efek jera, diumumkan saja nama namanya	<i>Positive</i>
3	Tuh para wakil rakyat pilihanmu	<i>Neutral</i>
4	Asik dapet bansos	<i>Neutral</i>
5	Maling mana yang mau ngaku	<i>Negative</i>

3.4 Pre-processing

Pre-processing adalah langkah penting dalam analisis data yang bertujuan untuk mempersiapkan data mentah agar siap digunakan dalam analisis atau pemodelan. *Pre-processing* memastikan bahwa data yang dianalisis bebas dari kesalahan dan bias, sehingga meningkatkan akurasi dan keandalan hasil analisis atau model pembelajaran mesin. Tahap *Pre-processing* yang dapat dilakukan yaitu *case folding*, *tokenization*, dan penghapusan *stopword*.

1. Case folding

Case folding adalah proses dalam pemrosesan teks yang bertujuan untuk mengubah semua huruf dalam teks menjadi huruf kecil (lowercase). Langkah ini dilakukan untuk memastikan bahwa teks diperlakukan secara konsisten dalam analisis, tanpa membedakan antara huruf besar dan huruf kecil. Teknik ini digunakan dalam analisis teks atau pemrosesan bahasa alami untuk mengurangi variasi yang tidak relevan dalam data teks.

2. Filtering

filtering adalah langkah untuk menyaring data agar hanya informasi yang relevan digunakan dalam analisis. Proses ini mencakup penghilangan bagian-bagian yang tidak diperlukan, seperti data yang kosong, teks yang tidak berkaitan, atau elemen pengganggu seperti simbol dan angka yang tidak mendukung tujuan penelitian.

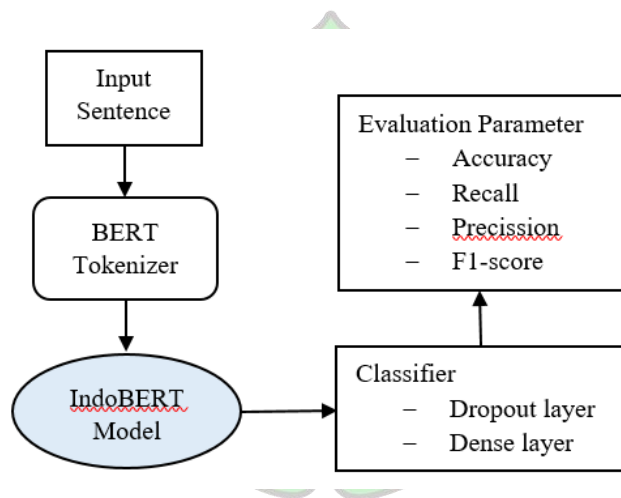
3. Normalisasi Teks

Normalisasi teks adalah tahap penting dalam prapemrosesan data teks yang bertujuan untuk menyederhanakan dan menyeragamkan format teks agar lebih konsisten serta mudah diolah oleh model pemrosesan bahasa alami (NLP). Proses ini mencakup berbagai langkah, seperti mengubah semua huruf menjadi

huruf kecil (case folding) untuk menghindari perbedaan karena kapitalisasi, serta menghapus karakter khusus atau tanda baca yang tidak relevan. Selain itu, normalisasi juga melibatkan penyederhanaan pengulangan karakter, misalnya mengubah "haaappyy" menjadi "happy", dan mengganti ekspresi informal atau singkatan seperti "gak" menjadi "tidak". Dalam beberapa kasus, dilakukan juga stemming atau lemmatisasi untuk mengembalikan kata ke bentuk dasarnya, seperti mengubah "berlari" menjadi "lari" atau "makanlah" menjadi "makan". Dengan normalisasi ini, teks menjadi lebih seragam dan siap untuk dianalisis

3.5 Implementasi Model Arsitektur

3.5.1 IndoBERT



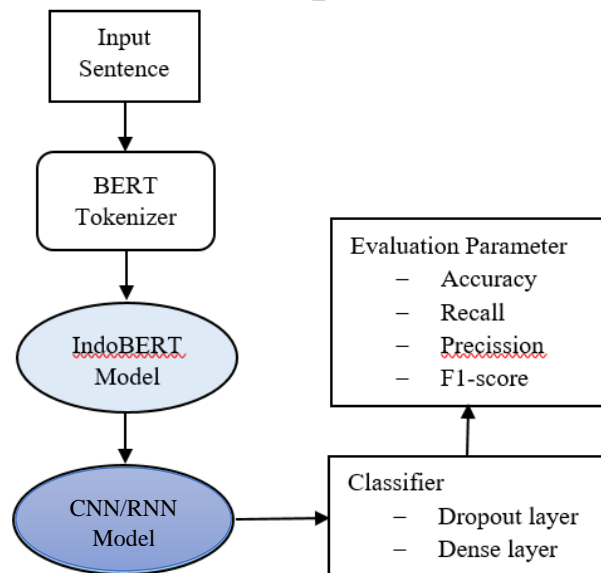
Gambar 3.2 IndoBERT Model

Implementasi model IndoBERT melibatkan penerapan dan *fine-tuning* model BERT yang dirancang khusus untuk bahasa Indonesia dalam tugas-tugas NLP seperti klasifikasi teks dan analisis sentimen. Proses ini mencakup penyesuaian model untuk dataset spesifik dan integrasinya ke dalam aplikasi yang memerlukan pemahaman mendalam terhadap bahasa Indonesia, sehingga meningkatkan akurasi dan efektivitas analisis teks.

Dataset yang digunakan dalam penelitian ini merupakan komentar pengguna Instagram yang dihasilkan dengan melakukan scraping data pada Instagram. Selanjutnya, dilakukan tahap pre-processing sebagai tahap pembersihan teks pada data lalu diberi label pada setiap teks.

Selanjutnya dilakukan *splitting* atau pemisahan data dengan perbandingan 70% untuk data latih, 20% untuk data valid, dan 10% untuk data testing. Data latih merupakan dataset yang digunakan untuk melatih sistem model dengan tujuan agar model dapat mengenali pola dari data, data testing digunakan untuk mengukur performa akhir model setelah proses pelatihan selesai.

3.5.2 Model Hybrid



Gambar 3. 3 Model Hybrid IndoBERT dengan CNN/RNN

Implementasi model yang menggabungkan IndoBERT dan CNN untuk analisis sentimen melibatkan integrasi kekuatan kedua model tersebut untuk menghasilkan prediksi yang lebih akurat. Proses dimulai dengan Indobert, yang digunakan untuk memproses teks berbahasa Indonesia dan menghasilkan representasi vektor yang kaya akan konteks. Vektor ini kemudian diteruskan ke CNN, yang memiliki kemampuan untuk mendeteksi pola-pola lokal dalam teks, seperti frasa penting yang mungkin menandakan sentimen tertentu. Setelah CNN mengekstrak fitur-fitur penting dari representasi tersebut, hasilnya digunakan untuk melakukan klasifikasi sentimen,

misalnya menentukan apakah teks memiliki sentimen positif, negatif, atau netral. Dengan menggabungkan IndoBERT yang memahami konteks dan CNN yang fokus pada pola-pola spesifik, model ini mampu memberikan hasil yang lebih kuat dan akurat dalam berbagai situasi analisis sentimen. Hal yang sama juga dilakukan untuk model IndoBERT dengan RNN.

3.6 Evaluasi Model

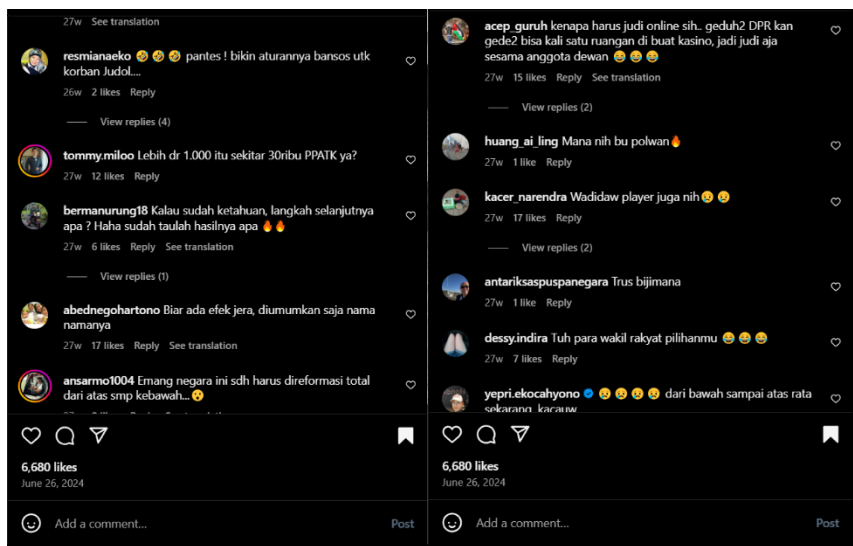
Evaluasi model dilakukan untuk menilai kinerja suatu model pembelajaran mesin setelah dilatih dengan data. Evaluasi ini dilakukan dengan menggunakan metrik tertentu, seperti akurasi, presisi, recall, atau F1-score, untuk mengukur seberapa baik model tersebut memprediksi atau mengklasifikasikan data baru. Evaluasi model bertujuan untuk memastikan bahwa model tidak hanya bekerja dengan baik pada data latih, tetapi juga mampu generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.



BAB IV HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

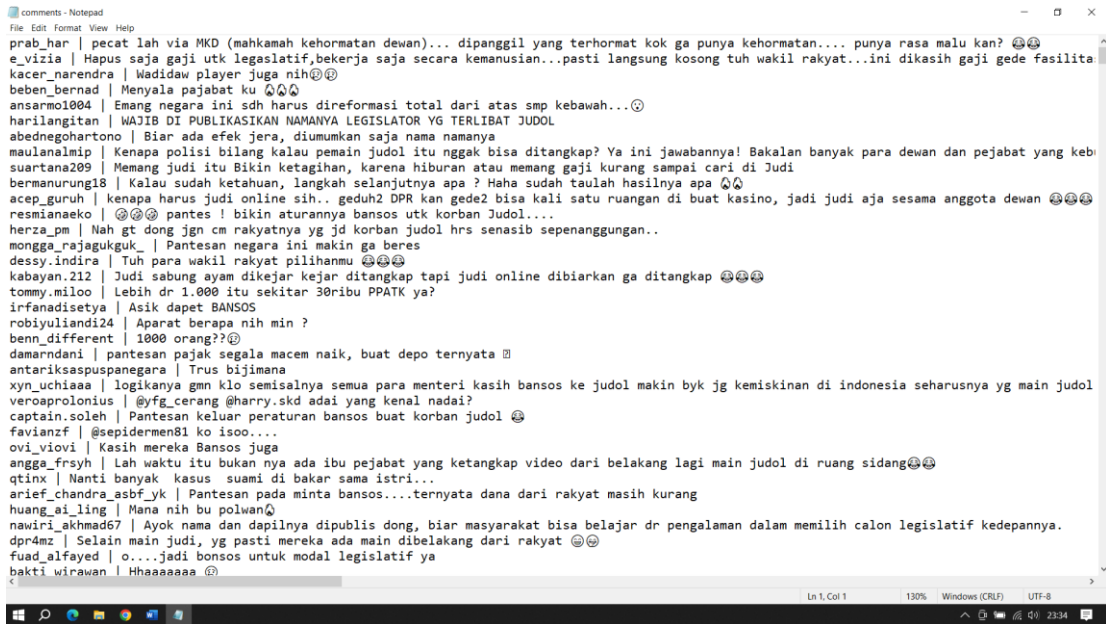
Dataset yang digunakan pada penelitian ini bersumber dari komentar beberapa postingan di aplikasi media sosial Instagram. Jumlah data komentar yang diambil sebanyak 1.811 komentar yang disimpan ke dalam format file csv. Berikut contoh salah satu komentar pada postingan tersebut seperti pada gambar 4.1.



Gambar 4. 1 Komentar Netizen Terhadap Judi Online

Proses pengumpulan data dilakukan dengan scraping menggunakan library Instaloader dengan atribut username dan komentarnya. Pengambilan data komentar menggunakan perintah `comment_gen = post.get_comments()`, perintah ini untuk mengambil data komentar yang tersedia pada postingan tersebut. Kemudian dataset yang terkumpul disimpan ke dalam format file csv untuk mempermudah proses pengolahan data.

Untuk mempermudah dan memaksimalkan penggunaan data, maka peneliti mengurangi atribut dengan menghapus atribut username karena tidak relevan digunakan dalam analisis sentiment. Berikut contoh hasil pengumpulan data komentar yang ditampilkan pada gambar 4. 2 dibawah ini.



Gambar 4. 2 Hasil Pengumpulan Data

4.2 Pelabelan Data

Pelabelan Data dalam penelitian ini menggunakan tiga jenis label yaitu *positive*, *neutral*, dan *negative*. Data yang diperoleh dari hasil pengumpulan data komentar postingan Instagram ini kemudian dilabel secara manual seluruh teks dataset nya. Jumlah seluruh data yang digunakan pada penelitian ini adalah sebanyak 1.811 dataset.

4.3 Pre-Processing Data

Selanjutnya masuk ke tahap *pre-processing* data, tahap ini dilakukan untuk menjadikan data lebih efektif dan dapat dibaca dengan baik oleh model saat pemrosesan nantinya. Data komentar mentah tersebut masih terdapat beberapa hal seperti tanda baca, angka, emoji, dan lainnya yang tidak memuat opini sama sekali. Maka data harus melewati tahap ini terlebih dahulu agar pelatihan model nantinya menjadi lebih efisien dan akurat. Dalam tahapan ini melewati beberapa proses tahapan. Antara lain yaitu, *case folding*, normalisasi teks dan *filtering*.

4.3.1 Case Folding

Case folding adalah salah satu langkah dalam proses prapemrosesan teks yang bertujuan untuk menyeragamkan bentuk huruf dalam data teks. Proses ini mengubah semua huruf menjadi huruf kecil (lowercase), tanpa memedulikan apakah aslinya

berupa huruf kapital atau tidak. Misalnya, kata "AI", "Ai", dan "ai" akan diubah menjadi bentuk seragam, yaitu "ai". Langkah ini penting untuk menghilangkan perbedaan yang disebabkan oleh kapitalisasi, sehingga teks dapat diproses secara konsisten.

4.3.2 Filtering

Filtering adalah proses menyaring teks untuk menghilangkan elemen yang dianggap tidak relevan atau mengganggu dalam analisis. Tahap ini sering mencakup penghapusan *stopwords*, yaitu kata-kata umum seperti "dan", "di", atau "yang" yang tidak memberikan informasi penting. Filtering juga dapat mencakup penghapusan angka, simbol, atau emoji jika tidak diperlukan, serta penyaringan berdasarkan panjang kata, misalnya menghapus kata yang terlalu pendek atau panjang tanpa arti. Selain itu, filtering dapat menggunakan kamus atau daftar tertentu untuk mempertahankan hanya kata-kata yang relevan.

4.3.3 Normalisasi Teks

Secara khusus, bagian ini bertujuan untuk menghilangkan karakter yang berulang secara berlebihan dan menyederhanakannya menjadi bentuk dasar, serta menghapus karakter khusus atau tanda baca yang tidak relevan. Selain itu, normalisasi juga melibatkan penyederhanaan ^Δ pengulangan karakter, misalnya mengubah "haaapyyy" menjadi "happy", dan mengganti ekspresi informal atau singkatan seperti "gak" menjadi "tidak".

```

def repeatcharClean(text):
    for i in range(len(character)):
        charac_long = 5
        while charac_long > 2:
            char = character[i]*charac_long
            text = text.replace(char, character[i])
            charac_long -= 1
    return text

```

Gambar 4.3 Kode Normalisasi Teks

```

def clean_review(text):
    # Ubah teks menjadi huruf kecil
    text = text.lower()
    # Ubah enter menjadi spasi
    text = re.sub(r'\n', ' ', text)
    # Hapus emoji
    text = emoji.demojize(text)
    text = re.sub(r':[a-zA-Z_]+:', ' ', text) # Hapus nama emoji yang diconvert
    # Hapus emoticon
    text = re.sub(r"([xX;]?[dDpPvVoO3\])", ' ', text)
    # Hapus link
    text = re.sub(r"https?:\/\/[^\s]+", "", text)
    # Hapus username
    text = re.sub(r"@[^\s]+\s?", ' ', text)
    # Hapus hashtag
    text = re.sub(r"#(\s+)", r'\1', text)
    # Hapus angka dan beberapa simbol
    text = re.sub(r'[a-zA-Z,?!]+', ' ', text)
    # Hapus karakter berulang
    text = repeatcharClean(text)
    # Bersihkan spasi berlebih
    text = re.sub(r'[ ]+', ' ', text)
    return text

```

Gambar 4.4 Kode Case Folding dan Filtering

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	review_text,label												
2	pecat lah via mkd mahkamah kehormatan dewan dipanggil yang terhormat kok ga punya kehormatan punya rasa malu kan, negative												
3	hapus saja gaji utk legaslatifbekerja saja secara kemanusiaanpasti langsung kosong tuh wakil rakyatini dikasih gaji gede fasilitas enakpensi												
4	wadidaw player juga nih, negative												
5	menyala pajabat ku, negative												
6	emang negara ini sdh harus direformasi total dari atas smp kebawah, positive												
7	wajib di publikasikan namanya legislator yg terlibat judul, neutral												
8	biar ada efek jera diumumkan saja nama namanya, positive												
9	kenapa polisi bilang kalau pemain judul itu nggak bisa ditangkap ya ini jawabannya bakalan banyak para dewan dan pejabat yang kebuka												
10	memang judi itu bikin ketagihan karena hiburan atau memang gaji kurang sampai cari di judi, positive												
11	kalau sudah ketahuan langkah selanjutnya apa haha sudah taulah hasilnya apa, negative												
12	kenapa harus judi online sih geduh dpr kan gede bisa kali satu ruangan di buat kasino jadi judi aja sesama anggota dewan, negative												
13	pantes bikin aturannya bansos utk korban judul, positive												
14	nah gt dong jgn cm rakyatnya yg jd korban judul hrs senasib sepenanggungan, negative												
15	pantesan negara ini makin ga beres, negative												
16	tuh para wakil rakyat pilihanmu, neutral												
17	judi sabung ayam dikejar kejar ditangkap tapi judi online dibiarkan ga ditangkap, negative												
18	lebih dr itu sekitar ribu ppatk ya, neutral												
19	asik dapet bansos, neutral												

Gambar 4. 5 Teks setelah melewati Pre-Processing

4.4 Split Data

Dataset yang sudah melewati tahap *pre-processing* akan dibagi menjadi tiga bagian, yaitu data training sebanyak 70% dari total data, data validasi sebanyak 20 % dari total data. Dan data test sebanyak 10% dari total data. Data pelatihan digunakan untuk melatih model, yaitu membantu model belajar pola atau hubungan dalam data. Selama pelatihan, model mengoptimalkan parameter berdasarkan data ini. Data validasi, di sisi lain, digunakan untuk mengevaluasi kinerja model secara berkala selama pelatihan, memastikan model tidak terlalu cocok (*overfitting*) pada data pelatihan. Set ini juga membantu dalam penyesuaian hiperparameter. Terakhir, data pengujian adalah data yang sepenuhnya baru untuk model, digunakan setelah pelatihan selesai untuk mengukur kemampuan model dalam membuat prediksi pada data yang belum pernah dilihat sebelumnya. Jumlah setiap data yang telah di bagi dapat dilihat pada tabel 4. 1 berikut.

Table 4. 1 jumlah Split Dataset

Dataset	Jumlah Dataset
Dataset Train	1209
Dataset Validation	347
Dataset Test	172

4.5 Perancangan Model IndoBERT

Hasil yang telah melewati tahap sebelumnya selanjutnya akan dimasukkan kedalam jaringan IndoBERT. Perancangan model yang akan digunakan yaitu menggunakan model IndoBERT base p1. IndoBERT Base P1 memiliki spesifikasi yang sama dengan BERT Base secara arsitektur, tetapi dilatih menggunakan data khusus dalam bahasa Indonesia. Kode untuk penerapan model dapat di lihat pada gambar 4.6 dibawah ini.

```
# Load Tokenizer and Config
tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-base-p1')
config = BertConfig.from_pretrained('indobenchmark/indobert-base-p1')
config.num_labels = DocumentSentimentDataset.NUM_LABELS

# Instantiate model
model = BertForSequenceClassification.from_pretrained('indobenchmark/indobert-base-p1', config=config)
```

Gambar 4. 6 Kode Penerapan IndoBERT base p1

Kode di atas digunakan untuk mengatur dan mempersiapkan model BERT untuk tugas klasifikasi sekuens, klasifikasi sekuens sendiri merupakan suatu tugas pembelajaran mesin di mana model diberikan sebuah urutan data (seperti teks, audio, atau waktu). Setiap baris kode diatas memiliki fungsi untuk menginisialisasi tokenizer untuk model IndoBERT base P1, memuat konfigurasi model IndoBERT base P1, menyesuaikan konfigurasi model untuk tugas klasifikasi sekuens dengan sejumlah label tertentu, dan membuat model BERT untuk Klasifikasi Sekuens berdasarkan model IndoBERT base P1.

4.6 Perancangan Model IndoBERT dengan CNN

Pada perancangan model hybrid yang menggabungkan antara *IndoBERT Base P2* dengan *Convolutional Neural Network* (CNN) dirancang untuk memanfaatkan kemampuan transformer dalam memahami konteks teks secara mendalam sekaligus mengintegrasikan kekuatan CNN dalam menangkap pola spasial lokal dalam data sekuensial. Dengan pendekatan ini, model dapat menghasilkan representasi teks yang kaya dan efisien untuk berbagai tugas klasifikasi teks berbahasa Indonesia. Penerapan model tersebut dapat dilihat pada gambar berikut ini.

```

# Pastikan Anda memiliki IndoBERT pretrained model
MODEL_NAME = "indobenchmark/indobert-base-p2"
tokenizer = BertTokenizer.from_pretrained(MODEL_NAME)
bert_model = TFBertModel.from_pretrained(MODEL_NAME)]

# Model definition
input_ids = tf.keras.layers.Input(shape=(max_len,), dtype=tf.int32, name="input_ids")
attention_mask = tf.keras.layers.Input(shape=(max_len,), dtype=tf.int32, name="attention_mask")

bert_output = bert_model(
    input_ids,
    attention_mask=attention_mask
)

# Combine BERT with CNN
sequence_output = bert_output.last_hidden_state
conv1 = tf.keras.layers.Conv1D(128, kernel_size=3, activation='relu')(sequence_output)
conv1 = tf.keras.layers.GlobalMaxPooling1D()(conv1)

output = tf.keras.layers.Dense(num_classes, activation='softmax')(conv1)
model = tf.keras.models.Model(inputs=[input_ids, attention_mask], outputs=output)

# Compile model
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=5e-5),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

```

Gambar 4. 7 Penerapan model IndoBERT + CNN

Langkah pertama adalah memuat tokenizer dan model pretrained *IndoBERT Base P2* menggunakan pustaka Hugging Face Transformers. *Tokenizer* berfungsi mengubah teks mentah menjadi token numerik yang kompatibel dengan model BERT, sedangkan IndoBERT digunakan untuk mengekstraksi embedding teks dari konteks inputnya. Dalam model ini, output dari IndoBERT berupa representasi last hidden state, yaitu embedding token-token dalam teks input.

Arsitektur *hybrid* ini memanfaatkan output embedding dari IndoBERT sebagai input untuk lapisan CNN. Representasi sekuensial dari teks diproses oleh lapisan 1D Convolutional Layer dengan kernel ukuran 3, yang berfungsi mendeteksi pola-pola lokal dalam teks, seperti frasa penting atau kombinasi kata spesifik. Aktivasi ReLU digunakan dalam lapisan ini untuk menambahkan non-linearitas. Setelah itu, output dari CNN diringkas menggunakan Global Max Pooling, yang menangkap nilai fitur maksimum dalam setiap filter untuk menghasilkan representasi yang padat.

Hasil dari lapisan CNN ini dilewatkan ke lapisan dense dengan aktivasi softmax untuk menghasilkan probabilitas prediksi kelas. Model ini kemudian dikompilasi menggunakan optimasi Adam dengan fungsi loss sparse categorical crossentropy, yang

cocok untuk klasifikasi multikelas. Pelatihan dilakukan selama beberapa epoch dengan menggunakan dataset pelatihan dan validasi, yang memungkinkan model untuk belajar pola dalam data.

Pendekatan hybrid ini sangat efektif karena memanfaatkan kemampuan BERT dalam memahami konteks global teks dan kecepatan CNN dalam menangkap pola lokal. Kombinasi ini menjadikannya pilihan yang ideal untuk berbagai tugas NLP, seperti klasifikasi sentimen, analisis topik, atau deteksi kategori dokumen, terutama dalam konteks bahasa Indonesia. Model ini tidak hanya akurat tetapi juga efisien dalam mengolah teks panjang dengan struktur kompleks.

4.7 Perancangan Model IndoBERT dengan RNN

Pada perancangan model kali ini menggunakan *IndoBERT base p2* yang digabungkan dengan model RNN. Model hybrid yang menggabungkan *IndoBERT Base P2* dan RNN dirancang untuk memanfaatkan keunggulan masing-masing metode dalam pemrosesan bahasa alami (NLP). Model ini memanfaatkan kemampuan transformer dalam IndoBERT untuk menghasilkan representasi teks yang mendalam, lalu menambahkan lapisan RNN untuk menangkap pola sekuensial yang lebih kompleks. Dengan pendekatan ini, model mampu menganalisis teks berbahasa Indonesia secara lebih mendalam dan kontekstual. Penerapan model nya dapat dilihat pada gambar berikut ini.

```
# Load tokenizer dan model IndoBERT
MODEL_NAME = "indobenchmark/indobert-base-p2"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
bert_model = TFAutoModel.from_pretrained(MODEL_NAME)

# Tokenisasi input
input_ids, attention_masks = preprocess_data(X, tokenizer, MAX_LEN)
```

```

# Hybrid Model: BERT + RNN
def create_hybrid_model(bert_model, max_len, num_classes):
    # Input BERT
    input_ids = tf.keras.Input(shape=(max_len,), dtype=tf.int32, name="input_ids")
    attention_masks = tf.keras.Input(shape=(max_len,), dtype=tf.int32, name="attention_masks")

    # Freeze BERT layers
    for layer in bert_model.layers:
        layer.trainable = False

    # BERT embeddings
    bert_output = bert_model(input_ids, attention_mask=attention_masks)
    bert_pooled_output = bert_output.last_hidden_state

    # RNN Layer
    rnn_layer = tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128, return_sequences=False))(bert_pooled_output)

    # Fully connected layer
    dense = tf.keras.layers.Dense(128, activation="relu")(rnn_layer)
    dropout = tf.keras.layers.Dropout(0.3)(dense)
    output = tf.keras.layers.Dense(num_classes, activation="softmax")(dropout)

    # Define model
    model = tf.keras.Model(inputs=[input_ids, attention_masks], outputs=output)
    return model

# Compile model
num_classes = y.shape[1]
model = create_hybrid_model(bert_model, MAX_LEN, num_classes)
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=2e-5),
              loss="categorical_crossentropy",
              metrics=["accuracy"])

```

Gambar 4. 8 Model Hybrid IndoBERT + RNN

Tahap pertama adalah memuat tokenizer dan model pretrained IndoBERT Base P2 menggunakan pustaka *Hugging Face Transformers*. Tokenizer bertugas mengonversi teks mentah menjadi token numerik, sementara IndoBERT digunakan untuk mengekstraksi embedding teks berdasarkan konteksnya. Data teks kemudian ditokenisasi, menghasilkan representasi berupa *input IDs* dan *attention masks*. Kedua representasi ini digunakan sebagai input utama untuk model.

Arsitektur model *hybrid* terdiri dari dua komponen utama. Pertama, model IndoBERT bertugas menghasilkan embedding dari teks input. Lapisan-lapisan dalam IndoBERT dibekukan agar tidak dilatih ulang, sehingga mengurangi kebutuhan komputasi. Kedua, output dari IndoBERT diteruskan ke lapisan *Bidirectional LSTM* untuk memahami hubungan antar-token secara dua arah, sehingga pola-pola dalam teks dapat dimodelkan dengan lebih baik. Setelah itu, representasi dari lapisan RNN dilewatkan ke lapisan fully connected dengan dropout untuk menghindari overfitting, dan diakhiri dengan lapisan softmax untuk menghasilkan prediksi akhir.

Model ini dikompilasi menggunakan optimasi *Adam* dengan laju pembelajaran rendah untuk stabilitas, serta fungsi loss *categorical_crossentropy* yang sesuai untuk klasifikasi multikelas. Kombinasi ini menghasilkan model yang mampu memanfaatkan kekuatan IndoBERT dalam memahami konteks bahasa dan RNN dalam menangkap pola sekuensial.

4.8 Hyperparameter

Untuk seluruh model dalam penelitian ini peneliti menggunakan *hyperparameter* yang sama, *hyperparameter* yang digunakan yaitu menggunakan *batch size* 32, *learning rate* $2e^{-5}$, dan dengan *epoch* 10. Untuk lebih jelasnya dapat dilihat pada tabel 4.2 dibawah ini.

Table 4. 2 Hyperparamater

Hyperparameter	Skala
Batch size	32
Learning rate	$2e^{-5}$
Epoch	10

Dari tabel di atas menampilkan *hyperparameter batch size* diatur pada nilai 32, yang berarti model akan memproses 32 data sekaligus dalam satu iterasi pelatihan. Selanjutnya, *learning rate* ditetapkan sebesar 2×10^{-5} , menunjukkan seberapa besar langkah pembaruan bobot model dilakukan selama pelatihan. Nilai ini penting untuk menjaga keseimbangan antara kecepatan konvergensi dan stabilitas proses pelatihan. Terakhir, *epoch* bernilai 10, menandakan bahwa seluruh data pelatihan akan dilewatkan sebanyak 10 kali melalui model. Kombinasi *hyperparameter* ini dirancang untuk mengoptimalkan kinerja model dalam menghasilkan prediksi yang akurat tanpa menyebabkan overfitting atau underfitting.

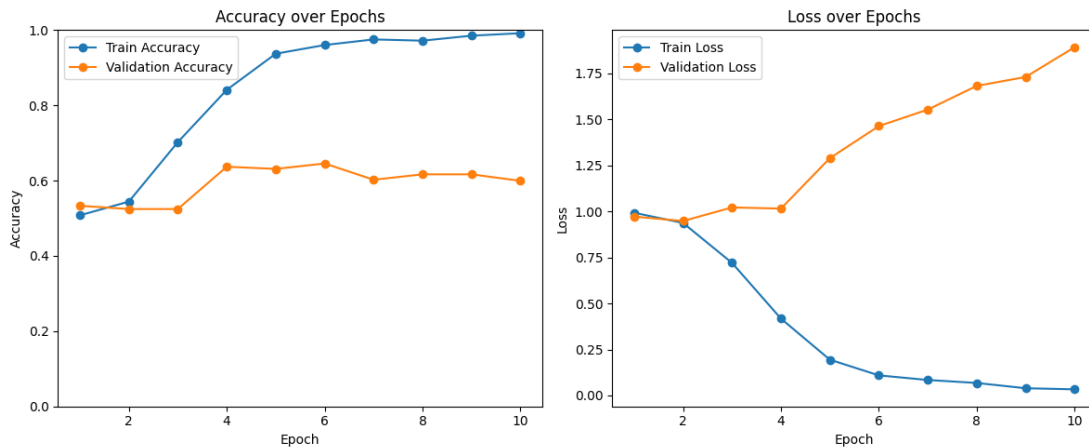
4.9 Hasil Pelatihan

Dalam melakukan analisis sentimen pada teks dalam penelitian ini, peneliti membagi dataset menjadi tiga bagian dalam analisis IndoBERT biasa menjadi data train, data valid, dan data test dengan porsi 70% untuk data train, 20% untuk data validasi, dan 10% untuk data test. Sedangkan untuk model IndoBERT dengan RNN,

dan IndoBERT dengan CNN peneliti membagi menjadi dua bagian, yaitu 80% untuk data train, dan 20% untuk data test.

4.9.1 Hasil Train validation Accuracy dan Loss Model IndoBERT

Hasil train validation accuracy dan loss model IndoBERT dengan menggunakan epoch 10 pada dataset judul dapat dilihat pada gambar 4.9. dibawah ini.



Gambar 4.9 Grafik Akurasi dan loss

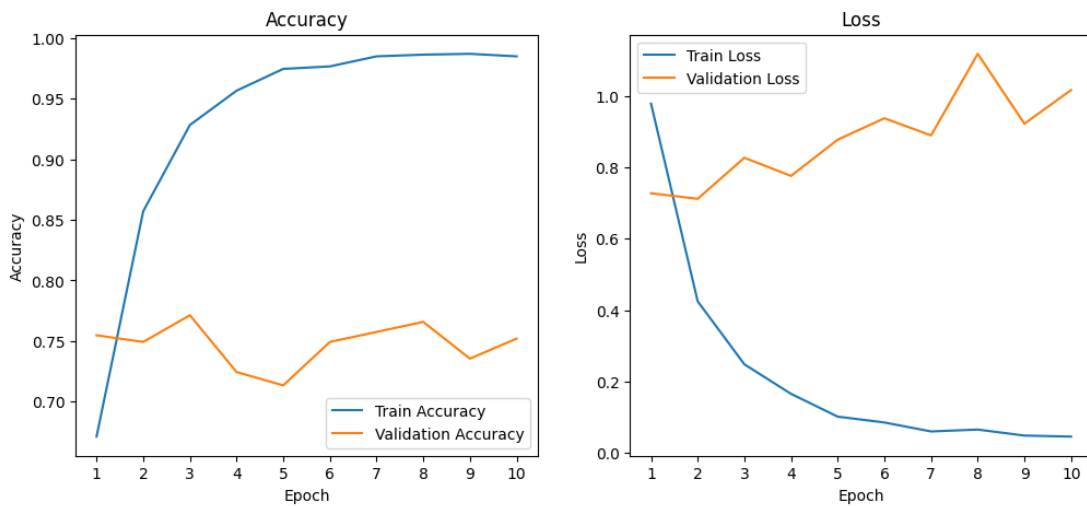
Pada gambar grafik di atas dapat kita lihat bahwa grafik tersebut menunjukkan perubahan akurasi dan loss selama proses pelatihan model dalam 10 *epoch*. Pada grafik akurasi, terlihat bahwa akurasi pada data pelatihan (Train Accuracy) meningkat tajam hingga mencapai hampir 100% (tepatnya diangka 0.94) di *epoch* ke-5, kemudian cenderung stabil di *epoch-epoch* berikutnya. Sebaliknya, akurasi pada data validasi (Validation Accuracy) meningkat hingga sekitar 60% pada awal pelatihan, tetapi setelah itu stagnan dan tidak menunjukkan peningkatan signifikan.

Sementara itu, pada grafik loss, terdapat penurunan yang konsisten pada loss data pelatihan (Train Loss) seiring bertambahnya *epoch*, menunjukkan bahwa model semakin mampu memprediksi data pelatihan dengan baik. Namun, loss pada data validasi (Validation Loss) justru mengalami kenaikan setelah beberapa epoch awal. Ini mengindikasikan bahwa model mulai mengalami overfitting, yaitu ketika model terlalu terlatih pada data pelatihan sehingga performanya pada data validasi menurun.

Kesimpulannya, grafik ini menunjukkan bahwa meskipun model memiliki performa yang sangat baik pada data pelatihan, terdapat masalah overfitting atau data pelatihan tidak representatif terhadap data validasi.

4.9.2 Hasil Train-validation Accuracy dan Loss Model IndoBERT dengan Model CNN

Penelitian ini menggabungkan antara model IndoBERT dengan model CNN. Hasil train validation accuracy dan loss model IndoBERT dengan model hybrid tersebut dengan menggunakan *epoch* 10 pada dataset judul dapat dilihat dalam tampilan dalam bentuk grafik pada gambar 4.10 dibawah ini.



Gambar 4. 10 Gambar Grafik Akurasi dan Loss

Diketahui bahwa grafik di atas menunjukkan tren akurasi dan loss selama pelatihan dan validasi untuk 10 *epoch*. Pada grafik sebelah kiri, terlihat bahwa akurasi pelatihan meningkat tajam hingga mencapai hampir 100% (di angka 0.97) pada *epoch* ke-5 dan kemudian cenderung stabil. Sebaliknya, akurasi validasi awalnya meningkat, namun mulai stagnan dan sedikit menurun setelah beberapa *epoch*, menunjukkan potensi overfitting.

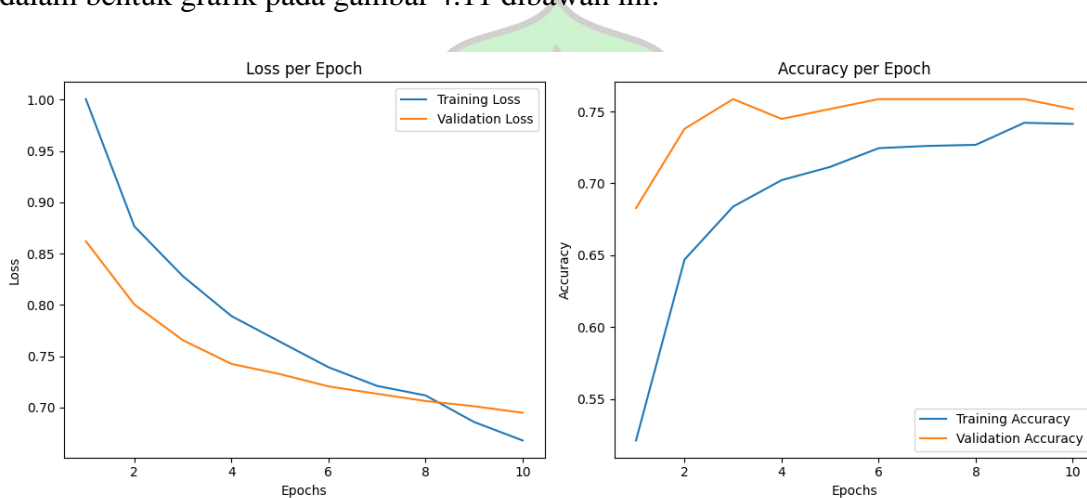
Grafik di sebelah kanan menggambarkan tren loss. Loss pada data pelatihan terus menurun secara konsisten hingga hampir mendekati nol, mencerminkan bahwa model semakin baik dalam memprediksi data pelatihan. Namun, loss pada data validasi

justru meningkat setelah beberapa epoch awal, menunjukkan bahwa model kurang generalisasi pada data yang tidak terlihat selama pelatihan.

Secara keseluruhan, grafik ini mengindikasikan bahwa model mengalami overfitting, di mana performa pada data pelatihan sangat baik, tetapi tidak terjaga pada data validasi.

4.9.3 Hasil Train-validation Accuracy dan Loss Model IndoBERT dengan Model RNN

Penelitian ini menggabungkan antara model IndoBERT dengan model RNN. Hasil train validation accuracy dan loss model IndoBERT dengan model hybrid tersebut dengan menggunakan *epoch* 10 pada dataset judul dapat dilihat pada tampilan dalam bentuk grafik pada gambar 4.11 dibawah ini.



Gambar 4. 11 Gambar Grafik Akurasi dan Loss

Gambar grafik di atas menunjukkan performa model berdasarkan metrik akurasi dan loss yang sama seperti sebelumnya, yaitu hasil selama pelatihan dan validasi dalam 10 *epoch*. Pada grafik sebelah kiri, yang menggambarkan loss, terlihat bahwa baik loss pelatihan maupun validasi terus menurun secara konsisten seiring bertambahnya *epoch*. Hal ini mengindikasikan bahwa model berhasil belajar dari data dengan lebih baik seiring waktu, baik pada data pelatihan maupun validasi.

Di grafik sebelah kanan, yang menunjukkan akurasi, akurasi pelatihan meningkat secara bertahap hingga mencapai sekitar 75%. Akurasi validasi juga meningkat dengan cepat pada epoch awal dan tetap stabil setelah beberapa epoch, menunjukkan bahwa

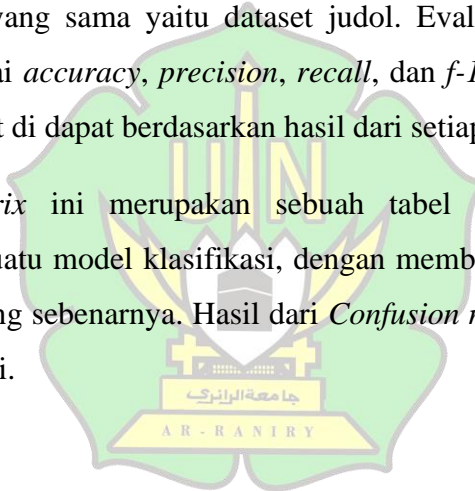
model memiliki performa yang cukup baik untuk data yang tidak dilihat selama pelatihan.

Secara keseluruhan, tren pada kedua grafik ini menunjukkan bahwa model tidak mengalami overfitting yang signifikan. Penurunan loss yang konsisten pada kedua dataset, serta stabilitas akurasi validasi, mengindikasikan bahwa model memiliki generalisasi yang cukup baik.

4.10 Hasil Confusion Matrix dan Hasil Evaluasi Model

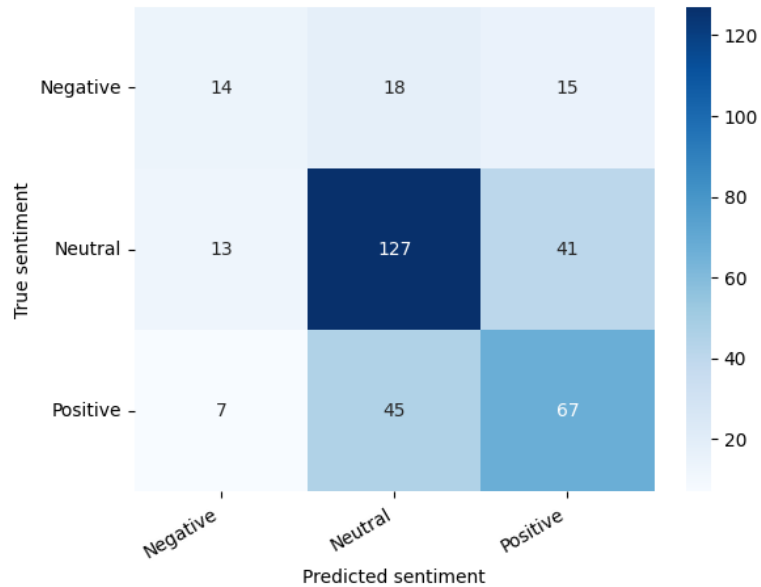
Setelah melewati tahap fine-tuning dari setiap model yang telah di uji, selanjutnya akan dilakukan perbandingan terhadap performa setiap model yang telah di latih. Perbandingan dilakukan antara model IndoBERT dengan model *Hybrid* yang menggunakan dataset yang sama yaitu dataset judul. Evaluasi model akan dinilai dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *f-1 score* dari dataset yang diberikan. Hasil tersebut di dapat berdasarkan hasil dari setiap confusion matrix.

Confusion matrix ini merupakan sebuah tabel yang digunakan untuk mengevaluasi kinerja suatu model klasifikasi, dengan membandingkan hasil prediksi model terhadap data yang sebenarnya. Hasil dari *Confusion matrix* pada setiap model dapat dilihat dibawah ini.



4.10.1 Confusion Matrix model IndoBERT

Untuk melihat hasil confusion matrix dari model IndoBERT base biasa dapat dilihat pada gambar 4.12 di bawah ini.

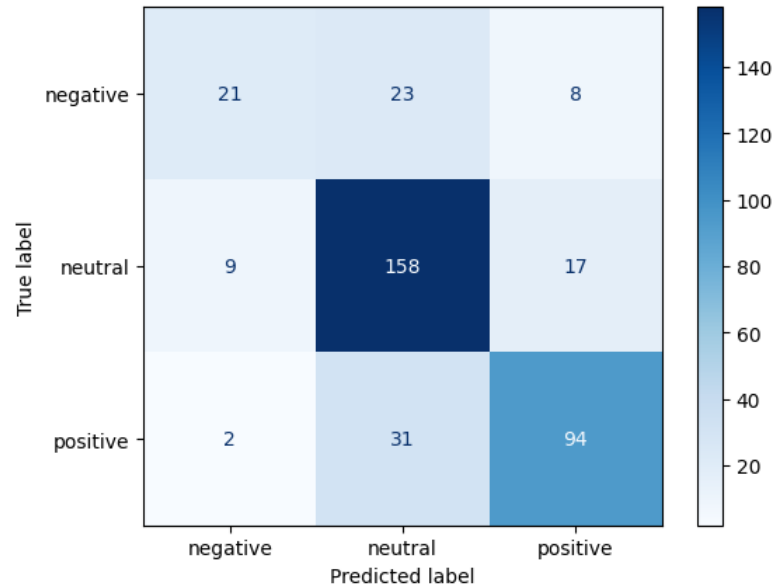


Gambar 4. 12 Confusion Matrix Model IndoBERT

Pada gambar di atas, model hanya mampu memprediksi dengan baik dua label saja (*neutral* dan *positive*). Label *neutral* mendominasi prediksi dengan angka akurasi tertinggi sebesar 127, tetapi masih terdapat kesalahan yang signifikan pada label *positive* yang sering salah diklasifikasikan sebagai *neutral*. Situasi ini menunjukkan keterbatasan model dalam mengenali semua kategori secara seimbang, sehingga distribusi prediksi menjadi tidak proporsional.

4.10.2 Confusion Matrix Model IndoBERT dengan Model CNN

Untuk melihat hasil *confusion matrix* dari model IndoBERT dengan CNN terhadap dataset judul dapat dilihat pada gambar 4.13 di bawah ini.

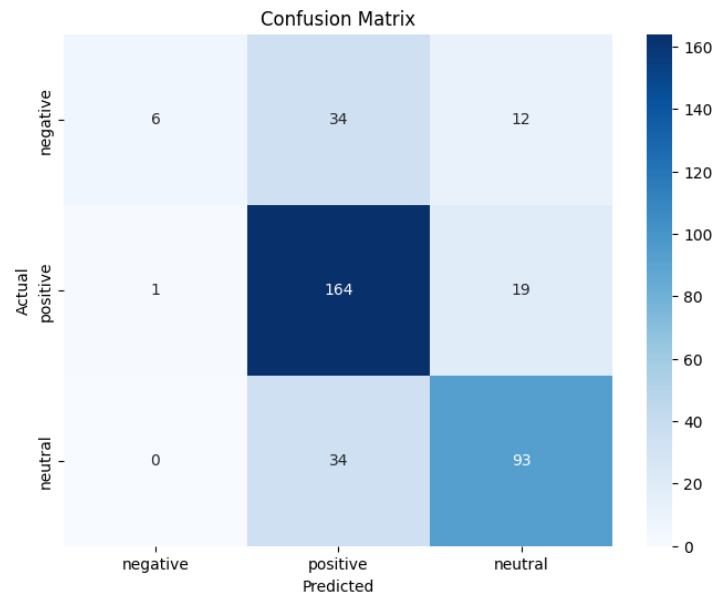


Gambar 4. 13 Confusion Matrix Model IndoBERT dengan Model CNN

Grafik ini secara keseluruhan, menunjukkan akurasi sedang dalam prediksi. Label *neutral* memiliki kinerja terbaik dengan 158 prediksi benar, sementara *negative* dan *positive* memiliki tingkat kesalahan lebih tinggi. Hal ini terlihat dari jumlah signifikan salah prediksi pada label lain, misalnya 23 kali *negative* diprediksi sebagai *neutral* dan 31 kali *positive* diprediksi sebagai *neutral*. Ini mengindikasikan bahwa model sering kali salah mengenali polaritas emosi pada data.

4.10.3 Confusion Matrix Model IndoBERT dengan Model RNN

Untuk melihat hasil confusion matrix dari model IndoBERT dengan RNN terhadap dataset judul dapat dilihat pada gambar 4.14 di bawah ini.



Gambar 4. 14 Confusion Matrix Model IndoBERT dengan Model RNN

Pada Confusion Matrix ini kategori negatif hanya memiliki prediksi yang benar sebanyak 6 sampel, dengan 34 sampel salah diklasifikasikan sebagai *positive* dan 12 sebagai *neutral*. Untuk kategori positif, model memberikan prediksi yang sangat akurat dengan 164 klasifikasi benar, meskipun terdapat 19 kesalahan pada kategori *neutral*. Untuk kategori *neutral*, model mengklasifikasikan 93 sampel dengan benar tetapi keliru mengklasifikasikan 34 sampel sebagai *positive*.

Dari ketiga confusion matrix di atas maka di dapatkan hasil evaluasi dari setiap model sebagai berikut.

4.10.4 Hasil Pengujian Model IndoBERT

Pada tahap pertama ini dilakukan evaluasi performa model terhadap dataset yang telah diberikan yaitu dataset judul. Berikut hasil performa model terhadap data yang diberikan. Hasil tersebut dapat dilihat pada gambar 4.15 berikut ini.

	precision	recall	f1-score	support
Negative	0.41	0.30	0.35	47
Neutral	0.67	0.70	0.68	181
Positive	0.54	0.56	0.55	119
accuracy			0.60	347
macro avg	0.54	0.52	0.53	347
weighted avg	0.59	0.60	0.59	347

Gambar 4. 15 Performa Model Terhadap Dataset Judul

Dari gambar di atas hasil evaluasi model menunjukkan performa yang relatif moderat dengan akurasi keseluruhan sebesar 60%, dengan tingkat akurasi yang cukup memadai dalam pengklasifikasian data. Pada kelas *Negative*, model menunjukkan *presisi* sebesar 0,41, *recall* 0,30, dan *f1-score* 0,35 dengan total 47 sampel. Meskipun ada *presisi* yang lebih tinggi, nilai *recall* yang rendah mengindikasikan bahwa model belum mampu mengenali sebagian besar sampel dari kelas ini dengan baik, dan ini menunjukkan bahwa model kesulitan dalam mengidentifikasi data yang benar-benar negatif.


Pada kelas *Neutral*, performa model lebih baik, dengan *presisi* 0,67, *recall* 0,70, dan *f1-score* 0,68 untuk 181 sampel yang diprediksi sebagai *neutral*. Kelas ini menunjukkan keseimbangan yang lebih baik antara *presisi* dan *recall*, yang menunjukkan bahwa model cukup baik dalam mengenali sentimen *neutral* dan menghasilkan prediksi yang akurat. Untuk kelas *Positive*, model mencatatkan *presisi* 0,54, *recall* 0,56, dan *f1-score* 0,55 dari 119 sampel yang diprediksi sebagai *positive*. Meskipun nilai-nilai ini lebih baik dibandingkan dengan kelas *Negative*, model masih

memiliki ruang untuk meningkatkan kinerjanya, terutama dalam hal *recall*, yang menunjukkan bahwa model masih melewatkan beberapa sampel dari kelas ini.

Secara keseluruhan, model ini memiliki akurasi 60%, yang mengindikasikan bahwa prediksi yang benar hanya terjadi pada sekitar 60% dari seluruh sampel. Dalam hal rata-rata makro (*macro avg*), *presisi* mencapai 0,54, *recall* 0,52, dan *f1-score* 0,53, yang menunjukkan bahwa model secara umum memiliki performa yang cukup baik, meskipun ada ketidakseimbangan antara kelas-kelas yang ada. Skor rata-rata berbobot (*weighted avg*) untuk *presisi* 0,59, *recall* 0,60, dan *f1-score* 0,59 menunjukkan bahwa meskipun ada ketidakseimbangan antar kelas, model memberikan hasil yang lebih baik secara keseluruhan, dengan kontribusi terbesar berasal dari kelas *Neutral* yang memiliki jumlah sampel terbanyak.

4.10.5 Hasil Pengujian Model IndoBERT dengan Model CNN

Sama seperti pada tahap sebelumnya, pada tahap kedua ini akan dilakukan evaluasi performa model terhadap dataset yang sama yaitu dataset judul. Namun pada tahap ini hasil evaluasi berasal dari model *Hybrid*, yaitu antara model IndoBERT dengan CNN. Berikut hasil performa model terhadap data yang diberikan. Hasil tersebut dapat dilihat pada gambar 4.16 berikut ini.



	precision	recall	f1-score	support
negative	0.66	0.40	0.50	52
neutral	0.75	0.86	0.80	184
positive	0.79	0.74	0.76	127
accuracy			0.75	363
macro avg	0.73	0.67	0.69	363
weighted avg	0.75	0.75	0.74	363

Gambar 4. 16 Performa Model Hybrid Terhadap Dataset Judol

Dapat dilihat bahwa hasil evaluasi model menggunakan pendekatan *hybrid* antara IndoBERT dan CNN menunjukkan peningkatan performa yang signifikan dibandingkan dengan model sebelumnya. Dengan akurasi keseluruhan mencapai 75%, model ini mampu meningkatkan prediksi sentimen dengan cukup baik pada berbagai kelas dibandingkan dengan model sebelumnya yang hanya mencatatkan akurasi 60%.

Pada kelas *Negative*, model mencatatkan *presisi* sebesar 0,66, *recall* 0,40, dan *f1-score* 0,50 dengan 52 sampel yang diprediksi sebagai *negative*. Meskipun *presisi* cukup tinggi, *recall* yang lebih rendah menunjukkan bahwa model masih kesulitan untuk mengenali sebagian besar sampel dari kelas ini. Hal ini menunjukkan bahwa meskipun model cukup akurat dalam memprediksi kelas *Negative*, masih ada beberapa sampel yang salah diklasifikasikan atau terlewat.

Pada kelas *Neutral*, model menunjukkan performa yang sangat baik, dengan *presisi* 0,75, *recall* 0,86, dan *f1-score* 0,80 untuk 184 sampel. Nilai *recall* yang tinggi mengindikasikan bahwa model sangat baik dalam mengenali sampel yang sebenarnya *neutral*. Skor *f1-score* yang juga tinggi menunjukkan keseimbangan yang baik antara *presisi* dan *recall*, yang mencerminkan bahwa model sangat efektif dalam mengklasifikasikan kelas ini dengan akurasi yang tinggi.

Untuk kelas *Positive*, model memperoleh *presisi* 0,79, *recall* 0,74, dan *f1-score* 0,76 dari 127 sampel yang diprediksi sebagai *positive*. Angka-angka ini menunjukkan bahwa model cukup kuat dalam memprediksi kelas *Positive*, dengan *presisi* yang lebih tinggi daripada *recall*. Meskipun ada sedikit perbedaan antara *recall* dan *presisi*, model ini masih menunjukkan performa yang solid dalam mengidentifikasi sentimen *positive*.

Secara keseluruhan, model ini mencapai akurasi 75%, yang menunjukkan bahwa 75% prediksi dari model adalah benar. Rata-rata makro (*macro avg*) untuk *presisi*, *recall*, dan *f1-score* adalah 0,73, 0,67, dan 0,69, yang mengindikasikan bahwa meskipun ada perbedaan dalam kinerja antara kelas-kelas, model ini masih memberikan hasil yang cukup baik secara keseluruhan. Rata-rata berbobot (*weighted avg*) menunjukkan skor 0,75 untuk *presisi*, 0,75 untuk *recall*, dan 0,74 untuk *f1-score*, yang menunjukkan bahwa model ini memberikan performa yang stabil, dengan kontribusi terbesar datang dari kelas *Neutral* dan *Positive*, yang memiliki lebih banyak sampel.

Secara keseluruhan, penggunaan model *hybrid* IndoBERT dengan CNN berhasil meningkatkan performa secara signifikan dibandingkan dengan hasil awal yang hanya mencapai akurasi 60%, terutama pada kelas *Neutral* dan *Positive*. Walaupun ada beberapa tantangan pada kelas *Negative*, model ini sudah menunjukkan

kemampuan yang baik dalam mengidentifikasi dan mengklasifikasikan berbagai jenis sentiment.

4.10.6 Hasil Pengujian Model IndoBERT dengan Model RNN

Masih sama seperti pada tahap sebelumnya, pada tahap ketiga ini juga dilakukan evaluasi performa model terhadap dataset judul. Namun pada tahap ketiga ini hasil evaluasi berasal dari model *Hybrid* yang berbeda, yaitu antara model IndoBERT dengan RNN. Berikut merupakan hasil performa model *hybrid* terhadap dataset yang diberikan. Hasil tersebut dapat dilihat pada gambar 4.17 berikut ini.

	precision	recall	f1-score	support
0	0.86	0.12	0.20	52
1	0.71	0.89	0.79	184
2	0.75	0.73	0.74	127
accuracy			0.72	363
macro avg	0.77	0.58	0.58	363
weighted avg	0.74	0.72	0.69	363

Gambar 4. 17 Performa Model Hybrid Terhadap Dataset Judol

Dari gambar di atas dapat dilihat bahwa hasil evaluasi model menggunakan pendekatan hybrid antara IndoBERT dan RNN menunjukkan akurasi keseluruhan sebesar 72%, yang sedikit lebih rendah dibandingkan dengan model *hybrid* IndoBERT dan CNN yang sebelumnya mencapai 75%. Meskipun demikian, model ini tetap memberikan hasil yang cukup baik dalam mengklasifikasikan data dengan berbagai tingkat sentimen.

Pada kelas 0 (*Negative*), model mencatatkan *presisi* 0,86, *recall* 0,12, dan *f1-score* 0,20 untuk 52 sampel. Meskipun *presisi* tinggi, nilai *recall* yang sangat rendah menunjukkan bahwa model kesulitan untuk mengidentifikasi sebagian besar sampel dari kelas ini. Artinya, meskipun ketika model mengklasifikasikan sampel sebagai kelas 0 (*negative*), prediksinya cenderung benar, namun model gagal mengenali sebagian besar sampel *negative* yang sebenarnya ada dalam data. Hal ini menunjukkan adanya ketidakseimbangan dalam pengklasifikasian kelas 0 (*negative*), yang mempengaruhi kinerja model secara keseluruhan.

Pada kelas 1 (*neutral*), model menunjukkan performa yang sangat baik, dengan *presisi* 0,71, *recall* 0,89, dan *f1-score* 0,79 untuk 184 sampel. Nilai *recall* yang tinggi menunjukkan bahwa model sangat efektif dalam mengenali sebagian besar sampel dari kelas ini, yang memberikan kontribusi besar terhadap akurasi keseluruhan model. *F1-score* yang juga tinggi menunjukkan keseimbangan yang baik antara *presisi* dan *recall* pada kelas ini, mencerminkan kemampuan model yang solid dalam mengklasifikasikan sentimen *neutral*.

Untuk kelas 2 (*positive*), model memperoleh *presisi* 0,75, *recall* 0,73, dan *f1-score* 0,74 dari 127 sampel. Model ini menunjukkan kemampuan yang cukup baik dalam mengklasifikasikan kelas *Positive*, dengan nilai *recall* dan *presisi* yang cukup seimbang. Meskipun *recall* sedikit lebih rendah daripada *presisi*, model masih memberikan hasil yang baik dalam mengenali sentimen *positive*.

Secara keseluruhan, model ini mencapai akurasi 72%, yang menunjukkan bahwa 72% dari prediksi model adalah benar. Rata-rata makro (*macro avg*) untuk *presisi*, *recall*, dan *f1-score* adalah 0,77, 0,58, dan 0,58, yang mencerminkan bahwa meskipun *presisi* rata-rata cukup tinggi, *recall* yang rendah menunjukkan adanya kesulitan model dalam mengenali sampel dari kelas yang lebih kecil atau tidak dominan (terutama kelas 0). Skor rata-rata berbobot (*weighted avg*) untuk *presisi* 0,74, *recall* 0,72, dan *f1-score* 0,69 menunjukkan bahwa meskipun ada ketidakseimbangan antara kelas-kelas tersebut, model ini menunjukkan performa yang lebih baik secara keseluruhan, dengan kontribusi terbesar datang dari kelas 1 (*Neutral*) dan kelas 2 (*Positive*), yang memiliki jumlah sampel lebih banyak.

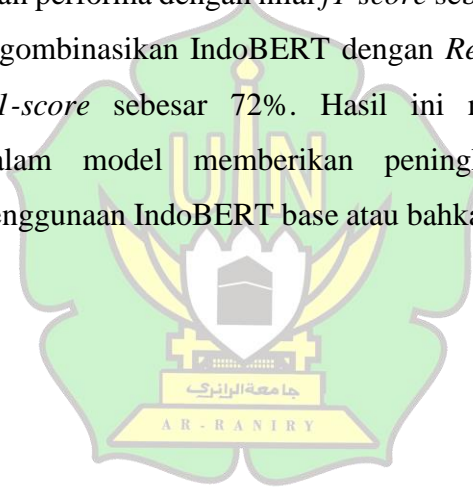
Secara keseluruhan, model hybrid berbasis IndoBERT dan RNN menghasilkan performa yang lebih rendah dibandingkan kombinasi IndoBERT dengan CNN pada dataset yang sama yaitu dataset judul. Kelemahan utamanya adalah kegagalan dalam mengenali sampel dari kelas *negative*, yang menurunkan performa keseluruhan model. Meskipun model ini cukup baik dalam memprediksi kelas *neutral* dan *positive*, hasil ini menunjukkan bahwa RNN mungkin kurang mampu menangani pola kompleks pada dataset ini dibandingkan CNN, terutama ketika digabungkan dengan IndoBERT.

Dari seluruh hasil akurasi yang didapat, berikut perbandingan nilai *f1-score* dari setiap akurasi yang di dapat pada setiap model yang digunakan, dapat dilihat dalam tabel 4.3 berikut

Table 4. 3 Hasil perbandingan Akurasi setiap Dataset

No	Model	Nilai f1-score Dataset Judol
1	IndoBERT base	60%
2	IndoBERT dengan CNN	75%
3	IndoBERT dengan RNN	72%

Dari tabel di atas dapat kita lihat jika model pertama, yang menggunakan IndoBERT base, menghasilkan *f1-score* sebesar 60%. Model kedua, yang menggabungkan IndoBERT dengan *Convolutional Neural Network* (CNN), menunjukkan peningkatan performa dengan nilai *f1-score* sebesar 75%. Sementara itu, model ketiga yang mengombinasikan IndoBERT dengan *Recurrent Neural Network* (RNN) memperoleh *f1-score* sebesar 72%. Hasil ini mengindikasikan bahwa penggunaan CNN dalam model memberikan peningkatan yang signifikan dibandingkan dengan penggunaan IndoBERT base atau bahkan penggabungan dengan RNN.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan pada bab sebelumnya diketahui bahwa pada penelitian ini dilakukan analisis sentimen opini masyarakat terhadap judi online di media sosial Instagram. Data yang diambil sebanyak 1.811 data komentar yang kemudian diberikan label berupa *positive*, *negative*, dan *neutral*. Penelitian ini di uji dengan menggunakan model IndoBERT dan model *Hybrid*, yaitu gabungan antara model IndoBERT dengan CNN dan model IndoBERT dengan RNN. Dari hasil penelitian yang diperoleh, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Proses analisis sentimen dengan menggunakan model IndoBERT dan model *Hybrid* menggunakan parameter yang sama yaitu dengan *learning rate* $2e-5$, *epoch* 10, dan *batch size* 32 pada pelatihan.
2. Dari hasil pengujian yang dilakukan pada bab sebelumnya maka diperoleh hasil sebagai berikut: Analisis sentimen yang menggunakan model IndoBERT base menghasilkan akurasi sebesar 60%, analisis sentimen yang menggunakan model *hybrid* antara model IndoBERT dengan CNN memperoleh hasil akurasi sebesar 75%, dan model IndoBERT dengan RNN memperoleh hasil akurasi sebesar 72%.
3. Berdasarkan hasil seluruh akurasi yang diperoleh menunjukkan bahwa metode *hybrid* yang menggabungkan model IndoBERT dengan CNN mendapatkan hasil terbaik dalam melakukan analisis sentimen dibandingkan kedua model lainnya. Nilai akurasi yang didapat yaitu sebesar 75%.

5.2 Saran

Terdapat beberapa saran yang diberikan untuk pengembangan penelitian selanjutnya:

1. Memperhatikan keseimbangan terhadap jumlah data pada setiap label positif, negative, dan netral agar menghasilkan akurasi yang lebih baik pada model.

2. Menggunakan metode pelabelan dataset yang lain untuk mengetahui perbedaan dan pengaruh terhadap performa model.
3. Melakukan penelitian dengan memanfaatkan model IndoBERT dan model hybrid lainnya untuk mengembangkan penelitian dalam pemrosesan bahasa alami (NLP) yang berbahasa Indonesia.



DAFTAR PUSTAKA

- Aulia, K., & Amelia, L. (2020). Analisis Sentimen Twitter Pada Isu Mental Health Dengan Algoritma Klasifikasi Naive Bayes. *Siliwangi Journal (Seri Sains and Teknologi)*, 6(2), 60–65.
- Khairani, U., Mutiawani, V., & Ahmadian, H. (2024). Pengaruh Tahapan Preprocessing Terhadap Model Indobert Dan Indobertweet Untuk Mendeteksi Emosi Pada Komentar Akun Berita Instagram. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(4), 887–894. <https://doi.org/10.25126/jtiik.1148315>
- Lambang, S., Pradana, S., Sains, F., & Teknologi, D. (2024). Analisis sentimen masyarakat media sosial twitter terhadap kinerja pejabat gubernur DKI Jakarta menggunakan model indobert. <https://repository.uinjkt.ac.id/dspace/handle/123456789/77071>
- Merdiansah, R., & Ali Ridha, A. (2024). Sentiment Analysis of Indonesian X Users Regarding Electric Vehicles Using IndoBERT. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 7(1), 221–228.
- Viana, F. N. (2024). Klasifikasi Emosi Opini Twitter Menggunakan Model Benchmark Indonlu. [https://repository.ar-raniry.ac.id/id/eprint/35631/%0Ahttps://repository.ar-raniry.ac.id/id/eprint/35631/1/Farla Nara Viana, 190705037, FST, TI.pdf](https://repository.ar-raniry.ac.id/id/eprint/35631/%0Ahttps://repository.ar-raniry.ac.id/id/eprint/35631/1/Farla%20Nara%20Viana,%20190705037,%20FST,%20TI.pdf)
- Uswatun, K., "Analisis Sentimen Terhadap Tempat Wisata Menggunakan Metode Naïve Bayes Classifier dan Support Vector Machine," Universitas Islam Syarif Hidayatullah, Jakarta, 2023.
- Alvianda, F., Indriati, I., & Adikara, P. P. (2019). "Analisis Sentimen Konten Radikal Di Media Sosial Twitter Menggunakan Metode Support Vector Machine (SVM)." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(1), 241-246.
- Mailoa, F. F. (2019). "Analisis sentimen data twitter menggunakan metode text mining tentang masalah obesitas di indonesia." *Journal of Information Systems for Public Health*, 6(1), 44-51.
- Ryanizar, M. B. "Analisis sentimen pada media sosial twitter terhadap produk mixue dengan metode naïve bayes classifier dan support vector machine (SVM)" (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).
- Setianingsih, D. R. "ANALISIS OPINI PUBLIK MENGENAI PEMBELAJARAN DARING DIMASA PANDEMI COVID 19 PADA MEDIA SOSIAL TWITTER." *PERANG OPINI DI MEDIA SOSIAL*, 252.

- Yanuarti, R. (2021). "Analisis Media Sosial Twitter Terhadap Topik Vaksinasi Covid-19." *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 6(2), 121-130.
- Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2023). "Pre-trained language models and their applications." *Engineering*, 25, 51-65.
- Hendri, A., Taufik, F, A., Hammam, R., & Kahlil, M. (2024). "Hybrid Models for Emotion Classification and Sentiment Analysis in Indonesian Language." Wiley, 2-6.
- Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, 8(2), 1888.
- Putra, T. I. Z. M., Suprpto, S., & Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 1-28.
- Putra, K. B. D., Kurniawan, V. A., Maharani, A. B., & Fudholi, D. R. (2024). Pelaporan Jalan Rusak dengan Deteksi Citra dan Uji Viralitas dari Data Twitter. *Buletin Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi*, 2(1), 47-52.
- Setyawan, B. A. *Analisis sentimen multilabel terhadap ujaran kebencian menggunakan gabungan metode indo bert lite dan bidirectional lstm-cnn dengan grid search hyperparameter optimization* (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).
- Putra, T. I. Z. M., Suprpto, S., & Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 1-28.
- Iftikar, M. A., & Sibaroni, Y. (2022). Analisis Sentimen Twitter: Penanganan Pandemi Covid-19 Menggunakan Metode Hybrid Naïve Bayes, Decision Tree, dan Support Vector Machine. *eProceedings of Engineering*, 9(3).
- Hidayat, M. N., & Pramudita, R. (2024). Analisis Sentimen Terhadap Pembelajaran Secara Daring Pasca Pandemi Covid-19 Menggunakan Metode IndoBERT. *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS: Journal of Information Management*, 8(2), 161-170.
- Guridno, C., Azimah, A., & Ningsih, S. (2024). ANALISIS HYBRID METODE CNN DAN LSTM DALAM MEDIA BERITA ONLINE INDONESIA. *Jurnal Sistem Informasi Bisnis (JUNSIBI)*, 5(1), 86-101.

- Imron, S., Setiawan, E. I., & Santoso, J. (2023). Deteksi Aspek Review E-Commerce Menggunakan IndoBERT Embedding dan CNN. *INSYST: Journal of Intelligent System and Computation*, 5(1), 10-16.
- Sinapoy, M. I. K., Sibaroni, Y., & Prasetyowati, S. S. (2023). Comparison of lstm and indobert method in identifying hoax on twitter. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 7(3), 657-662.
- Yefferson, D. Y., Lawijaya, V., & Girsang, A. S. (2024). Hybrid model: IndoBERT and long short-term memory for detecting Indonesian hoax news. *Int J Artif Intell*, 13(2), 1913-1924.




```

Case Folding

[] # Fungsi untuk membersihkan teks
def clean_review(text):
    # Ubah teks menjadi huruf kecil
    text = text.lower()
    # Ubah enter menjadi spasi
    text = re.sub('^\n', ' ', text)
    # Hapus emoji
    text = emoji.demojize(text)
    text = re.sub('[:p-z-7,;,:]', '', text) # Hapus nama emoji yang dicover
    # Hapus smilie
    text = re.sub('[:x;:]?([dpvwo01])', '', text)
    # Hapus link
    text = re.sub('https://v[^\s]*', '', text)
    # Hapus isorime
    text = re.sub('@[^\s]+[\s]', '', text)
    # Hapus hashtag
    text = re.sub('#([a-z-2-7])', '#', text)
    # Hapus angka dan beberapa simbol
    text = re.sub('[^a-zA-Z-2-7]', '', text)
    # Hapus karakter berbaris
    text = re.sub('\n', '\n', text)
    # Bersihkan spasi berlebih
    text = re.sub('[ ]+', ' ', text)
    return text

Pre Processing

[] def preprocess_vi(df):
    df_pp = df.copy()
    df_pp['review_text'] = df_pp['review_text'].map(clean_review)
    # Hapus baris yang kosong atau hanya berisi spasi
    df_pp['review_text'] = df_pp['review_text'].replace('', np.nan, inplace=True)
    df_pp['review_text'] = df_pp['review_text'].replace(' ', np.nan, inplace=True)
    # Hapus baris dengan nilai NaN pada kolom 'review_text'
    df_pp.dropna(subset=['review_text'], inplace=True)
    return df_pp

[] # Hasil pre processing
df_vi = preprocess_vi(df)
print(df_vi)

df_vi.to_csv('dataset_clean.tsv', sep='\t', header=None, index=False)

[] # Mengecek jumlah sampel per kelas
class_counts = df_vi['label'].value_counts()

# Menampilkan kelas yang hanya memiliki satu sampel
print('Kelas dengan satu sampel:', class_counts[class_counts == 1])

# Menghapus kelas yang hanya memiliki 1 sampel
df_vi_filtered = df_vi[df_vi['label'].isin(class_counts[class_counts > 1].index)]

# Membagi dataset menjadi train, val, dan test set
train_set, val_set = train_test_split(df_vi_filtered, test_size=0.3, stratify=df_vi_filtered['label'], random_state=1)
val_set, test_set = train_test_split(val_set, test_size=0.33, stratify=val_set['label'], random_state=1)

[] print('Train shape: (train_set.shape)')
print('Val shape: (val_set.shape)')
print('Test shape: (test_set.shape)')

[] # Export to tsv
train_set.to_csv('train_set.tsv', sep='\t', header=None, index=False)
val_set.to_csv('val_set.tsv', sep='\t', header=None, index=False)
test_set.to_csv('test_set.tsv', sep='\t', header=None, index=False)

Training dataset

###
# Common functions
###
def set_seed(seed):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)

def count_param(module, trainable=False):
    if trainable:
        return sum(p.numel() for p in module.parameters() if p.requires_grad)
    else:
        return sum(p.numel() for p in module.parameters())

def get_lr(optimizer):
    for param_group in optimizer.param_groups:
        return param_group['lr']

def metrics_to_string(metric_dict):
    string_list = []
    for key, value in metric_dict.items():
        string_list.append('{}: {:.2f}'.format(key, value))
    return ' '.join(string_list)

[] # Set random seed
set_seed(20692020)

Load Model

[] # Load Tokenizer and Config
tokenizer = BertTokenizer.from_pretrained('Indobenchmark/Indobert-base-p1')
config = BertConfig.from_pretrained('Indobenchmark/Indobert-base-p1')
config.num_labels = DocumentSentimentDataset.NLP_LABELS

# Instantiate model
model = BertForSequenceClassification.from_pretrained('Indobenchmark/Indobert-base-p1', config=config)

[] model

[] count_param(model)
124443651

Prepare Dataset

[] train_dataset_path = './content/train_set.tsv'
valid_dataset_path = './content/val_set.tsv'
test_dataset_path = './content/test_set.tsv'

[] train_loader = DocumentSentimentDataLoader(dataset=train_dataset, max_seq_len=512, batch_size=32, num_workers=16, shuffle=True)
valid_loader = DocumentSentimentDataLoader(dataset=valid_dataset, max_seq_len=512, batch_size=32, num_workers=16, shuffle=False)
test_loader = DocumentSentimentDataLoader(dataset=test_dataset, max_seq_len=512, batch_size=32, num_workers=16, shuffle=False)

[] W01, I2w = DocumentSentimentDataset.LABEL2INDEX, DocumentSentimentDataset.INDEX2LABEL
print(W01)
print(I2w)

```

```

Fine Tuning & Evaluation

[] # Update learning rate and number of epochs
optimizer = optim.Adam(model.parameters(), lr=2e-5)
model = model.cuda()

[] # Train
n_epochs = 10
history = defaultdict(list)

for epoch in range(n_epochs):
    model.train()
    torch.set_grad_enabled(True)

    total_train_loss = 0
    list_hyp_train, list_label = [], []

    train_pbar = tqdm(train_loader, leave=True, total=len(train_loader))
    for i, batch_data in enumerate(train_pbar):
        # Forward model
        loss, batch_hyp, batch_label = forward_sequence_classification(model, batch_data[-1], i2w=i2w, device='cuda')

        # Update model
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        tr_loss = loss.item()
        total_train_loss += tr_loss

    # Calculate metrics
    list_hyp_train += batch_hyp
    list_label += batch_label

[] # train_pbar.set_description("{} TRAIN LOSS: {:.4f} | {:.4f}".format(
    epoch + 1, total_train_loss / (i + 1), get_lr(optimizer)))

# Calculate train metric
metrics = document_sentiment_metrics_fn(list_hyp_train, list_label)
print("{} TRAIN LOSS: {:.4f} | {:.4f}".format(
    epoch + 1, total_train_loss / (i + 1), metrics_to_string(metrics), get_lr(optimizer)))

# Save train accuracy and loss for learning curve
history['train_acc'].append(metrics['ACC'])
history['train_loss'].append(total_train_loss / len(train_loader))

# Evaluate on validation
model.eval()
torch.set_grad_enabled(False)

total_loss = 0
list_hyp, list_label = [], []

pbar = tqdm(valid_loader, leave=True, total=len(valid_loader))
for i, batch_data in enumerate(pbar):
    batch_size = batch_data[-1]
    loss, batch_hyp, batch_label = forward_sequence_classification(model, batch_data[-1], i2w=i2w, device='cuda')

    # Calculate total loss
    valid_loss = loss.item()
    total_loss += valid_loss

# Calculate evaluation metrics
list_hyp += batch_hyp
list_label += batch_label
metrics = document_sentiment_metrics_fn(list_hyp, list_label)

pbar.set_description("{} VALID LOSS: {:.4f} | {:.4f}".format(total_loss / (i + 1), metrics_to_string(metrics)))
metrics = document_sentiment_metrics_fn(list_hyp, list_label)
print("{} VALID LOSS: {:.4f} | {:.4f}".format(epoch + 1, total_loss / (i + 1), metrics_to_string(metrics)))

# Save validation accuracy and loss for learning curve
history['val_acc'].append(metrics['ACC'])
history['val_loss'].append(total_loss / len(valid_loader))

Visualisasi hasil training & Evaluasi

[] # Plot training and validation accuracy and loss
plt.figure(figsize=(12, 8))

# Accuracy plot
plt.subplot(1, 2, 1)
epochs = range(1, 11) # Adjusting to show 10 epochs
plt.plot(epochs, history['train_acc'], label='Train Accuracy', markers='s')
plt.plot(epochs, history['val_acc'], label='Validation Accuracy', markers='o')
plt.title('Accuracy over epochs')
plt.xlabel('epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.ylim(0, 1)

# Loss plot
plt.subplot(1, 2, 2)
plt.plot(epochs, history['train_loss'], label='Train Loss', markers='o')
plt.plot(epochs, history['val_loss'], label='Validation Loss', markers='o')
plt.title('Loss over epochs')
plt.xlabel('epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()

Classification report Confusion Matrix

Validation Dataset

[] #confusion matrix
def show_confusion_matrix(confusion_matrix):
    heatmap = sns.heatmap(confusion_matrix, annot=True, fmt="d", cmap="Blues")
    heatmap.set_xticks(heatmap.get_xticks(), rotation=0, ha='right')
    heatmap.set_yticks(heatmap.get_yticks(), rotation=0, ha='right')
    plt.xlabel('True sentiment')
    plt.ylabel('Predicted sentiment')

cm = confusion_matrix(val_real, val_pred)
df_cm = pd.DataFrame(cm, index=['Negative', 'Neutral', 'Positive'], columns=['Negative', 'Neutral', 'Positive'])
show_confusion_matrix(df_cm)

```

Berikut *Source Code* untuk IndoBERT dengan CNN

```

import pandas as pd
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
from transformers import BertTokenizer, TFSeq2VecWrapper

# Bertlike mode memilik IndoBERT pretrained model
MODEL_NAME = "IndoBERT/indobert-base-p2"
tokenizer = BertTokenizer.from_pretrained(MODEL_NAME)
bert_model = TFSeq2VecWrapper.from_pretrained(MODEL_NAME)

```



```

[] # Load data
data_path = "/content/labelled_comments_FTK.xlsx" # Ganti dengan path file Anda
df = pd.read_excel(data_path)

# Memberikan data di kolom 'label'
df['label'] = df['label'].str.strip() # Menghapus spasi di awal dan akhir
df['label'] = df['label'].str.lower() # Mengubah menjadi huruf kecil

# Hitung ulang value counts()
df_count = df['label'].value_counts()
print("Shape: ", df.shape)
print(df_count)

[] # Label encoding
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df_filtered['label'] = label_encoder.fit_transform(df_filtered['label'])
num_classes = len(label_encoder.classes_)

[] # Split data
from sklearn.model_selection import train_test_split
train_texts, test_texts, train_labels, test_labels = train_test_split(
    df_filtered['review_text'], df_filtered['label'], test_size=0.2, random_state=42
)

[] # Tokenization
def tokenize_texts(texts, tokenizer, max_len=128):
    return tokenizer(
        list(texts),
        max_length=max_len,
        padding='max_length',
        truncation=True,
        return_tensors='tf'
    )

max_len = 128
train_encodings = tokenize_texts(train_texts, tokenizer, max_len)
test_encodings = tokenize_texts(test_texts, tokenizer, max_len)

[] # Prepare datasets
train_dataset = tf.data.Dataset.from_tensor_slices((
    dict(train_encodings, train_labels)
)).shuffle(1000).batch(32)
test_dataset = tf.data.Dataset.from_tensor_slices((
    dict(test_encodings, test_labels)
)).batch(32)

[] # Model definition
input_ids = tf.keras.layers.Input(shape=(max_len,), dtype=tf.int32, name='input_ids')
attention_mask = tf.keras.layers.Input(shape=(max_len,), dtype=tf.int32, name='attention_mask')

bert_output = bert_model(
    input_ids,
    attention_mask*attention_mask
)

[] # Combine BERT with CNN
sequence_output = bert_output.last_hidden_state
conv1 = tf.keras.layers.Conv1D(32, kernel_size=3, activation='relu')(sequence_output)
conv1 = tf.keras.layers.GlobalMaxPooling1D()(conv1)
output = tf.keras.layers.Dense(num_classes, activation='softmax')(conv1)
model = tf.keras.models.Model(inputs=[input_ids, attention_mask], outputs=output)

# Compile model
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=2e-5),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

[] # Train model
epochs = 10
history = model.fit(
    train_dataset,
    validation_data=test_dataset,
    epochs=epochs
)

[] # Evaluate model
test_loss, test_acc = model.evaluate(test_dataset)
print("Test accuracy: ", test_acc)

[] # Predictions
predictions = model.predict(test_dataset)
predicted_labels = np.argmax(predictions, axis=-1)

[] # Classification report
report = classification_report(test_labels, predicted_labels, target_names=label_encoder.classes_)
print(report)

[] # Confusion matrix
cm = confusion_matrix(test_labels, predicted_labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=label_encoder.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.show()

[] # Plot training history
def plot_training_history(history, epochs=10):
    plt.figure(figsize=(12, 5))

    # Accuracy plot
    plt.subplot(1, 2, 1)
    plt.plot(range(1, epochs+1), history.history['accuracy'], label='Train Accuracy') # Explicit range
    plt.plot(range(1, epochs+1), history.history['val_accuracy'], label='Validation Accuracy') # Explicit range
    plt.title('Accuracy')
    plt.xlabel('Epoch')
    plt.xticks(ticks=range(1, epochs+1)) # Set ticks to 1 to 10
    plt.ylabel('Accuracy')
    plt.legend()

    # Loss plot
    plt.subplot(1, 2, 2)
    plt.plot(range(1, epochs+1), history.history['loss'], label='Train Loss') # Explicit range
    plt.plot(range(1, epochs+1), history.history['val_loss'], label='Validation Loss') # Explicit range
    plt.title('Loss')
    plt.xlabel('Epoch')
    plt.xticks(ticks=range(1, epochs+1)) # Set ticks to 1 to 10
    plt.ylabel('Loss')
    plt.legend()

    plt.show()

# Ensure you pass the correct number of epochs
plot_training_history(history, epochs=10)

```

Berikut *Source Code* untuk IndoBERT dengan RNN

```

[] # Instalasi pustaka yang dibutuhkan
!pip install transformers tensorflow scikit-learn matplotlib seaborn opencv

# Import pustaka yang diperlukan
import tensorflow as tf
from transformers import AutoTokenizer, TFAutoModel
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, precision_recall_fscore_support
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

[] # Load tokenizer dan model IndoBERT
MODEL_NAME = "IndoBERT/indobert-base-p2"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
bert_model = TFAutoModel.from_pretrained(MODEL_NAME)

```

```

[] # load dataset excel (pastikan dataset memiliki nama kolom "review_text" dan "label")
# ganti path di bawah dengan path dataset Anda
data_path = "content/labelled_comments_FIR.xlsx"
df = pd.read_excel(data_path)

# Pembersihan data di kolom "label"
df["label"] = df["label"].str.strip() # Menghapus spasi di awal dan akhir
df["label"] = df["label"].str.lower() # Mengubah menjadi huruf kecil

# Hitung ulang value counts()
df_count = df["label"].value_counts()
print(df_count)
print(df_count)

[] # Preprocessing data
def preprocess_data(data, tokenizer, max_len):
    input_ids, attention_masks = [], []
    for text in data:
        encoded = tokenizer.encode_plus(
            text,
            add_special_tokens=True,
            max_length=max_len,
            pad_to_max_length=True,
            truncation_strategy="only_first",
            return_attention_mask=True,
        )
        input_ids.append(encoded["input_ids"])
        attention_masks.append(encoded["attention_mask"])
    return np.array(input_ids), np.array(attention_masks)

[] # Definisikan model
MAX_LEN = 128
X = df_filtered["review_text"].astype(str)
y = pd.get_dummies(df_filtered["label"]).values

# Tokenisasi input
input_ids, attention_masks = preprocess_data(X, tokenizer, MAX_LEN)

# Split data menjadi training dan testing
X_train_ids, X_test_ids, X_train_masks, X_test_masks, y_train, y_test = train_test_split(
    input_ids, attention_masks, y, test_size=0.1, random_state=0
)

[] # Hybrid Model: BERT + RNN
def create_hybrid_model(bert_model, max_len, num_classes):
    # Input BERT
    input_ids = tf.keras.Input(shape=(max_len,), dtype=tf.int32, name="input_ids")
    attention_masks = tf.keras.Input(shape=(max_len,), dtype=tf.int32, name="attention_masks")

    # Freeze BERT layers
    for layer in bert_model.layers:
        layer.trainable = False

    # BERT embeddings
    bert_output = bert_model(input_ids, attention_mask=attention_masks)
    bert_pooled_output = bert_output.last_hidden_state

    # RNN layer
    rnn_layer = tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128, return_sequences=False))(bert_pooled_output)

    # Fully connected layer
    dense = tf.keras.layers.Dense(128, activation="relu")(rnn_layer)
    dropout = tf.keras.layers.Dropout(0.3)(dense)
    output = tf.keras.layers.Dense(num_classes, activation="softmax")(dropout)

    # Define model
    model = tf.keras.Model(inputs=[input_ids, attention_masks], outputs=output)
    return model

[] # Compile model
num_classes = y.shape[1]
model = create_hybrid_model(bert_model, MAX_LEN, num_classes)
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=2e-5),
              loss="categorical_crossentropy",
              metrics=["accuracy"])

[] # Train model
history = model.fit(
    [X_train_ids, X_train_masks], y_train,
    validation_split=0.1,
    epochs=10,
    batch_size=32
)

[] # Evaluasi model
y_pred = model.predict([X_test_ids, X_test_masks])
y_pred_labels = np.argmax(y_pred, axis=-1)
y_test_labels = np.argmax(y_test, axis=-1)
print(classification_report(y_test_labels, y_pred_labels))

[] # Menampilkan Confusion Matrix
conf_matrix = confusion_matrix(y_test_labels, y_pred_labels)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
           xticklabels=df_filtered["label"].unique(),
           yticklabels=df_filtered["label"].unique())
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

[] # Menampilkan Grafik Loss dan Akurasi
plt.figure(figsize=(12, 9))

# Grafik Loss
plt.subplot(1, 2, 1)
plt.plot(range(1, 11), history.history['loss'], label='Training Loss')
plt.plot(range(1, 11), history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss per Epoch')
plt.legend()

# Grafik Akurasi
plt.subplot(1, 2, 2)
plt.plot(range(1, 11), history.history['accuracy'], label='Training Accuracy')
plt.plot(range(1, 11), history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Accuracy per Epoch')
plt.legend()

plt.tight_layout()
plt.show()

```