

Analisis Teks Narasi Kebijakan Tapera di Media Sosial Twitter (X) Menggunakan Model IndoBERT untuk Pemetaan Persepsi Publik

TUGAS AKHIR

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)
dalam Teknologi Informasi



Oleh : **GUNTUR SYARIFUDDIN PUTRA**

NIM : 2108096045

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
TAHUN 2025**

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama : Guntur Syarifuddin Putra

NIM : 2108096045

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

**Analisis Teks Narasi Kebijakan Tapera di Media Sosial
Twitter (X) Menggunakan Model IndoBERT Untuk
Pemetaan Persepsi Publik**

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

Semarang, 10 Maret 2025
Pembuat Pernyataan,



Guntur Syarifuddin Putra
NIM: 2108096045



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601259 Fax. 7615387

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul : Analisis Teks Narasi Kebijakan Tapera di Media Sosial Twitter (X) Menggunakan Model IndoBERT untuk Pemetaan Persepsi Publik
Penulis : Guntur Syarifuddin Putra
NIM : 2108096045
Jurusan : Teknologi Informasi

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

Semarang, 23 April 2025

DEWAN PENGUJI

Ketua Sidang

Hery Mustofa, M.Kom.

NIP. 198703172019031007

Sekretaris Sidang

Siti Nur'aini, M.Kom.

NIP. 198401312018012001

Penguji Utama I

Dr. Masy Ari Ulincha, M.T.

NIP. 19810812201101107

Penguji Utama II

Nur Cahyo Hendro Wibowo, M.Kom

NIP. 197312222006041001

Pembimbing I

Dr. Khotibul Umam, M.Kom.

NIP. 197908272011011007

Pembimbing II

Siti Nur'aini, M.Kom.

NIP. 198401312018012001

NOTA PEMBIMBING I

Semarang, 10 Maret 2025

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

Assalamu'alaikum. Wr. Wb.

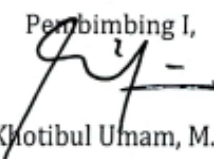
Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Analisis Teks Narasi Kebijakan
Tapera di Media Sosial Twitter (X)
Menggunakan Model IndoBERT
Untuk Pemetaan Persepsi Publik
Nama : **Guntur Syarifuddin Putra**
NIM : 2108096045
Jurusan : Teknologi Informasi

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

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,


Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

NOTA PEMBIMBING II

Semarang, 10 Maret 2025

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

Assalamu'alaikum. Wr. Wb.

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

Judul : Analisis Teks Narasi Kebijakan
Tapera di Media Sosial Twitter (X)
Menggunakan Model IndoBERT
Untuk Pemetaan Persepsi Publik
Nama : **Guntur Syarifuddin Putra**
NIM : 2108096045
Jurusan : Teknologi Informasi

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

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,



Siti Nur'aini, M.Kom
NIP. 198401312018012001

MOTTO

“Wong jujur bakalan mujur, wong tulus dalane bakalan mulus,
wong asor ora bakal ndlosor. Utamakan 3S bolo
(Senyum, Sopan, SANGAR)”

(danielbudianto28)

ABSTRAK

Nama : Guntur Syarifuddin Putra
NIM : 2108096045
Judul : **Analisis Teks Narasi Kebijakan Tapera di Media Sosial Twitter (X) Menggunakan Model IndoBERT untuk Pemetaan Persepsi Publik**

Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap kebijakan Tabungan Perumahan Rakyat (Tapera) di media sosial Twitter (X) menggunakan model IndoBERT. Penelitian didasarkan pada tingginya penggunaan media sosial di Indonesia, khususnya Twitter, sebagai platform untuk mengekspresikan opini publik, serta kontroversi yang muncul terkait implementasi kebijakan Tapera yang dianggap memberatkan masyarakat. Permasalahan yang dikaji meliputi kecenderungan sentimen masyarakat, penerapan model IndoBERT, dan efektivitas model dalam menganalisis narasi kebijakan Tapera. Pengumpulan data dilakukan melalui proses *crawling* menggunakan *Tweet Harvest* dengan kata kunci "tapera" dari Januari hingga Oktober 2024, menghasilkan 2.604 *tweet*. Data kemudian dilabeli secara manual oleh pakar bahasa ke dalam tiga kategori sentimen, dilanjutkan dengan tahap *pre-processing*, serta augmentasi data menggunakan *Easy Data Augmentation* (EDA) untuk menyeimbangkan distribusi kelas. Model IndoBERT diimplementasikan dengan *fine-tuning* menggunakan *learning rate* $2e-5$, *batch size* 16, dan 5 *epochs*. Hasil penelitian menunjukkan akurasi model sebesar 94% dengan *precision*, *recall*, dan *F1-score* rata-rata 94%.

Kata kunci : Tapera, Analisis Sentimen, IndoBERT

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah SWT atas limpahan nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Teks Narasi Kebijakan Tapera di Media Sosial Twitter (X) Menggunakan Model IndoBERT untuk Pemetaan Persepsi Publik" dengan baik. Skripsi ini disusun sebagai salah satu syarat kelulusan dalam program sarjana (S1) pada Program Studi Teknologi Informasi di Universitas Islam Negeri Walisongo, Kota Semarang. Selain itu, penelitian ini juga bertujuan untuk memberikan wawasan bagi pembaca mengenai analisis narasi kebijakan Tapera di media sosial.

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan dan bantuan, baik dalam pelaksanaan penelitian maupun dalam penyusunan skripsi ini. Penulis menyadari bahwa tanpa bimbingan, arahan, serta motivasi dari berbagai pihak, proses penyelesaian skripsi ini tidak akan berjalan dengan lancar. Oleh karena itu, penulis mengucapkan penghargaan dan rasa terima kasih yang mendalam kepada:

1. Bapak Prof. Dr. Nizar, M.Ag selaku Rektor Universitas Islam Negeri Walisongo Semarang.

2. Bapak Prof. Dr. H. Musahadi, M.Ag selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
3. Bapak Dr. Khotibul Umam, M.Kom selaku Ketua Program studi Teknologi Informasi sekaligus dosen wali dan pembimbing pertama yang telah memberikan bimbingan sejak awal mengikuti perkuliahan hingga skripsi ini terselesaikan.
4. Ibu Siti Nur'aini, M,Kom, selaku dosen Pembimbing II yang telah banyak memberi masukan dan arahan yang membuka pikiran penulis dalam penyusunan skripsi dengan sabar.
5. Staff, karyawan dan dosen di lingkungan Universitas Islam Negeri Walisongo Semarang.
6. Orang tua tercinta dan keluarga yang selalu menemani dalam membantu penulis dan selalu mendo'akan serta memberikan dukungan kepada penulis.
7. Teman-teman Teknologi Informasi yang selalu memberi dukungan.
8. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang terlibat dalam pembuatan skripsi ini sehingga dapat terselesaikan dengan baik.

Dalam proses pelaksanaan dan penyusunan skripsi ini, penulis menyadari bahwa masih terdapat berbagai kekurangan dan ketidaksempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik serta saran yang membangun

guna meningkatkan kualitas penulisan ini. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Semarang, 10 Maret 2025

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN	iii
LEMBAR PENGESAHAN.....	v
NOTA PEMBIMBING I	vii
NOTA PEMBIMBING II.....	ix
MOTTO	xi
ABSTRAK.....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI	xix
DAFTAR GAMBAR.....	xxiii
DAFTAR TABEL	xxvii
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah	8
C. Batasan Masalah	8
D. Tujuan Penelitian	9
E. Manfaat Penelitian.....	10
BAB II LANDASAN PUSTAKA	13
A. Kajian Teori	13
1. Kebijakan Tapera.....	13
2. Twitter	15
3. Analisis Sentimen	16
4. Easy Data Augmentation (EDA).....	17

5. Natural Language Processing (NLP)	20
6. Transformers	22
7. Bidirectional Encoder Representations from Transformers (BERT).....	26
8. IndoBERT.....	36
9. Confusion Matrix	38
B. Kajian Penelitian yang Relevan.....	42
BAB III METODOLOGI PENELITIAN.....	49
A. Metode Pengumpulan Data	49
1. Studi Pustaka.....	49
2. Observasi Lapangan Digital.....	49
B. Tahapan Penelitian.....	50
C. Uraian Metodologi	53
1. Pengambilan Data Twitter	53
2. Pelabelan Data.....	54
3. Pre-Processing Data	55
4. Easy Data Augmentation (EDA).....	57
5. Splitting Data.....	61
6. Implementasi Model IndoBERT.....	62
7. Pengujian Akhir Model	65
BAB IV HASIL DAN PEMBAHASAN	67
A. Pengambilan Data Twitter.....	67
B. Pelabelan	70
C. Pre-Processing.....	76

1. Case Folding.....	76
2. Spell Normalized.....	79
3. Cleaned text	83
D. Easy Data Augmentation (EDA).....	87
E. Splitting Data	91
F. Implementasi IndoBERT	93
1. Tokenizing.....	93
2. Fine-Tuning IndoBERT.....	95
3. Evaluasi Matrix.....	109
G. Pengujian Akhir Model.....	114
1. Pengujian dengan Dataset Berlabel.....	115
2. Pengujian dengan Kalimat Acak	117
3. Hasil Keseluruhan Pengujian.....	120
BAB V KESIMPULAN DAN SARAN.....	121
A. Kesimpulan.....	121
B. Saran.....	122
DAFTAR PUSTAKA	125
LAMPIRAN-LAMPIRAN	133

DAFTAR GAMBAR

Gambar	Judul	Halaman
Gambar 2.1	Arsitektur Tranformer	23
Gambar 2.2	<i>Pre-training</i> dan <i>Fine-tuning</i> BERT	27
Gambar 2.3	Representasi <i>Input</i> BERT	29
Gambar 2.4	Proses Tokenisasi BERT <i>Tokenizer</i>	31
Gambar 2.5	<i>Output</i> dan <i>Input</i> dalam BERT	32
Gambar 2.6	Ilustrasi Proses Klasifikasi BERT	33
Gambar 2.7	Tabel <i>Summary</i> Pelatihan IndoBERT	37
Gambar 3.1	<i>Flowchart</i> Alur Penelitian	50
Gambar 3.2	Hasil <i>Crawling</i> Data	54
Gambar 4.1	<i>Source Code</i> Penerapan <i>Tweet</i> <i>Harvest</i>	67
Gambar 4.2	Grafik Data Per Bulan	70
Gambar 4.3	Grafik Distribusi Label Per Bulan	73
Gambar 4.4	Grafik Keseluruhan Distribusi Label	75
Gambar 4.5	<i>Source Code</i> Proses <i>Case</i> <i>Folding</i>	76
Gambar 4.6	<i>Source Code</i> Memuat Kamus Alay	80
Gambar 4.7	<i>Source Code</i> Proses <i>Spell</i> <i>Normalized</i>	81
Gambar 4.8	<i>Source Code</i> Proses <i>Cleaned</i> <i>Text</i>	84
Gambar 4.9	<i>Source Code</i> Empat Teknik EDA	88
Gambar 4.10	Inisialisasi Parameter	90
Gambar 4.11	Perbandingan Distribusi Label Sebelum dan Sesudah Proses EDA	91

Gambar 4.12	<i>Source Code</i> Proses <i>Splitting</i> Data	91
Gambar 4.13	Distribusi Label pada Data Latih, Data Validasi dan Data Uji	92
Gambar 4.14	<i>Source Code</i> Proses Tokenisasi	94
Gambar 4.15	<i>Source Code</i> Fase Pelatihan	96
Gambar 4.16	<i>Source Code</i> Fase Validasi	97
Gambar 4.17	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 1	99
Gambar 4.18	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 1	99
Gambar 4.19	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 2	101
Gambar 4.20	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 2	101
Gambar 4.21	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 3	102
Gambar 4.22	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 3	103
Gambar 4.23	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 4	104
Gambar 4.24	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 4	105
Gambar 4.25	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 5	106
Gambar 4.26	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 5	106
Gambar 4.27	Grafik Akurasi <i>Training</i> dan <i>Validation</i> Skenario 6	108
Gambar 4.28	Grafik <i>Training Loss</i> dan <i>Validation Loss</i> Skenario 6	108
Gambar 4.29	Diagram <i>Confusion Matrix</i> Data Validasi	110
Gambar 4.30	<i>Source Code</i> Perhitungan Otomatis Performa Model	113

Gambar 4.31	Hasil Perhitungan Otomatis Performa Model	114
Gambar 4.32	<i>Source Code</i> Pengujian Data Uji	116
Gambar 4.33	Diagram <i>Confusion Matrix</i> Data Uji	116
Gambar 4.34	Hasil Performa Model Dataset Uji	117
Gambar 4.35	<i>Source Code</i> Menyimpan Model	118
Gambar 4.36	<i>Source Code</i> Memuat Model	118
Gambar 4.37	<i>Source Code</i> Klasifikasi Kalimat Acak	119
Gambar 4.38	Hasil Pengujian Kalimat Pertama	119
Gambar 4.39	Hasil Pengujian Kalimat Kedua	120
Gambar 4.40	Hasil Pengujian Kalimat Ketiga	120

DAFTAR TABEL

Tabel	Judul	Halaman
Tabel 2.1	<i>Confusion Matrix 3x3</i>	38
Tabel 2.2	Penelitian yang Relevan	41
Tabel 3.1	Ilustrasi <i>Case Folding</i>	55
Tabel 3.2	Ilustrasi <i>Spell Normalized</i>	56
Tabel 3.3	Ilustrasi <i>Cleaned Text</i>	56
Tabel 3.4	Contoh Hasil <i>Synonym Replacement</i>	57
Tabel 3.5	Contoh Hasil <i>Random Insertion</i>	58
Tabel 3.6	Contoh Hasil <i>Random Swap</i>	58
Tabel 3.7	Contoh Hasil <i>Random Deletion</i>	59
Tabel 3.8	Ilustrasi <i>Tokenizing</i> IndoBERT	61
Tabel 4.1	Contoh Dataset yang Digunakan	67
Tabel 4.2	Contoh Dataset yang Sudah Diberi Label	69
Tabel 4.3	Contoh Hasil Proses <i>Case Folding</i>	75
Tabel 4.4	Contoh Isi ' <i>colloquial-indonesian-lexicon.csv</i> '	77
Tabel 4.5	Contoh Isi ' <i>kbba.txt</i> '	77
Tabel 4.6	Contoh Isi ' <i>slangword.csv</i> '	78
Tabel 4.7	Contoh Isi ' <i>slang.txt</i> '	78
Tabel 4.8	Contoh Hasil Proses <i>Spell Normalized</i>	79
Tabel 4.9	Contoh Hasil Proses <i>Cleaned Text</i>	83
Tabel 4.10	Contoh Hasil Proses EDA	87
Tabel 4.11	Contoh Hasil Proses Tokenisasi	93
Tabel 4.12	Skenario <i>Hyperparameter</i>	96
Tabel 4.13	Hasil Skenario <i>Hyperparameter</i>	96
Tabel 4.14	<i>Confusion Matrix</i> Data Validasi	108

Tabel 4.15	Hasil Perhitungan Performa Model	Manual	111
------------	-------------------------------------	--------	-----

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam era digital saat ini, perkembangan teknologi informasi dan komunikasi telah menghadirkan perubahan fundamental dalam berbagai aspek kehidupan masyarakat. Salah satu manifestasi paling signifikan dari transformasi digital ini adalah penetrasi media sosial yang memungkinkan pertukaran informasi dan opini secara real-time (Aditya Permana et al., 2021). Media sosial telah berkembang menjadi platform yang tidak hanya menghubungkan antar individu, tetapi juga menjadi arena diskusi publik yang merefleksikan respons masyarakat terhadap berbagai isu sosial dan kebijakan pemerintah (Alfarizy, 2024).

Tingkat penetrasi media sosial di Indonesia menunjukkan angka yang signifikan. Dalam data HootSuite (*We Are Social*) pada Januari 2024 mencatat ada 139,0 juta pengguna media sosial di Indonesia, atau 49,9% dari total populasi yaitu 278,7 juta jiwa. Lebih jauh lagi, survei Status Literasi Digital Indonesia tahun 2022 mengungkapkan bahwa 72,6% responden menjadikan media sosial sebagai sumber utama informasi, melampaui televisi (60%) dan

berita online (27,5%). Dalam konteks ini, Twitter (X) memiliki posisi strategis sebagai platform kelima terpopuler di dunia, dengan rata-rata lebih dari 190 juta pengunjung per bulan. Sedangkan pengguna Twitter di Indonesia mencapai sekitar 27,5 juta pengguna, menempatkan Indonesia di peringkat keempat negara dengan pengguna terbanyak di dunia setelah Amerika Serikat, Jepang, dan India (We Are Social & Meltwater, 2023).

Twitter (X) merupakan media sosial dengan format *microblogging* yang khas, menawarkan fitur unik berupa *tweets* atau cuitan dengan batas 280 karakter (Susanto, 2023). Platform ini telah berkembang menjadi barometer opini publik yang signifikan, dimana pengguna dapat secara bebas mengekspresikan pandangan mereka tentang berbagai isu, termasuk kebijakan pemerintah (Sunitha et al., 2022). Karakteristik Twitter yang memungkinkan interaksi *real-time* dan penyebaran informasi yang cepat menjadikannya sumber data yang berharga untuk analisis sentimen publik (Garcia & Berton, 2021).

Salah satu kebijakan pemerintah yang belakangan ini menjadi sorotan publik di Twitter adalah program Tabungan Perumahan Rakyat (Tapera). Program ini diatur dalam PP Nomor 25 Tahun 2020 yang didasarkan pada Undang-Undang Nomor 4 Tahun 2016. Tapera dirancang

sebagai sistem tabungan wajib untuk pembiayaan perumahan, dengan target utama mengatasi masalah backlog perumahan di Indonesia yang mencapai 12,7 juta unit per Juni 2024 (Tim Riset IDX melalui BPS dan Kementerian PUPR). Angka backlog ini mengindikasikan kesenjangan yang signifikan dalam akses masyarakat terhadap hunian layak, yang berdampak pada kualitas hidup dan kesejahteraan sosial.

Meskipun Tapera dirancang sebagai solusi untuk memperluas akses pembiayaan perumahan, tetapi implementasinya menuai berbagai respons penolakan dari masyarakat. Berdasarkan survei yang dilakukan oleh Damayanti (2024) terhadap 102 pekerja dari berbagai sektor dan wilayah di Indonesia, menunjukkan 81,4% responden menolak program ini. Meski survei ini memiliki keterbatasan dalam hal ukuran sampel, hasil ini mengindikasikan adanya kekhawatiran signifikan di kalangan masyarakat.

Alasan penolakan yang pertama ialah, sesuai Pasal 5 Angka 2 PP No. 25 Tahun 2020, seluruh pekerja dan pekerja mandiri diwajibkan mengikuti program Tapera. Tetapi manfaat dari program Tapera hanya dapat dirasakan oleh Masyarakat Berpenghasilan Rendah (MBR) dengan penghasilan setara UMR atau penghasilan di bawah 8 juta yang belum mempunyai rumah dan/atau untuk pembelian

rumah pertama, pembangunan rumah pertama, atau perbaikan rumah pertama. Manfaat tersebut dapat dirasakan jika sudah memenuhi syarat yaitu sudah menjadi peserta selama paling singkat 12 bulan (Pasal 36, 37, 38 PP 25/2020). Sementara pekerja non-MBR atau yang telah memiliki rumah tetap diwajibkan menjadi peserta tetapi hanya akan mendapat pengembalian pokok tabungan beserta hasil pemupukannya saat kepesertaan taperanya berakhir.

Kedua, yaitu besaran tabungan yang harus dikeluarkan peserta Tapera sebagaimana yang tertera dalam Pasal 15 PP No. 25 Tahun 2020 Tentang Penyelenggaraan Tabungan Perumahan Rakyat, besaran tabungan peserta sebesar 3% dari gaji atau penghasilan pekerja. 3% ini ditanggung bersama oleh pekerja dan pemberi kerja masing-masing 2,5% dan 0,5%, sedangkan untuk pekerja mandiri ditanggung sendiri oleh pekerja secara pribadi.

Tidak hanya itu Badan Penyelenggara Jaminan Sosial Ketenagakerjaan (BPJS Ketenagakerjaan) juga turut menjalankan program serupa. Hal itu tertuang dalam Peraturan Menteri Ketenagakerjaan Nomor 17 Tahun 2021 yang berisi tentang Tata Cara Pemberian, Persyaratan, dan Jenis MLT dalam Program Jaminan Hari Tua (JHT). Akibatnya terjadi tumpang tindih kebijakan dan

kewenangan penyelenggaraan program pembiayaan rumah antar instansi pemerintah. Selain itu, seluruh pekerja yang terdaftar dalam program BPJS Ketenagakerjaan nantinya akan diwajibkan ikut menjadi peserta program Tapera juga. Hal ini menyebabkan besarnya potongan gaji yang didapat oleh para pekerja dan dikhawatirkan akan menambah beban para pekerja.

Ketiga, kewajiban mengikuti program Tapera membebani pekerja yang sudah memiliki rumah sebelumnya. Keikutsertaan eksklusif pekerja yang sudah memiliki rumah sebagai peserta Tapera nampaknya merupakan sebuah paksaan kemauan yang dibebankan kepada pekerja Indonesia (Sulistyowati et al., 2024). Hingga saat ini, pemerintah belum membuka opsi kepada pekerja yang sudah memiliki rumah untuk memilih menjadi peserta atau tidak di program Tapera ini.

Di sisi lain, terdapat juga suara-suara yang mendukung program ini, seperti yang disampaikan oleh Rizki Aljupri dari DPP PAN, yang melihat potensi program ini dalam membantu guru honorer memiliki rumah. Direktur Jenderal Pembiayaan Infrastruktur Pekerjaan Umum dan Perumahan, Triono Junoasmono juga mengatakan “Program Tapera tidak dapat diartikan sebagai beban finansial, namun justru harus diartikan sebagai tabungan bagi peserta yang memberikan banyak manfaat”.

Beragamnya tanggapan atau narasi terhadap kebijakan Tapera membutuhkan analisis yang lebih mendalam dan sistematis. Pendekatan manual dalam menganalisis data volume besar dari media sosial tidak lagi efektif. *Natural Language Processing* (NLP), sebagai cabang Artificial Intelligence, menawarkan solusi untuk menganalisis opini publik secara efisien dan komprehensif (Ramadhan et al., 2020). Dalam konteks ini NLP dimanfaatkan sebagai alat klasifikasi teks, yang kemudian mengklasifikasi narasi masyarakat menjadi positif, netral atau negatif. Salah metode atau model yang biasa digunakan dalam persoalan klasifikasi teks yaitu BERT.

Bidirectional Encoder Representations from Transformers (BERT) merupakan model *Deep Learning* yang didasarkan pada *Transformers*. BERT dapat memahami konteks sebuah kata dari teks yang ambigu berdasarkan konteks di sekitarnya dengan karakteristik bidireksionalnya (Sofi et al., 2023). Model BERT telah melewati masa pelatihan yang menggunakan 104 bahasa termasuk bahasa Indonesia yang dikenal sebagai varian *BERT base multilingual*.

Terdapat model BERT yang dikhususkan untuk pemrosesan bahasa Indonesia, yaitu IndoBERT. Efektifitas IndoBERT dalam memahami konteks dan makna kalimat kompleks yang berbahasa Indonesia sudah terbukti dalam

penelitian-penelitian sebelumnya. Salah satunya ialah dalam penelitian yang dilakukan Juarto dan Yulianto pada tahun 2023 yang membandingkan model IndoBERT dengan metode lain seperti XLNET, BERT *multilingual* dan XLMRoberta, serta metode lain yang berbasis TF-IDF seperti XGBoots, LGB dan *Random Forest* dalam mengklasifikasikan berita yang ada di Indonesia. Dataset yang digunakan pada penelitian tersebut ialah teks yang berupa berita dalam bahasa Indonesia. Dari penelitian tersebut menunjukkan hasil bahwa model IndoBERT memiliki akurasi paling tinggi, yaitu mencapai 94%. Hasil ini menunjukkan model IndoBERT lebih unggul dalam keefektifitasannya dibandingkan dengan metode lain. Hal ini lah yang menjadi dasar mengapa penelitian ini menggunakan model IndoBERT dalam pelaksanaannya.

Tidak hanya memberikan manfaat bagi pembuat kebijakan dalam memahami narasi masyarakat, analisis ini juga sejalan dengan Surah Asy-Syuura ayat 38 yang berbunyi :

وَالَّذِينَ اسْتَجَابُوا لِرَبِّهِمْ وَأَقَامُوا الصَّلَاةَ وَأَمْرُهُمْ شُورَى بَيْنَهُمْ
وَمِمَّا رَزَقْنَاهُمْ يُنفِقُونَ

'Dan (bagi) orang-orang yang menerima (mematuhi) seruan Tuhannya dan mendirikan shalat, sedang urusan mereka (diputuskan) dengan musyawarah antara mereka; dan mereka menafkahkan sebagian

dari rezeki yang Kami berikan kepada mereka' (QS. Asy-Syuura: 38).

Ayat ini menjelaskan pentingnya mendengarkan dan memahami aspirasi masyarakat. Ayat ini menekankan pentingnya musyawarah atau diskusi dalam mengambil keputusan, termasuk dalam kebijakan publik.

Penelitian ini diharapkan dapat memberikan pemahaman mendalam mengenai kecenderungan persepsi publik terhadap kebijakan Tapera dan menjadi masukan bagi pemerintah dalam mengevaluasi serta menyesuaikan kebijakan agar lebih diterima masyarakat.

B. Rumusan Masalah

Berdasarkan uraian latar belakang, rumusan masalah dari penelitian tugas akhir ini adalah:

1. Bagaimana mengimplementasikan model IndoBERT dalam menganalisis teks narasi mengenai kebijakan Tapera pada *tweets* di Twitter?
2. Bagaimana hasil performa model IndoBERT untuk analisis terhadap teks narasi mengenai kebijakan Tapera pada *tweets* di Twitter?

C. Batasan Masalah

Agar penelitian lebih terarah dan jelas, maka dibuatlah batasan permasalahan yang mengacu pada indikator yang akan difokuskan yaitu:

1. Data yang digunakan dalam penelitian ini hanya data *tweets* yang dihasilkan dari scraping pada media sosial Twitter.
2. Data yang diambil hanya berdasarkan kata kunci 'tapera' dari bulan Januari sampai bulan Oktober tahun 2024.
3. Data yang digunakan hanya *tweets* yang menggunakan bahasa Indonesia.
4. Data akan dikategorikan menjadi 3 kategori sentimen yaitu negatif, netral dan positif.

D. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang sudah dijabarkan diatas, pada penelitian ini memiliki beberapa tujuan sebagai berikut:

1. Menerapkan model IndoBERT untuk menganalisis teks narasi mengenai kebijakan Tapera pada *tweets* di Twitter (X).
2. Mengetahui performa model IndoBERT dalam melakukan analisis terhadap teks narasi mengenai kebijakan Tapera pada *tweets* di Twitter (X).
3. Menganalisis kecenderungan sentimen masyarakat terhadap kebijakan Tapera berdasarkan analisis *tweets* di Twitter (X).

E. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat secara praktis dan teoritis, diantaranya:

1. Manfaat Praktis

- a. Hasil analisis menggunakan model IndoBERT memberikan wawasan bagi pengembang sistem analitik berbasis kecerdasan buatan untuk mendukung evaluasi kebijakan publik, khususnya kebijakan Tapera, agar lebih responsif terhadap kebutuhan masyarakat.
- b. Meningkatkan kesadaran tentang pentingnya penerapan teknologi pengolahan bahasa alami (*Natural Language Processing*) dalam analisis data media sosial untuk membantu proses pengambilan keputusan yang berbasis data di sektor publik.

2. Manfaat Teoritis

- a. Menjadi referensi bagi penelitian-penelitian di bidang data *science*, khususnya dalam pengembangan dan implementasi model NLP berbasis IndoBERT untuk analisis kebijakan publik yang menggunakan bahasa Indonesia.
- b. Memberikan kontribusi pada pengembangan teori dan aplikasi teknologi NLP dalam analisis narasi

media sosial, yang dapat mendukung berbagai domain, seperti pemetaan opini publik, analisis sentimen, dan penyusunan rekomendasi kebijakan berbasis data.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

1. Kebijakan Tapera

Tabungan Perumahan Rakyat (TAPERA) merupakan kebijakan dalam bidang perumahan yang sudah lama dirancang oleh pemerintah Indonesia sebagai upaya untuk meningkatkan akses masyarakat terhadap kepemilikan rumah, namun baru dapat diwujudkan secara formal pada tahun 2018 dengan disahkannya Undang-Undang Nomor 4 Tahun 2016 tentang Tabungan Perumahan Rakyat (Pramudia et al., 2024). Tapera dirancang dengan sistem tabungan wajib dengan tujuan penyediaan biaya untuk perumahan yang terjangkau bagi masyarakat yang memiliki penghasilan rendah (Tania et al., 2021).

Berdasarkan Pasal 1 Angka 1 Undang-Undang Nomor 4 Tahun 2016, Tapera adalah adalah simpanan periodik peserta yang digunakan untuk pembiayaan perumahan atau dikembalikan dengan hasilnya setelah kepesertaan berakhir. Sesuai Pasal 1 Angka 3, peserta mencakup WNI dan WNA pemegang visa kerja minimal

enam bulan. Program ini wajib bagi seluruh pekerja, termasuk ASN, TNI, Polri, pejabat negara, serta pekerja di BUMN, BUMD, BUMDes, dan swasta, baik yang bergaji maupun tidak. (Pasal 7 PP 25/2020).

Berdasarkan Peraturan Pemerintah Nomor 25 Pasal 15 Tahun 2020, besaran simpanan Tapera yang harus dibayarkan ialah sebesar dari 3% dari gaji atau penghasilan pekerja. Untuk peserta pekerja, 2,5% ditanggung oleh pekerja itu sendiri dan 0,5% ditanggung oleh pemberi kerja. Sedangkan untuk pekerja mandiri ditanggung semuanya sendiri. Berikutnya dana tersebut akan dikelola oleh Badan Pengelola Tabungan Perumahan Rakyat (BP Tapera) yang kemudian diperuntukan sebagai berikut :

1. Memastikan hak atas kehidupan layak, termasuk tempat tinggal dan lingkungan yang sehat.
2. Menyediakan rumah bagi masyarakat berpenghasilan rendah untuk mengurangi kesenjangan kebutuhan dan pembangunan rumah.
3. Mengatasi masalah pembiayaan perumahan, seperti daya beli, ketersediaan dana, akses, dan keberlanjutan pembiayaan.
4. Menawarkan pembiayaan rumah jangka panjang dengan suku bunga terjangkau bagi masyarakat berpenghasilan rendah.

Pada pasal 70, dijelaskan bahwa peraturan pemerintah tersebut mulai berlaku pada tanggal diundangkan, yakni 20 Mei 2024. Dengan kata lain, program Tapera saat ini sudah disahkan oleh pemerintah. Perusahaan wajib mendaftarkan karyawannya paling lambat 7 tahun sejak PP tentang Penyelenggaraan Tapera ditetapkan. Kebijakan Penyelenggaraan Tapera sendiri pertama kali ditetapkan pada 20 Mei 2020 melalui PP 25 Tahun 2020. Artinya, perusahaan wajib mendaftarkan pegawainya dalam program Tapera paling lambat pada 20 Mei 2027.

2. Twitter

Twitter merupakan salah satu platform media sosial yang populer di Indonesia yang menyediakan layanan jejaring sosial dan blog ringkas (*microblogging*) yang digunakan sebagai sarana untuk bertukar pesan antar individu dengan individu lainnya (Pradana, 2024). Twitter dapat menyebarkan opini atau cuitan atau dikenal dengan istilah *tweet* secara real time dengan batas maksimal sebanyak 280 karakter yang dapat berupa teks, foto, video atau tautan (Dinilah, 2021). Hal ini memungkinkan *tweet* menjadi sarana untuk mendapatkan informasi secara cepat dan efisien.

Pada media sosial lain, cukup sulit untuk mendapatkan data secara terbuka karena berbedanya kebijakan privasi antar media sosial. Sedangkan di Twitter, data dapat diambil dengan mudah. Data yang diambil berupa opini dari sebuah *tweet* yang berbentuk teks bebas yang tidak terstruktur dan tidak berstandar. Selain itu, Twitter juga cocok untuk data mining karena format datanya mudah dan cocok untuk dianalisis, peraturan data yang fleksibel dan mudah di akses bagi pengguna (Wandani et al., 2021).

3. Analisis Sentimen

Analisis sentimen merupakan proses pengolahan bahasa alami untuk menganalisis dan mengkategorikan pendapat atau opini yang terdapat di dalam suatu kalimat yang bertujuan untuk mengetahui persepsi penulis terhadap suatu topik tertentu, produk, layanan atau isu aktual sehingga dapat digolong menjadi sentimen positif, negatif atau netral (Zein et al., 2022). Analisis sentimen pada *tweet* di Twitter dapat ditandai dengan berbagai cara. Jika didalam *tweet* mengandung *polar word* atau kata yang berlawanan, maka *tweet* tersebut dapat digolongkan menjadi sentimen positif atau sentimen negatif. Jika didalam *tweet* tidak

mengandung polar word, maka dapat digolongkan menjadi sentimen netral.

Berikut merupakan tingkatan yang dilakukan dalam analisis sentimen sebagai berikut (Wandani et al., 2021):

1. Level Dokumen (*Document Level*):

Pada tingkat ini adalah mengklasifikasi apakah seluruh isi dokumen mengekspresikan sentimen positif atau negatif.

2. Level Kalimat (*Sentence Level*):

Pada tingkat ini, analisis sentimen menentukan apakah setiap kalimat menyatakan pendapat positif, negatif, atau netral.

3. Level Aspek (*Aspect Level*):

Pada tingkat ini, analisis menentukan sentimen berdasarkan aspek tertentu dalam kalimat, seperti pada kalimat '*makanannya enak tapi pelayanannya kurang bagus*', aspek '*makanan*' memiliki sentimen positif, sedangkan '*pelayanan*' negatif.

4. Easy Data Augmentation (EDA)

Exploratory Data Analysis (EDA) merupakan salah satu teknik augmentasi teks yang dapat digunakan untuk mengatasi ketidakseimbangan kelas dalam suatu dataset

(Wei & Zou, 2019). Metode ini diperkenalkan oleh Wei dan Zou (2019) dengan tujuan untuk meningkatkan variasi data tanpa mengubah makna aslinya, sehingga dapat membantu dalam meningkatkan performa model pembelajaran mesin, khususnya dalam tugas klasifikasi teks.

Dalam penerapan EDA terdapat empat teknik utama yang digunakan augmentasi data, yaitu sebagai berikut :

1. *Synonym Replacement (SR)*

Mengganti beberapa kata dalam teks dengan sinonimnya yang diperoleh dari kamus sinonim, seperti WordNet.

2. *Random Insertion (RI)*

Menyisipkan kata-kata baru secara acak ke dalam teks berdasarkan sinonim dari kata-kata yang sudah ada.

3. *Random Swap (RS)*

Menukar posisi dua kata secara acak dalam kalimat untuk menghasilkan variasi baru.

4. *Random Deletion (RD)*

Menghapus beberapa kata secara acak dari teks untuk menguji seberapa penting kata-kata tersebut dalam klasifikasi.

Selain berguna dalam menangani dataset yang kecil dan tidak seimbang, EDA juga dapat meningkatkan generalisasi model terhadap data baru, mengurangi risiko *overfitting*, dan memperkaya representasi teks dalam skenario dengan keterbatasan data. Teknik ini sering digunakan dalam pemrosesan bahasa alami (NLP) sebagai alternatif dari metode augmentasi berbasis pembelajaran mendalam yang lebih kompleks, seperti *back-translation* atau *word embedding-based augmentation* (Shorten et al., 2021).

Untuk menentukan berapa banyak kata yang akan diganti dengan sinonim, disisipkan, ditukar, atau dihapus dalam satu kalimat, digunakan parameter α (*alpha*). Parameter ini mengatur seberapa banyak perubahan yang dilakukan pada kalimat tersebut. Jumlah perubahan kata, yang dilambangkan dengan n , dihitung berdasarkan panjang kalimat (l) dan nilai parameter α . Secara matematis, hubungan ini dapat dirumuskan sebagai berikut:

$$n = \alpha \times l \quad (2.1)$$

Semakin besar nilai α , semakin banyak perubahan yang terjadi pada kalimat tersebut. Dengan kata lain, nilai α yang tinggi akan menghasilkan lebih banyak modifikasi pada kalimat, sedangkan nilai α yang rendah akan menghasilkan perubahan yang lebih sedikit. Selain

α , terdapat juga parameter *naug* (*Number of Augmented Sentences*) yang digunakan untuk menentukan berapa banyak kalimat augmentasi yang akan dihasilkan untuk setiap kalimat asli.

5. Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan salah satu cabang dari *Artificial Intelligence* (AI) yang memungkinkan komputer dapat memahami serta mengolah teks seperti yang dilakukan manusia dalam berbagai Bahasa (Amien, 2023). NLP berfokus pada pemahaman cara manusia berkomunikasi sehari-hari. Tujuannya adalah menciptakan metode dan perangkat yang memungkinkan komputer untuk mengerti dan berinteraksi dengan bahasa manusia secara wajar. Dengan pemahaman ini, sistem komputer dapat melaksanakan berbagai tugas linguistik sesuai kebutuhan pengguna. Proses ini melibatkan pengumpulan dan analisis pengetahuan tentang struktur dan penggunaan bahasa manusia, yang kemudian diterapkan untuk mengembangkan teknologi yang mampu memproses dan merespons bahasa alami secara efektif.

NLP dapat diintegrasikan dengan berbagai teknologi seperti *machine learning*, *deep learning*,

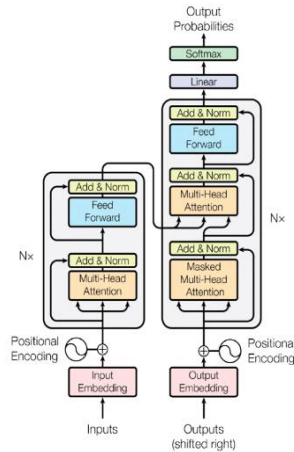
statistik, dan *computational linguistic*. Tujuannya adalah untuk mencapai pemahaman, interpretasi, dan manipulasi bahasa manusia secara komprehensif. Perkembangan signifikan dalam NLP didorong oleh beberapa faktor, terutama ketersediaan berlimpah dokumen digital di internet. Situasi ini menciptakan kebutuhan akan teknologi yang mampu memproses dan menganalisis data dalam jumlah besar, serta melakukan komputasi kompleks untuk memahami dan mengekstrak informasi dari korpus dokumen yang masif (Gangal et al., 2023).

NLP berfungsi sebagai instrumen untuk mengolah data buatan manusia menjadi informasi yang bermakna. *Output* dari NLP sangat bermanfaat dalam analisis dan text mining, memanfaatkan pola komunikasi manusia untuk menghasilkan insights yang berharga (Torfi et al., 2020). Aplikasi NLP sangat beragam, termasuk deteksi spam pada email, sentiment analysis, peringkasan otomatis teks kompleks, pemahaman dan terjemahan pesan suara, serta ekstraksi wawasan dari data teks dalam jumlah besar. Kemampuan NLP dalam memproses bahasa manusia secara natural menjadikannya teknologi kunci di era informasi digital saat ini.

6. Transformers

Transformers merupakan terobosan dalam desain jaringan saraf tiruan yang mengatasi keterbatasan dalam pemahaman konteks jarak jauh dan pemodelan relasi antar kata dalam teks panjang (Vaswani et al., 2017). Arsitektur ini telah memberikan dampak signifikan pada perkembangan model NLP terkini. Dirancang untuk mengolah data sekuensial seperti teks atau audio, *Transformers* telah menunjukkan keunggulannya dalam berbagai tugas, termasuk terjemahan mesin dan pengenalan suara.

Inovasi utama *Transformers* ialah mekanisme *self-attention* yang canggih. Fitur ini memungkinkan model untuk menentukan tingkat kepentingan kata-kata dalam teks *input* saat menghasilkan representasi yang lebih kaya. Berikut merupakan arsitektur yang terdapat pada *Transformers*.



Gambar 2. 1. Arsitektur *Transformer*

Arsitektur *Transformers* terdiri dari dua komponen utama yaitu *encoder* dan *decoder*. *Encoder* terdiri dari beberapa lapisan yang secara bertahap dan berulang memproses masukan, sementara *decoder* juga memiliki struktur berlapis serupa yang bekerja dengan keluaran dari *encoder*. Pendekatan ini memungkinkan *Transformers* untuk menangani berbagai tugas transduksi dengan efektif (Hovipah et al., 2023). Selain *encoder* dan *decoder*, arsitektur *Transformers* memiliki komponen penting lainnya yang disebut *positional encoding*. Fungsi utama dari *positional encoding* adalah untuk memberikan informasi urutan kepada model. Hal ini penting karena *Transformers* tidak memproses *input* secara berurutan seperti model rekuren tradisional (Hidayat & Maharani, 2022).

Untuk mengimplementasikan *positional encoding*, *Transformers* mengadopsi pendekatan matematika. Model ini memanfaatkan fungsi trigonometri, khususnya sinus dan kosinus, untuk menghasilkan representasi posisi yang unik. *Sin* dan *cos* digunakan bersamaan untuk dimensi yang berpasangan dalam *vektor encoding*. Untuk setiap posisi, dimensi genap menggunakan *sin*, sedangkan dimensi ganjil menggunakan *cos*. Penggunaan fungsi trigonometri ini memungkinkan model untuk memahami urutan dan posisi relatif setiap elemen dalam sekuens *input*. Berikut rumus sinus dan kosinus untuk *positional encoding* dalam *Transformers*:

Rumus Sinus :

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_model}}\right) \quad (2.2)$$

Rumus Kosinus :

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_model}}\right) \quad (2.3)$$

Keterangan : PE : *Positional Encoding*

pos : posisi

i : dimensi ke-i

d_model : dimensi model

Untuk mengatasi keterbatasan model sebelumnya, *Transformers* menggantikan proses sekuensial RNN (*Recurrent Neural Network*) dengan mekanisme self-attention. *Self-attention* merupakan mekanisme dari

attention memanfaatkan posisi setiap kata untuk menganalisis hubungan antar kata dalam kalimat. Setiap lapisan *encoder* pada *Transformers* terdiri dari komponen *self-attention* dan *feed-forward*. Berbeda dengan CNN yang umumnya mereduksi fitur di tiap lapisan, *Transformers* mempertahankan panjang *output* yang sama dengan panjang *input* (Vaswani et al., 2017).

Multi-head self-attention merupakan komponen utama penyusun arsitektur *Transformers*. *Multi-head self-attention* menganggap *input* sebagai pasangan kunci-nilai (key dan value), dinotasikan sebagai (K, V), dengan masukannya yaitu kueri dan kunci dimensi d_k (panjang urutan *input*) yang prosesnya berlangsung di *encoder*. Selanjutnya di *decoder*, fungsi *attention* dihitung menggunakan sekumpulan queri yang diorganisir dalam matriks Q. *Output* dihasilkan dengan memetakan Q terhadap K dan V. Hasil dari proses ini adalah rangkaian vektor *output* yang memiliki korespondensi 1:1 dengan token *input* pada indeks yang sama (Juarto & Yulianto, 2023). Proses perhitungan *output* melibatkan penjumlahan nilai-nilai yang telah diberi bobot, dimana bobot untuk setiap nilai ditentukan oleh fungsi kompatibilitas antara query dan key yang bersesuaian. Adapun rumusnya sebagai berikut :

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.4)$$

Keterangan : Q : Query

K : Key

V : Value

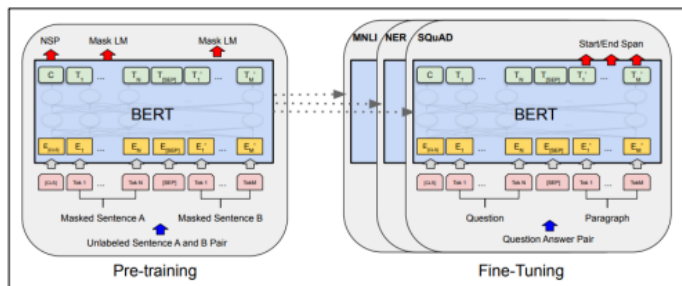
d_k : dimensi dari vektor key

Dalam implementasi *Transformers*, rumus ini diterapkan beberapa kali secara paralel dalam konteks *multi-head attention*, memungkinkan model untuk fokus pada aspek-aspek berbeda dari *input* secara bersamaan.

7. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers atau yang biasa dikenal BERT, merupakan evolusi dari model *Transformers*. Arsitektur BERT dibentuk dengan mengadopsi lapisan *encoder* dari *Transformers* (Rayyan, 2022). BERT hadir sebagai solusi atas keterbatasan teknik-teknik sebelumnya dalam memanfaatkan representasi yang telah dilatih. Salah satu kendala utama pada model-model terdahulu ialah sifatnya yang unidireksional, yang membatasi fleksibilitas arsitektur selama tahap *pre-training*. Sebagai contoh, OpenAI GPT menggunakan pendekatan *left-to-right*, dimana setiap token hanya dapat memperhatikan token-token sebelumnya dalam lapisan

self-attention Transformer (Hafidhigbal, 2024). Inovasi utama BERT terletak pada kemampuannya untuk melatih representasi teks secara mendalam dan bidireksional, dengan mempertimbangkan konteks dari kedua arah yaitu kiri dan kanan di seluruh lapisan modelnya, bahkan ketika bekerja dengan teks yang tidak berlabel (Devlin et al., 2019). BERT menerapkan *self-attention* dalam lapisan Transformer *encoder* melalui proses *pre-training* dan *fine-tuning*. Berikut merupakan gambaran proses *pre-training* dan *fine-tuning*.



Gambar 2. 2. Ilustrasi *Pre-training* dan *Fine-tuning*

7.1. Pre-training

Tahap *pre-training* bertujuan untuk membangun dasar pengetahuan model BERT yang awalnya belum memiliki pemahaman khusus, misalnya tentang teks (Setiawan et al., 2023). Pada tahap ini, *self-attention* dimanfaatkan untuk mempelajari representasi kata yang mampu menangkap keterkaitan antar kata dalam suatu

kalimat. Hal ini dicapai melalui perhitungan attention diantara kata-kata dalam kalimat dengan mempertimbangkan konteks kata-kata lainnya.

Dalam proses *pre-training*, BERT mengandalkan dua tugas utama untuk melatih modelnya. Pertama, *Masked Language Modeling* (MLM) bekerja dengan menyembunyikan beberapa kata dalam kalimat dan meminta model untuk memprediksi kata-kata tersebut. Token [MASK] digunakan untuk menyamarkan kata-kata tertentu selama proses MLM. Kedua, *Next Sentence Prediction* (NSP) melibatkan penyajian dua kalimat kepada model, yang kemudian harus menentukan apakah kalimat-kalimat tersebut berurutan (Devlin et al., 2019). Kedua tugas ini memanfaatkan mekanisme *self-attention* untuk menghitung hubungan perhatian antar kata dalam kalimat, dengan mempertimbangkan *token embeddings*, *segment embeddings*, dan *position embeddings*.

Ketiga embedding tersebut memiliki tugas yang berbeda. *Token Embeddings* merepresentasikan makna setiap kata dalam kalimat melalui vektor unik yang dipelajari selama pelatihan, memungkinkan model memahami arti

kata-kata dalam *input*. *Segment Embeddings* digunakan untuk membedakan segmen teks yang berbeda, seperti dalam tugas tanya-jawab atau inferensi bahasa alami, dengan menambahkan embedding khusus ke setiap token dari segmen yang berbeda. *Position Embeddings* memberikan informasi tentang posisi setiap token dalam urutan *input*, yang penting karena model *transformer* seperti BERT tidak memiliki pemahaman bawaan tentang urutan. Ini dilakukan dengan menggunakan vektor yang dipelajari untuk setiap posisi potensial dalam batas panjang urutan maksimum yang dapat diterima model. (Kurniasih & Manik, 2022).

<i>Input</i>	[CLS]	acting	was	not	good	beautiful	visual	and	songs	[SEP]	[PAD]
<i>Token Embedding</i>	E[CLS]	Eacting	Ewas	Enot	Egood	Ebeautiful	Evisual	Eand	Esongs	E[SEP]	E[PAD]
<i>Segment Embedding</i>	+	+	+	+	+	+	+	+	+	+	+
<i>Position Embedding</i>	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10

Gambar 2. 3. Representasi *Input* BERT

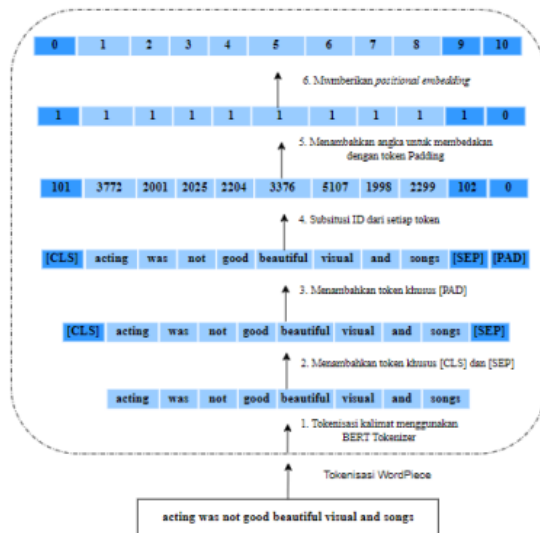
Sebelum proses *training* dimulai, data perlu dikonversi ke format yang dapat diproses oleh BERT menggunakan *tokenizer*. Proses tokenisasi ini mengubah kata menjadi token yang dapat diinterpretasi oleh model, dengan memanfaatkan

kosakata (*vocabulary*) spesifik berbasis model WordPiece.

BERT memiliki ketentuan bahwa setiap *input* kalimat harus memiliki panjang yang seragam, dengan batas maksimum 512 kata (Devlin et al., 2019). Batasan ini muncul karena *encoder Transformers* hanya dapat menghasilkan *output* dengan dimensi 512 kata. Ketika panjang kalimat *input* kurang dari 512 kata, diperlukan penambahan token. Sebaliknya, kalimat yang melebihi 512 kata akan mengalami pemotongan.

Dalam proses tokenisasi BERT, terdapat beberapa token khusus yang berfungsi untuk menandai hubungan antar kata maupun antar kalimat. Kata-kata yang tidak terdapat dalam *vocabulary* akan ditandai sebagai *unknown* (UNK). Setiap kalimat diawali dengan token [CLS] yang berfungsi sebagai penanda awal kalimat. Token [SEP] ditempatkan di akhir kalimat untuk memisahkan satu kalimat dengan kalimat berikutnya. Sementara itu, token [PAD] digunakan untuk mengisi kalimat yang memiliki panjang di bawah batas yang ditentukan (Karimah & Baita, 2024).

Proses tahapan tokenisasi akan menghasilkan *input* pada BERT. Berikut merupakan gambar keseluruhan proses tokenisasi menggunakan BERT *Tokenizer*.



Gambar 2. 4. Proses Tokenisasi BERT *Tokenizer*

Rangkaian langkah yang disebutkan sebelumnya menggambarkan cara *input* diolah untuk digunakan oleh model BERT. Setelah persiapan ini, data *input* akan diproses oleh BERT, kemudian memproses *input* melalui serangkaian lapisan *encoder*. Setiap lapisan *encoder* ini mengandung dua komponen utama yaitu mekanisme *self-attention* dan jaringan *feed-forward*. Proses ini tidak hanya dilakukan sekali,

melainkan diulang sebanyak 12 kali, membentuk tumpukan 12 lapisan *encoder* (Pradana, 2024).

Setelah pemrosesan melalui rangkaian *encoder*, setiap token dalam *input* mengalami transformasi menjadi representasi vektor. Representasi ini dikenal sebagai vektor *hidden layer*, yang memiliki dimensi 768. Dalam konteks tugas klasifikasi menggunakan model berbasis BERT seperti IndoBERT, perhatian khusus diberikan pada token spesial [CLS]. Representasi vektor dari token [CLS] ini yang akan dimanfaatkan sebagai masukan utama untuk lapisan klasifikasi (Karimah & Baita, 2024).



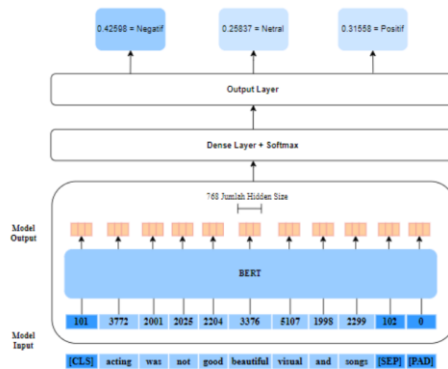
Gambar 2. 5. *Input dan Output dalam BERT*

Pre-training BERT menghasilkan model dengan bobot terlatih dan representasi vektor kata yang kaya akan konteks linguistik. Selama proses ini, BERT mengembangkan pemahaman mendalam tentang hubungan antar kata dalam

konteks bahasa, menghasilkan representasi kata yang kompleks dan peka konteks. Model yang dihasilkan ini menjadi fondasi kuat untuk berbagai aplikasi pemrosesan bahasa alami. (Rahmatullah, 2021).

7.2. Fine-Tuning

Fine-tuning merupakan proses adaptasi model untuk tugas spesifik. Dalam penelitian kali ini fine-tuning digunakan untuk melakukan klasifikasi teks. Tahap fine-tuning melibatkan pelatihan model yang sudah melalui *pre-training*.



Gambar 2. 6. Ilustrasi Proses Klasifikasi BERT

Dalam proses *fine-tuning* terdapat penambahan *Dense Layer* (*Fully Connected Layer*). *Dense Layer* merupakan lapisan di jaringan saraf yang terhubung penuh, dimana setiap *neuron* dalam lapisan tersebut terhubung ke setiap *neuron*

di lapisan sebelumnya dan lapisan berikutnya. Hubungan penuh ini berarti bahwa setiap unit dalam lapisan menerima *input* dari semua unit di lapisan sebelumnya dan mengirim *output* ke semua unit di lapisan berikutnya.

Cara kerja lapisan ini yaitu dengan menerima vektor *input*, yang bisa berasal dari berbagai sumber, seperti *output* dari lapisan lain atau dari model yang telah dilatih sebelumnya. Misalnya, *output* dari lapisan BERT biasanya berupa vektor berdimensi tinggi, biasanya 768 dimensi dari token [CLS]. Kemudian setiap koneksi dalam *dense layer* memiliki bobot (*weight*), yang merupakan nilai numerik yang mewakili kekuatan hubungan antara *neuron*. Setiap *neuron* memiliki bias, yaitu nilai yang ditambahkan untuk menggeser hasil total agar model bisa lebih fleksibel dalam belajar. Misalnya, jika vektor *input* berukuran 768 dimensi dan lapisan *dense* memiliki 3 *neuron*, maka akan ada 768 bobot untuk masing-masing dari 3 *neuron*, ditambah masing-masing *neuron* akan memiliki bias.

Setiap *neuron* di *dense layer* menghitung hasil dari kombinasi linier semua *input* yang

masuk. Rumus matematisnya adalah sebagai berikut:

$$z_j = \sum_i (w_{ij} \cdot x_i) + b_j \quad (2.5)$$

Dimana:

- z_j adalah hasil total untuk *neuron* ke-j.
- w_{ij} adalah bobot dari *neuron* ke-i menuju *neuron* ke-j.
- x_i adalah nilai *input* dari *neuron* ke-i.
- b_j adalah bias untuk *neuron* ke-j

Setelah mendapatkan *output* dari dense layer, fungsi *softmax* diterapkan untuk mengubah nilai *logits* tersebut menjadi probabilitas. *Softmax* merupakan fungsi aktivasi yang digunakan dalam jaringan saraf untuk tugas klasifikasi *multi-class*. Setiap kelas diberikan probabilitas antara 0 dan 1, dan total dari semua probabilitas tersebut selalu berjumlah 1. Misalkan ada n kelas, dan $z_1, z_2, \dots, z_n$ adalah *output logits* dari lapisan terakhir (*dense layer*) jaringan saraf. *Softmax* untuk kelas ke- i dihitung sebagai berikut:

$$P(y = i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (2.6)$$

Dimana :

- $P(y = i)$ adalah probabilitas bahwa *input* termasuk dalam kelas ke- i .

- z_i adalah logit (nilai *output*) untuk kelas ke- i .
- e adalah basis logaritma natural (sekitar 2.718).
- $\sum_{j=i}^n e^{z_j}$ adalah total dari eksponensial dari semua logits, yang digunakan untuk normalisasi.

Setelah probabilitas diperoleh kemudian kelas dengan probabilitas tertinggi lah yang dipilih.

8. IndoBERT

IndoBERT merupakan inovasi penting dalam bidang Pemrosesan Bahasa Alami (NLP) untuk Bahasa Indonesia. Meskipun Bahasa Indonesia termasuk bahasa yang banyak digunakan di internet, perkembangan riset NLP untuk bahasa ini masih terbatas, terutama karena keterbatasan sumber daya komputasi. Untuk mengatasi hal ini, IndoNLU mengembangkan IndoBERT, adaptasi model BERT yang dilatih menggunakan lebih dari 220 juta kata dalam Bahasa Indonesia, bersumber dari Wikipedia Indonesia (74 juta kata), artikel berita seperti Kompas, Tempo, dan Liputan6 (55 juta kata), serta Indonesia *Web Corpus* (90 juta kata) (Koto et al., 2020).

IndoBERT memiliki 4 varian model yang berbeda, yaitu IndoBERT-BASE, IndoBERT-liteBASE, IndoBERT-

LARGE dan IndoBERT-liteLARGE. Dalam proses pelatihannya IndoBERT menggunakan dataset Indo4B dan TPUv3-8, dilaksanakan dalam dua tahap. Tahap pertama menggunakan *input* data berukuran 128, dengan durasi pelatihan bervariasi untuk setiap varian model yaitu 35 jam untuk IndoBERTBASE, 38 jam untuk IndoBERT-liteBASE, 89 jam untuk IndoBERTLARGE, dan 134 jam untuk IndoBERT-liteLARGE. Pada tahap kedua, jumlah sekuens ditingkatkan menjadi 512, memerlukan waktu pelatihan 9, 23, 32, dan 45 jam untuk masing-masing model tersebut. Selama kedua tahap, IndoBERTBASE menggunakan *batch size* 256 dan *learning rate* $2e-5$. Sementara itu, IndoBERTLARGE pada tahap pertama menggunakan *batch size* 256 dan *learning rate* $1e-4$, yang kemudian dimodifikasi pada tahap kedua

Model	Maximum Sequence Length = 128				Maximum Sequence Length = 512			
	Batch Size	Learning Rate	Steps	Duration (Hr.)	Batch Size	Learning Rate	Steps	Duration (Hr.)
IndoBERT-liteBASE	4096	0.00176	112.5 K	38	1024	0.00088	50 K	23
IndoBERT _{BASE}	256	0.00002	1 M	35	256	0.00002	68 K	9
IndoBERT-liteLARGE	1024	0.00044	500 K	134	256	0.00044	129 K	45
IndoBERT _{LARGE}	256	0.0001	1 M	89	128	0.00008	120 K	32

Gambar 2. 7. Tabel *Summary* Pelatihan IndoBERT menjadi *batch size* 128 dan *learning rate* $8e-5$ (Wilie et al., 2020).

IndoBERT dirancang untuk memberikan solusi efektif bagi berbagai tugas pemahaman bahasa alami dalam konteks Bahasa Indonesia. Model ini menunjukkan kemampuan yang beragam, mulai dari

klasifikasi kalimat tunggal hingga pelabelan urutan kalimat berpasangan, mencakup berbagai tingkat kompleksitas. Hasil penelitian menunjukkan bahwa IndoBERT mampu mengungguli sebagian besar model pra-latih yang ada, memberikan kemajuan signifikan dalam pemrosesan dan pemahaman teks Bahasa Indonesia (Wilie et al., 2020). Dengan demikian, IndoBERT menjadi landasan kuat untuk kemajuan NLP dalam konteks Bahasa Indonesia, menawarkan potensi besar untuk aplikasi dan penelitian lebih lanjut.

Dalam implementasi model IndoBERT dibutuhkan beberapa *hyperparameter* pada saat pre-training dan fine-tuning. Nilai *hyperparameter* yang optimal untuk setiap task pada NLP selalu berbeda. Berdasarkan penelitian yang dilakukan oleh (Devlin et al., 2019) merekomendasikan *hyperparameters* yang menunjukkan performa yang optimal untuk tugas spesifik, yaitu:

1. *Batch size*: 16 dan 32
2. *Epoch*: 2, 3, dan 4
3. *Learning rate* (AdamW): $5e-5$, $3e-5$, dan $2e-5$

9. Confusion Matrix

Confusion Matrix adalah tabel evaluasi yang membandingkan hasil prediksi model dengan nilai

aktual dalam empat kategori yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Alat ini membantu menilai akurasi model klasifikasi. Untuk kasus *multi-class*, *Confusion Matrix Multiclass* digunakan untuk mengukur kinerja model dalam lebih dari dua kategori. Menurut Grandini et al. (2020), alat ini sangat bermanfaat dalam analisis sentimen, dimana model diuji menggunakan dataset pelatihan dan validasi dengan label yang sudah diketahui. Tabel ini memberikan gambaran menyeluruh tentang bagaimana performa model dalam mengklasifikasikan berbagai kategori sentimen, memungkinkan evaluasi yang lebih mendalam terhadap akurasi dan kesalahan klasifikasi di setiap kelas. Berikut merupakan contoh *confusion matrix* untuk *multi-class classification*.

Table 2. 1. *Confusion Matrix 3x3*

Classes		Predicted Classification		
		X	Y	Z
Actual Classification	X	TP	FN	FN
	Y	FN	TP	FN
	Z	FN	FN	TP

Dalam tabel ini X diibaratkan sebagai *class* negatif, Y diibaratkan sebagai *class* netral dan Z diibaratkan

sebagai *class* positif. Berikut keterangan dari istilah pada tabel diatas :

1. TP (*True Positive*): data yang diklasifikasikan dengan benar ke dalam kelas yang sesuai.
2. FN (*False Negative*): dari suatu kelas yang salah diklasifikasikan sebagai kelas lain.

Dalam evaluasi sistem klasifikasi *multi-class*, *confusion matrix* berperan sebagai alat utama untuk menghitung berbagai matrik kinerja model. Beberapa matrik penting yang dapat dihitung dari *confusion matrix* yaitu *accuracy*, *precision*, *recall*, dan *f1-score*.

Accuracy merupakan indikator yang mengukur proporsi prediksi yang tepat dari keseluruhan data yang dievaluasi (Grandini et al., 2020). *Accuracy* dihitung menggunakan rumus :

$$Accuracy = \frac{TP}{Total\ Data} \quad (2.7)$$

Precision merupakan matrik yang berfokus pada keakuratan model dalam membandingkan antara jumlah data yang dengan tepat diidentifikasi sebagai positif (*true positives*) dengan total keseluruhan data yang diprediksi positif oleh model, termasuk yang sebenarnya negatif namun salah diklasifikasikan sebagai positif (*false positives*) (Grandini et al., 2020). *Precision* dihitung menggunakan rumus:

$$Precision = \frac{TP}{TP+FP} \quad (2.8)$$

Recall merupakan matrik menilai sejauh mana model berhasil mengenali kasus positif dari keseluruhan kasus positif yang ada. Dalam perspektif analitis, recall dapat dipahami sebagai perbandingan antara jumlah kasus positif yang berhasil diidentifikasi dengan tepat oleh model (*true positives*) terhadap total keseluruhan kasus positif yang sebenarnya ada dalam dataset, termasuk yang gagal diidentifikasi oleh model (*false negatives*) (Grandini et al., 2020). *Recall* dihitung menggunakan rumus :

$$Recall = \frac{TP}{TP+FN} \quad (2.9)$$

F1-score merupakan matrik yang dirancang untuk mencerminkan keseimbangan antara *precision* dan *recall*, memberikan gambaran yang lebih holistik tentang performa model. Secara matematis, *F1-score* dapat dipahami sebagai rata-rata harmonik dari *precision* dan *recall*. Karakteristik khusus dari rata-rata harmonik ini adalah kecenderungannya untuk memberikan hasil yang lebih rendah jika terdapat ketidakseimbangan yang signifikan antara *precision* dan *recall*. Akibatnya, *F1-score* mencapai nilai maksimalnya hanya ketika baik *precision* maupun *recall* berada pada level yang sama-sama tinggi (Grandini et al., 2020). *F1-score* dihitung menggunakan rumus :

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.10)$$

B. Kajian Penelitian yang Relevan

Penelitian ini memerlukan referensi sebagai sumber informasi tambahan yang bertujuan untuk mendukung penelitian dan memperoleh perbandingan dengan penelitian-penelitian lainnya. Beberapa penelitian yang memiliki relevansi dengan penelitian ini.

Table 2. 2 Penelitian yang Relevan

No	Judul	Penulis	Algoritma	Hasil
1	Analisis Sentimen Masyarakat Media Sosial Twitter Terhadap Kinerja Pejabat Gubernur DKI Jakarta Menggunakan Model IndoBERT	Lambang Surya Pradana, (2024)	IndoBERT	Penelitian ini menganalisis sentimen masyarakat terhadap kinerja Pejabat Gubernur DKI Jakarta Heru Budi Hartono melalui data Twitter, menggunakan kombinasi analisis eksplorasi data dan algoritma IndoBERT. Data yang digunakan dalam penelitian ini sebanyak 6.648 data yang didapat dari <i>crawling</i> di media sosial twitter. Model IndoBERT-large-p2 yang digunakan menunjukkan performa baik dengan akurasi

				90.5% dan f1-score 90.49%.
2	Klasifikasi Sentimen Terhadap Kebijakan Tapera Menggunakan Komparasi Machine Learning dan SMOTE	Henny Leidiyan , Titik Misriati, Riska Aryanti (2024)	K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multinomial Naïve Bayes (NB), and Decision Tree	Penelitian ini menggunakan analisis sentimen dengan algoritma K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multinomial Naïve Bayes (NB), dan Decision Tree. Data dibagi dalam tiga skenario: 70-30%, 80-20%, dan 90-10% untuk data latih dan uji, dengan penyeimbangan data menggunakan SMOTE. Evaluasi kinerja dilakukan berdasarkan akurasi, precision, recall, dan F1 Score. Hasilnya, SVM menunjukkan performa terbaik dengan akurasi hingga 88% dan F1 Score 85%, diikuti Multinomial Naïve Bayes yang stabil, sementara KNN dan Decision Tree memiliki kinerja lebih rendah.
3	Analisis Sentimen	Muhammad	K-Nearest	Penelitian ini menganalisis

	Masyarakat Terhadap Tapera Pada Media Sosial X Menggunakan Metode K-Nearest Neighbor	Iksan Alfandi, Pitrasac ha Adytia, Wahyuni (2024)	Neighbor	sentimen pengguna Media Sosial X terhadap kebijakan Tapera menggunakan algoritma K-NN (K-Nearest Neighbor). Sebanyak 958 tweet berbahasa Indonesia, dikumpulkan pada 19-23 Juli 2024, dikategorikan ke dalam sentimen positif, netral, dan negatif menggunakan bobot TF-IDF. Pengujian dengan nilai K=3, K=5, dan K=7 menunjukkan akurasi tertinggi 62,75% pada K=3. Hasilnya mengungkapkan bahwa persepsi masyarakat terhadap Tapera cenderung kurang baik.
4	Analisis Sentimen Masyarakat terhadap Kebijakan Tabungan Perumahan Rakyat	Anis Maulidatur Rizqiyah, I Kadek Dwi Nuryana (2024)	Naïve Bayes dan Support Vector Machine	Penelitian ini bertujuan mengklasifikasikan tweet terkait kebijakan Tapera menggunakan algoritma Naïve Bayes dan Support Vector Machine (SVM) untuk

	(Tapera) pada Platform X Menggunakan Algoritma Naïve Bayes Classifier dan Support Vector Machine			menganalisis sentimen masyarakat. Sebanyak 1.280 tweet dikumpulkan dari X, dilabeli menggunakan InSet Lexicon, dan diseimbangkan dengan SMOTE. Klasifikasi dilakukan dengan rasio data latih-uji 80:20, 70:30, dan 60:40, serta dievaluasi menggunakan confusion matrix dan k-fold cross validation. Hasilnya menunjukkan sentimen masyarakat cenderung negatif terhadap Tapera, dengan SVM unggul memiliki akurasi 84% sebelum SMOTE (80:20, kernel linear, C=2) dan meningkat menjadi 93% setelah SMOTE (80:20, kernel rbf, C=3). Sementara itu, Naïve Bayes memiliki akurasi 81% sebelum
--	--	--	--	--

				SMOTE (80:20, Complement, alpha 0.01) dan meningkat menjadi 89% setelah SMOTE (60:40, Complement, alpha 0.1).
5	Indonesia News Classification Using IndoBERT	Budi Juarto, Yulianto	IndoBERT, XLNet, BERT-Multilingual, XLM-Robert, XGBoost, LightGBM, Random Forest	Penelitian ini bertujuan untuk melakukan klasifikasi berita berbahasa Indonesia menggunakan model IndoBERT, Penelitian ini juga membandingkan kinerja IndoBERT dengan model pre-trained lain seperti XLNet, BERT multilingual, dan XLM-Roberta, serta algoritma lain seperti XGBoost, LightGBM, dan Random Forest. Hasilnya, IndoBERT menunjukkan akurasi terbaik sebesar 94,5% serta waktu pelatihan, validasi, dan komputasi yang paling singkat dibandingkan metode lain. Data yang digunakan

				pada penelitian ini berupa dataset berita yang berbahasa Indonesia sebanyak 8.000 baris.
6	Easy Data Augmentation Untuk Data Yang Imbalance Pada Konsultasi Kesehatan Daring	Anisa Nur Azizah, Misbachl Falach Asy'ari, Ifnu Wisma Dwi Prastya, Diana Purwitasari	Easy Data Augmentation (EDA), Latent Dirichlet Allocation (LDA)	Penelitian ini membuktikan bahwa Easy Data Augmentation (EDA) meningkatkan kinerja klasifikasi teks Konsultasi Kesehatan Daring (KKD), dengan model Random Forest mencapai akurasi 88,86%. Dari empat teknik yang diuji (SR, RI, RS, RD), Random Deletion (RD) dengan parameter 0,4 memberikan hasil terbaik. Augmentasi juga meningkatkan prediksi pada kelas minoritas, terutama dengan model AdaBoost yang naik 3,78%. Secara keseluruhan, EDA efektif menangani data tidak seimbang, dengan

				RD sebagai teknik paling optimal.
--	--	--	--	-----------------------------------

Analisis sentimen terhadap kebijakan Tapera telah banyak diteliti, namun penggunaan model IndoBERT masih jarang diterapkan. Penelitian ini mengisi gap tersebut dengan memanfaatkan IndoBERT, yang mampu memahami konteks dan hubungan antar kata lebih baik dibandingkan tools otomatis yang hanya menilai bobot kata secara terpisah. Pendekatan otomatis sering kali menghasilkan pelabelan yang kurang akurat, sehingga penelitian ini menerapkan pelabelan manual oleh pakar bahasa. Dengan metode ini, data yang digunakan lebih akurat dan representatif dalam menggambarkan persepsi publik terhadap kebijakan Tapera.

Selain itu, penelitian ini juga menerapkan *Easy Data Augmentation* (EDA) untuk menangani data yang tidak seimbang dalam analisis sentimen. Dengan teknik seperti *Synonym Replacement* (SR), *Random Insertion* (RI), *Random Swap* (RS), dan *Random Deletion* (RD), EDA dapat meningkatkan kualitas dataset dan kinerja model. Kombinasi antara pelabelan manual oleh pakar dan analisis menggunakan IndoBERT diharapkan memberikan hasil yang lebih mendalam serta berkontribusi signifikan dalam studi analisis narasi kebijakan di media sosial.

BAB III

METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

1. Studi Pustaka

Peneliti melaksanakan kajian literatur komprehensif dengan memanfaatkan berbagai sumber ilmiah seperti buku, jurnal akademik, dan karya tulis ilmiah lainnya. Tujuannya adalah untuk memperdalam pemahaman tentang konsep-konsep kunci, metode analisis, dan berbagai aspek terkait topik penelitian. Selain itu, peneliti juga melakukan pencarian daring menggunakan mesin pencari internet untuk mengakses situs web yang relevan dengan analisis sentimen, penambangan teks, dan metode klasifikasi IndoBERT.

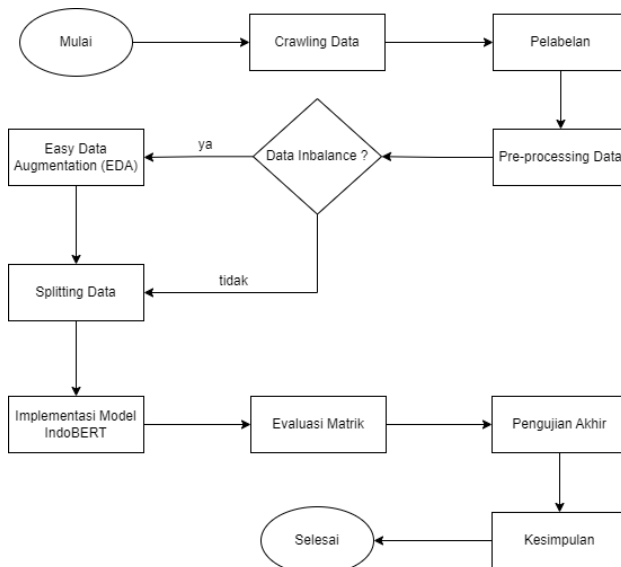
2. Observasi Lapangan Digital

Dalam pendekatan ini, peneliti melakukan pengamatan sistematis terhadap dinamika percakapan di media sosial, khususnya terkait isu yang sedang menjadi pusat perhatian publik. Fokus utama penelitian adalah menganalisis respons pengguna Twitter di Indonesia terhadap kebijakan Tabungan Perumahan Rakyat (TAPERA). Peneliti mengamati secara cermat

pola interaksi, sentimen, dan opini yang berkembang di platform tersebut terkait implementasi kebijakan TAPERA.

B. Tahapan Penelitian

Tahapan penelitian terdiri dari urutan aktivitas sistematis yang harus dijalankan oleh seorang peneliti merancang, melaksanakan, dan mengevaluasi sebuah penelitian. Setiap fase dalam alur ini penting untuk memastikan penelitian dilakukan dengan metode yang terstruktur dan dapat dipertanggungjawabkan. Berikut merupakan diagram alur dari tahapan-tahapan yang akan dilakukan dalam penelitian ini.



Gambar 3. 1. *Flowchart* Alur Penelitian

Penelitian ini dimulai dengan tahap pengumpulan data yang berfokus pada opini dan tanggapan terkait kebijakan Tabungan Perumahan Rakyat (TAPERA) dari media sosial Twitter (X). Proses pengumpulan data dilakukan menggunakan teknik *crawling* data yang memanfaatkan API Twitter dan Tweet Harvest sebagai alat dalam melakukan *crawling*. Metode ini memungkinkan pengambilan data yang efisien dan terstruktur, khususnya untuk mencari dan mengumpulkan *tweet-tweet* yang membahas tentang Tapera.

Setelah data terkumpul secara komprehensif, langkah berikutnya adalah melakukan pelabelan pada data ke dalam tiga kategori yaitu positif, negatif dan netral. Proses pelabelan ini akan dilakukan oleh seorang pakar atau ahli bahasa Indonesia. Pada tahap ini, distribusi label dalam dataset akan dianalisis untuk melihat apakah data seimbang atau tidak. Jika distribusi tidak seimbang, maka akan diterapkan *Easy Data Augmentation* (EDA) guna mengurangi ketimpangan jumlah label. Tetapi sebelum dilakukan proses EDA data harus melewati proses *pre-processing* terlebih dahulu.

Tahap *pre-processing* bertujuan untuk membersihkan data mentah. Proses ini terdiri dari beberapa langkah penting. Dimulai dengan *case folding*, dimana seluruh teks diubah menjadi huruf kecil untuk keseragaman.

Selanjutnya, tahap *cleansing* menghapus elemen-elemen yang tidak diperlukan seperti tanda baca, tautan URL, emoji, dan tagar untuk mengurangi noise dalam teks. Kemudian dilakukan *normalization* untuk mengubah kata-kata tidak baku atau singkatan menjadi bentuk baku sesuai KBBI, terutama yang berkaitan dengan istilah dalam kebijakan perumahan. Setelah ketiga tahap *pre-processing* dilakukan, *remove duplicate* akan diterapkan untuk menghilangkan opini atau *tweet* yang berulang tentang kebijakan Tapera

Setelah tahap *pre-processing*, jika dianalisis terjadi ketimpangan jumlah label, langkah selanjutnya adalah menerapkan *Easy Data Augmentation* (EDA) diterapkan untuk menyeimbangkan distribusi label dengan memperbanyak data pada kelas minoritas. Teknik yang digunakan meliputi *Synonym Replacement* (SR), *Random Insertion* (RI), *Random Swap* (RS), dan *Random Deletion* (RD). Dengan EDA, dataset menjadi lebih seimbang, memungkinkan IndoBERT untuk melatih model secara lebih optimal dan meningkatkan akurasi analisis sentimen terhadap kebijakan Tapera.

Setelah augmentasi data, langkah berikutnya adalah pembagian dataset dengan rasio 70% data digunakan untuk pelatihan, 20% data digunakan untuk validasi dan 10% data dialokasikan sebagai data uji. Pembagian ini memastikan model IndoBERT dapat menghasilkan prediksi sentimen

yang lebih akurat dan dapat diandalkan dalam analisis kebijakan Tapera.

Langkah berikutnya, dataset yang sudah dibagi menjadi data latih, data validasi dan data uji akan masuk ke tahap *tokenizing* yang bertujuan untuk mengubah kalimat menjadi sub-kata yang kemudian akan di dipetakan ke ID numerik atau vektor yang sesuai dalam kosakata model atau biasa disebut dengan proses *encoding*. Kemudian data numerik tersebut digunakan dalam proses pra-training dan fine-tuning dalam IndoBERT. Hasilnya adalah sebuah matrik yang berisi nilai nilai yang penting untuk mengukur akurasi dari model IndoBERT.

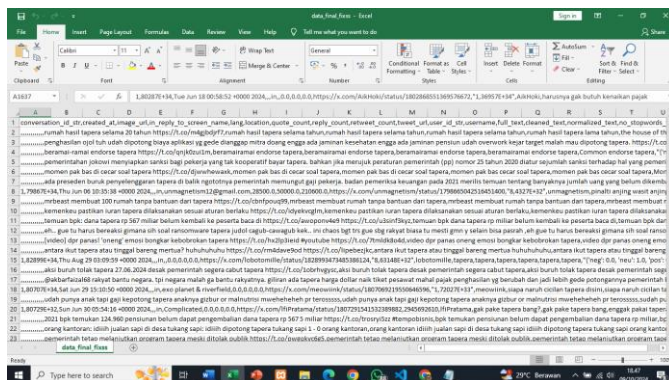
Kemudian, model IndoBERT yang telah dilatih akan diuji lebih lanjut. Model ini akan digunakan untuk melakukan klasifikasi terhadap teks atau kalimat tanpa label yang belum pernah dipelajari sebelumnya. Dengan demikian, kinerja model IndoBERT tidak hanya diukur berdasarkan keakuratannya dalam mengklasifikasikan teks terkait kebijakan Tapera, tetapi juga diuji kemampuannya sebagai alat untuk mengklasifikasikan teks secara acak.

C. Uraian Metodologi

1. Pengambilan Data Twitter

Data dalam penelitian ini berasal dari opini pengguna Twitter mengenai kebijakan Tapera yang

dikumpulkan melalui teknik *crawling* menggunakan Tweet Harvest. Proses *crawling* dilakukan dengan memasukkan kata kunci "tapera" untuk mencari *tweets* yang relevan, termasuk yang menyebutkan atau menggunakan tagar terkait. Hasilnya, diperoleh 2.604 tweet, yang kemudian disimpan dalam file dengan format *csv* sebagai dataset utama untuk analisis selanjutnya. Berikut merupakan contoh hasil *crawling* data.



Gambar 3. 2. Hasil *Crawling* Data

2. Pelabelan Data

Setelah data terkumpul, langkah selanjutnya adalah proses pelabelan untuk memungkinkan model memahami pola dan karakteristik sentimen dalam teks. Pelabelan dilakukan oleh pakar atau ahli bahasa Indonesia yang memahami berbagai aspek bahasa guna memastikan akurasi yang maksimal. Data dikategorikan

ke dalam tiga kelas yaitu negatif, netral, dan positif. Setelah pelabelan, dilakukan analisis distribusi label untuk menentukan apakah data seimbang atau tidak seimbang. Jika distribusi label tidak merata, maka diperlukan penerapan EDA untuk mengatasi ketidakseimbangan dan memastikan model dapat belajar secara optimal.

3. Pre-Processing Data

Tahap ini bertujuan untuk membersihkan data mentah dan mempersiapkannya agar siap untuk proses klasifikasi. Langkah-langkah *pre-processing* diterapkan secara berurutan pada data yang telah diperoleh melalui proses pengambilan sebelumnya. Serangkaian tahapan ini dirancang untuk mengolah data Twitter yang masih mentah menjadi format yang lebih terstruktur dan bersih. Berikut urutan dari tahap *pre-processing* ini yang disesuaikan dengan karakteristik unik data Twitter dan kebutuhan analisis yang akan dilakukan:

3.1. Case Folding

Dalam fase ini, dilakukan proses normalisasi teks dimana seluruh karakter dalam kata, frasa, atau paragraf dikonversi menjadi huruf kecil (*lowercase*). Tujuan utama dari langkah ini adalah

menciptakan keseragaman dalam representasi kata-kata. Berikut merupakan ilustrasi dari *case folding*.

Tabel 3. 1. Ilustrasi *Case Folding*

Input	Output
Untuk apa Tapera kalau presiden terpilih aja berjanji akan bangun 3 juta rumah https://t.co/BJKNw6f3jx	untuk apa tapera kalau presiden terpilih aja berjanji akan bangun 3 juta rumah https://t.co/bjknw6f3jx

3.2. Spell Normalization

Proses ini melibatkan konversi kata-kata non-standar, singkatan, dan istilah slang ke dalam bentuk baku. Tujuannya adalah mengurangi variasi kata dan meningkatkan akurasi analisis. Dalam konteks ini, referensi untuk normalisasi diambil dari kamus yang sering digunakan untuk melakukan normalisasi yaitu '*colloquial-indonesian-lexicon.csv*', '*kbba.txt*', '*slangword.txt*', '*slang.csv*' yang tersedia di GitHub. Berikut merupakan ilustrasi dari *spell normalization*.

Tabel 3. 2. Ilustrasi *Spell Normalization*

Input	Output
untuk apa tapera kalau presiden terpilih aja berjanji akan bangun 3 juta rumah https://t.co/bjknw6f3jx	untuk apa tapera kalau presiden terpilih saja berjanji akan bangun 3 juta rumah https://t.co/bjknw6f3jx

3.3. Cleaned Text

Proses ini berfokus pada penghapusan elemen-elemen yang dianggap tidak relevan atau mengganggu dalam analisis teks. Elemen-elemen tersebut dapat mencakup tanda baca, angka, URL, emoji, dan tagar. Pembersihan ini bertujuan untuk meningkatkan kualitas dan kejelasan data teks. Berikut merupakan ilustrasi dari *cleaned text*.

Tabel 3. 3. Ilustrasi *Cleaned Text*

Input	Output
untuk apa tapera kalau presiden terpilih saja berjanji akan bangun 3 juta rumah https://t.co/bjknw6f3jx	untuk apa tapera kalau presiden terpilih saja berjanji akan bangun juta rumah

4. Easy Data Augmentation (EDA)

Untuk meningkatkan kualitas dan kuantitas dataset, terutama pada kelas minoritas teknik *Easy Data Augmentation* (EDA) diterapkan. EDA merupakan metode augmentasi data teks yang sederhana namun efektif, terdiri dari empat teknik utama yaitu *Synonym Replacement* (SR), *Random Insertion* (RI), *Random Swap* (RS), dan *Random Deletion* (RD). Teknik ini bertujuan untuk menghasilkan variasi kalimat baru yang tetap mempertahankan makna asli dan label sentimennya.

4.1. **Synonym Replacement (SR)**

Pada teknik ini, beberapa kata dalam kalimat diganti dengan sinonimnya untuk menciptakan variasi kalimat. Sinonim diperoleh dari Tesaaurus Bahasa Indonesia yang telah disiapkan. Jumlah kata yang diganti ditentukan oleh parameter α (alpha), yang dihitung sebagai persentase dari panjang kalimat. Berdasarkan rekomendasi dari Wei dan Zou (2019), nilai $\alpha = 0.1$ digunakan, yang berarti sekitar 10% kata dalam kalimat akan diganti. Berikut merupakan contoh dari penerapan *Synonym Replacement*.

Tabel 3. 4. Contoh Hasil *Synonym Replacement*

Kalimat Asli	Kalimat Hasil SR
Kebijakan Tapera sangat membantu masyarakat	Kebijakan Tapera sangat menolong masyarakat

4.2. **Random Insertion (RI)**

Teknik ini menyisipkan sinonim dari kata yang sudah ada ke dalam kalimat pada posisi acak. Jumlah kata yang disisipkan juga diatur oleh parameter $\alpha = 0.1$. Operasi ini membantu meningkatkan variasi kalimat tanpa mengubah makna aslinya. Berikut merupakan contoh dari penerapan *Random Insertion*.

Tabel 3. 5. Contoh Hasil *Random Insertion*

Kalimat Asli	Kalimat Hasil RI
Kebijakan Tapera sangat membantu masyarakat	Kebijakan Tapera sangat amat membantu masyarakat

4.3. Random Swap (RS)

Pada teknik ini, dua kata dalam kalimat ditukar posisinya secara acak. Jumlah swap yang dilakukan ditentukan oleh parameter $\alpha = 0.1$. Teknik ini berguna untuk menciptakan variasi struktur kalimat. Berikut merupakan contoh dari penerapan *Random Swap*.

Tabel 3. 6. Contoh Hasil *Random Swap*

Kalimat Asli	Kalimat Hasil RS
Kebijakan Tapera sangat membantu masyarakat	Tapera Kebijakan sangat membantu masyarakat

4.4. Random Deletion (RD)

Teknik ini menghapus kata dalam kalimat dengan probabilitas $\alpha = 0.1$. Jika kalimat terlalu pendek (hanya terdiri dari satu kata), penghapusan tidak dilakukan. Operasi ini membantu meningkatkan robust model terhadap kalimat yang tidak lengkap. Berikut merupakan contoh dari penerapan *Random Deletion*.

Tabel 3. 7. Contoh Hasil *Random Deletion*

Kalimat Asli	Kalimat Hasil RD
Kebijakan Tapera sangat membantu masyarakat	Kebijakan Tapera membantu masyarakat

Data yang akan melalui proses EDA hanya data yang berkelas minoritas. Hal ini bertujuan untuk menyeimbangkan distribusi kelas dalam dataset, sehingga model klasifikasi tidak bias terhadap kelas mayoritas. Dengan meningkatkan jumlah sampel pada kelas minoritas melalui augmentasi, diharapkan model dapat belajar lebih baik dan menghasilkan prediksi yang lebih akurat untuk semua kelas.

Dalam proses augmentasi, dari keempat teknik EDA akan dipilih secara acak untuk setiap kalimat yang akan diaugmentasi. Pemilihan teknik secara acak ini bertujuan untuk menciptakan variasi yang lebih beragam dalam data augmentasi, sehingga model dapat belajar dari berbagai bentuk kalimat yang berbeda namun tetap mempertahankan makna dan label aslinya.

Parameter *naug* akan disetel ke 2, yang berarti setiap kalimat dari kelas minoritas akan dihasilkan menjadi dua versi augmentasi menggunakan teknik EDA yang dipilih secara acak. Pemilihan nilai ini bertujuan untuk menjaga keseimbangan antara penambahan jumlah data dan kualitas hasil augmentasi. Dengan jumlah augmentasi yang tidak berlebihan, diharapkan

kalimat yang dihasilkan tetap relevan dan tidak terlalu menyimpang dari makna aslinya. Selain itu, menetapkan $naug = 2$ juga membantu mencegah overfitting, yang bisa terjadi jika terlalu banyak kalimat augmentasi dibuat.

5. Splitting Data

Pada fase ini melibatkan pembagian dataset menjadi beberapa subset yang berbeda. Tujuan utama dari pemisahan data ini adalah untuk memungkinkan pelatihan, pengujian, dan validasi model atau algoritma secara efektif. Pada fase ini dataset akan dibagi menjadi 70% untuk pelatihan, 20% untuk validasi dan 10% untuk pengujian. Data pelatihan digunakan untuk mengajarkan model mengenali pola dari data, sedangkan data validasi berfungsi untuk mengevaluasi performa model selama proses pelatihan dan menyetel parameter agar tidak terjadi overfitting. Sementara itu, data pengujian digunakan pada tahap akhir untuk menilai kemampuan model dalam memprediksi data baru yang belum pernah dilihat sebelumnya. Pembagian dilakukan secara acak dengan bantuan *python*. Setiap dataset yang dibagi memiliki proporsi label yang sama agar model dapat belajar secara optimal dari setiap kelas yang ada.

6. Implementasi Model IndoBERT

Dalam implementasi model IndoBERT, melibatkan beberapa proses sebagai berikut :

6.1. Tokenizing

Tokenisasi merupakan teknik pengolahan teks yang membagi sebuah kalimat menjadi bagian-bagian kecil yang bermakna, yang dikenal sebagai token. Proses tokenisasi dilakukan dengan memanfaatkan *BERT Tokenizer*, yang merupakan komponen dari model bahasa BERT. BERT Tokenizer menerapkan teknik WordPiece dalam proses tokenisasinya, yang membagi kata-kata menjadi sub-kata atau token. Kemudian sub-kata atau token tersebut diubah menjadi representasi numerik atau vektor. Tujuannya adalah agar informasi tersebut dapat diinterpretasi dan diproses oleh komputer. Dalam tokenisasi ini melibatkan *token embedding*, *segment embedding* dan *position embedding* dalam prosesnya. Berikut ilustrasi dari tokenizing IndoBERT.

Tabel 3. 8. Ilustrasi *Tokenizing IndoBERT*

Input	untuk apa tapera kalau presiden terpilih saja berjanji akan bangun juta rumah
--------------	---

Tokenized	['untuk', 'apa', 'tapi', '##ra', 'kalau', 'presiden', 'terpilih', 'saja', 'berjanji', 'akan', 'bangun', 'juta', 'rumah']
Token IDs	[2, 90, 387, 16051, 553, 599, 1871, 6073, 377, 9593, 150, 3692, 1633, 448, 3]

6.2. Fine-Tuning IndoBERT

Pada tahap ini, dataset yang telah dikonversi ke format yang kompatibel dengan model IndoBERT menggunakan BERT Tokenizer digunakan untuk proses training. Proses fine-tuning dilakukan untuk mengadaptasi model IndoBERT yang telah di *pre-training* agar sesuai dengan tugas spesifik, yaitu analisis sentimen. Model IndoBERT yang digunakan dalam penelitian ini adalah *indobert-base-uncased*, sebuah varian model BERT yang telah di *pre-training* dengan korpus teks berbahasa Indonesia.

Selama proses fine-tuning, model dilatih menggunakan dataset yang telah dilabeli dengan memanfaatkan *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). MLM membantu model memahami konteks kata dalam kalimat dengan memprediksi kata-kata yang sengaja disembunyikan, sementara NSP melatih

model untuk memahami hubungan antara dua kalimat. Kedua teknik ini memungkinkan model untuk mempelajari representasi teks yang lebih kaya dan kontekstual.

Setelah proses *training* selesai, model IndoBERT dioptimalkan lebih lanjut untuk tugas analisis sentimen dengan menambahkan lapisan klasifikasi diatas *encoder*. Lapisan ini terdiri dari dense layer dan fungsi aktivasi *softmax* untuk menghasilkan probabilitas kelas sentimen. Dengan demikian, model IndoBERT diharapkan dapat mencapai performa yang optimal dalam mengklasifikasikan sentimen dari teks input.

6.3. Grafik dan Diagram

Hasil dari proses dan *fine-tuning* model IndoBERT akan disajikan dalam bentuk grafik dan diagram visual. Grafik akan memvisualisasikan tingkat akurasi *training* dan validasi model serta *train loss* dan *validation loss* berdasarkan nilai *epoch* selama proses pelatihan.

6.4. Evaluasi Matrik

Setelah pelatihan, validasi dan pengujian, kinerja model diukur menggunakan berbagai

matrik evaluasi yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Untuk menganalisis kemampuan model dalam mengklasifikasikan sentimen secara lebih mendalam, hasil klasifikasi dievaluasi menggunakan *confusion matrix*. Hal ini memungkinkan penilaian yang komprehensif terhadap keakuratan dan performa IndoBERT dalam tugas analisis sentimen.

7. Pengujian Akhir Model

Setelah Pada tahap ini, model IndoBERT yang telah melalui proses *fine-tuning* akan diuji untuk memastikan kinerjanya dalam mengelompokkan sentimen teks dengan dua cara. Yang pertama, model akan melakukan klasifikasi pada dataset yang sudah diberi label sebelumnya. Yang kedua, model akan digunakan untuk menganalisis teks acak yang tidak memiliki label. Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan model dalam mengklasifikasikan teks.

BAB IV

HASIL DAN PEMBAHASAN

A. Pengambilan Data Twitter

Data untuk penelitian ini diperoleh menggunakan *Tweet Harvest*, sebuah alat yang memungkinkan pengumpulan data dari platform Twitter/X. *Tweet Harvest* menggunakan *Playwright* untuk mengumpulkan tweet dari hasil pencarian Twitter berdasarkan kata kunci dan rentang tanggal tertentu. Berikut merupakan gambar penerapan *Tweet Harvest* pada proses pengambilan data di Twitter.

```
data = "tapera.csv"
search_keyword = "tapera lang:id until:2024-10-31 since:2024-01-01"
limit = 2000

!npm --yes tweet-harvest@latest -o "{data}" -s "{search_keyword}" -l {limit} --token ""
```

Gambar 4. 1. *Source Code* Penerapan *Tweet Harvest*

Untuk menggunakan *Tweet Harvest* diharuskan terlebih dahulu menginstall Node.js di komputer atau *jupyter notebook*. Pada penelitian ini *jupyter notebook* yang digunakan adalah *google colaboratory* yang merupakan sebuah *tools* berbentuk cloud yang dijalankan menggunakan browser untuk menulis dan menjalankan program *python*.

Setelah terinstall, siapkan beberapa variabel yang akan digunakan dalam proses pengambilan data yaitu "*data*" untuk menyimpan *output* yang akan dihasilkan oleh

Tweet Harvest menjadi sebuah file berbentuk .csv, 'search_keyword' untuk memasukan kata kunci, bahasa, dan rentang waktu atau tanggal *tweet* yang akan diambil dan 'limit' untuk menetapkan jumlah maksimal *tweet* yang akan diambil.

Kemudian untuk menjalankan package *Tweet Harvest* melalui Node.js, gunakan 'npx tweet-harvest@latest'. Setelah menjalankan baris program tersebut, *Tweet Harvest* akan meminta sebuah token. Token ini didapat dari 'aunt token' akun Twitter yang digunakan untuk melakukan pengambilan data. Jika token valid, *Tweet Harvest* akan langsung melakukan proses pengambilan data yang kemudian data tersebut akan disimpan ke dalam sebuah file berbentuk dengan format csv.

Proses pengambilan data dilakukan dengan menggunakan kata kunci 'tapera', yang relevan dengan topik penelitian mengenai narasi kebijakan Tapera di media sosial. Data yang diambil memiliki rentang waktu dari bulan Januari sampai bulan Oktober tahun 2024. Dari proses pengumpulan data tersebut terkumpul data sebanyak 2.604 data.

Terdapat 15 atribut yang diambil dari sebuah *tweet* yaitu 'conversation_id_str', 'created_at', 'favorite_count', 'full_text', 'image_url', 'in_reply_to_screen_name', 'lang', 'location', 'quote_count', 'reply_count', 'retweet_count',

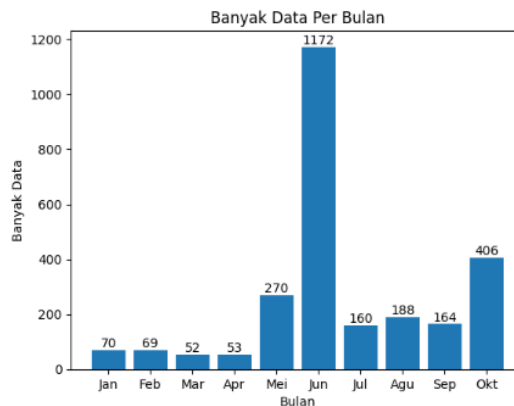
'tweet_url', 'user_id_str', 'username', dan 'id_str'. Agar data lebih efisien, dari 15 atribut diatas hanya 'created_at' dan 'full_text' yang akan digunakan. Berikut merupakan contoh data yang telah terkumpul.

Tabel 4. 1. Contoh Dataset yang Digunakan

No	created_at	full_text
1	Tue Jun 04 15:30:39 +0000 2024	Rumah hasil TAPERA selama 20 tahun https://t.co/m4gJbdJRf7
2	Tue Jun 04 02:48:42 +0000 2024	Keren ini. Rohingya ditampung Gaza ditampung Rakyat sendiri menghadapi harga beras naik BBM naik UKT naik Sembako naik transport naik cicilan naik pajak naik potongan Tapera potongan BPJS Kesehatan potongan ketenagakerjaan terkena pinjol dll https://t.co/wZCIWAsPRG
3	Sat Jun 01 06:36:03 +0000 2024	Penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong Tapera. https://t.co/vfHIPomXY2
.....		
2602	Thu Apr 25 02:01:33 +0000 2024	Selain skema KPR dengan tenor 35 tahun ada peluang KPR Tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/DTIGxwW9e3 #Opini #AdadiKompas
2603	Sat Apr 27 15:07:31 +0000 2024	@BachtarNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtarNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile.

		Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan
2604	Mon Apr 15 05:10:47 +0000 2024	@Lialarsen @Alfred67686158 @BlessingMombes3 @KingJayZim @classic263 @TeamFuloZim Apa hanzi tapera

Berikut merupakan visualisasi banyak data per bulan yang dikumpulkan.



Gambar 4. 2. Grafik Data Per Bulan

Dari gambar grafik data per bulan diatas, terlihat lonjakan signifikan pada jumlah data di bulan Juni, mencapai 1.172 *tweet*. Ini menunjukkan bahwa topik kebijakan Tapera menjadi sangat dibicarakan atau *trending* pada bulan tersebut.

B. Pelabelan

Dalam penelitian ini, proses pelabelan data dilakukan secara manual oleh seorang pakar bahasa, Ibu Nurul Ainunnisa, S.Pd., yang merupakan guru mata pelajaran

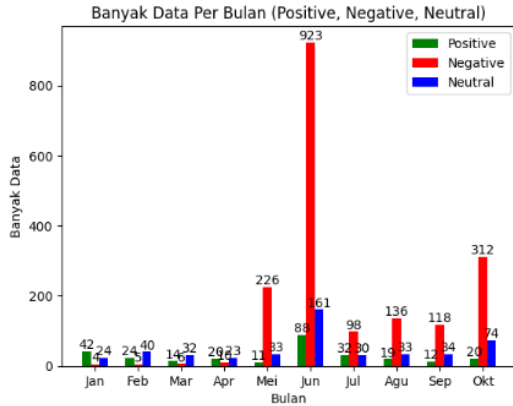
Bahasa Indonesia di MAN 2 Kota Semarang. Sebagai seorang ahli dalam bidang bahasa, beliau memiliki pemahaman mendalam terhadap struktur dan makna kalimat dalam Bahasa Indonesia, sehingga dapat memberikan analisis yang akurat terhadap sentimen dari teks yang dianalisis. Berikut merupakan contoh data yang sudah dilabeli.

Tabel 4. 2. Contoh Dataset Yang Sudah Diberi Label

No	created_at	full_text	Label
1	Tue Jun 04 15:30:39 +0000 2024	Rumah hasil TAPERA selama 20 tahun https://t.co/m4gJbdJRf7	neutral
2	Tue Jun 04 02:48:42 +0000 2024	Keren ini. Rohingya ditampung Gaza ditampung Rakyat sendiri menghadapi harga beras naik BBM naik UKT naik Sembako naik transport naik cicilan naik pajak naik potongan Tapera potongan BPJS Kesehatan potongan ketenagakerjaan terkena pinjol dll https://t.co/wZCIWAsPRG	negative
3	Sat Jun 01 06:36:03 +0000 2024	Penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong Tapera. https://t.co/vfHIPomXY2	negative
.....			

2602	Thu Apr 25 02:01:33 +0000 2024	Selain skema KPR dengan tenor 35 tahun ada peluang KPR Tapera bagi masyarakat berpenghasilan rendah dan milenial. #Opini #AdadiKompas	positive
2603	Sat Apr 27 15:07:31 +0000 2024	@BachtiarNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtiarNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan	neutral
2604	Mon Apr 15 05:10:47 +0000 2024	@Lialarsen @Alfred67686158 @BlessingMombes3 @KingJayZim @classic263 @TeamFuloZim Apa hanzi tapera	neutral

Hasil dari proses pelabelan ini kemudian divisualisasikan dalam bentuk grafik untuk melihat distribusi sentimen dari data yang telah dikumpulkan. Berikut merupakan grafik distribusi sentiment per bulan dari dataset yang digunakan.



Gambar 4. 3. Grafik Distribusi Label Per Bulan

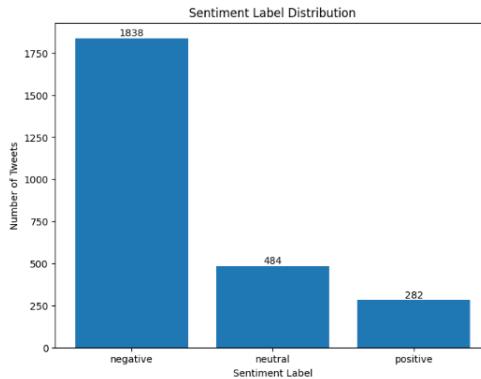
Berdasarkan grafik distribusi sentimen diatas, dapat diketahui dari bulan Januari sampai bulan April jumlah data yang diperoleh masih relatif rendah dibandingkan dengan jumlah data dari bulan Mei sampai bulan Oktober, sentimen yang muncul cukup seimbang, dengan dominasi sentimen netral dan positif dengan beberapa unggahan negatif. Hal ini menunjukkan belum ramainya perbincangan terkait isu Tapera.

Pada bulan Mei, jumlah unggahan mulai meningkat menjadi 270 unggahan, yang didominasi dengan sentimen negatif sebanyak 226 unggahan, sementara positif dan netral relatif rendah. Hal ini menandakan mulai ada respons yang cukup kuat dari masyarakat terhadap isu kebijakan Tapera, dengan kecenderungan opini yang kurang mendukung.

Puncak dari perbincangan terkait isu Tapera terjadi pada bulan Juni, yang ditandai dengan jumlah unggahan melonjak tajam hingga 1.172 unggahan. Pada bulan ini sentimen negatif mendominasi sangat signifikan dengan 923 unggahan, diikuti oleh netral sebanyak 161 unggahan dan positif sebanyak 88 unggahan. Lonjakan ini menandakan bahwa, kebijakan Tapera memicu kontroversi atau ketidakpuasan yang meluas di masyarakat dan menimbulkan banyak reaksi dalam bentuk kritik yang bersifat kontra terhadap kebijakan tersebut. Setelah itu, pada bulan Juli hingga bulan September, jumlah tweet menurun menjadi 160 sampai 188 unggahan per bulan, meskipun sentimen negatif tetap dominan.

Pada Oktober, terjadi kenaikan kembali dalam jumlah unggahan menjadi 406, dengan 312 diantaranya bersentimen negatif, sementara sentimen netral dan positif juga mengalami sedikit peningkatan. Hal ini menunjukkan bahwa isu Tapera kembali menjadi perhatian publik, mungkin karena adanya perkembangan terbaru atau respons pemerintah terhadap kritik sebelumnya.

Berikut merupakan grafik jumlah keseluruhan distribusi sentimen dari dataset.



Gambar 4. 4. Grafik Keseluruhan Distribusi Label

Secara keseluruhan, grafik menggambarkan bahwa kebijakan Tapera lebih banyak didominasi oleh respons negatif dengan 1.838 tweet, jauh lebih banyak dibandingkan sentimen netral yang hanya 484 tweet dan positif yang berjumlah 282 tweet. Hal ini menunjukkan bahwa kebijakan ini masih menjadi perdebatan di kalangan masyarakat, dengan sentimen yang cenderung kritis terhadap implementasi atau dampaknya.

Dari grafik ini pula, terlihat adanya ketidakseimbangan distribusi kelas dalam dataset, yang ditandai dengan *tweet* sentimen negatif jauh lebih tinggi dibandingkan dengan sentimen netral maupun positif. Ketidakseimbangan ini dapat berdampak pada analisis lebih lanjut, terutama dalam penerapan model pembelajaran mesin, karena model cenderung lebih terpengaruh oleh kelas yang dominan, dalam hal ini sentimen negatif.

Oleh karena itu, diperlukan proses *Easy Data Augmentasi* (EDA) untuk mengurangi ketidakseimbangan distribusi kelas. Tetapi sebelum data masuk dalam proses EDA, harus terlebih dahulu melalui tahap *pre-processing* untuk memastikan data bersih, terstruktur, dan siap untuk diproses.

C. Pre-Processing

Data yang sudah dilabeli tersebut kemudian masuk ke tahap *pre-processing*. Pada tahap ini terdiri dari 3 proses yang berurutan yaitu *case folding*, *cleaning* dan *spell normalized*. Ketiga proses ini diperlukan untuk memastikan data yang akan diolah oleh model lebih terstruktur dan bersih dari *noise*. Berikut merupakan tahapan dari *pre-processing* yang dilakukan pada penelitian ini.

1. Case Folding

Pada proses ini, data atau teks yang memiliki huruf kapital akan diubah menjadi huruf kecil yang bertujuan untuk menyamaratakan semua teks yang ada. Berikut *source code* yang digunakan untuk melakukan proses *case folding*.

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

Gambar 4. 5. *Source Code* Proses *Case Folding*

Berdasarkan *source code* diatas menunjukan bahwa hanya atribut '*full_text*' dari data akan digunakan dalam proses ini. Hasil *case folding* akan disimpan di kolom '*case_folding*'. Berikut merupakan contoh hasil dari proses *case folding*.

Tabel 4. 3. Contoh Hasil Proses *Case Folding*

No	full_text	case_folding
1	Rumah hasil TAPERA selama 20 tahun https://t.co/m4gJbdJRf7	rumah hasil tapera selama 20 tahun https://t.co/m4gjbdrjf7
2	Keren ini. Rohingya ditampung Gaza ditampung Rakyat sendiri menghadapi - harga beras naik - BBM naik - UKT naik* - Sembako naik - transport naik - cicilan naik - pajak naik - potongan Tapera - potongan BPJS kesehatan - potongan ketenagakerjaan - terjerat pinjol - dll. https://t.co/wZCIWAsP RG	keren ini. rohingya ditampung gaza ditampung rakyat sendiri menghadapi - harga beras naik - bbm naik - ukt naik* - sembako naik - transport naik - cicilan naik - pajak naik - potongan tapera - potongan bpjs kesehatan - potongan ketenagakerjaan - terjerat pinjol - dll. https://t.co/wzclwaspr g
3	Penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong Tapera.	penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong tapera.

	https://t.co/vfHIPomXY2	https://t.co/vfhipomxy2
****	*****	*****
2602	Selain skema KPR dengan tenor 35 tahun ada peluang KPR Tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/DTIGxwW9e3 #Opini #AdadiKompas	selain skema kpr dengan tenor 35 tahun ada peluang kpr tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/dtlgxww9e3 #opini #adadikompas
2063	@BachtianNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtianNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan	@bachtiarnu assalamu alaikum wr.wb sahabat btn syariah @bachtiarnu kami informasikan saat ini pengajuan pembiayaan kpr btn subsidi ib dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi sikasep online/tapera mobile. terima kasih. wassalamu alaikum wr.wb. ^joan
2604	@Lialarsen @Alfred67686158 @BlessingMombes3 @KingJayZim @classic263 @TeamFuloZim Apa hanzi tapera	@lialarsen @alfred67686158 @blessingmombes3 @kingjayzim @classic263 @teamfulozim apa hanzi tapera

2. Spell Normalized

Proses normalisasi ejaan dilakukan untuk mengubah kata-kata tidak baku atau slang menjadi kata baku. Langkah ini bertujuan agar model dapat lebih mudah memahami konteks dan makna dari teks yang dianalisis. Sumber data untuk kata slang dalam penelitian ini berasal dari beberapa kamus slang yang tersedia secara publik di GitHub, yaitu '*colloquial-indonesian-lexicon.csv*' terdiri dari 15.397 baris kata, '*kbba.txt*' terdiri dari 1.317 baris kata, '*slangword.txt*' terdiri dari 286 baris kata dan '*slang.csv*' terdiri dari 1.543 baris kata. Dengan menggunakan sumber ini, penelitian dapat memastikan bahwa teks yang dianalisis telah memenuhi standar bahasa baku, sehingga hasil analisis menjadi lebih akurat dan relevan. Kamus-kamus tersebut terdiri. Berikut merupakan contoh isi dari kamus kamus tersebut.

Tabel 4. 4. Contoh Isi '*colloquial-indonesian-lexicon.csv*'

No	Slang	Normal
1	met	selamat
2	woww	wow
.....		
15397	gaharus	enggak harus

Tabel 4. 5. Contoh Isi '*kbba.txt*'

No	Slang	Normal
1	abis	habis
2	7an	tujuan

.....		
1317	adalah	adalah

Tabel 4. 6. Contoh Isi 'slangword.txt'

No	Slang	Normal
1	&am;	dan
2	=	Sama dengan
.....		
286	wtf	Apa-apaan

Tabel 4. 7. Contoh Isi 'slang.csv'

No	Slang	Normal
1	udah	sudah
2	gau	aku
.....		
1543	tntang	tentang

Berikut *source code* yang digunakan untuk melakukan proses *spell normalized*.

```
# Load colloquial-indonesian-lexicon.csv
kamus_alay = pd.read_csv('colloquial-indonesian-lexicon.csv')
alay_dict = pd.Series(kamus_alay['formal'].values, index=kamus_alay['slang']).to_dict()

# Load kamus kbba.txt
kamus_kbba = pd.read_csv('kbba.txt', sep='\t')
kbba_dict = pd.Series(kamus_kbba['tujuan'].values, index=kamus_kbba['?an']).to_dict()

# Load kamus slang.csv
kamus_slang = pd.read_csv('slang.csv')
slang_dict = pd.Series(kamus_slang['formal'].values, index=kamus_slang['slang']).to_dict()

# Load kamus slangword.txt
kamus_sw = pd.read_csv('slangword.txt', sep=':')
sw_dict = pd.Series(kamus_sw['formal'].values, index=kamus_sw['slang']).to_dict()
```

Gambar 4. 6. Source Code Memuat Kamus Alay

```

# Gabungkan semua kamus
all_dicts = [alay_dict, kbba_dict, slang_dict, sw_dict]
combined_dict = {}
for d in all_dicts:
    combined_dict.update(d)

# Fungsi untuk normalisasi teks
def normalize_text(text):
    normalized_text = []
    for word in text.split():
        if word in combined_dict:
            normalized_text.append(combined_dict[word])
        else:
            normalized_text.append(word)
    return ' '.join(normalized_text)

df['normalized_text'] = df['case_folding'].apply(normalize_text)
df

```

Gambar 4. 7. *Source Code Proses Spell Normalized*

Pada bagian pertama dilakukan pembacaan file kamus yaitu '*colloquial-indonesian-lexicon.csv*', '*kbba.txt*', '*slang.csv*', dan '*slangword.txt*'. Kemudian semua kamus digabung menjadi satu *dictionary* besar (*combined_dict*). Selanjutnya dibuat fungsi *normalize_text* yang akan mengganti kata-kata dalam teks dengan versi formalnya jika ditemukan dalam kamus gabungan. Hasil normalisasi kemudian diterapkan ke *DataFrame* menggunakan kolom '*case_folding*' dan disimpan dalam kolom baru bernama '*normalized_text*'. Hasil proses diatas menunjukkan bahwa sebanyak 7.183 kata telah berhasil dinormalisasi dari 1.855 data. Berikut merupakan contoh hasil dari proses *spell normalized*.

Tabel 4. 8. Contoh Hasil *Spell Normalized*

No	case_folding	normalized_text
1	rumah hasil tapera selama 20 tahun https://t.co/m4gibdjrf7	rumah hasil tapera selama 20 tahun https://t.co/m4gibdjrf7

2	keren ini. rohingya ditampung gaza ditampung rakyat sendiri menghadapi - harga beras naik - BBM naik - ukt naik* - sembako naik - transport naik - cicilan naik - pajak naik - potongan tapera - potongan bpjs kesehatan - potongan ketenagakerjaan - terjerat pinjol - dll. https://t.co/wzclwasprg	keren ini. rohingya ditampung gaza ditampung rakyat sendiri menghadapi - harga beras naik - BBM naik - ukt naik* - sembilan bahan pokok naik - transport naik - cicilan naik - pajak naik - potongan tapera - potongan bpjs kesehatan - potongan ketenagakerjaan - terjerat pinjol - dll. https://t.co/wzclwasprg
3	penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong tapera. https://t.co/vfhipomxy2	penghasilan ojol itu sudah dipotong biaya aplikasi yang gede dianggap mitra saja tidak ada jaminan kesehatan tidak ada jaminan pensiun sudah overwork kejar target bahkan mau dipotong tapera. https://t.co/vfhipomxy2
....		
2602	selain skema kpr dengan tenor 35 tahun ada peluang kpr tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/dtlgxww9e3 #opini #adadikompas	selain skema kpr dengan tenor 35 tahun ada peluang kpr tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/dtlgxww9e3 #opini #adadikompas
2603	@bachtiarnu assalamu alaikum wr.wb sahabat btn syariah	@bachtiarnu assalamu alaikum wr.wb sahabat btn syariah @bachtiarnu

	@bachtiarnu kami informasikan saat ini pengajuan pembiayaan kpr btn subsidi ib dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi sikasep online/tapera mobile. terima kasih. wassalamu alaikum wr.wb. ^joan	kami informasikan saat ini pengajuan pembiayaan kpr btn subsidi ib dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi sikasep online/tapera mobile. terima kasih. wassalamu alaikum wr.wb. ^joan
2604	@lialarsen @alfred67686158 @blessingmombes3 @kingjayzim @classic263 @teamfulozim apa hanzi tapera	@lialarsen @alfred67686158 @blessingmombes3 @kingjayzim @classic263 @teamfulozim apa hanzi tapera

3. Cleaned text

Pada proses ini, data atau teks akan dibersihkan dari simbol simbol yang tidak dibutuhkan seperti tanda baca, hastag, mention dan sebagainya. Untuk mendapatkan hasil terbaik, proses membersihkan dilakukan dalam 6 tahapan, yaitu sebagai berikut :

1. Menghapus link atau URL
2. Menghapus mention
3. Menghapus hastag atau tagar
4. Menghapus spesial karakter
5. Menghapus karakter selain a-z
6. Menghapus ekstra spasi

Setiap tahapan dilakukan secara berurutan untuk memastikan teks yang dihasilkan benar-benar bersih dan terstruktur, sehingga memungkinkan analisis lebih akurat dan relevan. Berikut *source code* yang digunakan untuk melakukan proses *cleaned text*.

```
def clean_text(text):
    if isinstance(text, str):
        # URLs
        text = re.sub(r'http\S+', '', text)
        # mentions
        text = re.sub(r'@\S+', '', text)
        # hashtags
        text = re.sub(r'#\S+', '', text)
        # special characters
        text = re.sub(r'[\W\s]', '', text)
        # characters other than a-z
        text = re.sub(r'[^a-zA-Z]', '', text)
        # extra spaces
        text = re.sub(r'\s+', ' ', text).strip()
    else:
        # non-string values
        text = ''
    return text

df['cleaned_text'] = df['normalized_text'].apply(clean_text)
```

Gambar 4. 8. *Source Code Proses Cleaned Text*

Berdasarkan *source code* diatas, terdapat fungsi ‘*clean_text*’ yang digunakan untuk menampung ke-6 tahapan proses pembersihan teks. Pertama, fungsi ‘*clean_text*’ akan memeriksa apakah teks yang akan dibersihkan bertipe *string*. Jika benar teks akan masuk dalam 6 tahapan proses pembersihan, sebaliknya jika bukan bertipe *string*, teks akan dikembalikan sebagai data kosong. Setelah semua tahap selesai, fungsi akan mengembalikan teks yang sudah dibersihkan. Fungsi ini kemudian diterapkan pada kolom ‘*normalized_text*’ dan hasilnya disimpan dalam kolom baru bernama ‘*cleaned_text*’. Hasil proses diatas menunjukkan bahwa

sebanyak 4.603 simbol telah berhasil dihilangkan dari 2.391 data. Berikut merupakan contoh hasil dari proses *cleaned text*.

Tabel 4. 9. Contoh Hasil Proses *Cleaned Text*.

No	normalized_text	cleaned_text
1	rumah hasil tapera selama 20 tahun https://t.co/m4gjbdrf7	rumah hasil tapera selama tahun
2	keren ini. rohingya ditampung gaza ditampung rakyat sendiri menghadapi - harga beras naik - bbm naik - ukt naik* - sembilan bahan pokok naik - transport naik - cicilan naik - pajak naik - potongan tapera - potongan bpjs kesehatan - potongan ketenagakerjaan - terjerat pinjol - dll. https://t.co/wzclwasprg	keren ini rohingya ditampung gaza ditampung rakyat sendiri menghadapi harga beras naik bbm naik ukt naik sembilan bahan pokok naik transport naik cicilan naik pajak naik potongan tapera potongan bpjs kesehatan potongan ketenagakerjaan terjerat pinjol dll
3	penghasilan ojol itu sudah dipotong biaya aplikasi yang gede dianggap mitra saja tidak ada jaminan kesehatan tidak ada jaminan pensiun sudah overwork kejar target bahkan mau dipotong tapera. https://t.co/vfhipomxy2	penghasilan ojol itu sudah dipotong biaya aplikasi yang gede dianggap mitra saja tidak ada jaminan kesehatan tidak ada jaminan pensiun sudah overwork kejar target bahkan mau dipotong tapera
....		

2602	selain skema kpr dengan tenor 35 tahun ada peluang kpr tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/dtlgxww9e3 #opini #adadikompas	selain skema kpr dengan tenor tahun ada peluang kpr tapera bagi masyarakat berpenghasilan rendah dan milenial
2603	@bachtiarnu assalamu alaikum wr.wb sahabat btn syariah @bachtiarnu kami informasikan saat ini pengajuan pembiayaan kpr btn subsidi ib dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi sikasep online/tapera mobile. terima kasih. wassalamu alaikum wr.wb. ^joan	assalamu alaikum wrwb sahabat btn syariah kami informasikan saat ini pengajuan pembiayaan kpr btn subsidi ib dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi sikasep onlinetapera mobile terima kasih wassalamu alaikum wrwb joan
2604	@lialarsen @alfred67686158 @blessingmombes3 @kingjayzim @classic263 @teamfulozim apa hanzi tapera	apa hanzi tapera

Setelah tahap pre-processing, data duplikat dihapus untuk menghindari redudansi yang dapat mempengaruhi hasil analisis. Dari 2.604 entri awal, jumlah data berkurang menjadi 2.523 setelah penyaringan. Langkah ini memastikan setiap sampel

unik, sehingga proses augmentasi dan pemodelan lebih optimal.

D. Easy Data Augmentation (EDA)

Sebelum melakukan proses eksplorasi data (EDA), tahap awal yang dilakukan adalah membersihkan dataset dari kolom-kolom yang tidak digunakan dalam proses ini. Kolom yang dihapus mencakup '*full_text*', '*case_folding*', dan '*normalized_text*'. Dan kolom yang dipertahankan adalah '*cleaned_text*' dan '*label*', yang masing-masing merepresentasikan teks hasil *pre-processing* dan kategori sentimen.

Proses augmentasi ini hanya diterapkan pada data yang memiliki kelas minoritas, dalam penelitian ini yaitu kelas *neutral* dan *positive*, dengan tujuan untuk mengurangi ketidakseimbangan distribusi kelas dalam dataset sebelum masuk ke tahap pemodelan. Data yang memiliki kelas minoritas akan melewati salah satu dari keempat teknik EDA secara acak.

Teknik pertama yaitu *Synonym Replacement* (SR) yang mengganti suatu kata dengan sinonimnya. Kata sinonim didapat dari Tesaurus Bahasa Indonesia dalam format *json* yang berasal dari Pusat Bahasa karya Departemen Pendidikan Nasional pada tahun 2008. Teknik kedua yaitu *Random Insertion* (RI) untuk menyisipkan

suatu kata sinonim dari salah satu kata yang ada dalam data secara acak pada kalimat. Teknik ketiga yaitu *Random Swap* (RS) untuk mengubah urutan suatu kata secara acak pada suatu kalimat. Teknik keempat yaitu *Random Deletion* (RD) untuk menghapus suatu kata secara acak pada suatu kalimat. Berikut *source code* dari keempat teknik EDA yang akan digunakan dalam penelitian ini.

```
# 1. Synonym Replacement (SR)
def synonym_replacement(text, alpha=alpha):
    words = text.split()
    n = max(1, int(alpha * len(words)))
    indices = random.sample(range(len(words)), n)
    for i in indices:
        words[i] = get_synonym(words[i])
    return " ".join(words)

# 2. Random Insertion (RI)
def random_insertion(text, alpha=alpha):
    words = text.split()
    n = max(1, int(alpha * len(words)))
    for _ in range(n):
        word = random.choice(words)
        synonym = get_synonym(word)
        position = random.randint(0, len(words))
        words.insert(position, synonym)
    return " ".join(words)

# 3. Random Swap (RS)
def random_swap(text, alpha=alpha):
    words = text.split()
    n = max(1, int(alpha * len(words)))
    for _ in range(n):
        if len(words) < 2:
            break
        idx1, idx2 = random.sample(range(len(words)), 2)
        words[idx1], words[idx2] = words[idx2], words[idx1]
    return " ".join(words)

# 4. Random Deletion (RD)
def random_deletion(text, alpha=alpha):
    words = text.split()
    if len(words) == 1:
        return text
    words = [word for word in words if random.random() > alpha]
    return " ".join(words) if words else random.choice(text.split())
```

Gambar 4. 9. *Source Code* Empat Teknik EDA

Dari keempat teknik tersebut, masing-masing teknik memerlukan parameter α (*alpha*) untuk menentukan bagaimana augmentasi diterapkan pada teks. Pada teknik *Synonym Replacement* (SR), *Random Insertion* (RI) dan *Random Swap* (RS) parameter α digunakan untuk menentukan proporsi kata yang dimodifikasi. Sedangkan

untuk teknik *Random Deletion* (RD), parameter α digunakan untuk menentukan probabilitas penghapusan setiap kata dalam kalimat. Dalam penelitian ini, nilai α ditetapkan sebesar 0,1 sesuai dengan rekomendasi dari Wei dan Zou (2019). Berikut *source code* inialisasi parameter yang digunakan.

```
# Inialisasi parameter
alpha = 0.1
n_aug = 2
```

Gambar 4. 10. Inialisasi Parameter

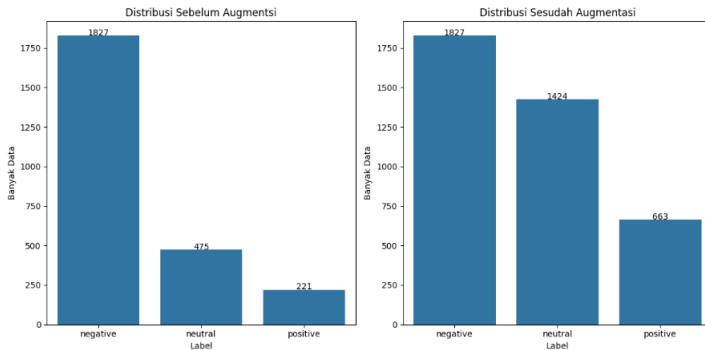
Selain parameter α , parameter lain yang digunakan dalam proses EDA adalah *naug*. Parameter ini digunakan untuk menentukan banyak kalimat baru yang dibuat dari satu data asli. Pada penelitian ini, nilai parameter *naug* ditetapkan sebesar 2. Jadi setiap data yang melewati proses EDA akan di ubah menjadi dua versi augmentasi. Nilai ini ditentukan berdasarkan kebutuhan dari dataset agar mencegah overfitting, yang bisa terjadi jika terlalu banyak kalimat augmentasi dibuat. Berikut merupakan contoh hasil dari proses EDA.

Tabel 4. 10. Contoh Hasil Proses EDA

cleaned_text	Augmentasi 1	Augmentasi 2
beramairamai endorse tapera	beramairamai tapera endorse	endorse beramairamai tapera
cek fakta benarkah tapera bisa atasi backlog perumahan	cek fakta benarkah tapera atasi backlog perumahan	cek fakta benarkah atasi tapera bisa atasi

		backlog perumahan
dukung dana tapera pengembangan siap kebut bangun rumah murah	dukung dana tapera pengembang siap kencang bangun rumah murah	siap dana tapera pengembang dukung kebut bangun rumah murah
rakyat sejahtera lewat program tapera bisa	rakyat senang lewat program tapera bisa	rakyat sejahtera lewat cadangan tapera bisa
jadi ini tujuan tapera buat rumah ini ternyata masuk akal sih kalau ini	jadi ini tujuan tapera buat rumah sebenarnya ini ternyata masuk akal sih kalau ini	jadi ini tujuan tapera buat rumah ini ternyata masuk akal kalau sih ini

Setelah proses ini, didapati jumlah data meningkat dari 2.523 menjadi 3.914. Sebelum dilakukan EDA, data terdiri dari 1.827 data berlabel negatif, 475 data berlabel netral, dan 221 data berlabel positif. Setelah EDA, distribusi label berubah menjadi 1.827 data berlabel negatif, 1.424 data berlabel netral, dan 663 data berlabel positif. Berikut adalah grafik yang menunjukkan perbandingan distribusi label sebelum dan sesudah EDA.



Gambar 4. 11. Perbandingan Distribusi Label Sebelum dan Sesudah Proses EDA

E. Splitting Data

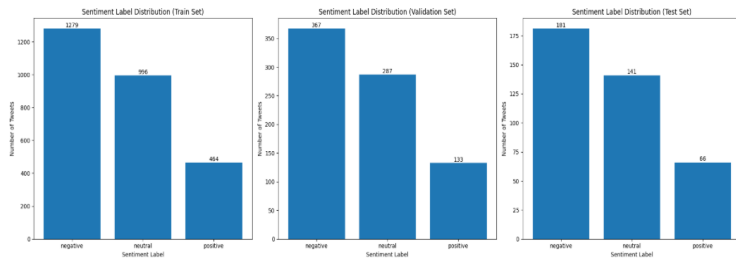
Setelah augmentasi selesai, data dibagi menjadi tiga kelompok, yaitu data latih (*training set*) sebesar 70%, data validasi (*validation set*) sebesar 20%, dan data uji (*test set*) sebesar 10%. Data latih digunakan untuk melatih model, data validasi digunakan untuk mengevaluasi performa model selama proses pelatihan, dan data uji digunakan untuk mengukur performa akhir model setelah pelatihan selesai. Proses pembagian data dilakukan secara acak (*random sampling*) untuk memastikan distribusi data pada masing-masing kelompok tetap representatif. Berikut *source code* yang digunakan untuk melakukan *splitting* data.

```
# Splitting data 70% (train), 20% (val), 10% (test)
train_set, val_set = train_test_split(df, test_size=0.3, random_state=42, stratify=df['label'])
val_set, test_set = train_test_split(val_set, test_size=0.33, random_state=42, stratify=val_set['label'])

print(f"Jumlah data train: {len(train_set)}")
print(f"Jumlah data validasi: {len(val_set)}")
print(f"Jumlah data test: {len(test_set)}")
```

Gambar 4. 12. *Source Code Splitting Data*

Dari proses pembagian data tersebut, diperoleh data latih sebanyak 2.739 data yang terdiri dari label negatif sebanyak 1.279 data, label netral sebanyak 996 data dan label positif sebanyak 464 data. Untuk data validasi, diperoleh sebanyak 787 data yang terdiri dari label negatif sebanyak 367 data, label netral sebanyak 287 data dan label positif sebanyak 133 data. Sementara itu, data uji terdiri dari 388 data dengan label negatif sebanyak 181 data, label netral sebanyak 141 data dan label positif sebanyak 66 data. Berikut merupakan grafik distribusi label dari data latih, data validasi dan data uji.



Gambar 4. 13. Distribusi Label pada Data Latih, Data Validasi dan Data Uji

Kemudian, data latih, data validasi, dan data uji akan disimpan dalam file berbentuk berformat *tsv* agar mudah diakses dan digunakan pada tahap pemrosesan atau analisis selanjutnya. Format *tsv* (*tab-separated values*) dipilih karena sederhana, terstruktur, dan kompatibel dengan berbagai alat serta pustaka pemrosesan data.

F. Implementasi IndoBERT

Setelah melalui proses *splitting*, data akan disesuaikan agar dapat diterima dan diolah oleh model. Setelah itu, proses klasifikasi dapat dilakukan. Berikut ini adalah rangkaian proses dalam implementasi model IndoBERT.

1. Tokenizing

Sebelum dataset digunakan dalam proses klasifikasi, dataset harus ditokenisasi terlebih dahulu agar sesuai dengan representasi input yang diterima oleh model IndoBERT. Dalam penelitian ini, proses tokenisasi dilakukan menggunakan BERT Tokenizer yang berasal dari *pre-trained* model *indobert-base-uncased*. Proses tokenisasi menggunakan BERT Tokenizer dari model *pre-trained indobert-base-uncased* menghasilkan representasi teks dalam bentuk token numerik yang sesuai dengan kosakata model. Tokenizer ini bekerja dengan memecah teks menjadi unit-unit kecil, baik berupa kata utuh maupun sub-kata, menggunakan teknik WordPiece *tokenization*. Hasil tokenisasi ini mencakup token khusus seperti [CLS] di awal teks dan [SEP] di akhir teks, yang diperlukan oleh arsitektur IndoBERT untuk memahami struktur input.

Setelah tokenisasi selesai, setiap token diubah menjadi indeks numerik sesuai dengan kosakata model. Representasi numerik ini kemudian digunakan untuk membentuk dua vektor utama, yaitu *Input Ids* untuk merepresentasi numerik dari token-token teks, termasuk token khusus [CLS] dan [SEP], dan *attention mask* yang merupakan vektor biner yang menunjukkan token mana yang perlu diperhatikan oleh model (1 untuk token yang relevan, 0 untuk *padding*). Berikut *source code* pada proses tokenisasi.

```
Subject = df['cleaned_text'].values

input_ids = []
attention_masks = []

for sent in Subject:
    encoded_sent = tokenizer.encode_plus(
        sent,
        add_special_tokens=True,
        max_length=512,
        padding='max_length',
        truncation=True,
        return_attention_mask=True,
        return_tensors='pt'
    )

    input_ids.append(encoded_sent['input_ids'].squeeze().tolist())
    attention_masks.append(encoded_sent['attention_mask'].squeeze().tolist())

# Pilih contoh teks ke-4
example_index = 4
print("Original :", Subject[example_index])
print("Tokenized :", tokenizer.tokenize(Subject[example_index]))
print("Token IDs :", input_ids[example_index])
print("Attention Mask :", attention_masks[example_index])
```

Gambar 4. 14. *Source Code* Tokenisasi

Berdasarkan *source code* diatas, model IndoBERT hanya menerima input dengan panjang maksimal 512 kata. Jika panjang kalimat lebih pendek dari batas tersebut, maka akan dilakukan *padding* dengan menambahkan token [PAD] untuk mengisi kekosongan. Sebaliknya, jika kalimat lebih panjang dari batas

maksimal, model akan melakukan *truncation* atau pemotongan kalimat.

Selain itu, untuk kata-kata yang tidak terdapat dalam kosakata model (*out-of-vocabulary*), IndoBERT akan memecahnya menjadi subword dengan menambahkan simbol ## di depan pecahan kata. Sebagai ilustrasi, proses penyesuaian input untuk IndoBERT dapat ditampilkan dalam bentuk tabel. Berikut contoh hasil dari proses tokenisasi.

Tabel 4. 11. Contoh Hasil Proses Tokenisasi

[illegible]

2. Fine-Tuning IndoBERT

Setelah proses penyesuaian, data kemudian dimasukkan ke dalam jaringan IndoBERT. Dalam proses

fine-tuning IndoBERT untuk tugas klasifikasi dilakukan dengan menambahkan lapisan klasifikasi pada model. Library *Transformers* menyediakan kelas *BertForSequenceClassification*, yang dirancang khusus untuk tugas klasifikasi teks. Model ini memanfaatkan keluaran dari *pooler* untuk menghitung nilai logits, yang kemudian diproses menggunakan fungsi *softmax* untuk menghasilkan prediksi akhir. Dalam penelitian ini, penulis menggunakan model *indobert-base-uncased* untuk melakukan *fine-tuning*. Berikut *source code* pada proses *fine-tuning* pada fase pelatihan dan fase validasi.

```
# Train
n_epochs = 5
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

        # Hitung skor train metrics
        list_hyp_train += batch_hyp
        list_label += batch_label

    train_pbar.set_description("(Epoch {}) TRAIN LOSS:{:.4f} LR:{:.8f}".format((epoch+1),
        total_train_loss/(i+1), get_lr(optimizer)))

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

# Simpan acc train
history['train_acc'].append(metrics['Acc'])
history['train_loss'].append(total_train_loss / (i + 1))
```

Gambar 4. 15. *Source Code* Fase Pelatihan

```

# Evaluasi di validation set
model.eval()
torch.set_grad_enabled(False)

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

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

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

    # Hitung skor evaluasi metrics
    list_hyp += batch_hyp
    list_label += batch_label
    metrics = document_sentiment_metrics_fn(list_hyp, list_label)

    pbar.set_description("VALID LOSS:{:.4f} {}".format(total_loss/(i+1), metrics_to_string(metrics)))

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

# Simpan acc val
history['val_acc'].append(metrics['ACC'])
history['val_loss'].append(total_loss / (i + 1))

```

Gambar 4. 16. Source Code Fase Validasi

Dalam proses *fine-tuning* untuk pelatihan dan validasi, membutuhkan beberapa *hyperparameter* seperti *batch size*, *learning rate* dan *epoch*. Berdasarkan referensi dari penelitian Devlin tahun 2020, *hyperparameter* yang direkomendasikan untuk *fine-tuning* agar mencapai performa optimal mencakup *batch size* sebesar 16 atau 32, serta *learning rate* dalam rentang 5e-5, 3e-5, hingga 2e-5. Selain itu, model dilatih menggunakan *optimizer Adam* dengan jumlah *epoch* sebanyak 2, 3, atau 4. Namun, dalam penelitian ini, jumlah *epoch* ditetapkan sebanyak 5 untuk memperoleh gambaran yang lebih luas terkait proses iterasi selama pelatihan. Skenario kombinasi *learning rate* dan *batch size* dapat dilihat pada tabel berikut.

Tabel 4. 12. Skenario *Hyperparameter*

Skenario	Learning Rate	Batch Size
Skenario 1	2e-5	16
Skenario 2	2e-5	32
Skenario 3	3e-5	16
Skenario 4	3e-5	32
Skenario 5	5e-5	16
Skenario 6	5e-5	32

Dari skenario-skenario diatas didapatkan hasil akhir sebagai berikut.

Tabel 4. 13. Hasil Skenario *Hyperparameter*

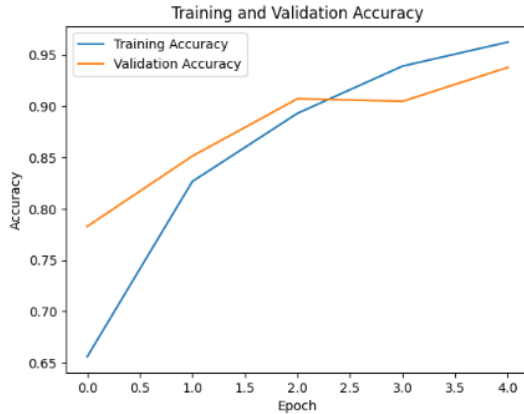
Skenario	Data	Accuracy	Precision	Recall	F1 score
Skenario 1	Training	0,96	0,96	0,96	0,96
	Validation	0,94	0,94	0,94	0,94
Skenario 2	Training	0,94	0,94	0,94	0,94
	Validation	0,94	0,94	0,93	0,94
Skenario 3	Training	0,98	0,98	0,98	0,98
	Validation	0,95	0,95	0,95	0,95
Skenario 4	Training	0,95	0,95	0,95	0,95
	Validation	0,93	0,94	0,93	0,93
Skenario 5	Training	0,96	0,96	0,96	0,96
	Validation	0,92	0,93	0,92	0,92
Skenario 6	Training	0,96	0,96	0,96	0,96
	Validation	0,94	0,94	0,94	0,94

Berikut merupakan analisis dari hasil keenam skenario *hyperparameter* diatas

2.1. Skenario 1

Pada Skenario 1, akurasi data *training* dan *validation* masing-masing sebesar 0,96 dan 0,94. Hasil ini menunjukkan bahwa model mampu belajar dengan baik tanpa indikasi *overfitting* yang signifikan. Selisih yang kecil antara kedua nilai ini

menunjukkan bahwa model memiliki kemampuan generalisasi yang baik. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 1.



Gambar 4. 17. Grafik Akurasi *Training* dan *Validation* Skenario 1

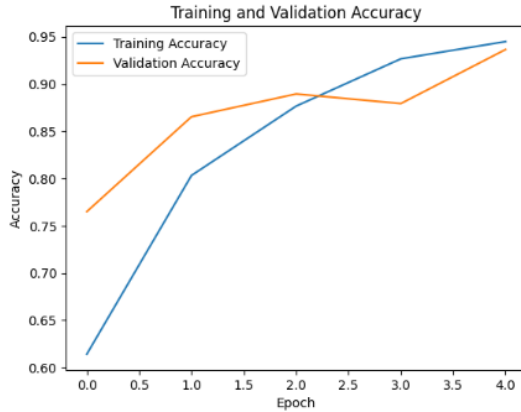


Gambar 4. 18. Grafik *Training Loss* dan *Validation Loss* Skenario 1

Grafik akurasi menunjukkan tren yang stabil dengan sedikit perbedaan antara data *training* dan *validation*, menandakan bahwa model tidak mengalami *overfitting*. Grafik loss juga menunjukkan penurunan yang konsisten tanpa fluktuasi besar, sehingga model dapat belajar secara optimal.

2.2. Skenario 2

Pada Skenario 2, akurasi data *training* sebesar 0,94 dan *validation* sebesar 0,94, yang menunjukkan stabilitas performa model seperti pada Skenario 1. Namun, sedikit penurunan akurasi pada data training mengindikasikan bahwa penggunaan *batch size* yang lebih besar dapat mempengaruhi kemampuan model dalam menangkap pola yang lebih kompleks. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 2.



Gambar 4. 19. Grafik Akurasi *Training* dan *Validation* Skenario 2



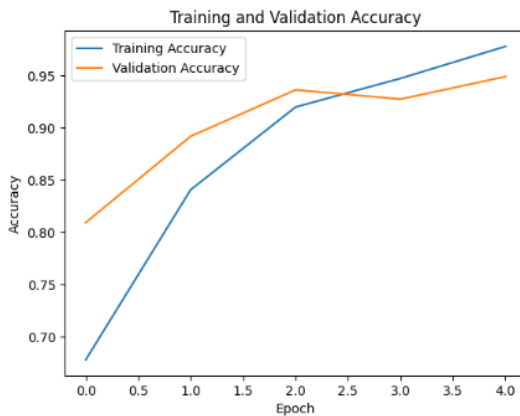
Gambar 4. 20. Grafik *Training* Loss dan *Validation* Loss Skenario 2

Grafik akurasi menunjukkan stabilitas yang serupa dengan Skenario 1, tetapi grafik loss menunjukkan sedikit fluktuasi yang mungkin disebabkan oleh penggunaan *batch size* yang lebih

besar, sehingga proses konvergensi menjadi kurang halus.

2.3. Skenario 3

Pada Skenario 3, akurasi data *training* mencapai 0,98 dengan akurasi *validation* sebesar 0,95. Perbedaan ini menunjukkan adanya indikasi awal *overfitting*, di mana model mulai terlalu menyesuaikan diri dengan data training sehingga kemampuan generalisasinya sedikit menurun. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 3.



Gambar 4. 21. Grafik Akurasi *Training* dan *Validation* Skenario 3



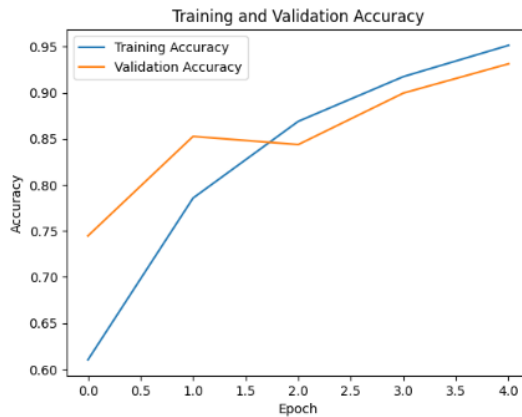
Gambar 4. 22. Grafik *Training Loss* dan *Validation Loss* Skenario 3

Grafik akurasi menunjukkan lonjakan yang lebih tinggi pada *training* dibandingkan dengan *validation*, yang memperkuat indikasi *overfitting*. Grafik *loss* juga menunjukkan bahwa *validation loss* tidak menurun secara signifikan, menandakan bahwa model lebih banyak belajar dari data training tetapi kurang efektif dalam menangani data baru.

2.4. Skenario 4

Pada Skenario 4, akurasi data *training* sebesar 0,95 dan *validation* sebesar 0,93. Model pada skenario ini masih menunjukkan keseimbangan yang cukup baik, tetapi performanya sedikit lebih rendah dibandingkan

dengan Skenario 1 dan 2. Hal ini menunjukkan bahwa *batch size* yang lebih besar tidak selalu memberikan peningkatan performa. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 4.



Gambar 4. 23. Grafik Akurasi *Training* dan *Validation* Skenario 4



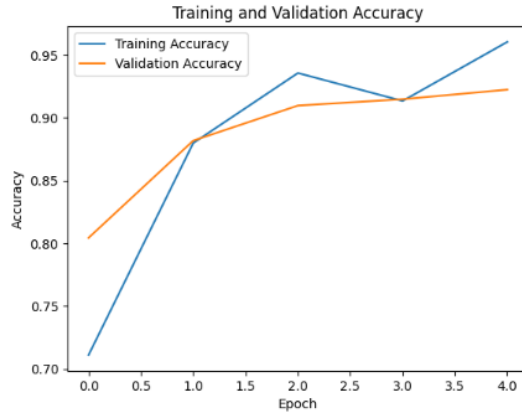
Gambar 4. 24. Grafik *Training Loss* dan *Validation Loss* Skenario 4

Grafik akurasi menunjukkan tren yang lebih fluktuatif dibandingkan skenario sebelumnya, sementara grafik *loss* menunjukkan bahwa *training loss* menurun dengan baik, tetapi *validation loss* tidak mengalami perbaikan yang signifikan, mengindikasikan potensi *overfitting* yang ringan.

2.5. Skenario 5

Pada Skenario 5, akurasi data *training* sebesar 0,96 dan *validation* sebesar 0,92. Hasil ini menunjukkan adanya *overfitting* yang cukup jelas. *Learning rate* yang lebih besar tampaknya menyebabkan model belajar terlalu cepat sehingga terlalu spesifik terhadap data *training*,

yang mengurangi kemampuan generalisasi terhadap data baru. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 5.



Gambar 4. 25. Grafik Akurasi *Training* dan *Validation* Skenario 5

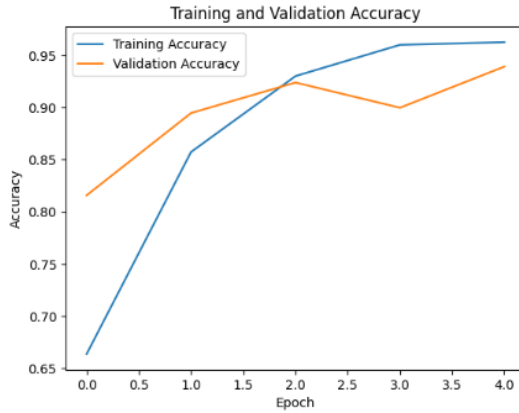


Gambar 4. 26. Grafik *Training Loss* dan *Validation Loss* Skenario 5

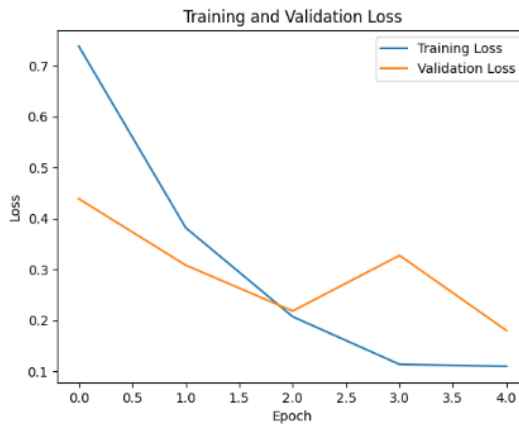
Grafik akurasi menunjukkan kesenjangan yang cukup besar antara training dan validation, yang menjadi indikasi utama dari *overfitting*. Grafik loss juga menunjukkan bahwa validation loss berhenti menurun lebih awal dibandingkan training loss, menandakan bahwa model tidak mampu belajar secara efektif pada data baru.

2.6. Skenario 6

Pada Skenario 6, akurasi data *training* sebesar 0,96 dan *validation* sebesar 0,94, menunjukkan bahwa *overfitting* masih terjadi, tetapi lebih terkendali dibandingkan dengan Skenario 5. Penggunaan *batch size* yang lebih besar tampaknya membantu dalam meningkatkan stabilitas, tetapi tetap belum cukup untuk sepenuhnya mengatasi *overfitting*. Berikut merupakan grafik akurasi *training* dan *validation* serta *training loss* dan *validation loss* pada skenario 6.



Gambar 4. 27. Grafik Akurasi *Training* dan *Validation* Skenario 6



Gambar 4. 28. Grafik *Training Loss* dan *Validation Loss* Skenario 6

Grafik akurasi menunjukkan pola yang hampir serupa dengan Skenario 5, tetapi dengan selisih yang lebih kecil antara *training* dan *validation*. Grafik *loss* menunjukkan bahwa model masih mengalami kesulitan dalam mencapai

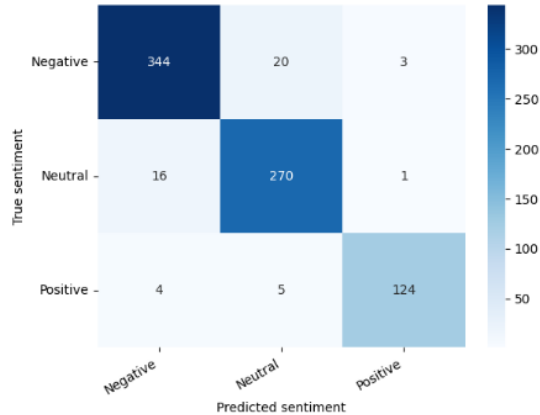
generalisasi yang optimal, meskipun tidak separah Skenario 5.

Secara keseluruhan, dari keenam skenario yang diuji, Skenario 3 memiliki akurasi *training* tertinggi tetapi menunjukkan indikasi awal *overfitting*. Skenario 1 menjadi kandidat terbaik karena menunjukkan keseimbangan optimal antara akurasi data *training* dan *validation*. Model yang dihasilkan dalam skenario ini memiliki kemampuan generalisasi yang baik dengan perbedaan yang kecil antara performa *training* dan *validation*, yang mengindikasikan minimnya risiko *overfitting*. Selain itu, grafik akurasi dan loss menunjukkan tren yang stabil, memastikan bahwa model belajar dengan baik tanpa mengalami penurunan performa pada data baru. Oleh karena itu, Skenario 1 direkomendasikan untuk digunakan dalam implementasi model guna mendapatkan hasil yang optimal.

3. Evaluasi Matrix

Setelah dilakukan pelatihan dan validasi menggunakan skenario 1, dengan *learning rate* sebesar $2e-5$, *batch size* 16 dan *epoch* 5, model menunjukkan performa yang sangat baik. Pada proses validasi, model mencapai akurasi awal sebesar 94% dengan presisi

sebesar 94%, *recall* sebesar 94% dan *f1-score* sebesar 94%. Berikut merupakan diagram *confusion matrix* dan performa dari model dengan menggunakan *learning rate* 2e-5, *batch size* 16 dan *epoch* 5.



Gambar 4. 29. Diagram *Confusion Matrix* Data Validasi

Dari *confusion matrix* tersebut jika digambarkan dalam bentuk tabel sebagai berikut.

Tabel 4. 14. *Confusion Matrix* Data Validasi

True \ Pred	Negatif	Netral	Positif	Total
Negatif	344 (TP)	20 (FN)	3 (FN)	367
Netral	16 (FN)	270 (TP)	1 (FN)	286
Positif	4 (FN)	5 (FN)	124 (TP)	133
Total	364	295	128	787

Untuk mengevaluasi performa model dalam penelitian ini, akan dihitung nilai *accuracy*, *precision*, *recall*, dan *f1-score* baik secara manual maupun otomatis. Perhitungan manual dilakukan menggunakan rumus

matematis, sedangkan perhitungan otomatis dibantu oleh program *python*.

Sebagai langkah awal dalam mengevaluasi performa model, perhitungan secara manual akan dilakukan terlebih dahulu sebelum beralih ke metode otomatis. Berikut merupakan penjabaran perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1-score* secara manual.

1. *Accuracy*

$$accuracy = \frac{TP}{Total\ Data}$$

$$accuracy = \frac{344+270+124}{787} = \frac{738}{787} = 0,9377 \text{ (93,77\%)}$$

2. *Precision* tiap kelas

- Negatif

$$precision\ neg = \frac{TP}{TP + FP}$$

$$precision\ neg = \frac{344}{344 + 16 + 4} = \frac{344}{364} = 0,9451 \text{ (94,51\%)}$$

- Netral

$$precision\ net = \frac{TP}{TP + FP}$$

$$precision\ net = \frac{270}{270 + 20 + 5} = \frac{270}{295} = 0,9153 \text{ (91,53\%)}$$

- Positif

$$precision\ pos = \frac{TP}{TP + FP}$$

$$precision\ pos = \frac{124}{124 + 3 + 1} = \frac{127}{128} = 0,9688 \text{ (96,88\%)}$$

3. *Recall* setiap kelas

- Negatif

$$recall\ neg = \frac{TP}{TP + FN}$$

$$recall\ neg = \frac{344}{344 + 20 + 3} = \frac{344}{367} = 0,9377 \text{ (93,77\%)}$$

- Netral

$$recall\ net = \frac{TP}{TP + FN}$$

$$recall\ net = \frac{270}{270 + 16 + 1} = \frac{270}{287} = 0,9408 \text{ (94,08\%)}$$

- Positif

$$recall\ pos = \frac{TP}{TP + FN}$$

$$recall\ pos = \frac{124}{124 + 4 + 5} = \frac{124}{133} = 0,9323 \text{ (93,23\%)}$$

4. *F1-score* setiap kelas

- Negatif

$$f1 - score\ neg = 2 \times \frac{precision \times recall}{precision + recall}$$

$$f1 - score\ neg = 2 \times \frac{0,9451 \times 0,9377}{0,9451 + 0,9377} = 0,9414 \text{ (94,14\%)}$$

- Netral

$$f1 - score\ net = 2 \times \frac{precision \times recall}{precision + recall}$$

$$f1 - score\ net = 2 \times \frac{0,9153 \times 0,9408}{0,9153 + 0,9408} = 0,9279 \text{ (92,79\%)}$$

- Positif

$$f1 - score\ pos = 2 \times \frac{precision \times recall}{precision + recall}$$

$$f1 - score\ pos = 2 \times \frac{0,9688 \times 0,9323}{0,9688 + 0,9323} = 0,9502 \text{ (95,02\%)}$$

Berikut merupakan hasil perhitungan manual dari performa model dalam bentuk tabel.

Tabel 4. 15. Hasil Perhitungan Manual Performa Model

Matrik	Negatif	Netral	Positif	Rata-rata Makro
Presisi	94,51%	91,53%	96,88%	94.18%
Recall	93,77%	94,08%	93,23%	93,69%
F1-Score	94,14%	92,79%	95,02%	93.98%
Akurasi	93.77%			

Kemudian, untuk perhitungan nilai *accuracy*, *precision*, *recall* dan *f1-score* secara otomatis, dapat menggunakan program bahasa *Python*. Berikut merupakan *source code* yang digunakan untuk menghitung nilai *accuracy*, *precision*, *recall* dan *f1-score* secara otomatis.

```
val_real = val_df.category
val_pred = val_df.pred
print(classification_report(val_real, val_pred, target_names=['Negative', 'Neutral', 'Positive']))
```

Gambar 4. 30. *Source Code* Perhitungan Otomatis Performa Model

Dari program tersebut akan menampilkan hasil dari performa model melalui nilai *accuracy* serta nilai *precision*, *recall* dan *f1-score* dari setiap kelas sentimen. Berikut merupakan hasil dari perhitungan nilai *accuracy*, *precision*, *recall* dan *f1-score* secara otomatis.

	precision	recall	f1-score	support
Negative	0.95	0.94	0.94	367
Neutral	0.92	0.94	0.93	287
Positive	0.97	0.93	0.95	133
accuracy			0.94	787
macro avg	0.94	0.94	0.94	787
weighted avg	0.94	0.94	0.94	787

Gambar 4. 31. Hasil Perhitungan Otomatis Performa Model

Berdasarkan hasil perhitungan manual dan otomatis, model klasifikasi sentimen menunjukkan performa yang sangat baik dengan akurasi tinggi dan keseimbangan yang baik antara presisi, recall, serta F1-score. Akurasi yang diperoleh secara manual sebesar 93,77%, sedangkan secara otomatis menunjukkan hasil 94%, dengan perbedaan yang sangat kecil yaitu 0.23% yang kemungkinan disebabkan oleh pembulatan angka.

Selain itu, nilai presisi, recall, dan F1-score rata-rata makro yang diperoleh juga hampir identik, yaitu sekitar 94%, menunjukkan bahwa model bekerja secara konsisten dalam mengklasifikasikan sentimen negatif, netral, dan positif. Secara keseluruhan, hasil ini menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan teks sesuai dengan sentimennya.

G. Pengujian Akhir Model

Pada tahap ini, model terpilih akan diuji menggunakan dua metode evaluasi yang berbeda. Pertama,

model diuji dengan dataset uji yang telah berlabel, yaitu data yang sebelumnya telah disiapkan melalui proses pembagian data. Pengujian ini bertujuan untuk mengukur sejauh mana model mampu mengklasifikasikan data dengan benar berdasarkan label yang telah diketahui.

Pengujian kedua dilakukan dengan menginputkan kalimat acak ke dalam model, tanpa label sebelumnya. Model diharapkan mampu memprediksi apakah kalimat tersebut memiliki sentimen negatif, netral, atau positif berdasarkan pola yang telah dipelajari selama proses pelatihan. Pengujian ini bertujuan untuk melihat performa model dalam situasi nyata dimana data baru yang tidak berlabel harus diklasifikasikan secara otomatis.

1. Pengujian dengan Dataset Berlabel

Pada pengujian pertama, model akan diuji menggunakan dataset uji yang terdiri dari 389 data, yang terdiri dari 181 data berlabel negatif, 141 data berlabel netral, dan 67 data berlabel positif. Berikut merupakan *source code* yang digunakan pada pengujian dengan dataset berlabel.

```

# Prediksi test set
model.eval()
torch.set_grad_enabled(False)

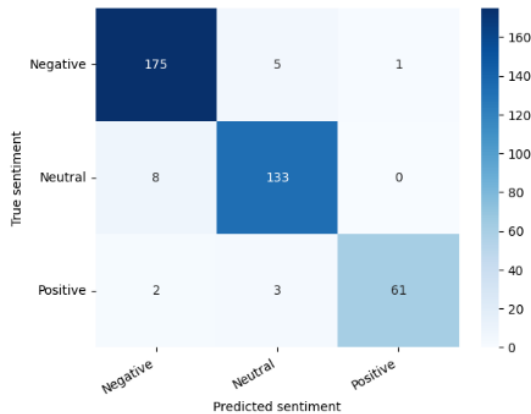
total_loss, total_correct, total_labels = 0, 0, 0
pred, list_label = [], []

pbar = tqdm(test_loader, leave=True, total=len(test_loader))
for i, batch_data in enumerate(pbar):
    _, batch_hyp, _ = forward_sequence_classification(model, batch_data[:-1], i2w=i2w, device="cuda")
    pred += batch_hyp

```

Gambar 4. 32. *Source Code* Pengujian Dataset Berlabel

Dari proses pengujian menggunakan dataset uji berlabel, didapatkan hasil *confusion matrix* sebagai berikut.



Gambar 4. 33. *Confusion Matrix* Dataset Uji

Berdasarkan *confusion matrix*, model berhasil memprediksi 175 dari 181 data negatif dengan benar, meskipun masih terdapat 5 data yang diklasifikasikan sebagai netral dan 1 data yang salah diklasifikasikan sebagai positif. Untuk label netral, model berhasil mengenali 133 dari 141 data dengan benar, dengan 8 data salah diklasifikasikan sebagai negatif. Sementara itu, untuk label positif, model berhasil

mengklasifikasikan 61 dari 66 data dengan benar, dengan 3 data salah diklasifikasikan sebagai netral dan 2 data sebagai negatif. Berikut merupakan hasil dari performa model menggunakan dataset uji.

	precision	recall	f1-score	support
Negative	0.95	0.97	0.96	181
Neutral	0.94	0.94	0.94	141
Positive	0.98	0.92	0.95	66
accuracy			0.95	388
macro avg	0.96	0.94	0.95	388
weighted avg	0.95	0.95	0.95	388

Gambar 4. 34. Hasil Performa Model Dataset Uji

Dari hasil pengujian pertama menggunakan dataset uji berlabel, diperoleh akurasi sebesar 95%, yang menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan teks sesuai dengan sentimennya. Secara keseluruhan, model memiliki *macro average precision*, *recall*, dan *f1-score* sebesar 0.95, serta *weighted average* sebesar 0.95, yang mengindikasikan performa model yang cukup stabil dalam mengenali ketiga kategori sentimen.

2. Pengujian dengan Kalimat Acak

Pada pengujian kedua, model diuji dengan kalimat acak dan belum berlabel yang tidak termasuk dalam dataset latih, dataset validasi dan dataset uji. Sebelum melakukan pengujian kedua, model dan tokenizer yang telah dilatih harus disimpan terlebih dahulu dan

kemudian akan dimuat kembali untuk melakukan klasifikasi sentimen pada kalimat acak tersebut. Berikut *source code* untuk menyimpan dan memuat model yang sudah dilatih.

```
# Simpan model dan tokenizer
output_dir = "./fine_tuned_indo_bert"
model.save_pretrained(output_dir)
tokenizer.save_pretrained(output_dir)

print(f"Model dan tokenizer berhasil disimpan di {output_dir}")
```

Gambar 4. 35. *Source Code* Menyimpan Model

Model dan tokenizer *fine-tuning* sebelumnya akan disimpan didalam folder *fine_tuned_indo_bert*. Setelah disimpan, model dan tokenizer akan dimuat ulang menggunakan *source code* berikut..

```
# Load model dan tokenizer
saved_model_path = "./fine_tuned_indo_bert"
tokenizer = BertTokenizer.from_pretrained(saved_model_path)
model = BertForSequenceClassification.from_pretrained(saved_model_path)

print("Model dan tokenizer berhasil dimuat!")
```

Gambar 4. 36. *Source Code* Memuat Model

Setelah dimuat model dan tokenizer akan digunakan untuk melakukan klasifikasi pada kalimat acak. Kalimat akan diinputkan manual, lalu model akan menebak kalimat tersebut termasuk ke dalam sentimen negatif, netral atau positif. Berikut *source code* yang digunakan untuk melakukan klasifikasi sentimen pada kalimat acak.

```

# Pengujian akhir
def load_model(model_path):
    tokenizer = BertTokenizer.from_pretrained(model_path)
    model = BertForSequenceClassification.from_pretrained(model_path)
    return tokenizer, model

def predict_sentiment(text, tokenizer, model):
    inputs = tokenizer(text, return_tensors='pt', padding=True, truncation=True, max_length=128)

    with torch.no_grad():
        outputs = model(**inputs)

    logits = outputs.logits
    predicted_class = torch.argmax(logits, dim=1).item()

    return predicted_class

if __name__ == "__main__":
    model_path = "./fine_tuned_indo_bert"
    tokenizer, model = load_model(model_path)
    text = input("Masukkan teks untuk analisis sentimen: ")
    sentiment = predict_sentiment(text, tokenizer, model)
    sentiment_labels = {0: "positive", 1: "neutral", 2: "negative"}
    sentiment_label = sentiment_labels.get(sentiment, "Unknown")

    print(f"Hasil analisis sentimen: {sentiment_label}")

```

Gambar 4. 37. Source Code Klasifikasi Kalimat Acak

Dalam pengujian kedua ini, kalimat yang digunakan adalah *"Gaji sudah pas-pasan, masih kena potongan Tapera"* yang memiliki ekspresi negatif, *"Kalau dikelola dengan baik, Tapera bisa jadi solusi sih."* yang memiliki ekspresi positif, dan *"Sistem Tapera ini sebenarnya seperti apa sih?"* yang memiliki ekspresi netral.

Dari ketiga kalimat yang diinputkan, model berhasil menebak dengan tepat sentimen masing-masing kalimat. Berikut adalah hasil pengujian yang menunjukkan klasifikasi sentimen yang dilakukan oleh model.

```

Masukkan teks untuk analisis sentimen: Gaji sudah pas-pasan, masih kena potongan Tapera
Hasil analisis sentimen: negative

```

Gambar 4. 38. Hasil Pengujian Kalimat Pertama

Masukkan teks untuk analisis sentimen: Kalau dikelola dengan baik, Tapera bisa jadi solusi sih.
 Hasil analisis sentimen: positive

Gambar 4. 39. Hasil Pengujian Kalimat Kedua

Masukkan teks untuk analisis sentimen: "Sistem Tapera ini sebenarnya seperti apa sih?
 Hasil analisis sentimen: neutral

Gambar 4. 40. Hasil Pengujian Kalimat Ketiga

Hasilnya menunjukkan bahwa model mampu mengklasifikasikan sebagian besar kalimat dengan akurat, terutama untuk kalimat yang memiliki ekspresi sentimen yang jelas. Namun, dalam beberapa kasus, model mengalami kesulitan dalam membedakan antara sentimen netral dan sentimen lainnya, yang dapat disebabkan oleh kesamaan pola kata atau frasa yang digunakan dalam kalimat.

3. Hasil Keseluruhan Pengujian

Secara keseluruhan, hasil pengujian menunjukkan bahwa model memiliki kemampuan klasifikasi yang baik dan generalisasi yang cukup kuat. Namun, untuk meningkatkan performa lebih lanjut, dapat dilakukan penambahan data pelatihan yang lebih beragam, terutama pada label yang memiliki distribusi data lebih kecil.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilaksanakan, dapat disimpulkan bahwa :

1. Dalam implementasi model IndoBERT untuk analisis teks narasi kebijakan Tapera, penelitian ini menggunakan model IndoBERT-Base-Uncased yang telah melalui proses *fine-tuning*. Tahapan utama dalam implementasi mencakup preprocessing data, yang meliputi *case folding*, *spell normalized*, serta *cleaned text*. Untuk menangani ketidakseimbangan kelas dalam dataset, dilakukan *Easy Data Augmentation* (EDA) guna menambah jumlah data pada kategori sentimen positif dan netral. Proses tokenisasi dilakukan dengan WordPiece Tokenization yang disediakan oleh IndoBERT. Model dilatih dengan enam skenario *hyperparameter*, dimana skenario terbaik diperoleh pada kombinasi *learning rate* 2e-5, *batch size* 16 dan *epoch* 5, yang memberikan hasil paling optimal.
2. Evaluasi performa model IndoBERT dilakukan melalui pengujian dengan data latih, data validasi dan data uji serta kalimat acak tanpa label. Hasil pengujian

menunjukkan bahwa model memiliki akurasi tinggi, dengan akurasi pada data latih sebesar 96% dan akurasi pada data validasi sebesar 94%, yang menunjukkan bahwa model mampu belajar dengan baik tanpa mengalami *overfitting* yang signifikan. Pada data uji, model mencapai akurasi sebesar 95%, dengan precision, recall, dan F1-score rata-rata makro sebesar 95%. *Confusion matrix* menunjukkan bahwa model dapat mengenali sentimen negatif dan positif dengan baik, tetapi masih mengalami sering kesalahan dalam mengklasifikasi sentimen netral. Dalam pengujian dengan kalimat acak, model mampu mengklasifikasikan sentimen dengan cukup akurat, tetapi tetap menghadapi tantangan dalam membedakan ekspresi netral dengan sentimen lainnya dan kalimat yang kurang jelas maknanya. Secara keseluruhan, penelitian ini menunjukkan bahwa IndoBERT dapat menjadi alat yang efektif dalam menganalisis sentimen masyarakat terhadap kebijakan publik di media sosial.

B. Saran

Berdasarkan hasil penelitian ini, terdapat beberapa rekomendasi untuk penelitian dan pengembangan lebih lanjut sebagai berikut.

1. Peningkatan kualitas data sangat diperlukan untuk memastikan representasi yang lebih seimbang antara sentimen negatif, netral, dan positif. Dataset dalam penelitian ini masih didominasi oleh sentimen negatif, sehingga pengumpulan data tambahan, baik dari Twitter maupun dari sumber lain seperti berita online dan forum diskusi, dapat membantu mendapatkan perspektif yang lebih beragam.
2. Model IndoBERT yang telah dikembangkan dalam penelitian ini dapat diuji pada platform media sosial lain, seperti Facebook, Instagram, atau YouTube. Hal ini penting untuk mengetahui bagaimana sentimen masyarakat terhadap kebijakan Tapera berkembang di berbagai kanal media sosial, yang mungkin memiliki karakteristik audiens yang berbeda. Dengan demikian, model dapat lebih fleksibel dalam menangkap opini publik dari berbagai sumber.

DAFTAR PUSTAKA

- Aditya Permana, A., Ajeng Kristiyanti, D., & Sihotang, M. (2021). Analisis Sentimen Pendapat Masyarakat Terhadap PPKM Darurat Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes Classifier (NBC). In *JIKA: Vol. ISSN*.
- Amien, M. (2023). Sejarah dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia: Tinjauan tentang sejarah, perkembangan teknologi, dan aplikasi NLP dalam bahasa Indonesia. *ArXiv Preprint ArXiv, 2007*, 1–7. <http://arxiv.org/abs/2304.02746>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1*, 4171–4186.
- Dinilah, A. (2021). Pornografi Pada Akun Media Sosial Twitter. *Program Studi Kriminologi Fakultas Ilmu Sosial Dan Ilmu Politik Universitas Islam Riau Pekanbaru*.
- Gangal, A., Shrivastava, A., Hussien, N. M., Singh, P., Diwakar, M., Joshi, K., Bisht, S., & Joshi, N. C. (2023). Natural language processing: A review. *AIP Conference Proceedings*,

- 2771(1). <https://doi.org/10.1063/5.0153960>
- Garcia, K., & Berton, L. (2021). Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing*, 101. <https://doi.org/10.1016/j.asoc.2020.107057>
- Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *ArXiv Preprint ArXiv*, 1–17. <https://doi.org/10.48550>
- Hafidhigbal, N. (2024). Implementasi model indobert untuk klasifikasi jenis tindak pidana korupsi berbasis web application. In *Repository.Uinjkt.Ac.Id*. [https://repository.uinjkt.ac.id/dspace/handle/123456789/79800%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/79800/1/NAUFALDI HAFIDHIGBAL-FST.pdf](https://repository.uinjkt.ac.id/dspace/handle/123456789/79800%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/79800/1/NAUFALDI%20HAFIDHIGBAL-FST.pdf)
- Hidayat, I. R., & Maharani, W. (2022). General Depression Detection Analysis Using IndoBERT Method. *International Journal on Information and Communication Technology (IJoICT)*, 8(1). <https://doi.org/10.21108/ijoict.v8i1.634>
- Hovipah, M., Hearani, E., Jasril, J., & Syafria, F. (2023). Klasifikasi Clickbait Menggunakan Transformers. *Jurnal CoSciTech (Computer Science and Information Technology)*, 4(1), 172–181. <https://doi.org/10.37859/coscitech.v4i1.4713>

- Juarto, B., & Yulianto. (2023). Indonesian News Classification Using IndoBert. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2).
- Karimah, N., & Baita, A. (2024). Multi-Aspect Sentiment Analysis Pada Review Film Menggunakan Metode Bidirectional Encoder Representations From Transformers (BERT) Multi-Aspect Sentiment Analysis of Film Review Using Bidirectional Encoder Representations from Transformers (BERT). *Jurnal Sistem Komputer*, 13(1), 2020. <https://doi.org/10.34010/komputika.v13i1.11098>
- Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 757–770. <https://doi.org/10.18653/v1/2020.coling-main.66>
- Kurniasih, A., & Manik, L. P. (2022). On the Role of Text Preprocessing in BERT Embedding-based DNNs for Classifying Informal Texts. *International Journal of Advanced Computer Science and Applications*, 13(6), 927–934. <https://doi.org/10.14569/IJACSA.2022.01306109>
- Pradana, L. S. (2024). Analisis Sentimen Masyarakat Media Sosial Twitter Terhadap Kinerja Penjabat Gubernur DKI Jakarta Menggunakan Model Indobert [UIN Syarif

Hidayatullah].

<https://repository.uinjkt.ac.id/dspace/handle/123456789/77071>

Pramudia, B. S., Setiyawati, D., Rinaryanta, N. D., & Savitri, S. J. (2024). Keadilan Sosial Dalam Program Tapera : Analisis Jurnal Hukum Terapan dan Inovasi Hukum. *Jurnal Hukum Terapan Dan Inovasi Hukum*, 6(3), 34–50.

Rahmatullah, B. (2021). Sentiment Analysis Pelaksanaan Work From Home di Indonesia pada Masa Pandemi COVID-19 Menggunakan IndoBERT. In *Institut Teknologi Sepuluh Nopember*.

https://repository.its.ac.id/91493/1/06111640000078-Undergraduate_Thesis.pdf

Ramadhan, Dimas, F., Noertjahjono, S., & Irawan, Joseph, D. (2020). Penerapan Chatbot Auto Reply Pada Whatsapp Sebagai Pusat Informasi Praktikum Menggunakan Artificial Intelligence Markup Language. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4(1), 198–205. <https://doi.org/10.36040/jati.v4i1.2375>

Rayyan, F. (2022). Pengembangan chatbot untuk aplikasi online chat telegram dengan pendekatan klasifikasi emosi pada teks menggunakan metode indobert-lite. In *Repository.Uinjkt.Ac.Id*.

Setiawan, D., Karuniawati, E. A. D., & Janty, S. I. (2023). Peran Chat Gpt (Generative Pre-Training Transformer) Dalam

Implementasi Ditinjau Dari Dataset. *INNOVATIVE: Journal of Social Science Research*, 3(3), 9527–9539. <https://j-innovative.org/index.php/Innovative/article/view/3286>

6

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text Data Augmentation for Deep Learning. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00492-0>

Sofi, N., Sulistyorini, T., & Nazaruddin, M. (2023). Analisis Sentimen Masyarakat Pengguna Media Sosial Twitter Terhadap Motogp Mandalika Lombok Menggunakan Metode Bidirectional Encoder Representation From Transformers (BERT). *Jurnal Informasi, Sains, Dan Teknologi*, 06(1).

Sulistyowati, Maharani, D. N., Maharaja, G. B. M., & Carnely Kahe, A. M. (2024). Hubungan Pemerintah Dan Rakyat Dalam Perspektif Hukum Administrasi Negara. *Journal of Indonesian Rural and Regional Government*, 8(1), 10–20. <https://doi.org/10.47431/jirreg.v8i1.1386>

Susanto, A. (2023). Keterlibatan Digital Presiden Joko Widodo Melalui Twitter Pada Seputar Pelantikannya Sebagai Presiden Indonesia 2019-2024. *JURNAL SISTEM INFORMASI UNIVERSITAS SURYADARMA*, 10(1). <https://doi.org/10.35968/jsi.v10i1.1003>

Tania, N., Novienko, J., & Sanjaya, D. (2021). Kajian Teori Hukum Progresif Terhadap Implementasi Produk

- Tabungan Perumahan Rakyat. *Perspektif*, 26(2).
<https://doi.org/10.30742/perspektif.v26i2.800>
- Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). Natural Language Processing Advancements By Deep Learning: A Survey. *The Scientific Temper*, 14(04).
<https://doi.org/10.48550>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem*, 5999–6009.
- Wandani, A., Fauziah, & Andrianingsih. (2021). Sentimen Analisis Pengguna Twitter pada Event Flash Sale Menggunakan Algoritma K-NN, Random Forest, dan Naive Bayes. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2).
- Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 6382–6388.
<https://doi.org/10.18653/v1/d19-1670>
- Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., &

- Purwarianti, A. (2020). *IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding*. <http://arxiv.org/abs/2009.05387>
- Zein, A., Farizy, S., & Suharyanto, E. (2022). Sentimen Analisis Pada Komentar Pendek Evaluasi Dosen Oleh Mahasiswa (Edom) Program Studi Sistem Informasi Universitas Pamulung. *Jurnal Ilmu Komputer*, 5(01).

LAMPIRAN-LAMPIRAN

Lampiran 1 : Contoh Dokumen Hasil *Crawling* Data Twitter

No	created_at	full_text
1	Tue Jun 04 15:30:39 +0000 2024	Rumah hasil TAPERA selama 20 tahun https://t.co/m4gJbdJRF7
2	Tue Jun 04 02:48:42 +0000 2024	Keren ini. Rohingya ditampung Gaza ditampung Rakyat sendiri menghadapi harga beras naik BBM naik UKT naik Sembako naik transport naik cicilan naik pajak naik potongan Tapera potongan BPJS Kesehatan potongan ketenagakerjaan terkena pinjol dll https://t.co/wZCIWAsPRG
3	Sat Jun 01 06:36:03 +0000 2024	Penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong Tapera. https://t.co/vfHIPomXY2
4	Sun Jun 02 07:19:16 +0000 2024	Beramai-ramai endorse Tapera https://t.co/qNjK0zu11M
5	Fri Jun 07 11:12:55 +0000 2024	Untuk apa Tapera kalau presiden terpilih aja berjanji akan bangun 3 juta rumah https://t.co/BJKNw6f3jx
*****	*****	*****
2600	Mon Apr 15 05:10:47 +0000 2024	@BachtiaNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtiaNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan

2601	Fri Apr 19 09:51:15 +0000 2024	Kepada seluruh pegawai agar segera memenuhi data yang diminta seperti BPJS Taspen dan Tapera sesuai dengan nota dinas yang telah di share melalui grup WA (whatsapp). Pengisian ini diharapkan segera diselesaikan mengingat ada deadline yang harus dipenuhi. Ucap Widhi https://t.co/IYPbIEdX6K
2602	Thu Apr 25 02:01:33 +0000 2024	Selain skema KPR dengan tenor 35 tahun ada peluang KPR Tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/DTlGxwW9e3 #Opini #AdadiKompas
2603	Sat Apr 27 15:07:31 +0000 2024	@BachtiaNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtiaNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan
2604	Mon Apr 15 05:10:47 +0000 2024	@Lialarsen @Alfred67686158 @BlessingMombes3 @KingJayZim @classic263 @TeamFuloZim Apa hanzi tapera

Lampiran 2 : Contoh Dokumen Data Yang Sudah Diberi Label

No	created_at	full_text	Label
1	Tue Jun 04 15:30:39 +0000 2024	Rumah hasil TAPERA selama 20 tahun https://t.co/m4gJbdJRf7	neutral
2	Tue Jun 04 02:48:42 +0000 2024	Keren ini. Rohingya ditampung Gaza ditampung Rakyat sendiri menghadapi harga beras naik BBM naik UKT naik Sembako naik transport naik cicilan naik pajak naik potongan Tapera potongan BPJS Kesehatan potongan ketenagakerjaan terkena pinjol dll https://t.co/wZCIWAsPRG	negative
3	Sat Jun 01 06:36:03 +0000 2024	Penghasilan ojol tuh udah dipotong biaya aplikasi yg gede dianggap mitra doang engga ada jaminan kesehatan engga ada jaminan pensiun udah overwork kejar target malah mau dipotong Tapera. https://t.co/vfHIPomXY2	negative
4	Sun Jun 02 07:19:16 +0000 2024	Beramai-ramai endorse Tapera https://t.co/qNjk0zu11M	neutral
5	Fri Jun 07 11:12:55 +0000 2024	Untuk apa Tapera kalau presiden terpilih aja berjanji akan bangun 3 juta rumah https://t.co/BJKNw6f3jx	negative
****	*****	*****	*****

2600	Mon Apr 15 05:10:47 +0000 2024	@BachtiarNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtiarNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan	neutral
2601	Fri Apr 19 09:51:15 +0000 2024	Kepada seluruh pegawai agar segera memenuhi data yang diminta seperti BPJS Taspen dan Tapera sesuai dengan nota dinas yang telah di share melalui grup WA (whatsapp). Pengisian ini diharapkan segera diselesaikan mengingat ada deadline yang harus dipenuhi. Ucap Widhi https://t.co/IYPbIEdX6K	neutral
2602	Thu Apr 25 02:01:33 +0000 2024	Selain skema KPR dengan tenor 35 tahun ada peluang KPR Tapera bagi masyarakat berpenghasilan rendah dan milenial. https://t.co/DTIGxwW9e3 #Opini #AdadiKompas	positive
2603	Sat Apr 27 15:07:31 +0000 2024	@BachtiarNU Assalamu alaikum Wr.Wb Sahabat BTN Syariah @BachtiarNU kami informasikan saat ini pengajuan Pembiayaan KPR BTN Subsidi iB dapat dilakukan oleh siapa saja dan dapat dilakukan	neutral

		melalui aplikasi SiKasep Online/Tapera Mobile. Terima kasih. Wassalamu alaikum Wr.Wb. ^Joan	
2604	Mon Apr 15 05:10:47 +0000 2024	@Lialarsen @Alfred67686158 @BlessingMombes3 @KingJayZim @classic263 @TeamFuloZim Apa hanzi tapera	neutral

Lampiran 3 : Dokumentasi Penyerahan Dataset Untuk Dilakukan Pelabelan Oleh Ahli Bahasa Indonesia



Lampiran 4 : Contoh Data Pendukung "colloquial-indonesian-lexicon.csv"

No	Slang	Normal
1	met	selamat
2	gaharus	enggak harus
3	aminn	amin
4	woww	wow
5	netass	menetas
6	eeeehhhh	eh
7	kata2nyaaa	kata-katanya
8	hallo	halo
9	kaka	kakak
10	ka	kak
11	daah	dah
12	aaaaahhhh	ah
13	yaa	ya
14	smga	semoga
15	slalu	selalu
.....		
15394	d1m	dalam
15395	wktu	waktu
15396	hr	hari
15397	gatau	enggak tau
15398	fans2	fan-fan
15397	gaharus	enggak harus

Lampiran 4 : Contoh Data Pendukung "kbba.txt"

No	Slang	Normal
1	abis	habis
2	7an	tujuan
3	ad	ada
4	adlah	adalah
5	adlh	adalah
6	adoh	aduh
7	afaik	as far as i know
8	aha	tertawa
9	ahaha	haha
10	aing	saya
11	aj	saja
12	aja	saja
13	ajep-ajep	dunia gemerlap
14	ajj	saja
15	ak	saya
*****	*****	*****
1313	almaidahyg	almaidah
1314	istaa	nista
1315	serasiii	serasi
1316	wkwkkw	tertawa
1317	sayan	sayang

Lampiran 5 : Contoh Data Pendukung "slang.csv"

No	Slang	Normal
1	udah	sudah
2	gua	aku
3	ntar	nanti
4	be like	seperti
5	aja	saja
6	dah	sudah
7	anjir	anjing
8	beginimana	bagaimana
9	online	daring
10	offline	luring
11	gpp	tidak apa
12	the	sebuah
13	struggle	perjuangan
14	is	adalah
15	real	nyata
*****	*****	*****
1541	tntang	tentang
1542	ripyuu	tinjau
1543	sinyal	jaringan
1544	meneh	lagi
1545	anaku	anak

Lampiran 6 : Contoh Data Pendukung "slangword.txt"

No	Slang	Normal
1	&	dan
2	+	tambah
3	/	atau
4	=	sama dengan
5	ababil	anak ingusan
6	abal2	palsu
7	abal	palsu
8	ad	ada
9	akooh	aku
10	alay	norak
11	albm	album
12	ampe	sampai
13	anjir	waw
14	anyway	ngomong-ngomong
15	aq	aku
.....		
282	woles	santai
283	wtf	apa-apaan
284	x	kali
285	y	ya
286	yg	yang

Lampiran 6 : Contoh Data Hasil Augmentasi

cleaned_text	Augmentasi 1	Augmentasi 2
beramairamai endorse tapera	beramairamai tapera endorse	endorse beramairamai tapera
cek fakta benarkah tapera bisa atasi backlog perumahan	cek fakta benarkah tapera atasi backlog perumahan	cek fakta benarkah atasi tapera bisa atasi backlog perumahan
dukung dana tapera pengembangan siap kebut bangun rumah murah	dukung dana tapera pengembang siap kencang bangun rumah murah	siap dana tapera pengembang dukung kebut bangun rumah murah
rakyat sejahtera lewat program tapera bisa	rakyat senang lewat program tapera bisa	rakyat sejahtera lewat cadangan tapera bisa
jadi ini tujuan tapera buat rumah ini ternyata masuk akal sih kalau ini	jadi ini tujuan tapera buat rumah sebenarnya ini ternyata masuk akal sih kalau ini	jadi ini tujuan tapera buat rumah ini ternyata masuk akal kalau sih ini
telah memiliki tabungan tapera dilaterbelakangi oleh kerja sama antara badan pengelola bp tapera dengan pt bank tabungan negara persero tbk atau btn untuk menyediakan kpr subsidi bagi pnsbumn untuk informasi	telah mengantongi tabungan tapera dilaterbelakangi akibat kerja sama rumpang badan pengelola bp tapera dengan pt bank tabungan negara persero tbk atau btn untuk menyediakan kpr subsidi bagi pnsbumn untuk informasi	telah memiliki tabungan tapera dilaterbelakangi oleh kerja sama antara badan pengelola bp tapera bersama- sama pt bank tabungan negara persero tbk atau btn untuk menyediakan kpr sumbangan bagi pnsbumn menurut informasi

Lampiran 7 : Contoh Data Hasil Klasifikasi dengan Model IndoBERT

No	Data evaluasi	Label	Prediksi
1	kyany ini sengaja kpi deh barengan tapera sama kpi	neutral	neutral
2	menteri hadimuljono melantik komisioner deputy komisioner bp tapera baru untuk masa jabatan	neutral	neutral
3	katanya uu tapera sudah dekat ketuk palu ya	neutral	negatif
4	makin jelas tapera uangnya buat apa dan siapa	negative	neutral
5	pemerintah pastikan tapera dilanjutkan meski ramai penolakan	negative	neutral
.....			
784	keputusan jokowi izinkan ormas agama kelola tambang jadi berita populer bersamaan dengan konferensi pers bp tapera	neutral	neutral
785	inti dari tapera iya pemerintah ngutang uang ke rakyat buat diputer lagi atau buat ikn sih soalnya kalau hutang ke china sama imf lagi sudah lumayan banyak wkww	negative	negative
786	sejumlah gaji jadi komite ida tapera yang menteri per bulananya mencapai rp ada basuki hadimuljono bp fauziyah dan sri mulyani	neutral	neutral
787	di kasih tapera iuran tapera	neutral	neutral

Lampiran 8 : Daftar Riwayat Hidup

Riwayat Hidup

A. Identitas Diri

1. Nama Lengkap : Guntur Syarifuddin Putra
2. Tempat & Tanggal Lahir : Belinyu, 23 Agustus 2003
3. Alamat : Ds. Pesawahan, RT 02, RW 05,
Kec. Rawalo, Kab. Banyumas
4. HP : 081387981288
5. Email : pepesolandep@gmail.com

B. Riwayat Pendidikan

1. Sekolah Dasar (SD) Negeri 2 Belinyu
2. Madrasah Tsanawiyah (MTs) Miftahul Huda Rawalo
3. Sekolah Menengah Atas (SMA) Negeri 1 Rawalo

Semarang, 10 Maret 2025
Pembuat Pernyataan,

Guntur Syarifuddin Putra
NIM: 2108096045