

KLASIFIKASI EMOSI PADA TEKS MENGGUNAKAN METODE DEEP LEARNING



Disusun Oleh:

N a m a : Alan Tusa Bagus W

NIM : 18523148

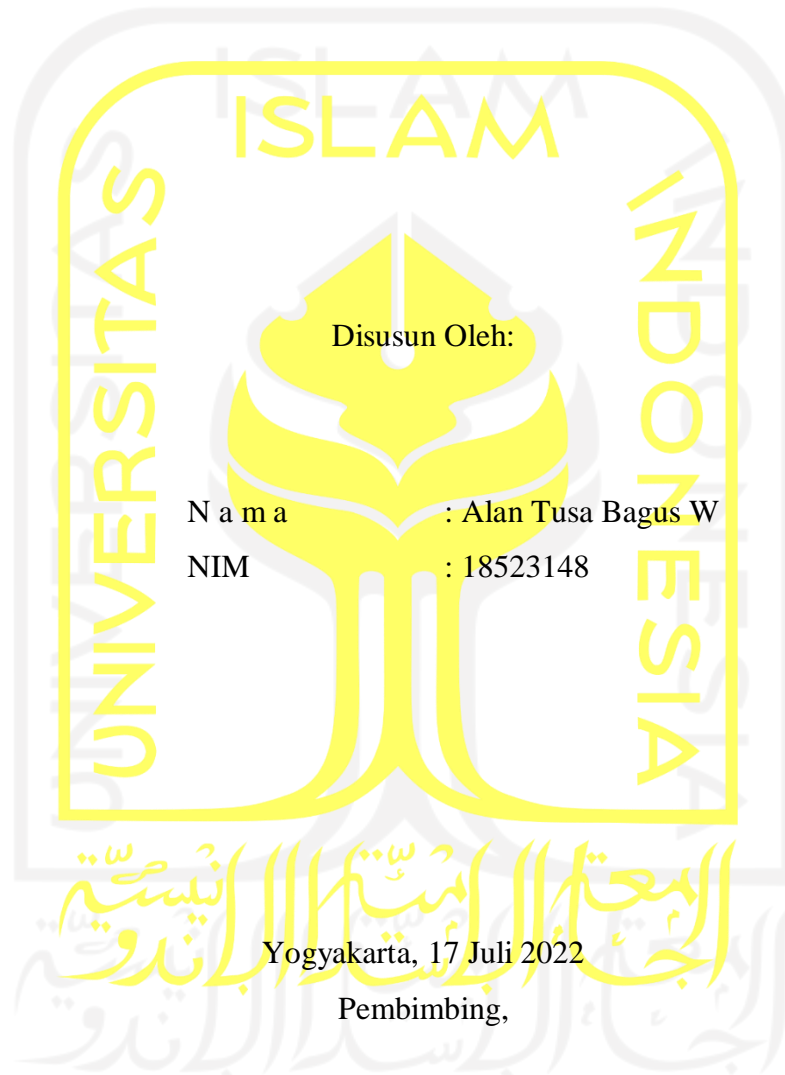
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

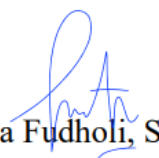
2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**KLASIFIKASI EMOSI PADA TEKS MENGGUNAKAN
METODE DEEP LEARNING**

TUGAS AKHIR




(Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D)

HALAMAN PENGESAHAN DOSEN PENGUJI

KLASIFIKASI EMOSI PADA TEKS MENGGUNAKAN METODE DEEP LEARNING

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 17 Juli 2022

Tim Penguji

Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D.

Anggota 1

Dr. Syarif Hidayat, S.Kom, M.IT.

Anggota 2

Sri Mulyati, S.Kom, M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Alan Tusa Bagus Widianto

NIM : 18523148

Tugas akhir dengan judul:

**KLASIFIKASI EMOSI PADA TEKS MENGGUNAKAN
METODE DEEP LEARNING**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 17 Juli 2022

(Alan Tusa Bagus Widainto)

HALAMAN PERSEMBAHAN

Assalamualaikum warahmatullahi wabarakatuh alhamdulillahirobbil'alamin, puji syukur kepada Allah SWT berkat ridho dan karunia-Nya yang telah memberikan kelancaran dan keberkahan selama proses pengerjaan tugas akhir ini. Semoga ilmu yang selama ini saya pelajari mendapatkan keberkahan dan akan berguna untuk orang-orang sekitar saya, Allahuma Amiin.

Terimakasih kepada Allah SWT, kedua orang tua saya yang telah banyak memberikan seluruh tenaganya sampai saya bisa diposisi seperti ini, semua proses dan perjalanan yang saya lalui tak akan lepas dari kerja keras kedua orang tua saya.

Terima kasih untuk Bapak Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D., selaku pembimbing, dan para dosen Informatika yang selalu mengajarkan ilmu baru yang tidak lupa menambahkan tentang ilmu keislaman di dalamnya yang sangat berharga bagi saya.

Untuk teman-teman saya terutama (Tio, Fitria, Albarra, Risca, Ridho, Akmal, Alm Jeje dan lainnya) yang saya banggakan, terima kasih atas segalanya, terima kasih telah memberikan saya warna baru dan pengalaman baru yang tidak akan terlupakan, semoga kita semua dapat menemukan jalannya masing-masing dan dilancarkan di dunia maupun di akhirat kelak.

Semoga Allah mengganti kebaikan-kebaikan yang kalian berikan dengan kebaikan yang lebih baik lagi. Semoga diberikan kekuatan, kelancaran, kesehatan dan kebahagiaan dunia akhirat untuk kita semua. Aamiin.

HALAMAN MOTO

“Jangan bunuh mimpimu, mimpi itu gak akan pernah mati, yang ada cuma pingsan, kemudian akan bangkit lagi dimasa tua dalam bentuk penyesalan” -Pandji Pragiwaksono

“Displin tidak mengekang, sebaliknya justru membebaskan” -Pandji Pragiwaksono

“Mentoleransi ketidaknyamanan membuat kita berkembang” -Dzawin Nur



KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan hidayah-Nya serta memberikan ridho dan kekuatan kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.

Skripsi berjudul “Klasifikasi Emosi Pada Teks Menggunakan Metode Deep Learning” yang disusun untuk memenuhi persyaratan guna mendapatkan gelar Sarjana Komputer (S.Kom) pada Program Studi Informatika di Universitas Islam Indonesia. Selama proses penyusunan skripsi, penulis banyak mendapatkan ilmu dan masukan serta dukungan dari berbagai pihak, maka dari itu penulis ingin menyampaikan terima kasih kepada:

1. Kedua orang tua penulis, Ibu dan Bapak yang telah bekerja keras dan mendidik penulis agar menjadi manusia yang bertanggung jawab dan bermanfaat bagi seluruh orang
2. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Dosen Pembimbing yang telah mengajarkan dan membimbing penulis sehingga penulis dapat menyelesaikan Penelitian ini dengan baik.
3. Seluruh Dosen dan Staff karyawan Program Studi Informatika Universitas Islam Indonesia yang telah banyak membantu dan memberikan ilmu-ilmu yang bermanfaat selama masa perkuliahan
4. Teman-teman penulis dan semua pihak yang terlibat dalam proses perkuliahan maupun di luar perkuliahan, terima kasih untuk segalanya yang telah diberikan kepada penulis

Pada akhirnya, penulis mengharapakan skripsi ini dapat bermanfaat dan berkah bagi penulis dan semua pihak yang akan menelaah di kemudian hari. Semoga Allah memberikan rahmat dan karunia-Nya untuk semua pihak yang terlibat dalam membantu tersusunnya skripsi ini, Aamiin ya Rabbal alamin.

Yogyakarta, 17 Juli 2022

(Alan Tusa Bagus Widiyanto)

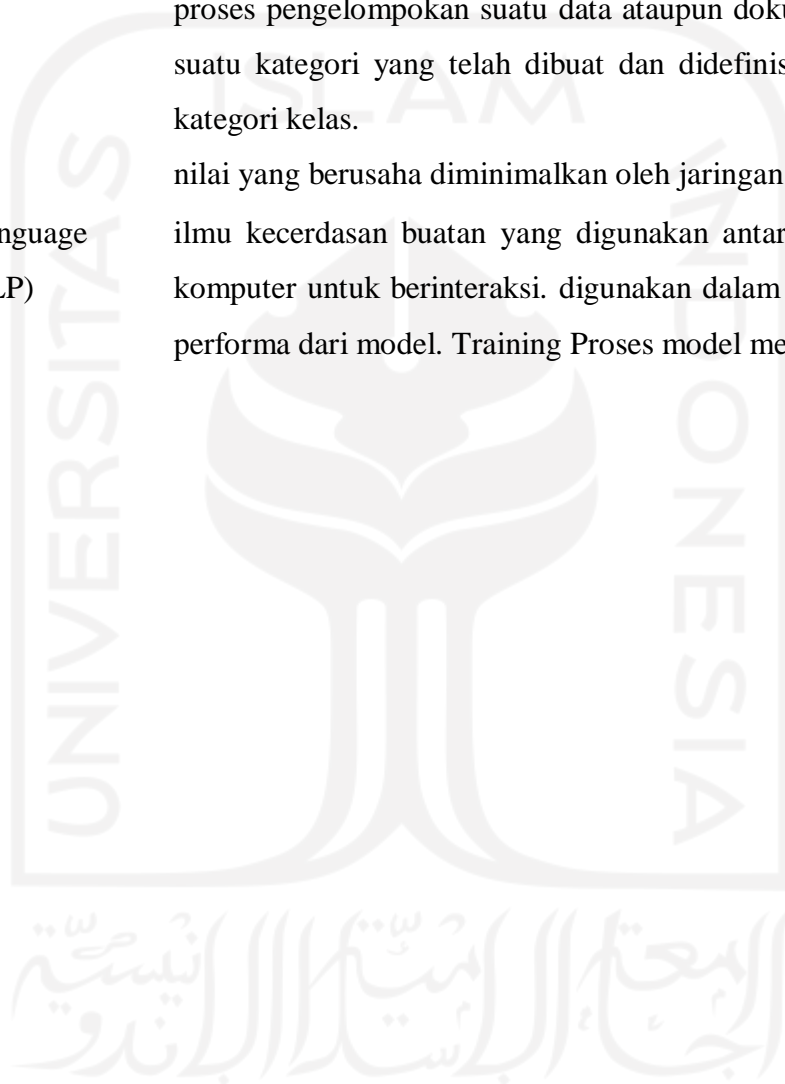
SARI

Emosi merupakan reaksi dari tubuh manusia ketika mendapatkan reaksi terhadap sesuatu yang dapat disampaikan secara nonverbal yaitu dengan menunjukkan ekspresi wajah ataupun perilaku. Dalam dunia digital, lebih tepatnya media sosial yang berbasis teks, manusia mengekspresikan emosinya melalui teks dan perlu dilakukan analisa untuk mengetahui emosi yang disampaikan oleh orang tersebut. Emosi memberikan informasi kepada seseorang untuk memahami dan mengambil keputusan yang sedang terjadi di lingkungan sekitar. Untuk mengatasi masalah di atas, perlu dilakukan metode klasifikasi pada kalimat-kalimat opini untuk mengetahui jenis emosi yang ada dalam teks tersebut, lalu emosi tersebut dikelompokkan menjadi sembilan kelas emosi yaitu marah, takut, sedih, netral, bahagia, tertarik, percaya, kaget, dan jijik. Metode yang diterapkan pada penelitian ini adalah metode klasifikasi menggunakan BERT (*Bidirectional Encoder Representations from Transformers*). BERT dirancang untuk melatih representasi dua arah. Sehingga model BERT *pre-train* dapat disesuaikan hanya dengan satu lapisan *output* tambahan untuk membuat model yang mutakhir. BERT adalah suatu model yang baru dikembangkan pada tahun 2019, dikarenakan model tersebut baru, masih perlu dilakukan pengembangan untuk mengoptimalkan kinerja model BERT. Dataset yang digunakan adalah data dari opini-opini seseorang di *Twitter* yang dikumpulkan menjadi satu ke dalam file csv. Dalam penelitian ini *hyperparameters* yang digunakan pada kedua model adalah 10 *epoch*, dan *validation* 0.1. Khusus untuk model IndoBERT menggunakan *hyperparameters* diantaranya *learning rate* 5e-05, *epsilon* 1e-08, *decay* 0.01, dan *clipnorm* 1.0. Klasifikasi emosi menggunakan BERT menghasilkan nilai akurasi sebesar 90% pada BERT *Uncased* dan menghasilkan akurasi 81% pada model indoBERT, sehingga model mampu mendeteksi emosi pada teks dengan tepat, parameter model mampu mendeteksi emosi pada teks dibuktikan pada proses pengujian *confussion matrix* dan pengujian model dengan menginputkan kalimat untuk mendeteksi emosinya. Terbukti model mampu mendeteksi emosi pada teks dengan tepat.

Kata kunci: Klasifikasi emosi, klasifikasi, BERT, opini

GLOSARIUM

Adam	algoritma optimisasi yang dapat digunakan sebagai ganti dari prosedur <i>classical stochastic gradient descent</i> untuk memperbarui bobot secara iteratif yang didasarkan pada data <i>training</i>
Batch size	jumlah sampel data yang diberikan pada jaringan.
Klasifikasi	proses pengelompokan suatu data ataupun dokumen ke dalam suatu kategori yang telah dibuat dan didefinisikan ke dalam kategori kelas.
Loss	nilai yang berusaha diminimalkan oleh jaringan saraf.
Natural Language Processing (NLP)	ilmu kecerdasan buatan yang digunakan antara manusia dan komputer untuk berinteraksi. digunakan dalam pengoptimalan performa dari model. Training Proses model mempelajari data.



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1 Penelitian Terdahulu.....	5
2.2 Analisis Emosi.....	8
2.3 Identifikasi Emosi Pada Teks.....	9
2.4 Klasifikasi Teks.....	9
2.5 <i>Natural Language Processing</i> (NLP).....	10
2.6 <i>Neural Network</i>	11
2.7 <i>Deep Learning</i>	16
2.8 <i>Bidirectional Encoder Representation From Transformers</i> (BERT).....	17
2.9 IndoBERT.....	22
BAB III METODE PENELITIAN	24
3.1 Tahapan Penelitian.....	24
3.2 Pengumpulan Data.....	24
3.3 <i>Pre-processing</i>	25

3.3.1	Menghapus Simbol-simbol	26
3.3.2	<i>Casefolding</i>	27
3.3.3	<i>Remove Slang</i>	28
3.3.4	<i>Stopword Removal</i>	28
3.3.5	<i>Stemming</i>	29
3.4	Pelabelan Data	30
3.6	Pemodelan	31
3.7	Melatih Model	32
3.8	Evaluasi	33
BAB IV HASIL DAN PEMBAHASAN		34
4.1	Dataset	34
4.2	<i>Pre-processing</i>	37
4.2.1	Menghapus Simbol-simbol	37
4.2.2	<i>Casefolding</i>	37
4.2.3	<i>Remove Slang</i>	38
4.2.4	<i>Stopword Removal</i>	38
4.2.5	<i>Stemming</i>	39
4.2.6	Hasil <i>Pre-processing</i>	39
4.3	Pelabelan Data	40
4.4	Perancangan Model	44
4.5	Melatih Model	46
4.5.1	BERT <i>Uncased</i>	47
4.5.2	IndoBERT	47
4.6	Hasil Penelitian	48
4.6.1	BERT <i>Uncased</i>	48
4.6.2	IndoBERT	49
4.7	Evaluasi	50
4.8	Komparasi Model	53
BAB V PENUTUP		56
5.1	Kesimpulan	56
5.2	Saran	56
DAFTAR PUSTAKA		58
LAMPIRAN		60

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu	6
Tabel 2.2 Perbandingan model BERT	22
Tabel 2.3 Perbandingan model pada analisis sentiment.....	22
Tabel 3.1 Contoh penerapan penghapusan simbol	26
Tabel 3.2 Contoh penerapan <i>casefolding</i>	27
Tabel 3.3 Proses <i>remove slang</i>	28
Tabel 3.4 Contoh daftar <i>stopword</i>	28
Tabel 3.5 Contoh penerapan <i>stopword removal</i>	29
Tabel 3.6 Proses <i>stemming</i>	29
Tabel 3.7 Contoh pelabelan data.....	31
Tabel 3.8 Contoh <i>confussion matrix</i>	34
Tabel 3.9 <i>Multiclass confussion matrix</i>	34
Tabel 4.1 Data twitter sebelum <i>pre-processing</i>	39
Tabel 4.2 Data twitter sesudah <i>pre-processing</i>	40
Tabel 4.3 Jumlah data per-label.....	41
Tabel 4.4 Contoh hasil pelabelan data	42
Tabel 4.5 Hasil dari model BERT <i>Uncased</i>	48
Tabel 4.6 Hasil dari model IndoBERT.....	49

DAFTAR GAMBAR

Gambar 2.1 Contoh tata bahasa	11
Gambar 2.2 <i>Pharse tree</i>	11
Gambar 2.3 <i>Feed-forward neural network</i> dan <i>Recurrent neural network</i>	13
Gambar 2.4 <i>Supervised learning</i>	16
Gambar 2.5 <i>Semi supervised</i>	17
Gambar 2.6 <i>Unsupervised</i>	17
Gambar 2.7 <i>Encoder</i> dan <i>decoder</i>	18
Gambar 2.8 Proses pada <i>self-attention</i>	19
Gambar 2.9 Proses pada <i>encoder</i>	20
Gambar 2.10 Perbedaan ukuran pada BERT.....	21
Gambar 2.11 Arsitektur BERT	21
Gambar 3.1 Tahapan penelitian	24
Gambar 3.2 Tahapan <i>pre-processing</i>	26
Gambar 3.3 <i>Plutchik's wheel of emotions</i>	32
Gambar 4.1 Contoh hasil pengumpulan data.....	36
Gambar 4.2 Kode program untuk menghapus simbol-simbol.....	37
Gambar 4.3 Kode pemrograman <i>casefolding</i>	38
Gambar 4.4 Kode pemrograman <i>remove slang</i>	38
Gambar 4.5 Kode pemrograman <i>stopword removal</i>	38
Gambar 4.6 Kode pemrograman <i>stemming</i>	39
Gambar 4.7 <i>Wordcloud</i> pada dataset	40
Gambar 4.8 Data pesebaran emosi.....	41
Gambar 4.9 Kode pemrograman model arsitektur BERT <i>Uncased</i>	44
Gambar 4.10 Model arsitektur BERT <i>Uncased</i>	45
Gambar 4.11 Kode pemrograman model arsitektur IndoBERT	46
Gambar 4.12 Model arsitektur IndoBERT	46
Gambar 4.13 Kode Pemrograman <i>Train Model BERT Uncased</i>	47
Gambar 4.14 Kode Pemrograman <i>Train Model IndoBERT</i>	47
Gambar 4.15 BERT <i>Uncased (Accuracy & Loss)</i>	49
Gambar 4.16 IndoBERT (<i>accuracy & Loss</i>).....	50
Gambar 4.17 Hasil dari evaluasi model	50

Gambar 4.18 Hasil dari <i>Predict single text</i>	51
Gambar 4.19 Hasil dari <i>Inference multiple predict</i>	51
Gambar 4.20 <i>Confussion matrix multiclass model IndoBERT</i>	52
Gambar 4.21 Perbandingan akurasi dan loss.....	53
Gambar 4.22 <i>Website</i> klasifikasi emosi	54
Gambar 4.23 Tampilan hasil prediksi	55



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kehidupan manusia erat kaitannya dengan emosi. Emosi merupakan reaksi terhadap seseorang atau suatu kejadian yang sedang terjadi. Emosi memiliki peranan penting terhadap kehidupan sehari-hari. Emosi dibagi menjadi dua luaran yaitu positif dan negatif. Dari dua luaran tersebut, ada beberapa kategori di dalamnya seperti marah, senang, sedih, takut, dan terkejut. Analisis emosi sudah menjadi pembahasan dalam berbagai disiplin ilmu seperti kognitif, psikologi, bahkan media sosial (Rohman et al., 2019).

Dengan adanya kemajuan teknologi, seseorang dapat dengan mudah mengekspresikan dirinya di sosial media, salah satunya beropini. Dalam bersosial media, kita tidak akan lepas dari beropini. Opini terjadi karena adanya pesan dari komunikator hingga kemudian terjadilah suatu diskusi ataupun reaksi terhadap isi pesan tersebut (Syarief, 2017). Dari opini tersebut biasanya orang-orang dapat mengeluarkan reaksi emosi dalam bentuk verbal maupun nonverbal.

Identifikasi emosi sendiri terbagi menjadi dua bentuk yaitu verbal dan non-verbal. Emosi verbal merupakan emosi dalam bentuk lisan ataupun tulisan, sedangkan non-verbal adalah emosi yang menggunakan bahasa tubuh untuk mengekspresikannya, seperti raut wajah, tindakan, gerak-gerik tangan, maupun kaki. Mendeteksi emosi dalam dialog tekstual adalah masalah yang menantang karena tidak adanya ekspresi wajah dan modulasi suara (Chatterjee et al., 2019). Untuk mendeteksi emosi tersebut perlu dilakukan klasifikasi emosi. Klasifikasi sendiri merupakan proses pengelompokan beberapa data sesuai kriteria.

Mendeteksi emosi memberikan peranan penting dalam berbagai aspek karena dapat diterapkan dalam berbagai bidang, seperti pengambilan keputusan dalam lingkungan sosial maupun lingkungan bisnis (Fera Fanesya dan Randy Cahya Wihandika, 2019). Pada era pandemi seperti ini, segala aktivitas yang biasanya tatap muka beralih ke daring (dalam jaringan), mulai dari bidang politik, ekonomi, hingga pendidikan. Di dalam dunia pendidikan, tenaga pengajar biasanya kesulitan untuk menyampaikan materinya dikarenakan *mood* murid atau siswa yang berbeda-beda, dan hasilnya akan buruk. Dengan demikian analisis emosi menjadi sangat penting untuk dilakukan. Analisis emosi dapat digunakan untuk mengetahui keadaan emosi siswa. Dari emosi tersebut tenaga pengajar dapat mengetahui *mood* siswanya. Hal tersebut berguna untuk meningkatkan kualitas belajar mengajar.

Dalam penelitian tentang klasifikasi emosi pada teks, banyak peneliti menggunakan beberapa metode, seperti LSTM, BiLSTM, word2vec, SVM, dan CNN. Dari penelitian-penelitian tersebut dihasilkan akurasi dari 70% hingga 90%. Diantaranya adalah klasifikasi emosi pada teks menggunakan metode LSTM oleh Riza & Charibaldi (2021) penelitian tersebut mendapatkan nilai akurasi sebesar 73,15%. Penelitian selanjutnya oleh Chiorrini et al (2021) dengan penelitian yang berjudul *emotion and sentiment analysis of tweet using BERT*, pada penelitian ini peneliti mendapatkan akurasi sebesar 92%. Hasil akurasi tersebut dipengaruhi oleh banyak data dan perbedaan label emosi yang dipakai. Pada penelitian ini, peneliti menggunakan sembilan label emosi diantaranya adalah marah, senang, sedih, takut, jijik, netral, tertarik, kaget, dan percaya.

Penelitian ini mengarah pada klasifikasi emosi pada teks opini yang digunakan untuk mengetahui jenis emosi dari opini yang telah diberikan. Opini tersebut diambil dari media sosial Twitter. Metode yang digunakan dalam penelitian ini adalah BERT (*Bidirectional Encoder Representations from Transformers*). BERT merupakan teknik berbasis jaringan saraf untuk *pre-training natural language* (Huang et al., 2019). Cara kerja BERT adalah dengan melatih model bahasa berdasarkan seluruh rangkaian kata dalam kalimat atau kueri. BERT memungkinkan model bahasa untuk memahami kata-kata ambigu dalam teks dan mengubahnya sesuai konteks yang benar dengan memproses seluruh kata-kata dalam teks secara bersamaan.

1.2 Rumusan Masalah

Klasifikasi emosi menjadi cara untuk menentukan emosi pada teks, dengan menggunakan berbagai metode seperti LSTM, BiLSTM, word2vec, SVM, dan CNN. Penggunaan metode tersebut sudah terbilang baik untuk mendeteksi emosi pada teks, karena pada penelitian sebelumnya, akurasi data yang dihasilkan mencapai rata-rata 70%-90%. Berdasarkan latar belakang yang telah dipaparkan oleh peneliti di atas, maka dapat diambil rumusan masalah bahwa belum adanya klasifikasi emosi pada teks menggunakan BERT dengan basis bahasa indonesia serta perlunya mengembangkan dan mengevaluasi model BERT.

1.3 Batasan Masalah

Untuk mendapatkan gambaran yang lebih jelas, maka diberikan batasan-batasan masalah sebagai berikut:

- a. Klasifikasi emosi per kalimat.

- b. Pengklasifikasian berdasarkan 9 emosi yaitu marah, bahagia, sedih, tertarik, netral, percaya, jijik, kaget, dan takut.
- c. Data diperoleh dari opini pada *platform* Twitter.
- d. Data yang digunakan dalam penelitian menggunakan bahasa Indonesia.

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu untuk mengembangkan analisis emosi pada teks, terutama untuk melakukan klasifikasi emosi pada teks dengan menggunakan pendekatan *deep learning*.

1.5 Manfaat Penelitian

Mendapatkan informasi mengenai emosi seseorang pada teks.

- a. Mengetahui performa BERT dalam menganalisis klasifikasi pada teks dalam Bahasa Indonesia.
- b. Hasil penelitian ini diharapkan bermanfaat dalam perkembangan ilmu *deep learning* dalam bidang klasifikasi emosi.
- c. Hasil penelitian ini diharapkan bisa menjadi acuan untuk penelitian-penelitian selanjutnya.

1.6 Sistematika Penulisan

Sistematika yang dibuat pada tugas akhir ini akan dibagi dalam lima bagian, yaitu:

a. BAB I PENDAHULUAN

Pada bab ini membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, dan manfaat penelitian serta sistematika penulisan yang merupakan gambaran menyeluruh dari penelitian ini.

b. BAB II LANDASAN TEORI

Pada bab ini membahas mengenai teori-teori yang menjadi dasar analisis permasalahan yang berhubungan dengan penelitian yaitu teori *deep learning*, klasifikasi emosi, emosi, CNN (*Convolutional Neural Network*), dan BERT (*Bidirectional Encoder Representations from Transformers*).

c. BAB III METODE PENELITIAN

Pada bab ini berisi pembahasan atau pemaparan metode yang peneliti gunakan dalam pencarian data maupun perancangan sistem yang dilakukan pada penelitian.

d. BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil dan pembahasan mengenai klasifikasi emosi pada teks menggunakan pendekatan *deep learning*.

e. BAB V PENUTUP

Pada bab ini berisi kesimpulan dari seluruh bab dan saran penelitian selanjutnya.



BAB II LANDASAN TEORI

2.1 Penelitian Terdahulu

Deteksi emosi merupakan aspek penting dalam berbagai bidang seperti pendidikan, bisnis, dan rekrutmen karyawan. Dalam penelitian ini, emosi akan dideteksi dengan teks yang berasal dari Twitter, karena media sosial membuat pengguna cenderung mengekspresikan emosi melalui postingan teks. Salah satu media sosial yang memiliki tingkat pertumbuhan pengguna tertinggi di Indonesia adalah Twitter. Penelitian mengenai klasifikasi emosi pada teks telah dilakukan sebelumnya oleh beberapa peneliti diantaranya oleh Riza dan Charibaldi (2021). Penelitian ini menggunakan metode LSTM karena metode ini terbukti lebih baik dari penelitian sebelumnya. Penyisipan *fasttext* juga digunakan dalam penelitian ini untuk meningkatkan *Word2Vec* dan *GloVe* yang tidak dapat menangani masalah *out of vocabulary* (OOV). Penelitian ini menghasilkan akurasi terbaik untuk setiap penyisipan kata sebagai berikut, *Word2Vec* menghasilkan akurasi 73,15%, *GloVe* menghasilkan akurasi 60,10%, teks cepat menghasilkan akurasi 73,15%. Kesimpulan dalam penelitian ini adalah akurasi terbaik diperoleh pada *Word2Vec* dan *fasttext*. *Fasttext* memiliki keunggulan dalam menangani masalah *out of vocabulary* (OOV), namun pada penelitian ini tidak dapat meningkatkan akurasi *word2vec*. Penelitian ini belum mampu menghasilkan akurasi yang sangat baik disebabkan oleh data yang digunakan.

Selanjutnya pada penelitian lainnya yang serupa juga dilakukan oleh Chiorrini et al (2021). Penelitian tersebut menyelidiki penggunaan model *Bidirectional Encoder Representations from Transformers* (BERT) yang digunakan untuk analisis sentimen dan pengenalan emosi data Twitter, dengan mengevaluasi kinerja model yang didapat pada kumpulan data Twitter. Eksperimen ini menunjukkan masing-masing model mendapatkan akurasi sebesar 92 % dan 90 % pada analisis sentimen dan deteksi emosi. Pada deteksi emosi peneliti mempertimbangkan kumpulan data intensitas emosi Twitter, yang terdiri dari 6755 data dan diberi label emosi dengan label diantaranya marah, bahagia, takut, dan sedih. Agar tiap-tiap kelas emosi terdistribusi secara rata, maka peneliti melakukan teknik *undersampling* untuk menyeimbangkan dataset. Peneliti mendapatkan 5200 data yang di distribusikan secara merata sebanyak 581 data tiap kelas. Pada tabel 2.1 berikut merupakan beberapa penelitian terdahulu yang terkait dengan klasifikasi emosi.

Tabel 2.1 Penelitian terdahulu

Judul penelitian	Metode	Penulis	Hasil
Klasifikasi Emosi Pada Teks Menggunakan Metode LSTM	LSTM	(Riza & Charibaldi, 2021)	Penelitian mengenai klasifikasi emosi pada teks yang telah dilakukan oleh (Riza & Charibaldi, 2021) menggunakan metode LSTM dengan menggunakan <i>word embedding</i> <i>Word2Vec</i> , <i>GloVe</i> , dan <i>Fast text</i> . Setelah melakukan percobaan, didapatkan hasil dengan akurasi tertinggi 73,15% pada saat menggunakan <i>Word2Vec</i>
Klasifikasi Emosi Pada Teks Menggunakan Metode Naïve Bayes	Naïve Bayes	(Fera Fanesya, Randy Cahya Wihandika, 2019)	Penelitian ini menggunakan metode naïve bayes dan kombinasi fitur dan didapatkan akurasi tertinggi pada metode kombinasi fitur sebesar 55,5%.
Aplikasi Pendeteksi Unsur Hinaan Dalam Komentar Di Media Sosial	SVM, <i>Logistic Regression</i> , dan <i>Naïve Bayes</i>	(W. et al., 2018)	Percobaan yang dilakukan dapat disimpulkan bahwa percobaan yang digunakan mampu memahami unsur hinaan dalam komentar pada media sosial. Pada implementasinya, sistem menghasilkan <i>output</i> positif dan negatif. Nilai positif akan muncul apabila terdapat unsur hinaan dalam kalimat.
SemEval-2019 Task 3:EmoContext Contextual Emotion Detection in Text	<i>Bi-directional</i> LSTM	(Chatterjee et al., 2019)	Penulis menggunakan metode <i>Bi-directional</i> LSTM dengan menggunakan <i>training dataset</i> sebanyak 30160 dialog, dan dua evaluasi <i>dataset</i> yang masing-masing terdapat 2755 dan 5509 dialog, data

			tersebut memiliki tiga label yang terdiri dari marah, sedih, dan senang. Dari penelitian ini dihasilkan nilai akurasi sebesar 79,59 %.
<i>Emotion and sentiment analysis of tweet using BERT</i>	BERT	(Chiorrini et al., 2021)	Penggunaan model <i>Bidirectional Encoder Representations from Transformers</i> (BERT) untuk analisis sentimen dan pengenalan emosi data Twitter, dengan mengevaluasi kinerja model yang didapat pada kumpulan data Twitter. Eksperimen ini menunjukkan masing-masing model mendapatkan akurasi sebesar 92 % dan 90 %
<i>Adopting Pre-trained BERT for Emotion Classification</i>	BERT	(Luo & Wang, 2019)	Penelitian ini menjelaskan bagaimana mendeteksi emosi pada dua dataset diantaranya <i>program</i> televisi dan percakapan di <i>facebook</i> . Penelitian ini menggunakan model arsitektur BERT dan menghasilkan akurasi skor sebesar 79,1% dan 86,2%.
Deteksi Emosi Media Sosial Menggunakan Pendekatan <i>Leksikon</i> dan NLP	Leksikon dan NLP	(Rohman et al., 2019)	Penelitian ini menghasilkan nilai akurasi sebesar 55,45% yang didapatkan dari unggahan status facebook. Selanjutnya diperbaiki dengan NLP, hasil dari evaluasi tersebut menghasilkan akurasi sebesar 61,53%.

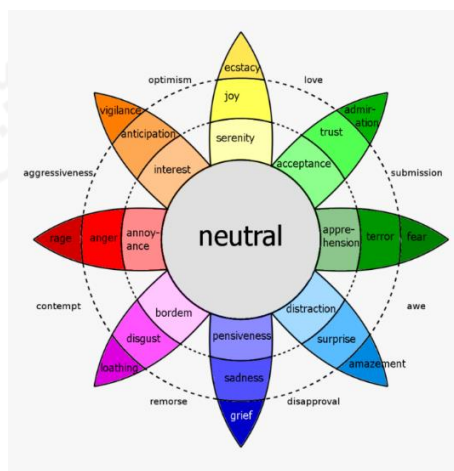
Penelitian yang dilakukan oleh peneliti terdahulu menghasilkan nilai akurasi antara 60% sampai dengan 90%. Belajar dari penelitian sebelumnya, penelitian ini menggunakan metode BERT berbahasa indonesia dengan kelas emosi yang lebih beragam dari penelitian sebelumnya. Seperti pada penelitian Riza & Charibaldi (2021) hanya melakukan klasifikasi

enam label emosi saja. Diantaranya bahagia, sedih, takut, jijik, marah, dan kaget. Pada penelitian ini, peneliti membuat data emosi yang lebih beragam dengan menambahkan lebih banyak kelas emosi. Dengan label emosi sebanyak sembilan emosi, diharapkan sistem klasifikasi emosi dapat lebih detil untuk mendeteksi emosi tersebut. Label emosi tersebut diantaranya:

- a. Marah
- b. Bahagia
- c. Sedih
- d. Tertarik
- e. Takut
- f. Netral
- g. Percaya
- h. Jijik
- i. Kaget

2.2 Analisis Emosi

Sistem survei analisis emosi dilakukan pertama kali pada tahun 1966, ini bisa dikatakan fase awal dari analisis emosi pada teks. Robert Plutchik membagi emosi menjadi delapan kategori utama. Separuhnya emosi positif dan separuhnya lagi emosi negatif. Setiap emosi memiliki subkelompok. Sebagai contoh pada emosi bahagia dapat menggambarkan keadaan kegembiraan atau kesenangan tergantung pada konteks yang diberikan. Robert Plutchik juga membuat roda emosi sebagai ilustrasi hubungan emosi satu dengan lainnya. (Abbasi & Beltiukov, 2019). Gambar 2.1 menunjukkan hubungan antara emosi satu dengan lainnya.



Gambar 2.1 *Plutchik wheel of emotions*

(Abbasi & Beltiukov, 2019)

Penelitian-penelitian yang sudah dilakukan sebelumnya menggunakan beberapa jenis emosi pada klasifikasi emosinya. Seperti pada penelitian Riza & Charibaldi (2021) menggunakan enam kelas emosi yang diantaranya bahagia, sedih, takut, jijik, marah, dan terkejut. Dari penelitian terdahulu tersebut perlu ditambahkan lagi kelas emosi agar saat melakukan identifikasi emosi lebih akurat lagi. Dengan acuan dari *Plutchik wheel of emotions*, perlu ditambahkan kelas emosi lagi agar setidaknya dapat mewakili emosi pada *Plutchik wheel of emotions*, sehingga pada saat mengidentifikasi emosi dapat lebih akurat.

2.3 Identifikasi Emosi Pada Teks

Emosi merupakan salah satu aspek dengan pengaruh besar terhadap sikap manusia dengan lingkungan. Emosi sendiri terbagi menjadi dua yaitu emosi positif dan negatif. Untuk mengidentifikasinya perlu mengetahui beberapa aspek pemicunya. Adanya media sosial di era digital ini, membuat seseorang sering berkomunikasi di dunia maya. Identifikasi emosi dilakukan dengan mengetahui pemicu yang muncul dan mengetahui respon seseorang tersebut (Rialdy Atmadja, 2017). Emosi sendiri dibagi menjadi dua yaitu verbal dan non verbal, emosi verbal dapat diartikan emosi langsung atau menggunakan lisan maupun tulisan sebagai mediana. Sedangkan non verbal biasanya menggunakan bahasa tubuh untuk mengekspresikannya.

Emosi verbal khususnya pada tulisan atau teks perlu dilakukan identifikasi lebih lanjut untuk mengetahui emosi yang benar-benar terjadi agar tidak terjadi kesalahan identifikasi karena pada tulisan bisa bermaksud multimakna di dalamnya (Agastya dan Haryanto, 2020). Teks sendiri adalah media utama dalam komunikasi pada sosial media, contoh media sosial yang hampir semua proses komunikasinya menggunakan teks adalah *twitter*. *Twitter* sendiri selain menjadi media komunikasi, pengguna juga dapat membagikan informasi, opini, curahan hati, dan berdiskusi. Identifikasi emosi pada teks dilakukan dengan menggunakan pendekatan klasifikasi emosi (Fera Fanesya, Randy Cahya Wihandika, 2019).

2.4 Klasifikasi Teks

Klasifikasi teks adalah proses pengelompokan suatu data ataupun dokumen ke dalam suatu kategori yang telah dibuat dan didefinisikan ke dalam kategori kelas (Torregrosa et al., 2022). Tujuan dari klasifikasi teks adalah untuk menganalisa, memproses, dan mengekstrak suatu informasi yang terkandung dalam teks. Dalam klasifikasi emosi pada teks, suatu teks akan diambil sebuah informasi yang terkandung di dalamnya untuk mengetahui emosi yang

terkandung dalam teks opini tersebut. Pada klasifikasi teks, terdapat beberapa kategori seperti analisis sentiment, kategorisasi berita, dan klasifikasi topik (Minaee et al., 2021).

Teks-teks opini cenderung berbentuk data yang tidak terstruktur, maka perlu dilakukan tahap *preprocessing* agar data teks tersebut lebih terstruktur dan dapat diserap informasi secara mudah. *Preprocessing* teks bertujuan untuk memecah setiap dokumen menjadi kata-kata individu dengan mewakili masing-masing sebagai vektor fitur. Untuk tujuan pengindeksan dokumen, proses pemilihan fitur dan fase *preprocessing* teks utama harus digunakan untuk memilih kata kunci. Tahap *preprocessing* teks, di sisi lain, membagi dokumen teks input menjadi fitur yang dikenal sebagai (*tokenization, words, terms* atau *attributes*)(Kadhim, 2018).

2.5 Natural Language Processing (NLP)

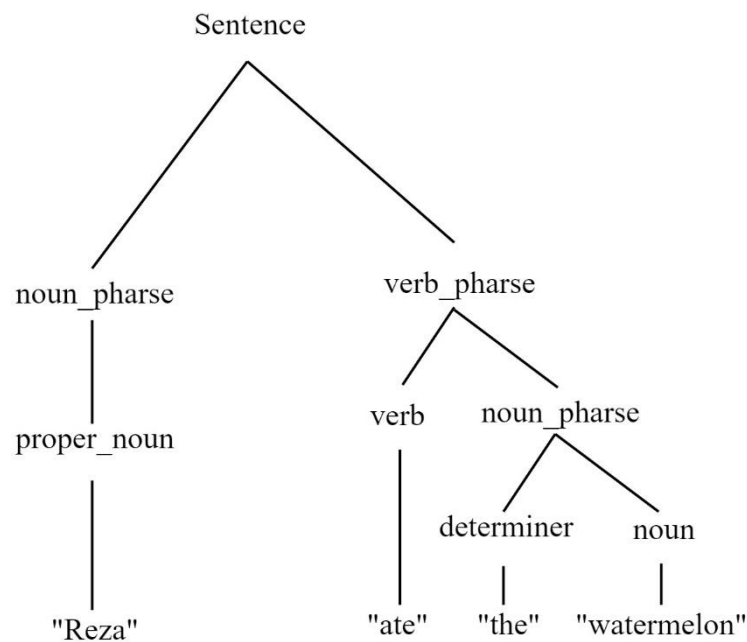
Natural language processing (Chowdhary, 2020) atau pengolahan bahasa alami adalah salah satu cabang ilmu yang meneliti dan mengembangkan cara kerja komputer untuk memahami dan memproses bahasa alami sebagai kalimat atau ucapan (Vig & Belinkov, 2019). Pada *natural language processing* (NLP) digunakan untuk mengevaluasi bagaimana bahasa dalam teks sesuai dengan aturan tata bahasa. Dalam penerapannya ada tahapan yang dipakai untuk menganalisa bahasa yaitu:

- a. *Stemming*, proses pemotongan kata diawal atau diakhir untuk mendapatkan kata baku sesuai dengan aturan tata bahasa. Tujuan dari *stemming* ini adalah menghilangkan imbuhan pada kata. Contohnya: perumahan menjadi rumah.
- b. *Tokenization*, proses pembagian sebagian kalimat menjadi kata per kata. “Rumah ini sebentar lagi akan ditinggalkan”. Setiap kata dalam kalimat tersebut akan dipisah satu per satu. “Rumah”, “ini”, “sebentar”, “lagi”, “akan”, “ditinggalkan”. Maka kalimat yang sudah dipisah perkata disebut token.
- c. *Parsing*, proses menentukan struktur pada teks dengan menganalisa kata penyusnnya berdasarkan tata hasa yang mendasarinya. Contoh pengaplikasiannya pada gambar 2.1. Gambar 2.1 merupakan contoh dari proses penguraian kata, setelah proses tersebut akan menjadi sebuah *pharse tree* seperti pada gambar 2.2 dimana kalimat akan diurai kata per kata sesuai tata bahasanya pada sebuah *pharse tree*.

“Reza ate the watermelon”

Sentence	>	noun_phrase, verb_phrase
Noun_phrase	>	proper_noun
Noun_phrase	>	determiner, noun
Verb_phrase	>	verb, noun_phrase
Proper_noun	>	[Reza]
Noun	>	[watermelon]
Verb	>	[ate]
Determiner	>	[the]

Gambar 2.1 Contoh tata bahasa



Gambar 2.2 Parse tree

Sumber: Chowdhary (2020)

2.6 Neural Network

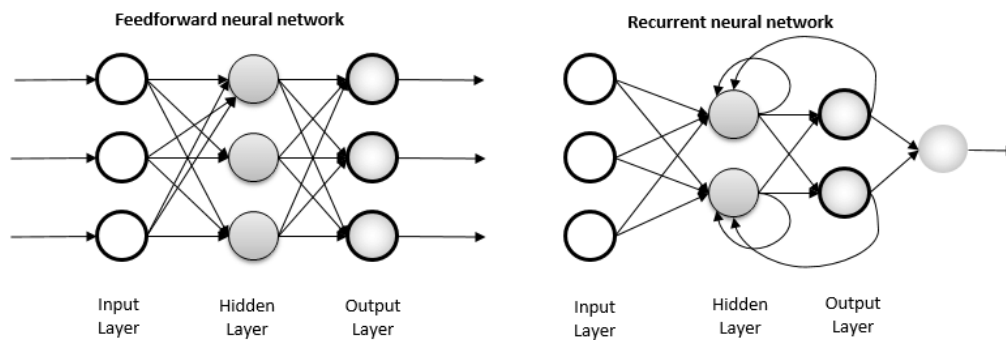
Neural network adalah cabang penelitian yang meniru bagaimana otak manusia memproses informasi dan menghasilkan hasil. Dengan meniru prinsip pembelajaran yang terinspirasi dari bagaimana sistem saraf manusia dan organisme biologis lainnya berfungsi. *Neural network*, juga dikenal sebagai jaringan saraf tiruan, adalah salah satu teknik pembelajaran mesin yang paling terkenal. *Neuron* adalah sel khusus yang menyusun sistem

saraf. Akson dan dendrit berfungsi sebagai mekanisme penghubung antara *neuron-neuron*. Hubungan antara akson dan dendrit disebut sinapsis. (Aggarwal, 2018)

Arsitektur jaringan yang menghubungkan tingkat yang berbeda, disebut sebagai arsitektur jaringan. Lapisan antara lapisan input dan output dikenal sebagai *hidden layer*, sedangkan lapisan di atas lapisan output dikenal sebagai *hidden units*. (Osinga, 2018). Unit-unit ini tidak dapat langsung dilihat sebagai *input* atau *output* dari luar, oleh karena itu disebut dengan istilah "*hidden*". Lapisan tersembunyi yang terdiri dari unit tersembunyi, yang masing-masing merupakan unit saraf, merupakan inti dari jaringan saraf. Ini menerapkan non-linearitas ke jumlah tertimbang dari input di lapisan ini. Setiap unit lapisan berkomunikasi satu sama lain dengan menggunakan semua unit di lapisan bawah sebagai *input* dan *output*, serta koneksi antara setiap pasangan unit dalam dua lapisan yang berdekatan. Unit *input* ditambahkan bersama oleh setiap unit tersembunyi. (Jurafsky & Martin, 2019).

Secara umum terdapat dua jenis arsitektur neural network, yaitu feed-forward network dan recurrent/recursive network.

1. *Feed-forward Network* atau *Multi-Layer Perceptrons* (MLP) adalah jaringan di mana komponen tidak digilir dan *output* dikirim kembali ke tingkat bawah. Hal ini memungkinkan jaringan untuk beroperasi dengan *input* ukuran tetap atau *input* panjang variabel dari serangkaian bagian yang dapat diabaikan. Jaringan belajar untuk mengintegrasikan komponen *input* saat mereka dimasukkan ke dalamnya. Data hanya mengalir dari *input* ke *output* dalam satu arah. Jaringan semacam ini digunakan untuk pengenalan pola karena lebih sederhana. Pengenalan gambar sering menggunakan jenis khusus jaringan umpan maju yang disebut *Convolutional Neural Network* (*CNN* atau *ConvNet*)
2. Dalam situasi di mana input berurutan. *Recurrent Neural Network* (*RNN*) sering digunakan ketika jaringan sedang memproses teks atau suara,. Urutan objek dimasukkan ke dalam RNN, dan menghasilkan vektor dengan ukuran tertentu yang berisi urutan tersebut. RNN memungkinkan *loop* dan memungkinkan data mengalir di kedua arah melalui jaringan. Jaringan ini lebih canggih dan kuat dari CNN. Untuk melihat prosesnya dapat dilihat pada gambar 2.3.



Gambar 2.3 *Feed-forward neural network* dan *Recurrent neural network*

Sumber: Pekel et al (2017)

Fungsi matematika yang disebut fungsi aktivasi, yang memetakan output dari satu lapisan ke input lapisan berikutnya, muncul setelah setiap lapisan dalam tumpukan. Fungsi softmax adalah komponen aktivasi. Fungsi aktivasi softmax memastikan bahwa vektor keluaran dijumlahkan tepat satu kali atau 1. Kemudian, simpul keluaran dengan kemungkinan tertinggi dipilih sebagai label prediktif untuk frasa masukan (Munikar et al., 2019). Jaringan yang membutuhkan keluaran dengan label tertentu dapat menggunakan *activation function*. Misalnya, dalam jaringan yang telah dilatih untuk mengenali anjing dan ayam. Jaringan mengenali ayam dengan kepastian 75 persen jika vektor input untuk identifikasi ayam memiliki vektor output 0,75. Softmax hanya berfungsi ketika hanya ada satu solusi. (Osinga, 2018). Output node dengan probabilitas tertinggi kemudian dipilih sebagai label prediksi untuk kalimat yang menjadi input (Munikar et al., 2019). Rumus *softmax function* dirumuskan pada persamaan 2.1

$$\text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.1)$$

Dengan $i = 1, \dots, K$ dimana:

1. $z = (z_1, \dots, z_k) \in \mathbb{R}^K$ adalah vektor input yang dimasukkan pada *softmax function* atau *output* dari *layer* terakhir yang disebut juga dengan *logits*. Inputan yang dapat diterima merupakan bilangan *real*.
2. e^{z_i} adalah eksponensial dari tiap elemen dari *input* vektor
3. $\sum_{j=1}^K e^{z_j}$ adalah proses normalisasi untuk memastikan semua nilai *output* dari *softmax* akan berjumlah tepat 1 dan masing-masing nilai dari antara kisaran (0,1)

Jaringan akan menunjukkan *loss* setelah prosedur pelatihan selesai. Setelah beberapa siklus, jika jumlah *loss* ini tidak berkurang, jaringan tidak memperoleh informasi baru dari operasi sebelumnya. Oleh karena itu, untuk memodifikasi keluaran *neural network* agar sesuai

dengan yang kita inginkan, diperlukan suatu *loss function* (optimasi). Fungsi *loss* menghasilkan skor jarak menggunakan prediksi yang dibuat oleh jaringan saraf dan target (apa yang ingin dihasilkan oleh jaringan). Hal ini memungkinkan fungsi *loss* untuk menentukan seberapa baik kinerja jaringan (Chollet, 2018). Hasil dari fungsi *loss* kemudian digunakan sebagai sinyal umpan balik untuk sedikit memodifikasi bobot untuk menurunkan skor *loss*. *Cross-entropy loss* adalah fungsi *loss* yang sering dikombinasikan dengan *softmax* dalam jaringan saraf (Aggarwal, 2018). Perhitungan antara dua distribusi probabilitas untuk variabel acak tertentu menghasilkan fungsi *loss* ini. Kemampuan model klasifikasi untuk menghasilkan keluaran probabilitas dengan nilai antara 0 dan 1 ditentukan oleh *cross-entropy loss*. Ketika *output* diberikan sebagai probabilitas, fungsi *loss* ini sesuai untuk digunakan (Chollet, 2018). Fungsi *loss* dapat dilihat pada rumus persamaan (2.2)

$$L = \sum_{i=1}^k y_i \log(o_i) \quad (2.2)$$

Dimana:

1. y_i adalah label dari klasifikasi
2. o_i adalah probabilitas yang diprediksi oleh model terhadap label
3. $\log(o_i)$ adalah nilai logaritma dari tiap probabilitas yang diprediksi oleh model

Banyak parameter jaringan saraf, seperti bobot W dan bias b , diselidiki melalui *gradient descent*. Sementara nilai terbaik untuk *hyperparameter* ditetapkan pada set pengembangan daripada di set data pelatihan menggunakan penurunan gradien, *hyperparameter* adalah parameter yang dipilih oleh perancang algoritma. Laju pembelajaran, ukuran mini-batch, arsitektur model (jumlah lapisan, jumlah node tersembunyi per lapisan, fungsi aktivasi yang dipilih), dan *hyperparameter* lainnya adalah contoh dari *hyperparameter* (Jurafsky & Martin, 2019).

Gradient descent (penurunan gradien) adalah metode pelatihan untuk model pembelajaran mesin yang menggunakan optimasi. Hanya nilai parameter fungsi (koefisien) yang meminimalkan biaya yang ditemukan menggunakan penurunan gradien. Dengan meminimalkan fungsi *loss*, penurunan gradien berusaha untuk menentukan bobot ideal. Laju pembelajaran, yang menunjukkan seberapa cepat atau lambat fungsi mencapai bobot ideal, menentukan berapa banyak langkah penurunan gradien. Tiga teknik penurunan gradien yang berbeda diantaranya *Batch Gradient Descent*, *Stochastic Gradient Descent*, dan *Mini-Batch Gradient Descent*.

1. *Batch Gradient Descent* (BGD) atau vanilla gradient descent menghitung kesalahan untuk setiap contoh pelatihan dalam kumpulan data. Masa pelatihan merupakan suatu proses yang menyerupai sebuah siklus. Pendekatan ini memiliki manfaat untuk mempercepat waktu pemrosesan jaringan syaraf tiruan dan menghasilkan gradien kesalahan yang stabil. Model mungkin tidak dapat memberikan kondisi konvergensi yang optimal jika gradien kesalahan stabil. Selanjutnya, set pelatihan lengkap perlu disimpan dalam memori.
2. *Stochastic Gradient Descent* (SGD) adalah sebuah algoritma yang menghitung gradiennya setiap kali pelatihan dilakukan, meminimalkan fungsi kerugian (Jurafsky & Martin, 2019). Karena hanya menggunakan satu sampel acak pada satu waktu dan menyesuaikan bobotnya untuk meningkatkan kinerja sampel itu, metode ini dikenal sebagai algoritma stokastik. Akibatnya, SGD akan melakukan prosedur pada setiap *instance* dari dataset secara individual. Peningkatan yang agak menyeluruh dimungkinkan berkat bobot yang terus diperbarui. Namun, karena tingkat kesalahan mungkin berbeda, pembaruan yang sering lebih mahal dan dapat menghasilkan gradien yang tidak merata.
3. *Mini Batch Gradient Descent* menggabungkan konsep dari SGD dan BGD. Mini batch adalah melatih serangkaian m kumpulan data sampel yang lebih kecil dari kumpulan data asli dan biasanya masing-masing memiliki 512 atau 1024 elemen. Algoritma ini akan mengubah bobot setiap *batch* dengan memecah dataset pelatihan menjadi *batch* yang lebih kecil.

Adam adalah salah satu metode optimasi yang paling banyak digunakan. Estimasi momen dihitung dan digunakan untuk mengoptimalkan fungsi melalui Adam atau Algoritma Estimasi Momen Adaptif (Tato & Nkambou, n.d, 2018) AdaGrad dan RMSProp adalah dua algoritma yang membentuk Adam. Untuk meningkatkan performa pada masalah dengan sparse gradients, AdaGrad akan terus mempertahankan kecepatan pembelajaran per parameter. RMSProp juga akan terus mempertahankan tingkat pembelajaran per parameter yang disesuaikan berdasarkan rata-rata magnitudo gradien untuk bobot (seberapa cepat berubah). Pendekatan ini mengkuadratkan gradien yang diperoleh setelah menghitung rata-rata eksponensial tertimbang bergerak. Adam menggunakan rata-rata momen pertama dan kedua saat menyesuaikan parameter kecepatan belajar. RMS Prop hanya menggunakan rata-rata momen pertama. Adam berhasil menyelesaikan masalah pembelajaran mendalam di dunia nyata menggunakan model

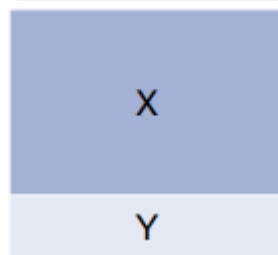
dan kumpulan data besar (Tato & Nkambou, 2018). Akibatnya, pengoptimal Adam adalah teknik komputasi hemat memori, mirip dengan gradien dengan skala diagonal, dan sesuai untuk masalah yang melibatkan sejumlah besar data dan/atau parameter.

2.7 Deep Learning

Deep learning adalah cabang ilmu dari *Machine Learning* yang menjadi bagian dari kecerdasan buatan. Teknik *Deep Learning* adalah bagian dari *neural network* yang terkenal ketika struktur multilayer, banyak dipakai karena dapat menangani banyak masalah sekaligus dan memberikan solusi yang unik (Amigo, 2021). *Deep Learning* dapat mempelajari suatu data agar dapat mempresentasikan data dengan baik, sehingga dapat melakukan prediksi dengan baik. *Deep learning* dibagi menjadi tiga metode diantaranya.

a. Supervised

Supervised adalah sebuah sistem yang memiliki variabel x sebagai *input* dan variabel y sebagai *output*, dengan persamaan $Y = f$, sistem ini memetakan variabel *input* ke variabel *output* menggunakan fungsi pemetaan (x). Tujuannya adalah untuk memperkirakan secara akurat fungsi pemetaan sehingga dapat memprediksi *output* dari variabel *input* baru. Metode *supervised* dapat dilihat pada gambar 2.4.

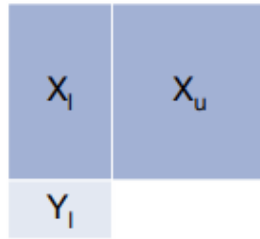


Gambar 2.4 *Supervised learning*

(Prasad & Chanussot, 2020)

b. Semi supervised

Algoritma *semi supervised* berada diantara algoritma *supervised* dan *unsupervised*. Cara kerjanya adalah menemukan dan menganalisis struktur dalam variabel input. Setelah itu, akan dibuat prediksi terbaik dari data yang tidak berlabel dan memberikan data tersebut ke algoritma supervised sebagai data latihan untuk membuat prediksi baru. Metode *semi supervised* dapat dilihat pada gambar 2.5.



Gambar 2.5 *Semi supervised learning*

Sumber: Prasad & Chanussot (2020)

c. *Unsupervised*

Algoritma yang tidak memiliki variabel *output* yang sesuai, hanya variabel *input*. Memodelkan struktur atau distribusi yang dapat memahami data dengan lebih baik adalah tujuan dari metode *unsupervised*. Untuk menemukan dan menunjukkan struktur data yang menarik, algoritma akan dibiarkan sendiri. Metode *unsupervised* dapat dilihat pada gambar 2.6.



Gambar 2.6 *Unsupervised learning*

Sumber: Prasad & Chanussot (2020)

2.8 *Bidirectional Encoder Representation From Transformers (BERT)*

Bidirectional Encoder Representation From Transformers (BERT) (Luo & Wang, 2019) merupakan teknik yang berbasis jaringan saraf untuk *pre-training natural language*. BERT dirancang untuk memahami bahasa pada kalimat yang ambigu menggunakan teks disekitarnya untuk membangun konteks yang lebih jelas.

BERT (Ganesh et al., 2022) mengandalkan transformer (mekanisme untuk mempelajari konteks kalimat dengan mempelajari hubungan kontekstual kata-kata dalam teks). Transformer dapat belajar dan mengubah pemahaman yang diperoleh dari mekanisme self-attention. Mekanisme self-attention adalah cara transformator memodifikasi kata terkait dan diubah oleh mekanisme tersebut. Transformer terdiri dari dua mekanisme *encoder* dan *decoder*.

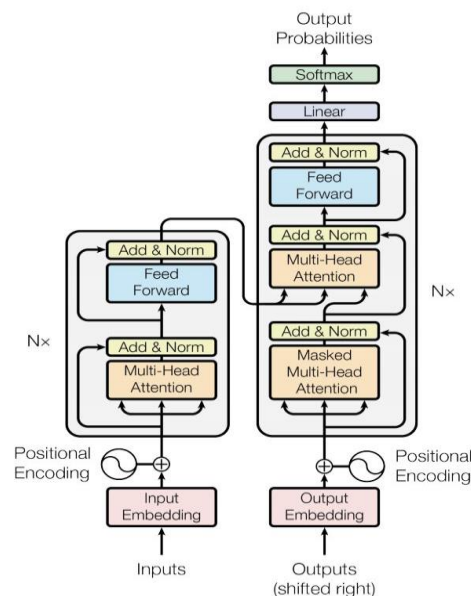
a. *Encoder*

Encoder digunakan untuk membaca data input teks. *Encoder* terdiri dari tumpukan $N = 6$ lapisan identik. Setiap lapisan memiliki dua sublapisan, lapisan *self-attention* dan jaringan saraf *feedforward*. Dengan lapisan *self-attention*, *encoder* dapat membantu node

yang tidak hanya fokus pada kata yang divisualisasikan, tetapi juga mendapatkan konteks dari kata tersebut. Setiap posisi di *encoder* dapat memproses semua posisi lapisan sebelumnya di *encoder*.

b. *Decoder*

Decoder berfungsi untuk menghasilkan urutan keluaran yang diprediksi. *Decoder* juga mencakup tumpukan $N = 6$ lapisan yang dapat diidentifikasi. Setiap lapisan terdiri dari dua sublapisan yang sama dengan lapisan *encoder*. Dengan *attention layer* tambahan di antara keduanya untuk membantu node saat ini mengakses konten utama yang diinginkan dengan melakukan perhatian *multi-head* pada *output encoder*. Seperti pada *encoder*, lapisan *self-attention* di *decoder* memungkinkan setiap posisi di *decoder* untuk menangani semua posisi sebelumnya dan saat ini.



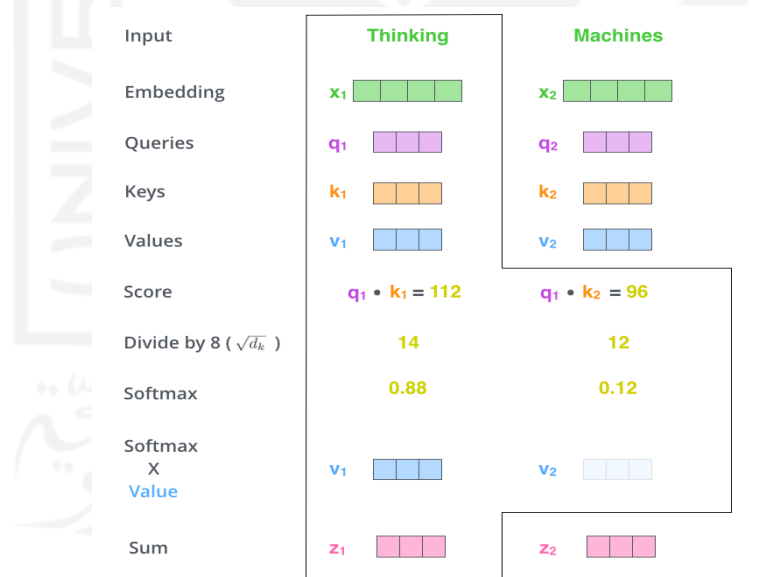
Gambar 2.7 *Encoder dan decoder*

Sumber: Vaswani et al (2017)

Langkah-langkah berikut adalah proses yang berjalan pada *encoder* dan *decoder* pada gambar 2.7 (Khan et al., 2022):

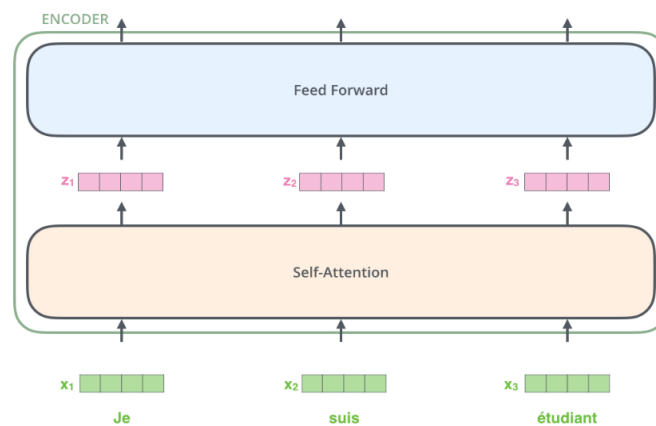
1. Setiap input istilah yang memasuki *encoder* diubah sebagai sebuah list vector memakai *embeddings*. Lantaran *self-attention layer* tidak membedakan urutan istilah-istilah dalam sebuah kalimat. *Positional encoding* dibubuhi untuk menampakan posisi berdasarkan istilah-istilahnya. Tiap vektor menurut input istilah mempunyai berukuran

512. Proses ini hanya terjadi pada *encoder* yang berada paling bawah, sebagai akibatnya *encoder* lainnya akan mendapat hasil menurut *encoder* yang pertama.
- Input vektor melewati dua layer yang terdapat dalam tiap *encoder* yaitu *self-attention layer* and *feed-forward neural network*. Pada *self-attention layer* dibentuk tiga vektor menurut masing-masing input vektor yaitu *Query*, *Key* dan *Value vector*. Ketiga vektor ini dibentuk menggunakan mengalikan *embedding*. Dimensi menurut tiap vektor merupakan 64. Setelah itu, nilai *self-attention* menurut tiap istilah dihitung menggunakan mengalikan *query vector* dan *key vector* misalnya yang terdapat dalam Gambar 2.8. Kemudian, nilai *self-attention* dibagi 8 lantaran 8 merupakan akar kuadrat menurut dimensi tiap vektor yaitu 64. Nilai *self-attention* juga dihitung menggunakan softmax. Sebagai akibatnya tiap *value vector* akan dikali menggunakan nilai menurut softmax. Akhirnya *value vector* dijumlahkan dan sebagai hasil menurut *self-attention layer*. Output menurut *self-attention layer* lalu masuk ke *feed-forward* untuk masing-masing posisi misalnya yang tertera dalam gambar 2.9.



Gambar 2.8 Proses pada *self-attention*

Sumber : Alammarr (2018)

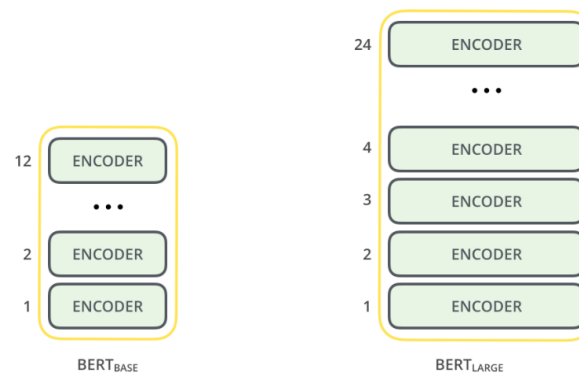


Gambar 2.9 Proses pada *Encoder*

Sumber: Alammr (2018)

3. Ketika setiap proses selesai pada *encoder*, output dari *encoder* yaitu *vector key* dan *vector value*, masuk ke dalam *decoder*. Setiap input dan output dari *encoder* dan *decoder self-awareness layer* dan *feedforward neural network* diproses oleh lapisan *add* dan *norm*, yang mencakup struktur residual dan lapisan normalisasi. Proses yang dilakukan oleh *decoder* serupa dengan *encoder*, tetapi terdapat *self attention* antara *layer self-aware* dan jaringan saraf *feedforward* yang membantu *decoder* untuk fokus pada bagian kata yang relevan. *Layer self-aware decoder* hanya bisa peduli dengan posisi sebelum output. Output dari setiap langkah terus disuplai ke *decoder* dan output *decoder* sama dengan output *encoder*. Akhirnya, output dari tumpukan *decoder* menghasilkan vektor dengan nilai *floating point*. Untuk memasukkannya ke dalam kata-kata, dibutuhkan lapisan tambahan dari lapisan yang terhubung penuh, bersama dengan lapisan softmax.

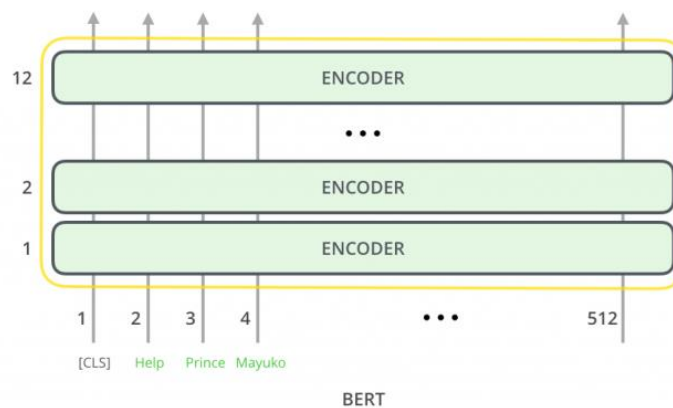
Arsitektur model BERT berupa *multi-layer bidirectional transformer* seperti yang dilakukan pada implementasi asli transformer tetapi hanya menggunakan proses sampai *encoder* saja. Ukuran pada BERT terbagi menjadi dua model yaitu BERT-*base* dan BERT-*large*.



Gambar 2.10 Perbedaan ukuran pada BERT

Sumber: Alammar (2018)

Pada implementasinya, terdapat dua ukuran model yang ada pada BERT, yaitu BERTBASE dan BERTLARGE seperti pada gambar 2.10. Kedua ukuran model BERT ini memiliki banyak lapisan *encoder* atau *Transformer Blocks*. BERTBASE memiliki encoder dengan 12 layers, 12 *self-attentions* heads, hidden size sebesar 768, dan 110M parameters. Sedangkan BERTLARGE terdapat 24 layers, 16 *self-attention heads*, hidden size sebesar 1024, dan 340M parameters. BERTBASE dilatih selama 4 hari menggunakan 4 cloud TPUs sedangkan BERTLARGE membutuhkan 4 hari menggunakan 16 TPUs.



Gambar 2.11 Arsitektur BERT

Sumber: Alammar (2018)

Sesuai dengan namanya, BERT hanya menggunakan *encoder*. Sehingga arsitektur BERT terlihat seperti Gambar 2.11 di atas. BERT berbeda dengan model terarah (*directional*) yang melihat urutan teks dari kiri-ke-kanan, kanan-ke-kiri, atau gabungan dari kiri-ke-kanan dan kanan-ke-kiri. Model bahasa yang dilatih secara *bidirectional* dapat memiliki pemahaman yang lebih dalam tentang konteks daripada model bahasa satu arah.

2.9 IndoBERT

Bahasa Indonesia adalah bahasa ke-10 yang paling banyak digunakan di dunia, dengan sekitar 200 juta orang menggunakannya sebagai bahasa pertama atau kedua mereka. Meskipun penggunaan bahasa Indonesia secara luas, masih sedikit praktisi NLP karena kurangnya sumber daya bahasa dan standar sumber daya. Penelitian *data science* dapat dibantu dengan penggunaan IndoBERT sebagai model bahasa pra-pelatihan baru untuk bahasa Indonesia.

IndoBERT adalah model berbasis *transformer* dengan mengadaptasi BERT itu sendiri, tetapi dilatih murni sebagai *masked language model* yang dilatih menggunakan *huggingface* dengan mengikuti konfigurasi *BERT base (uncased)* (Koto et al., 2020). IndoBERT memiliki 12 *hidden layer* dengan masing-masing 768d, 12 *attention head*, dan *feed-forward hidden layer* 3.072d. Koto et al., (2020) memodifikasi kerangka *huggingface* untuk membaca aliran teks terpisah untuk token dokumen yang berbeda dan mengatur pelatihan menggunakan 512 token per *batch*. IndoBERT mengumpulkan lebih dari 220 juta kata yang dikumpulkan dari tiga sumber utama diantaranya wikipedia Indonesia (74 juta kata), artikel berita Indonesia (55 juta kata), dan korpus web indonesia (90 juta kata).

Untuk membandingkan IndoBERT dibandingkan dengan dua model BERT yang sudah ada sebelumnya diantaranya *multilingual BERT (MBERT)* dan *monolingual BERT melayu (MALAYBERT)*. Perbandingan antara model dapat dilihat pada tabel 2.2 dan tabel 2.3.

Tabel 2.2 Perbandingan model BERT

Metode	Accuracy	F1
MBERT	96.8	71.6
MALAYBERT	96.8	73.2
INDOBERT	96.8	74.9

Tabel 2.3 Perbandingan model pada analisis sentimen

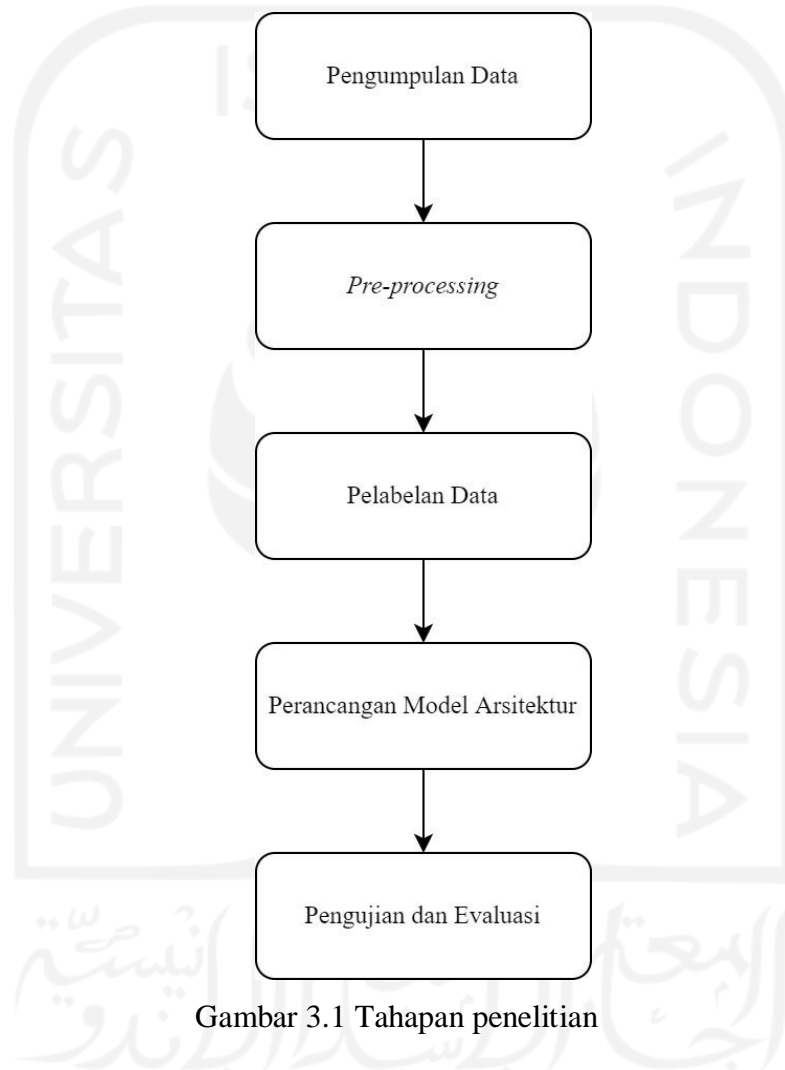
Metode	Analisis sentimen (F1)
MBERT	76.58
MALAYBERT	82.02
INDOBERT	84.13

IndoBERT mengungguli semua metode lain, baik untuk analisis sentimen dan akurasi model itu sendiri. Untuk analisis sentiment IndoBERT mengungguli kedua model dengan selisih +7.5 poin di atas MBERT dan +2.11 poin di atas MalayBERT.

BAB III METODE PENELITIAN

3.1 Tahapan Penelitian

Alur metode dalam penelitian ini dapat dilihat pada gambar 3.1. Penelitian ini memuat langkah-langkah yang diantaranya pengumpulan data, *preprocessing*, pelabelan data, modeling, pengujian, dan evaluasi.



Gambar 3.1 Tahapan penelitian

3.2 Pengumpulan Data

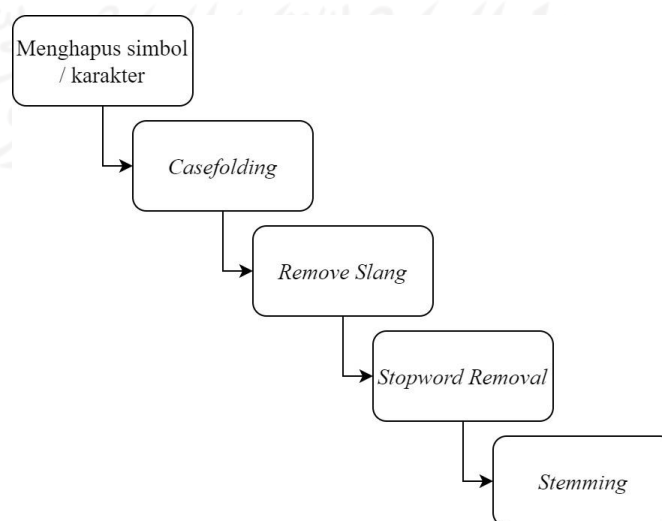
Data yang digunakan dalam penelitian ini adalah opini masyarakat di media sosial berupa teks. Proses pengambilan data diambil dengan menggunakan teknik *scraping*. *Scraping* data dilakukan agar data yang diambil lebih akurat. Peneliti menggunakan kata kunci untuk mencari opini pada *Twitter* sebagai contoh covid-19. Kemudian akan muncul semua opini terkait dengan covid-19, mulai dari opini yang positif hingga negatif. Setelah data dirasa sudah cukup, peneliti menjadikan data opini tersebut menjadi *file csv*. Data yang terkumpul sebanyak 2515

tweet. Data tersebut nantinya akan diberi label sesuai dengan emosi pada setiap *tweet*. Selain menggunakan teknik *scraping* untuk mengumpulkan data, peneliti juga mengumpulkan kembali data-data penelitian terdahulu pada penelitian yang sudah dilakukan oleh Riza & Charibaldi (2021). Data dari penelitian Riza & Charibaldi (2021) terdiri dari enam jenis emosi yaitu marah, senang, jijik, sedih, takut, dan kaget.

3.3 *Pre-processing*

Pada tahap ini data yang sudah dikumpulkan akan diproses untuk diambil informasi yang terkandung di dalamnya karena masih banyak data yang kotor dan adanya simbol-simbol yang tidak diperlukan. *Pre-processing* data dilakukan dengan cara menghapus data yang tidak perlu atau tidak sesuai agar membuat data lebih mudah untuk diproses, hingga menghasilkan data yang sesuai dengan kebutuhan penelitian. *Pre-processing* datang dalam beberapa macam. Hal ini kemudian disesuaikan dengan kebutuhan *dataset* yang digunakan.

Untuk setiap pencarian kemungkinan terdapat *pre-processing* yang berbeda. Namun, dalam pencarian ini, metode pemrosesan yang dilakukan adalah penghapusan simbol-simbol, pengubahan huruf menjadi kecil, *remove slang* yang berfungsi menghapus kata gaul dan merubahnya ke bentuk kata baku, dan *stemming* yang berfungsi untuk mengurangi jumlah indeks yang berbeda dari kumpulan data sehingga kata dengan akhiran atau awalan kembali ke bentuk dasarnya. Selain itu juga pengelompokan kata-kata lain yang memiliki kata dasar dan makna yang sama tetapi memiliki bentuk yang berbeda karena dilekatkan dengan imbuhan yang berbeda. Proses ini penting untuk dilakukan secara berurutan guna menghasilkan data yang sama dan optimal. Tahapan *pre-processing* yang dilakukan pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Tahapan *pre-processing*

3.3.1 Menghapus Simbol-simbol

Proses pengambilan data dengan menggunakan teknik *scraping* ternyata ditemukan simbol-simbol yang kurang tepat dan merubah makna dari kalimat aslinya. Salah satu contoh simbolnya seperti “@” pada nama akun yang membuat opini maupun orang yang sedang disebut dalam opini tersebut. Hal ini tidak hanya tidak memiliki makna dalam mengidentifikasi emosi tetapi juga membuat data tidak rapi. Simbol-simbol yang tidak bermakna dalam identifikasi emosi perlu dihilangkan agar data yang akan digunakan lebih jelas dan rapi saat peneliti akan melakukan pelabelan data. Contoh penghapusan simbol dapat dilihat pada tabel 3.1. diketahui pada tabel 3.1, kalimat opini hasil dari *scraping* memiliki simbol-simbol yang tidak memiliki makna yang mengakibatkan data sulit dibaca dan tidak rapi. Terlihat perbedaan data sebelum dan sesudah dibersihkan. Data yang sudah dibersihkan lebih rapi dan terstruktur.

Tabel 3.1 Contoh penerapan penghapusan simbol

Sebelum	Sesudah
@gumvbal Setauku pernah denger ada yg bilang di drama itu gatau episode berapa, katanya psikopat itu meskipun gapunya emosi tapi dia bisa memainkan emosi sesukanya, seakan akan dia kelihatan kalo dia punya emosi. Psikopat ga selamanya bermuka dingin. Gitu katanya	gumvbal Setauku pernah denger ada yg bilang di drama itu gatau episode berapa, katanya psikopat itu meskipun gapunya emosi tapi dia bisa memainkan emosi sesukanya, seakan akan dia kelihatan kalo dia punya emosi. Psikopat ga selamanya bermuka dingin. Gitu katanya
d sidang sebagai tersangka nolakðŸˆ,ðŸˆ,ðŸˆ, stress ðŸˆ,ðŸˆ,ðŸˆ, turunan 38 hoax ...nggak sakti .. cengeng..tekanan darah naik emosi liarðŸˆ”ðŸˆ”ðŸˆ”masih d pujaðŸˆ,ðŸˆ,ðŸˆ,ðŸˆ,ðŸˆ,ðŸˆ, hrs cuma ca keturunan Arab yaman anak penjual minyak wangiðŸˆ,ðŸˆ,ðŸˆ, PROSES HUKUM lanjutkan	sidang sebagai tersangka turunan 38 hoax nggak sakti cengeng tekanan darah naik emosi liar hrs cuma ca keturunan Arab yaman anak penjual minyak wangi PROSES HUKUM lanjutkan

3.3.2 Casefolding

Casefolding digunakan untuk menggeneralisasi penggunaan huruf kapital. Misalnya pada dataset terdapat tulisan “RuMahnya TerBAkar” maka dengan menggunakan proses *casefolding* data yang tidak beraturan akan dibenahi, *casefolding* akan mengubah semua huruf menjadi huruf kecil (*lowercase*). Guna dari penggunaan *casefolding* agar kalimat opini pada dataset memiliki bentuk atau jenis yang sama dan dapat memudahkan peneliti menentukan label emosi pada dataset karena data terlihat lebih rapi dan beraturan. Contoh penerapan *casefolding* dapat dilihat pada Tabel 3.2. Penerapan *casefolding* pada teks dengan menggeneralisasi bentuk huruf, dalam kasus ini peneliti merubah semua huruf dalam bentuk kecil (*lowercase*).

Tabel 3.2 Contoh penerapan *casefolding*

Sebelum	Sesudah
@ourzishu GA BISAA nanges. Jadi udah dikirim kodenya kan ke sms TAPI TULISANNYA SALAH KODENYA??? PADAHAL TUH BENERR EMOSI GA SIH kayaknya emang ga jodoh bgt sama y0uku ini,,	ourzishu ga bisaa nanges. jadi udah dikirim kodenya kan ke sms tapi tulisannya salah kodenya??? padahal tuh benerr emosi ga sih kayaknya emang ga jodoh bgt sama y0uku ini
Coba ya, sebagian nitizen yg lain awasi berita lain, selain berita HRS. Biasanya ADA MALING AMBIL KESEMPATAN di kala kita emosi dgn perlakuan tak adil pada HRS.	coba ya, sebagian nitizen yg lain awasi berita lain, selain berita hrs. biasanya ada maling ambil kesempatan di kala kita emosi dgn perlakuan tak adil pada hrs.
Sebenarnya emang mbak irene agak terkesan songong terkait isu dewa kipas, tapi gw yakin banyak orang yang hobi dan apdet dengan catur juga ngerasain panas emosi yang sama ngeliat komentar goblok netizen indo yang nunjukkin betapa awam dan sotoynya mereka So yea i dont blame her	sebenarnya emang mbak irene agak terkesan songong terkait isu dewa kipas, tapi gw yakin banyak orang yang hobi dan apdet dengan catur juga ngerasain panas emosi yang sama ngeliat komentar goblok netizen indo yang nunjukkin betapa awam dan sotoynya mereka so yea i dont blame her

3.3.3 Remove Slang

Remove slang adalah tahapan untuk mengubah kata-kata gaul menjadi kata baku yang sesuai dengan bahasa Indonesia pada kalimat opini. Penggunaan *remove slang* berfungsi untuk membuat teks opini lebih jelas saat proses deteksi teks. Contoh penerapan *remove slang* dapat dilihat pada tabel 3.3. Penerapan *remove slang* pada dataset yaitu merubah kata-kata yang kurang baku menjadi bentuk yang lebih baku. Penelitian ini menggunakan *dictionary slang* bahasa Indonesia.

Tabel 3.3 Proses *remove slang*

Sebelum	Sesudah
@ourzishu GA BISAA nanges. Jadi udah dikirim kodenya kan ke sms TAPI TULISANNYA SALAH KODENYA???	@ourzishu tidak bisa nangis. jadi sudah dikirim kodenya ke sms tapi tulisannya salah kodenya

3.3.4 Stopword Removal

Stopwords adalah kata-kata yang diabaikan selama pemrosesan dan biasanya dipertahankan dalam *stop lists* pada konteks pemrograman seperti NLP (*Natural Language Processing*). *Stopword* biasanya dipilih berdasarkan frekuensi kemunculannya yang tinggi, seperti kata penghubung seperti "dan", "atau", "tetapi", dan "akan". Untuk meningkatkan kecepatan dan efektivitas NLP, tujuan utama metode *stopwords* adalah membatasi jumlah kata dalam dokumen. Untuk contoh daftar *stopword* dapat dilihat pada tabel 3.4

Tabel 3.4 Contoh daftar *stopword*

Ada	Bagai	Dahulu	Kata	Masa
Adalah	Bagaikan	Dalam	Katakan	Masalah
Adanya	Bagaimana	Dan	Katakanlah	Masalahnya
Adapun	Bagaimanakah	Dapat	Katanya	Masih
Agak	Bagaimanapun	Dari	Ke	Masihkah
Agar	Bagi	Daripada	Keadaan	Masing
Akan	Bagian	Datang	Kebetulan	Masing-masing
Akhir	Bahkan	Dekat	Kecil	Mau
Akhiri	Bahwa	Demi	Kedua	Maupun
Akhirnya	Bahwasanya	Demikian	Keduanya	Melainkan
Aku	Baik	Demikianlah	Keinginan	Melakukan

Akulah	Bakal	Dengan	Kelamaan	Melalui
Amat	Bakalan	Depan	Kelihatan	Melihat
Amatlah	Balik	Di	Kelihatannya	Melihatnya
Anda	Banyak	Dia	Kelima	Memang
Andalah	Bapak	Diakhiri	Keluar	Memastikan
Antar	Baru	Diakhirinya	Kembali	Memberi
Antara	Bawah	Dialah	Kemudian	Memberikan

Tabel 3.5 Contoh penerapan *stopword removal*

Sebelum	Sesudah
Perusahaan kecil nih yang begini. Biasa kalo gede yang penting experince dan skill. Ga tau juga kalo mungkin hrnya masih kolot!!	Perusahaan kecil begini. kalo gede yang penting experince dan skill. Ga tau juga mungkin hrnya masih kolot!!

3.3.5 Stemming

Stemming adalah proses untuk mencari kata dasar dari sebuah kata. Faktanya sebagian besar kata memiliki karakteristik morfologis dan sifat semantik, membuatnya cocok untuk digunakan dalam aplikasi pencarian informasi. Stemmer menghilangkan istilah untuk membuat istilah. Misalnya, “pertimbangan” menjadi “timbang” (Haroon, 2018) dengan menghilangkan semua imbuhan termasuk awalan, sisipan, akhiran, dan kata turunan. Penerapan *stemming* dapat dilihat pada tabel 3.4.

Tabel 3.6 Proses *stemming*

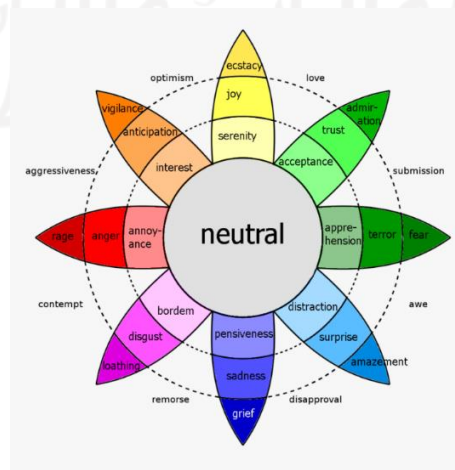
Sebelum	Sesudah
Sebenarnya emang mbak irene agak terkesan songong terkait isu dewa kipas, tapi gw yakin banyak orang yang hobi dan apdet dengan catur juga ngerasain panas emosi yang sama ngeliat komentar goblok netizen indo yang nunjukkin betapa awam dan sotoynya mereka So yea i dont blame her	Sebenarnya memang mbak irene agak kesan sombong terkait isu dewa kipas, tapi saya yakin banyak orang yang hobi dan apdet dengan catur juga merasakan panas emosi yang sama ngeliat komentar goblok netizen indo yang menunjukkan betapa awam dan sotoy mereka So yea i dont blame her

3.4 Pelabelan Data

Dataset yang sudah melalui proses *pre-processing* dan data sudah memenuhi kriteria peneliti selanjutnya dilakukan *labeling* satu persatu sesuai dengan label emosinya, ada sembilan jenis emosi yang dipakai pada penelitian ini, di antaranya sebagai berikut:

- a. Marah
- b. Bahagia
- c. Sedih
- d. Tertarik
- e. Takut
- f. Netral
- g. Percaya
- h. Jijik
- i. Kaget

Sembilan label emosi tersebut sudah mewakili setiap emosi pada *Plutchik wheel of emotions*. Seperti pada *Plutchik's wheel of emotions* di gambar 3.3, setiap emosi memiliki lapisan-lapisan emosi yang dimana setiap emosi memiliki kelas-kelas tingkatan, misalkan pada emosi marah yang memiliki kelas-kelas tingkatan yaitu terganggu, marah, dan murka. Walaupun terdapat pada satu kelas emosi yang sama, tetapi memiliki tingkatan sesuai apa yang dirasakan oleh seseorang. Jadi peneliti hanya mengambil salah satu emosi pada setiap kelas yang dapat mewakili seluruh tingkatan emosi lainnya. Menurut Robert Plutchik emosi dibagi menjadi delapan emosi utama yang mana setengah emosi ini adalah positif dan setengahnya lagi negatif. Emosi ini kemudian berkembang lagi menjadi emosi sekunder seperti jijik, antisipasi, kepercayaan, dan sebagainya (Mohsin & Beltiukov, 2019).



Gambar 3.3 *Plutchik's wheel of emotions*

Teknik pelabelan data menggunakan metode manual yang dilakukan dengan dengan menentukan kalimat opini dan emosi pada data. Pemilihan label emosi akan dibantu oleh mahasiswa psikologi guna mendapatkan hasil yang akurat. Sebagian data dalam penelitian ini diambil dari penelitian yang telah dilakukan oleh Riza & Charibaldi (2021). Dari penelitian tersebut diperoleh data sebanyak 1300 data yang berisikan beberapa emosi, di antaranya bahagia, sedih, takut, jijik, marah, dan kaget. Berikut adalah contoh kalimat yang sudah ditentukan emosinya pada dataset yang dapat dilihat pada tabel 3.7.

Tabel 3.7 Contoh pelabelan data

Data	Label
kurang ajar bgt sih lo!! masuk rumah orang seenaknya	Marah
seneng banget akhirnya lolos ptn	Bahagia
gak tau lagi mesti gimana, semua pada ninggalin gw	Sedih
wah produk terbaru apple keren banget jadi pengen beli	Tertarik
mendingan kalian ikut seminar untuk mengisi hari libur kalian	Netral
gw yakin dia gabakal ngelakuin itu	Percaya
ihh jorok banget sih main di tempat sampah	Jijik
wah ga nyangka bisa seramai ini	Kaget
jadi merinding sendiri denger suara aneh di luar	Takut

3.6 Pemodelan

Tahap pemodelan, data yang sebelumnya sudah di *pre-processing* dan ditentukan labelnya akan masuk ke tahap pemodelan. Tahap pemodelan dilakukan sebanyak dua kali untuk melihat model mana yang lebih baik dan akurat. Model yang dipakai adalah *BERT Uncased* dan *IndoBERT*. Kedua model tersebut memiliki perbedaan dalam *base* bahasa yang digunakan. *BERT Uncased* menggunakan bahasa Inggris sedangkan *IndoBERT* menggunakan bahasa Indonesia. Peneliti melakukan uji coba pertama menggunakan *BERT Uncased*, agar mendapatkan akurasi yang baik peneliti mengubah dataset yang sebelumnya bahasa Indonesia menjadi bahasa Inggris dengan menggunakan *tools* terjemahan.

a. *BERT Uncased*

BERT base adalah model *transformer* yang telah dilatih sebelumnya dengan kumpulan data bahasa Inggris, model *BERT base* dibagi menjadi dua yaitu *BERT Uncased* dan *BERT Cased*. Pada penelitian ini peneliti menggunakan *BERT Uncased*. *BERT Uncased* digunakan karena memerlukan dataset yang rapi. Dataset harus menggunakan huruf kecil (*lowercase*) sebelum masuk ke tahap tokenisasi dan menghapus simbol aksentuasi pada huruf agar data dapat diproses, hal ini dapat meningkatkan kinerja pada model. Tahapan pemodelan antara lain.

1. Memanggil model *BERT Uncased* di *tensorflow hub* (*bert_en_uncased_preprocess*).
2. Membuat *layer* diantaranya *layer BERT*, *neural network layer*, dan menggunakan *input* dan *output* untuk membangun model akhir.
3. Menampilkan hasil *model*.

b. *IndoBERT*

Bahasa Indonesia memiliki penutur yang cukup banyak, bahkan bahasa Indonesia merupakan bahasa terbesar nomor empat di dunia. Akan tetapi, dalam *resource NLP* bahasa Indonesia masih terbilang terbatas. *IndoBERT* adalah *pre-trained* model yang diadaptasi dari cara kerja *Convolutional Neural Network* dan dilatih dengan menggunakan *transformer*. Penelitian Koto et al, (2020) memperkenalkan *IndoBERT*, model bahasa *pre-trained* baru untuk bahasa Indonesia yang memiliki kinerja yang baik. Peneliti menggunakan model *IndoBERT* yang telah dibangun sebelumnya untuk melatih data yang sudah *dipreprocessing*. Proses yang dilakukan diantaranya adalah.

1. Memanggil model *BERT* dengan basis bahasa Indonesia.
2. Membangun model dengan menambahkan *layer* diantaranya *layer* utama, membuat *model input*, memuat *model BERT transformers* sebagai *layer* dalam *model keras*, membangun *model output*, dan mengkombinasikan semua pada objek *model*.
3. Menampilkan hasil *model*.

3.7 Melatih Model

Setelah arsitektur model terbentuk, kemudian model dilatih dengan *hyperparameter* sebagai berikut:

- a. *Optimizer (adam)*, dalam optimisasi peneliti menggunakan *Adam*. *Adam* merupakan gabungan dari *RMSprop* dan *Stochastic Gradient Descent*. *Adam* dipakai karena

memiliki kelebihan seperti mudah diterapkan, komputasi yang efisien, sesuai dalam hal parameter, dan sesuai untuk gradien dengan *noise* yang tinggi.

- b. *Loss (CategoricalCrossentropy)* digunakan untuk memecahkan masalah pada klasifikasi biner dengan nilai target berada di set (0, 1).
- c. *Validation_split* digunakan untuk membagi data uji dan data latih, pembagian data dengan proporsi 10% data uji dan 90% data latih. Untuk mengetahui kinerja model adalah dengan mengukur akurasi. *Split validation* memiliki dua konsep, yaitu *training error* dan *test error*. *Training error* didapatkan dengan menghitung kesalahan klasifikasi model pada data yang dilatih, sedangkan *test error* menggunakan dua data yang sepenuhnya terpisah. Satu untuk melatih dan satunya untuk menguji.
- d. *Batch_size = 32*, *batch size* merupakan jumlah sampel data yang akan disebarkan ke neural network. Pada penelitian ini peneliti memiliki 2515 data dan akan menggunakan 32 sampel data pertama dari 2515 data tersebut.
- e. *Epochs = 10*, penggunaan *epochs* 10 karena dapat menghasilkan nilai akurasi yang baik.

3.8 Evaluasi

Setelah model dilatih, tahap selanjutnya adalah evaluasi. Evaluasi di sini berguna untuk mengetahui kelayakan dari hasil model yang sudah dilatih dengan melihat hasil akurasi pada data tes. Pengujian model ini untuk mengetahui nilai akurasi pada model saat dievaluasi, selain itu juga akan dilakukan pengujian dengan memasukkan kalimat opini untuk mengetahui kelayakan model dalam menentukan label emosi.

Pada tahap evaluasi model, peneliti menguji model dengan membuat label emosi dan membuat kategori emosinya. Karena label dataset masih berbentuk *string*, peneliti akan mengubahnya ke dalam bentuk angka menggunakan *tokenizer*. Setelah diubah menjadi angka, model siap untuk dievaluasi dengan keterangan x adalah input kalimat dan y adalah outputnya yang berupa label emosi.

Setelah melalui tahapan evaluasi, peneliti akan melakukan *test predict* dan *confussion matrix*. *Test predict* dan *confussion matrix* berfungsi untuk mengetahui kelayakan model untuk menentukan label emosi pada kalimat yang dimasukan dan kalimat yang diprediksi dengan hasil aktualnya. Pengujian ini akan dilakukan dengan tahapan seperti di bawah ini:

- a. *Predict single text*, setelah hasil akurasi telah didapatkan, peneliti akan melakukan tes prediksi pada model, dengan memasukkan kalimat opini untuk mengetahui emosi yang terkandung dalam teks tersebut.

b. *Inference multiple text*, yaitu mencoba untuk menampilkan beberapa teks kalimat pada dataset untuk mengetahui apakah model yang telah dibangun berhasil. Keluaran yang dihasilkan berupa tabel yang berisikan tabel kalimat pada *data test*, *data test* digunakan untuk menguji kelayakan model, proposi yang digunakan pada *data test* adalah 10% dari dataset. Kemudian model akan berusaha untuk memprediksi emosi pada *data test* tersebut.

c. *Confussion matrix*

Confussion matrix adalah tabel yang merangkum kinerja model. Penggunaan *confussion matrix* dapat mengidentifikasi kelemahan dalam operasi algoritma. Ini digunakan untuk mengevaluasi eektivitas model klasifikasi. Dengan menggunakan *confussion matrix* peneliti dapat mengetahui *error* pada operasi algoritma yang dijalankan. Tabel 3.8 menunjukkan bentuk dasar *confussion matrix*.

Tabel 3.8 Contoh *confussion matrix*

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

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Tabel 3.9 *Multiclass confussion matrix*

		<i>Predicted Class</i>			
		C_1	C_2	C_N
<i>Actual Class</i>	C_1	$C_{1,1}$	FP	$C_{1,N}$
	C_2	FN	TP	FN

	C_N	$C_{N,1}$	FP	$C_{N,N}$

Keterangan:

- a. TP adalah *true positive* yaitu sebuah data dengan kondisi aktual yang mampu memprediksi dengan benar.
- b. TN adalah *true negative* adalah data negatif yang diprediksi dengan benar.
- c. FP adalah *false positive* yaitu sebuah data prediksi yang tidak sesuai.
- d. FN adalah *false negative* yaitu sebuah data negatif yang diprediksi tidak sesuai.
- e. C adalah kelas label.

Pada klasifikasi *multiclass* (Markoulidakis et al., 2021), *matrix* yang ditentukan untuk klasifikasi, tidak berlaku secara penuh. *Confussion matrix multiclass* seperti pada tabel 3.8 memiliki dimensi $N \times N$ dimana N adalah jumlah label kelas yang berbeda C_0, C_1, \dots, C_N oleh karena itu karakterisasi TP, TN, FP, dan FN tidak berlaku dalam kasus ini. Dengan demikian dimungkinkan untuk memberikan pengukuran *confussion matrix* penuh tergantung pada parameter. Maka akan menggunakan rumus

$$Accuracy = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N C_{i,j}} \quad (3.2)$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Dataset

Teknik *scraping* merupakan teknik yang peneliti lakukan pada pengumpulan data. Pengumpulan data pada teknik *scraping* yang peneliti lakukan ialah dengan menggunakan kata kunci untuk mendapatkan opini-opini masyarakat di *Twitter*. Setelah melakukan pengumpulan dataset, data telah terkumpul sebanyak 2515 data teks opini. Data ini diambil dari opini-opini para pengguna *Twitter* secara acak. Setelah dataset terkumpul lalu dipre-processing untuk membersihkan kata, simbol, dan tanda baca yang tidak perlu guna mempermudah proses pelabelan data. Jumlah data sebanyak 2515 sudah memenuhi kriteria untuk melakukan pelatihan model, karena pada penelitian-penelitian sebelumnya rata-rata peneliti membutuhkan 1000 sampai 2000 data dan sudah mendapatkan model dengan akurasi yang baik. Contoh hasil data yang terkumpul pada Gambar 4.1.

1	menurut saya sebagai negara non blok indo harus memiliki sikap netral
2	nonitrend multivitamin ekstrak buah mengkudu bahan baku alami harga murah riah pin
3	saran saya sebagai penengah hanya bisa menilai mana yg baik dan salah
4	b wouldaripada pikir seminar new generation pikir buka lebar
5	putus pacar
6	klinik tiebandung cari agen reseller indonesia hubung paruhwaktu
7	curiga
8	turun berat badan minum cahar milik efek samping mending minum double
9	follow motivatoralami motivasi
10	pacar hasil selingkuh be ahan tuju
11	susah bab sembelit pencaharalami serat ganda double cellulose tie botol langsing sehat pin
12	doublecellulose bersih salur cerna bentuk tablet hisap kayak permen murah langsing pin
13	kuliah bingung cari hasil agen tiebandung komisi
14	pacar goodnight sayang i love ldr goodnight sayang i miss tuju
15	langsing serat alami turun minggu bersih usus detox usus pin
16	jual online masker putih wajah chitosan tie kapsul free kuas wajah pin
17	langsing akupunktur akupunktur coba jalan braga no bandung pin
18	b wouldukung gera indonesiasehat follow tiebandung baca info sehat dukung indonesia sehat
19	orang orang nih love is blind
20	reseller tiebandung negeri hasil
21	baca info tips teknik badan
22	jomlo menang tonton film sedih peras gue netesin air mata
23	ketemu jodoh
24	tehtie bantu bersih salur darah hindar serang jantung box pin
25	kucing hanya bisa hidup 20 tahun
26	kucing kayak jodoh kali ya kejar
27	follow motivatoralami info motivasi galau

Gambar 4.1 Contoh hasil pengumpulan data

4.2 Pre-processing

Setelah dataset dikumpulkan, proses selanjutnya adalah *pre-processing*, tahapan ini memanfaatkan penggunaan *library* di pemrograman bahasa Python. Berikut adalah penjelasan dari setiap langkah *pre-processing* yang dapat dilakukan.

4.2.1 Menghapus Simbol-simbol

Data diambil dengan melakukan *scraping* pada Twitter. Setelah data dikumpulkan ke dalam bentuk *file* csv, ternyata masih ditemukan beberapa simbol-simbol yang kurang bermakna seperti (@, ^, \$, %, &, !). Hal ini membuat kalimat menjadi tidak rapi dan sulit untuk dibaca. Oleh karena itu hasil dari *scraping* tersebut harus melalui proses pembersihan data. Dengan memanggil *charcleaning* dan memasukan simbol-simbol yang perlu dihapus pada kalimat di dataset. Kode program yang digunakan untuk melakukan penghapusan simbol dapat dilihat pada gambar 4.2.

```
def charCleaning(sentence):
    sentence = re.sub(r'https?://\S+|www.\S+', r'', sentence)
    sentence = re.sub('\n', ' ', sentence)
    sentence = re.sub('rt', ' ', sentence)
    sentence = re.sub('<.*?>', r'', sentence)
    sentence = re.sub('\d+', '', sentence).strip()
    sentence = re.sub(r"[\w\s\d]", "", sentence)
    sentence = re.sub(r'@\w+', '', sentence)
    sentence = re.sub(r#\w+', '', sentence)
    sentence = re.sub(r"\s+", " ", sentence).strip()
    sentence = re.sub('user', ' ', sentence)
    sentence = re.sub("\S*\d\S*", "", sentence).strip()
    sentence = re.sub('[^A-Za-z]+', ' ', sentence)

    return sentence
```

Gambar 4.2 Kode program untuk menghapus simbol-simbol

4.2.2 Casefolding

Huruf terdiri dari dua jenis, huruf besar dan huruf kecil, di dalam teks opini pada dataset yang dikumpulkan, terdapat beberapa huruf besar dan kecil, agar data menjadi konsisten maka semua teks dalam dataset diubah menjadi satu jenis huruf yaitu *lowercase*. Untuk mengubah kalimat opini tersebut menjadi *lowercase*, dilakukan *casefolding* dimana karakter data dinormalisasi menjadi menggunakan *lowercase*. Gambar 4.3 menunjukkan kode program penggunaan *casefolding*.

```
def lowercase(sentence):
    return sentence.lower()
```

Gambar 4.3 Kode pemrograman *casefolding*

4.2.3 Remove Slang

Dalam sebuah kalimat opini terdapat kata-kata yang kurang bermakna dan bahasa yang tidak baku, maka dari itu kata tersebut harus dihilangkan dan diubah kedalam bentuk bahasa yang lebih baku. Tahapan ini memerlukan kamus daftar bahasa tidak baku dengan basis bahasa indonesia. Untuk menerapkannya peneliti perlu menyesuaikan leksikonnya ke dalam bahasa indonesia. Penerapan *remove slang* menggunakan kode pemrograman pada gambar 4.4.

```
# import Indonesian lexicon
spell.lex <- read.csv("data_input/colloquial-indonesian-lexicon.csv")

# replace internet slang
tweets <- replace_internet_slang(tweets, slang = paste0
("\b",spell.lex$slang,
"\b"),replacement = spell.lex$formal,
ignore.case = TRUE)
```

Gambar 4.4 Kode pemrograman *remove slang*

4.2.4 Stopword Removal

Stopword removal atau *filtering* yaitu memproses kata-kata penting dari hasil token, atau kata-kata yang akan menjadi representasi, karena pada proses klasifikasi teks *stopword* perlu dihilangkan atau disaring agar data yang diperoleh lebih jelas maknanya. Untuk menerapkannya perlu menggunakan kode pemrograman seperti pada gambar 4.5.

```
def stopWordRemover(sentence):
    text = ""
    for word in sentence.split():
        if word not in stop_word_dict.values:
            text = text + " " + word
    return text.strip()
```

Gambar 4.5 Kode pemrograman *stopword removal*

4.2.5 Stemming

Kalimat opini pada dataset tersebut beberapa kata masih memiliki imbuhan, fungsi *stemming* disini adalah untuk menghapus atau mengurangi suatu kata imbuhan dan mengubahnya menjadi bentuk dasar kata yang sama. Penerapan *stemming* dilakukan dengan menggunakan kode pemrograman pada gambar 4.6.

```
def stemmer(sentence):
    # create stemmer7
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(sentence)
```

Gambar 4.6 Kode pemrograman *stemming*

4.2.6 Hasil *pre-processing*

Pada tahap *preprocessing* yang dilakukan oleh peneliti, menghasilkan luaran yang dapat dilihat pada tabel 4.1 dan tabel 4.2. pada implementasinya peneliti menemukan perbedaan pada dataset sebelum dan sesudah di *preprocessing*. Data yang sudah di *preprocessing* cenderung lebih rapi dan tertata, seperti bentuk huruf yang berubah menjadi kecil (*lowercase*), kata imbuhan yang hilang, kata-kata kurang bermakna juga terhapus, dan kalimat menjadi rapi karena tidak adanya simbol-simbol yang mengurangi makna pada kalimat opini tersebut.

Tabel 4.1 Data twitter sebelum *pre-processing*

Username	Tweet
bestieyang	@yangifted dah, emosi gua
Nopipebriantii	@gumvbal Setauku pernah denger ada yg bilang di drama itu gatau episode berapa, katanya psikopat itu meskipun gapunya emosi tapi dia bisa memainkan emosi sesukanya, seakan akan dia keliatan kalo dia punya emosi. Psikopat ga selamanya bermuka dingin. Gitu katanya
Syxhistoire	sdank emosi
Hfdnrchf	Hiii abis kerja fulltime baru sadar kalo emosi jdi makin sulit ke kontrol, apalagi sama ade2 yg diajarin gapaham2, uda cape pulang malem, masi ditanya2 PR, disuruh jg ngga buru dikerjain, emosi doang rasanya jadinya ðŸ˜~ gabisa nih jdi ibu karir, I think I'll choose my career ofc
nishisyraf02	jangan cabar emosi orang yang tengah bekerja
deyss20	Halah lengkap bgt ini mau pulang ada aja yg bikin emosi
666baji	@juhgchrys sumpah dia batu banget gw emosi sendirinya liatnya

Tabel 4.2 Data twitter sesudah *pre-processing*

Username	Tweet
bestieyang	yangifted dah emosi gua
nopipebriantii	gumvbal setauku pernah denger ada yg bilang di drama itu gatau episode berapa katanya psikopat itu meskipun gapunya emosi tapi dia bisa memainkan emosi sesukanya seakan akan dia keliatan kalo dia punya emosi psikopat ga selamanya bermuka dingin gitu katanya
syxhistoire	sdank emosi
hfdnrchf	hiii abis kerja fulltime baru sadar kalo emosi jdi makin sulit ke kontrol apalagi sama ade yg diajarin gapaham uda cape pulang malem masi ditanya pr disuruh jg ngga buru dikerjain emosi doang rasanya jadinya gabisa nih jdi ibu karir i think ill choose my career ofc
nishisyraf02	jangan cabar emosi orang yang tengah bekerja
deyss20	halah lengkap bgt ini mau pulang ada aja yg bikin emosi
666baji	juhgechrys sumpah dia batu banget gw emosi sendirinya liatnya

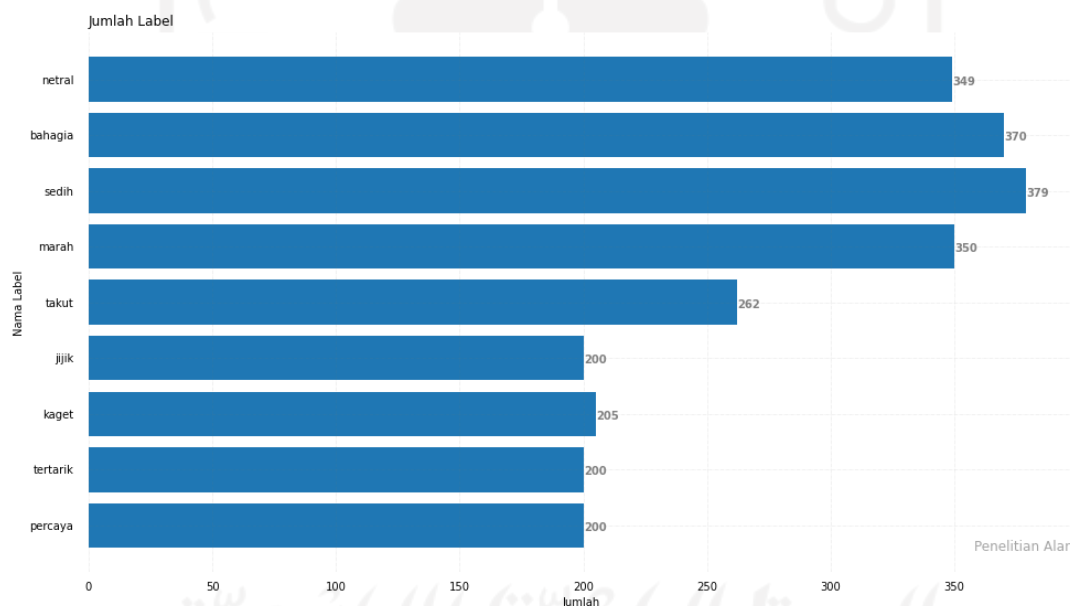
4.3 Pelabelan Data

Dari 2515 data yang sudah terkumpul kemudian dipisahkan sesuai kategori emosi masing-masing. Proses pelabelan data yang dilakukan oleh peneliti menggunakan metode manual, yaitu dengan membaca satu per satu dan menganalisa dataset tersebut untuk mengetahui emosi yang terkandung pada tiap kalimat opini tersebut. Tabel 4.3 menunjukkan jumlah hasil dari pelabelan data dan gambar 4.8 menunjukkan pesebaran emosi pada dataset dalam bentuk grafik per label emosi, fungsinya agar peneliti mengetahui jumlah data perlabel emosi untuk mengetahui apakah selisih data per emosi terlampau jauh atau tidak, ini untuk menghindari adanya ketidakseimbangan pada dataset. Selanjutnya pada gambar 4.7 adalah pesebaran kata-kata yang sering digunakan atau sering muncul pada dataset dalam bentuk *wordcloud* ini berguna untuk mengetahui kata apa saja yang sering digunakan untuk meluapkan emosi pada teks tersebut.

Gambar 4.7 *Wordcloud* pada dataset

Tabel 4.3 Jumlah data per-label

Label	Data
Marah	350
Bahagia	370
Sedih	379
Tertarik	200
Takut	262
Netral	349
Percaya	200
Jijik	200
Kaget	205



Gambar 4.8 Data pesebaran emosi

Proses menentukan emosi dengan menggunakan metode manual menghasilkan label data emosi seperti pada tabel 4.4. Penentuan emosi dengan cara manual adalah peneliti dengan dibantu oleh ahli contohnya seseorang yang bergerak dibidang psikologi. Dalam kasus ini peneliti dibantu oleh mahasiswa psikologi untuk menentukan emosi pada tiap kalimat opini agar tidak terjadi kesalahan dalam menentukan emosi pada teks tersebut.

Tabel 4.4 Contoh hasil pelabelan data

No	Data	Label
1	maaf bgt ya ini konten sensitif menurut gw bapak kaya gini ga pantes buat hidup kalo perlu di potong kelaminnya biar jera bikin emosi aja anj	Marah
2	terima kasih playstation telah banyak berbuat baik dengan hidupku sampai sekarang	Bahagia
3	bismillah, ini hari kedua sejak istriku dimakamkan di pemakaman khusus covid di macanda, gowa, sulawesi selatan. setelah pemakaman di malam itu, aku kembali lagi ke rumah isolasi mandiri. seorang diri.	Sedih
4	melihat semangat adik adik pita kuning membuat saya ingin untuk ikut bergabung	Tertarik
5	ini covid bikin gua jadi parno kemana-mana, sampe-sampe semingguan ini ga keluar rumah	Takut
6	kamu bisa menambah tinggi badan dengan olahraga dan memperbaiki pola makan dan cara tidurmu loh	Netral
7	saya mengandalkanmu segenap jiwa saya	Percaya
8	gara gara akun pesbuk jessica gue jadi keikut buka juga, untung masih inget dan anjir jijik banget yallah dari ngakak sampe merinding gue baca mess sama bacotan gue di pesbuk	Jijik
9	wah gak nyangka bisa serame ini kirain bakal ga laku ni warung	Kaget

Untuk mengidentifikasi emosi pada teks, peneliti menggunakan cara manual sebagai berikut:

1. Pada keterangan label pertama yaitu marah dengan kalimat *“maaf bgt ya ini konten sensitif menurut gw bapak kaya gini ga pantes buat hidup kalo perlu di potong kelaminnya biar jera bikin emosi aja anj”* Pada kalimat tersebut dapat dilihat pada kata *“bapak kaya gini gapantes buat hidup”*. Dimana kalimat tersebut menyampaikan emosi marah dan diperjelas lagi pada kalimat terakhirnya *“bikin emosi aja”*
2. Pada keterangan label kedua yaitu bahagia dengan kalimat *“terima kasih playstation telah banyak berbuat baik dengan hidupku sampai sekarang”*. Pada kalimat tersebut

penulis menyampaikan kebahagiaan dengan kata terima kasih dan diperkuat dengan kalimat selanjutnya yaitu *“telah banyak berbuat baik dengan hidupku”*

3. Pada keterangan label ketiga yaitu sedih dengan kalimat *“bismillah, ini hari kedua sejak istriku dimakamkan di pemakaman khusus covid di macanda, gowa, sulawesi selatan. setelah pemakaman di malam itu, aku kembali lagi ke rumah isolasi mandiri. seorang diri”* pada kalimat tersebut penulis menyampaikan emosi sedih karena sedang merasa kehilangan dan diperkuat lagi pada kalimat terakhirnya *“aku kembali lagi ke rumah isolasi mandiri. seorang diri”* yang menggambarkan jelas kesedihan yang dideritanya.
4. Pada keterangan label keempat yaitu tertarik, dengan kalimat *“melihat semangat adik-adik pita kuning membuat saya ingin untuk bergabung”*. Pada kalimat tersebut menggambarkan penulis tertarik untuk mengikuti sesuatu kegiatan karena melihat antusiasme suatu kelompok atau organisasi.
5. Pada keterangan label kelima yaitu takut, dengan kalimat *“ini covid bikin gua jadi parno kemana-mana, sampe-sampe semingguan ini ga keluar rumah”*. Pada kalimat tersebut menggambarkan ketakutan dikarenakan suatu pandemi hingga penulis takut untuk melakukan aktivitas diluar rumah.
6. Pada keterangan label keenam yaitu netral, dengan kalimat *“kamu bisa menambah tinggi badan dengan olahraga dan memperbaiki pola makan dan cara tidurmu loh”*. Pada kalimat ini mengandung suatu informasi untuk seseorang atau kelompok. Peneliti menilai kalimat tersebut termasuk netral.
7. Pada keterangan label ketujuh yaitu percaya, dengan kalimat *“saya mengandalkan segenap jiwa saya”*. Pada kalimat tersebut menggambarkan suatu emosi percaya karena penulis percaya pada dirinya sendiri.
8. Pada keterangan label kedelapan yaitu jijik, dengan kalimat *“gara gara akun pesbuk jessica gue jadi keikut buka juga, untung masih inget dan anjir jijik banget yallah dari ngakak sampe merinding gue baca mess sama bacotan gue di pesbuk”*. Pada kalimat tersebut peneliti mengidentifikasi emosi sebagai jijik karena pada kalimat tersebut penulis merasakan kejjjikan yang telah ia lakukan di masa lalu
9. Pada keterangan label kesembilan yaitu kaget atau terkejut, dengan kalimat *“wah gak nyangka bisa serame ini kirain bakal ga laku ni warung”*. Pada kalimat tersebut peneliti mengidentifikasikan sebagai emosi terkejut dikarenakan penulis tidak menyangka apa yang ia pikirkan dan yang terjadi sebenarnya.

4.4 Perancangan Model

Perancangan model adalah tahap selanjutnya setelah tahap *pre-processing*. Untuk melakukan klasifikasi peneliti harus merancang model terlebih dahulu. Terdapat dua model yang digunakan dalam penelitian ini yang bertujuan untuk membandingkan model mana yang lebih baik dan akurat. Model yang digunakan antara lain *IndoBERT* dan *BERT Uncased*.

BERT Uncased dan *IndoBERT* merupakan model yang diterapkan pada penelitian ini. Dua model ini digunakan untuk membandingkan nilai akurasi tertinggi dari kedua model tersebut.

a. *BERT Uncased*

Pada perancangan model pertama menggunakan *BERT Uncased*. Penerapan model dapat dilihat pada gambar 4.9. *BERT Uncased* disini memiliki beberapa *layer* diantaranya *layer BERT* untuk memasukan *data train*, yang kedua adalah *neural network layer* yang berisi *layer dropout* yang berfungsi untuk mencegah adanya *overfitting* dan berguna untuk mempercepat pengerjaan *learning*, selanjutnya *dense* dengan aktivasi *softmax*, dan terakhir *layer output*.

```
# Bert layers
text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
preprocessed_text = bert_preprocess(text_input)
outputs = bert_encoder(preprocessed_text)

# Neural network layers
l = tf.keras.layers.Dropout(0.1, name="dropout")(outputs['pooled_output'])
l = tf.keras.layers.Dense(8, activation='softmax', name="output")(l)

# Use inputs and outputs to construct a final model
model = tf.keras.Model(inputs=[text_input], outputs = [l])
```

Gambar 4.9 Kode pemrograman model arsitektur *BERT Uncased*


```

# Then build your model output
hidden_layer1 = Dense(units=64, name='hidden_layer1', activation='relu')(pooled_output)
hidden_layer2 = Dense(units=32, name='hidden_layer2', activation='relu')(hidden_layer1)

emosi = Dense(units=len(data.emosi_label.value_counts()), kernel_initializer=TruncatedNormal(stddev=config.initializer_range), name='emosi')(hidden_layer2)

outputs = {'emosi': emosi}

# And combine it all in a model object
model = Model(inputs=inputs, outputs=outputs, name='BERT_MultiLabel_MultiClass')
# Take a look at the model
model.summary()

```

Gambar 4.11 Kode pemrograman model arsitektur IndoBERT

Layer (type)	Output Shape	Param #
input_ids (InputLayer)	[(None, 200)]	0
bert (TFBertMainLayer)	TFBaseModelOutputWithPoolingAndCrossAttentions(1, past_hidden_state=(None, 200, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	124441344
pooled_output (Dropout)	(None, 768)	0
hidden_layer1 (Dense)	(None, 64)	49216
hidden_layer2 (Dense)	(None, 32)	2080
emosi (Dense)	(None, 9)	297

Total params: 124,492,937		
Trainable params: 124,492,937		
Non-trainable params: 0		

Gambar 4.12 Model arsitektur *IndoBERT*

4.5 Melatih Model

Setelah model terbangun, tahapan selanjutnya adalah menjalankan masing-masing model. Model akan dilatih untuk mendapatkan nilai akurasi yang baik dan layak untuk menguji emosi dengan tepat. Pada penelitian ini ada dua model yang dijalankan yaitu BERT Uncased dan IndoBERT.

4.5.1 BERT *Uncased*

Tahapan menjalankan model *BERT Uncased*, dengan tahapan pertama yaitu membuat matriks untuk akurasi, presisi, dan *recall*. Tahapan kedua adalah menyusun model dengan optimisasi *adam*, *loss categorical_crossentropy*, dan matriks adalah matriks pada tahapan pertama. Tahapan tersebut dapat dilihat pada gambar 4.13.

```
METRICS = [
    tf.keras.metrics.BinaryAccuracy(name='accuracy'),
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall')
]

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=METRICS)
```

Gambar 4.13 Kode pemrograman *train model BERT Uncased*

4.5.2 IndoBERT

Tahapan menjalankan model IndoBERT, menjalankan model tersebut dengan *hyperparameters* seperti pada gambar 4.14. dengan tahapan pertama yaitu mengatur pengoptimal model dengan *adam*, selanjutnya tahapan kedua yaitu mengatur *loss* dan matriks, tahapan ketiga adalah menyusun model, dan yang terakhir yaitu membuat *output* data untuk model.

```
optimizer = Adam(
    learning_rate=5e-05,
    epsilon=1e-08,
    decay=0.01,
    clipnorm=1.0)
# Set loss and metrics
loss = {'emosi': CategoricalCrossentropy(from_logits = True)}
metric = {'emosi': CategoricalAccuracy('accuracy')}
# Compile the model
model.compile(
    optimizer = optimizer,
    loss = loss,
    metrics = metric)
# Ready output data for the model
label2num = {'marah':0, 'bahagia':1, 'sedih':2, 'tertarik':3, 'net
ral':4, 'percaya':5, 'jijik':6, 'takut':7, 'kaget':8}
data.emosi_label = data.emosi_label.map(label2num)
y_emosi = to_categorical(data['emosi_label'])
```

Gambar 4.14 Kode pemrograman *train model IndoBERT*

4.6 Hasil Penelitian

Dalam melakukan klasifikasi emosi pada teks, peneliti membagi data latih dan data uji. Data latih (*training data*) adalah data yang digunakan untuk melatih sebuah model agar mendapatkan nilai akurasi yang sesuai. Data uji (*test data*) adalah data yang digunakan untuk menguji nilai akurasi model yang diperoleh pada data latih. Pembagian data latih dan data uji dengan total data yang peneliti gunakan sebesar 2515 data yang berisikan opini pengguna *twitter*. Dalam pembagiannya peneliti menggunakan proporsi 90% data latih dan 10% data uji.

$$\begin{aligned} \text{Data latih} &= 90\% \times 2515 \\ &= 2263 \end{aligned}$$

$$\begin{aligned} \text{Data uji} &= 10\% \times 2515 \\ &= 252 \end{aligned}$$

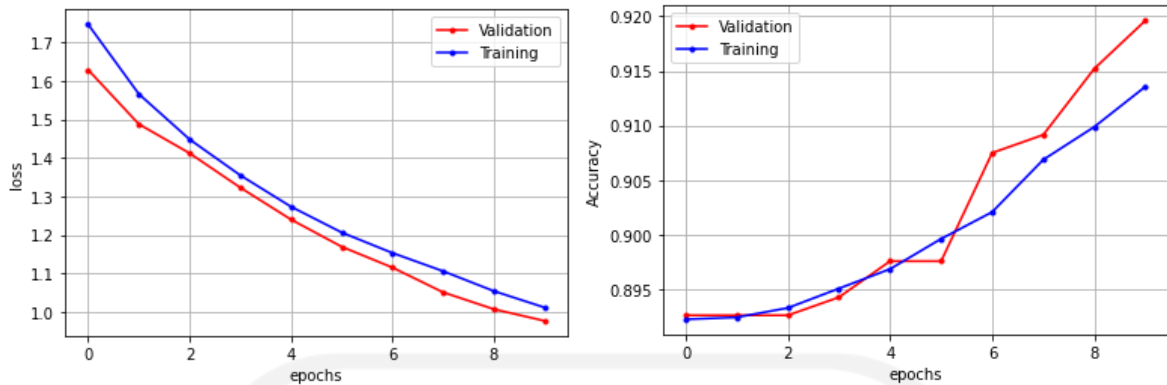
Di bawah ini adalah hasil dari penelitian dan perbandingan menggunakan BERT *Uncased* dan IndoBERT yakni sebagai berikut :

4.6.1 BERT *Uncased*

Pengujian model BERT *Uncased* dilakukan dengan menggunakan *epochs* 10. Penggunaannya menghasilkan nilai akurasi tertinggi pada *epochs* ke 10 sebesar 0.9196 dan *loss* sebesar 0.9763. Tabel 4.5 dan gambar 4.15 adalah hasil dari pelatihan model menggunakan model BERT *Uncased*.

Tabel 4.5 Hasil dari model BERT *Uncased*

Epoch	Loss	Accuracy	Val_Loss	Val_Accuracy
1	1.7326	0.8917	1.6287	0.8926
2	1.5488	0.8925	1.4875	0.8926
3	1.4174	0.8942	1.4119	0.8926
4	1.3206	0.8949	1.3226	0.8943
5	1.2390	0.8990	1.2396	0.8973
6	1.1706	0.9021	1.1689	0.8976
7	1.1115	0.9068	1.1148	0.9075
8	1.0462	0.9094	1.0505	0.9091
9	1.0171	0.9137	1.0066	0.9152
10	0.9626	0.9180	0.9763	0.9196



Gambar 4.15 BERT *Uncased* (Accuracy & Loss)

Model dilatih dengan epoch 10, grafik menunjukkan adanya peningkatan akurasi pada setiap epoch. Pada pengujian akurasi epoch pertama model menghasilkan nilai akurasi sebesar 0,8926, kenaikan yang signifikan pada epoch pertama dilanjutkan pada epoch-epoch selanjutnya hingga pada epoch ke 10 model menghasilkan nilai akurasi sebesar 0.9196 atau 91,96%.

Selanjutnya pada nilai loss, model bert uncased mendapatkan nilai loss sebesar 1,6287 pada epoch pertama. Model terus dilatih hingga epoch ke 10, terlihat pada grafik nilai loss pada model semakin menurun. Hingga pada epoch ke 10 model menghasilkan nilai sebesar 0,9793. nilai tersebut sudah bagus dengan akurasi sebesar 91%, dengan ini model dinilai dapat menentukan emosi pada teks dengan baik pada basis bahasa inggris.

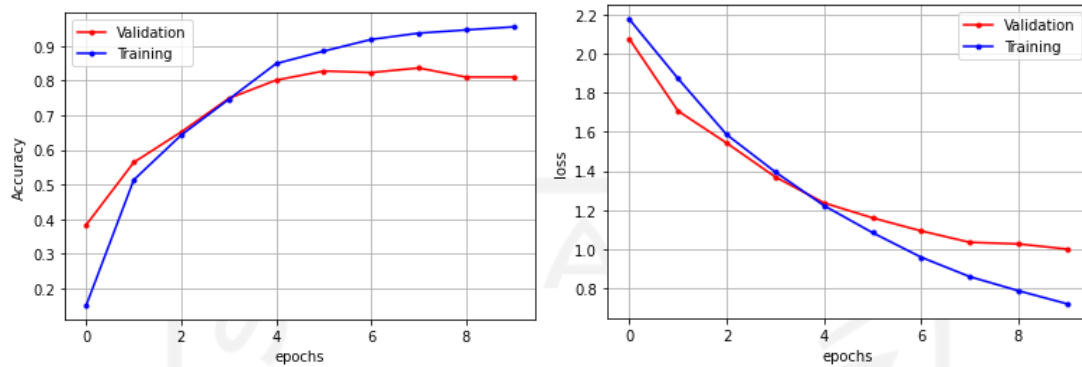
4.6.2 IndoBERT

IndoBERT sendiri adalah versi Indonesia dari model *BERT*, pengujian model diantaranya dengan *parameter validation split* dengan pembagian data *training* 90% dan *test* 10% , *batch size* sebesar 32, dan *epochs* 10. Hasil penelitian dapat dilihat pada tabel 4.6. Pengujian dengan *epochs* 10 mendapatkan nilai akurasi terbesar pada *epochs* ke delapan dengan nilai *validation accuracy* sebesar 0.8370 dan berakhir pada *epochs* ke 10 dengan *validation accuracy* sebesar 0.8106.

Tabel 4.6 Hasil dari model IndoBERT

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	2.1747	0.1508	2.0741	0.3833
2	1.8721	0.5133	1.7063	0.5639
3	1.5853	0.6429	1.5422	0.6539
4	1.3944	0.7461	1.3690	0.7489
5	1.2225	0.8497	1.2364	0.8018
6	1.0838	0.8856	1.1595	0.8282
7	0.9582	0.9194	1.0929	0.8238

8	0.8588	0.9376	1.0346	0.8370
9	0.7868	0.9470	1.0263	0.8106
10	0.7201	0.9558	1.0002	0.8106



Gambar 4.16 IndoBERT (*accuracy & Loss*)

Model indobert dilatih dengan epoch 10. Pelatihan ini menghasilkan nilai akurasi yang signifikan. Terlihat pada epoch pertama, model menghasilkan nilai akurasi sebesar 0.0881. Nilai tersebut semakin meningkat terus menerus saat pelatihan. Rata-rata peningkatan nilai akurasi adalah 10% pada setiap epoch, hingga pada epoch ke 10 model menghasilkan nilai akurasi sebesar 0.8106 atau sebesar 81,06%.

Selanjutnya nilai loss pada model indobert menghasilkan nilai 2,0741 pada epoch pertama. Penurunan nilai loss terlihat signifikan pada grafik, dengan rata-rata penurunan nilai loss sebesar 10% pada setiap epochnya. Maka pada epoch ke 10 model menghasilkan nilai loss sebesar 1.0002. Dengan nilai akurasi sebesar 81%, model dinilai menghasilkan model yang layak diuji untuk menentukan label emosi pada teks opini dengan basis bahasa Indonesia.

4.7 Evaluasi

Evaluasi dilakukan pada model IndoBERT. Pengujian dilakukan agar peneliti mengetahui apakah model sudah berjalan dengan baik dan sesuai. Dari evaluasi yang dilakukan model, pada gambar 4.17 peneliti mendapatkan nilai akurasi sebesar 97,6%.

```
# Run evaluation
model_eval = model.evaluate(
    x={'input_ids': test_x['input_ids']},
    y={'emosi': test_y_emosi}
)
```

8/8 [=====] - 11s 397ms/step - loss: 0.6601 - accuracy: 0.9762

Gambar 4.17 Hasil dari evaluasi model

Selanjutnya dilakukan dua tahap pengujian yaitu tahap *predict single text* dan *inference multiple text*. Dari hasil pengujian tersebut menghasilkan output sebagai berikut:

- a. *Predict Single Text*, tahap evaluasi yang pertama yaitu mengetahui label emosi yang ada pada text yang dimasukan. Peneliti melakukan pengujian dengan menginput teks opini untuk mengetahui emosi yang terkandung dalam teks tersebut. Hasil dari pengujian dapat dilihat pada gambar 4.18.

```
[ ] import numpy as np

def predict_string(text):
    predict_x = tokenizer(
        text=[text],
        add_special_tokens=True,
        max_length=max_length,
        truncation=True,
        padding='max_length',
        return_tensors='tf',
        return_token_type_ids = False,
        return_attention_mask = False,
        verbose = True)

    prediction = model.predict(x={'input_ids': predict_x['input_ids']})
    num2label = {v: k for k, v in label2num.items()}
    answer_real = []
    for i in prediction['emosi']:
        answer_real.append(np.argmax(i))
    string_answer = pd.DataFrame(answer_real).replace(num2label)
    return string_answer.values[0][0]

predict_string("hahh kok bisaa?")

'kaget'
```

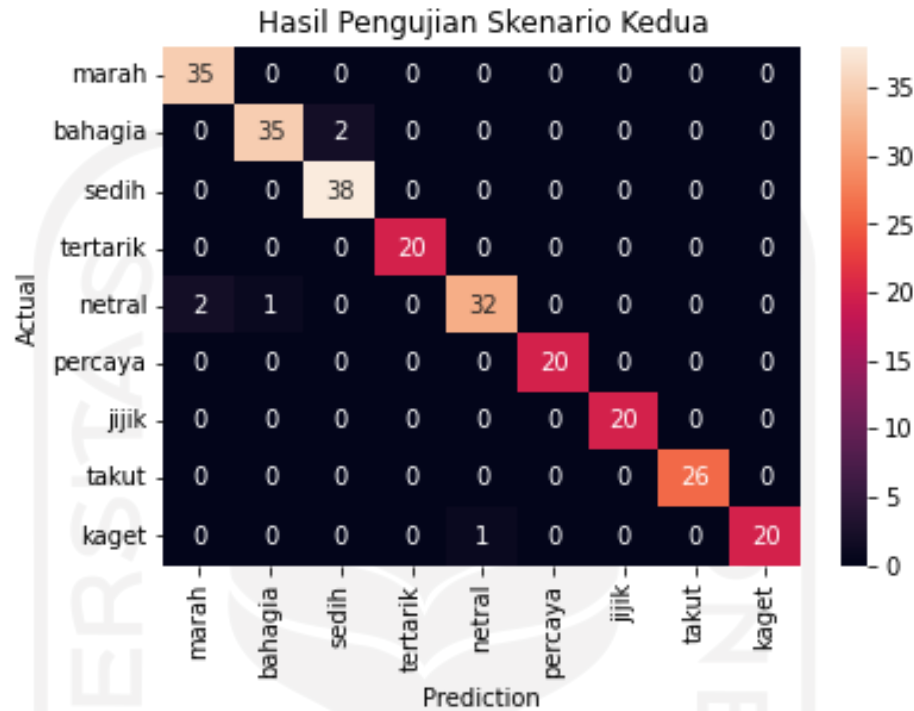
Gambar 4.18 Hasil dari *Predict single text*

- b. *Inference Multiple Text*, tahapan kedua adalah melakukan evaluasi pada dataset yang sudah memiliki label emosi untuk mengetahui keakuratan prediksi. Output yang dihasilkan berupa tabel daftar *data test* untuk menguji nilai aktual pada *data test*. Hasil dari pengujian dapat dilihat pada gambar 4.19.

Sentence	Predicted	Actual
merindukan dan setiap hari merindukanmu	sedih	sedih
ngakak emosi wkwkwkwk	bahagia	bahagia
harus nonton raya and the last dragon bagus bat demii ngakak iya emosi iya nangis iya	bahagia	bahagia
mantap kuy ada twitternya	bahagia	bahagia
goddessholics lu juga bikin emosi y ajg	marah	marah
jijik ya iya sama aku juga	jijik	jijik
percaya sama gw kita bakal menang	percaya	percaya
lebih yakin kalo mereka bersatu	percaya	percaya
auto jijik begitu denger nama nya disebut	jijik	jijik

Gambar 4.19 Hasil dari *Inference multiple predict*

- c. *Confusion matrix*, tahapan ketiga adalah mengukur performa model, model yang dilakukan evaluasi adalah model *IndoBERT*. Model ini akan digunakan karena berbasis bahasa indonesia. Hasil pengujian *confusion matrix* dapat dilihat pada gambar 4.20.



Gambar 4.20 *Confusion matrix multiclass* model *IndoBERT*

Confusion matrix menjelaskan data prediksi per tiap label marah, bahagia, sedih, tertarik, netral, percaya, jijik, takut, dan kaget didapatkan hasil yang sesuai dengan data uji sebesar 250 data. Akan tetapi mesin mendapatkan empat label yang memiliki *error* yaitu marah, sedih, bahagia, dan netral. Dimana label marah memiliki bobot 35 data dan dua data terprediksi ke dalam label netral, label sedih memiliki bobot 38 data tetapi ada dua data yang terdeteksi ke label bahagia, selanjutnya pada label bahagia yang memiliki bobot 35 data terdapat satu data terprediksi netral, dan pada label netral yang memiliki bobot 32 data terdapat satu data terprediksi kaget.

$$Accuracy = \frac{\sum_{i=1}^9 TP(35+35++38+20+32+20+20+29+20)}{\sum_{i=1}^9 \sum_{j=1}^9 (252)} \quad (4.1)$$

$$Accuracy = \frac{\sum_{i=1}^9 TP(246)}{\sum_{i=1}^9 \sum_{j=1}^9 (252)} = 0,976 \quad (4.2)$$

$$Accuracy = 0,984 \times 100 = 97,6\% \quad (4.3)$$

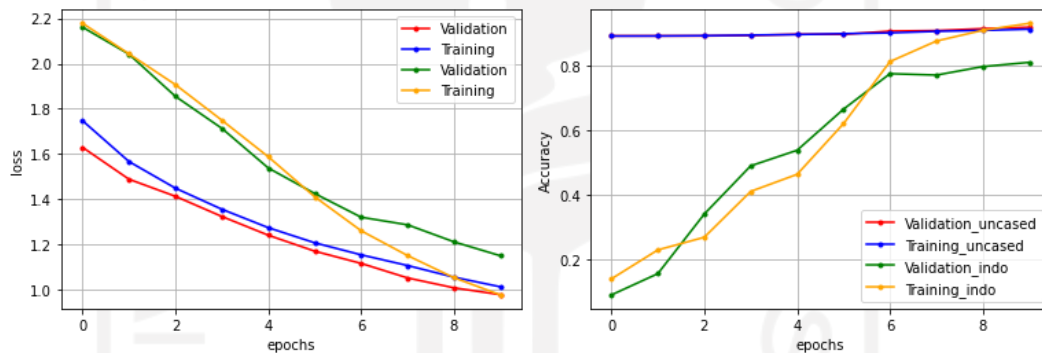
Berdasarkan perhitungan diatas, model menghasilkan nilai akurasi sebesar 97,6%. Berdasarkan nilai akurasi yang diperoleh, model klasifikasi tersebut mendapatkan nilai akurasi yang baik dan layak untuk mengidentifikasi emosi pada teks berbahasa Indonesia.

4.8 Komparasi Model

Hasil pada pengujian dua model yang dilakukan oleh peneliti yaitu model *BERT Uncased* dan *IndoBERT*. Pengujian performa pada model bertujuan untuk mengkomparasi model satu dengan yang lain. Pengujian antara lain dengan:

a. Perbandingan Grafik

Pada proses pertama yaitu membandingkan performa kedua model untuk melihat perbedaan pada tiap model, apakah cukup signifikan atau memiliki perbedaan yang sangat sedikit. Berikut adalah grafik perbandingan akurasi dan loss antara model *BERT Uncased* dan *IndoBERT* pada gambar 4.21.



Gambar 4.21 Perbandingan akurasi dan loss

Keterangan:

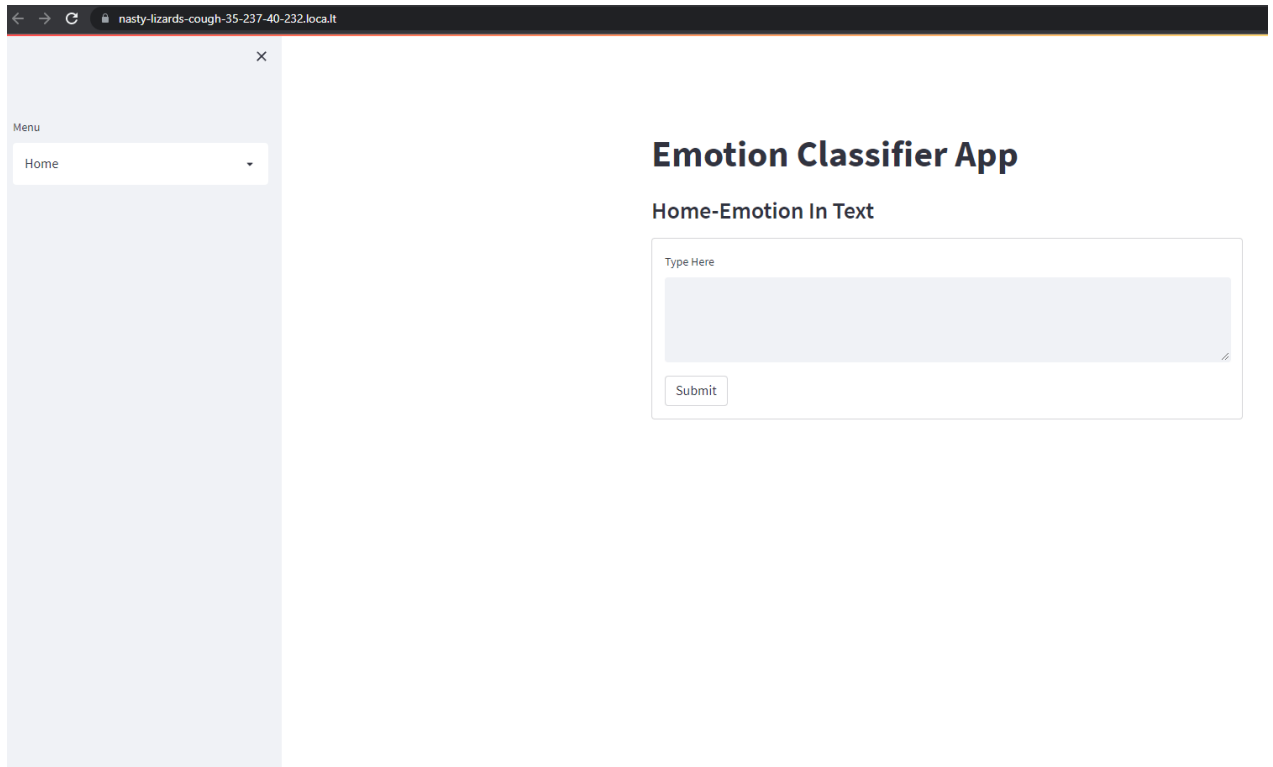
- Merah : *Validation BERT Uncased*
- Biru : *Training BERT Uncased*
- Hijau : *Validation IndoBERT*
- Kuning : *Training IndoBERT*
- Grafik : *Loss (kiri) & Akurasi (kanan)*

Pada proses penampilan akurasi antara *BERT Uncased* dan *IndoBERT*, terlihat perbedaan yang signifikan antara kedua model. Terlihat pada grafik *validation BERT Uncased* (merah), dari *epochs* pertama langsung diawali dengan akurasi diatas 80% dan terus naik hingga 90% pada *epochs* ke-10. Sedangkan pada *validation IndoBERT* (hijau), pada *epochs* pertama bahkan diawali dengan nilai akurasi dibawah 20% namun performanya terus meningkat terus 10% setiap kenaikan *epochnya* hingga pada *epochs* ke-10, nilai akurasi menyentuh 81%. Begitu pula pada nilai *loss* pada kedua model, pada *BERT*

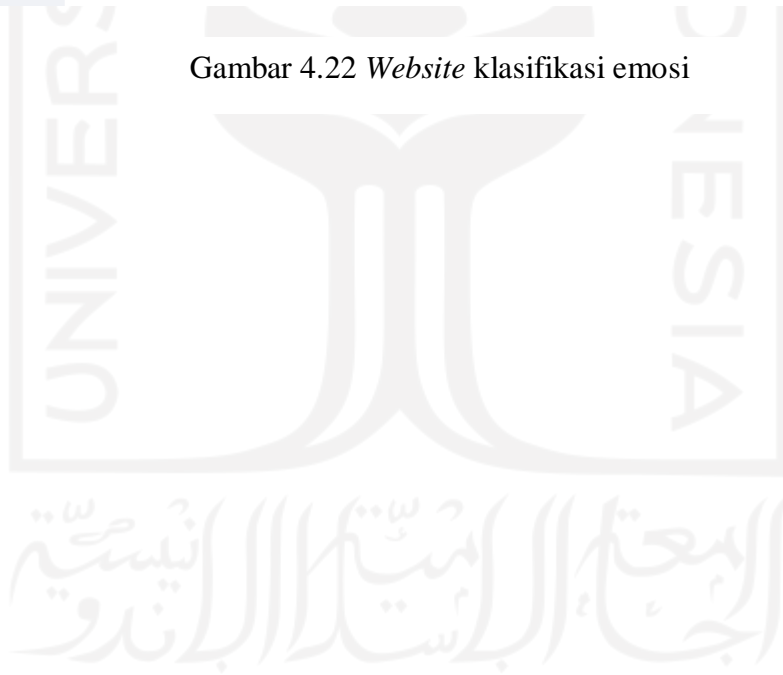
Uncased di epoch pertama menghasilkan nilai 1.6287. penurunan nilai *loss* terlihat bertahap dengan rata rata selisih penurunan pada tiap epochnya adalah 10%, hingga pada epoch ke 10 nilai *loss* pada *BERT Uncased* menghasilkan nilai sebesar 0.9763. Sedangkan pada nilai *loss* pada model *IndoBERT* mendapatkan nilai yang cukup berbeda dari model *BERT Uncased* yaitu sebesar 2.0741. Nilai ini terus turun pada setiap epochnya dengan rata-rata penurunan nilai sebesar 10%-15%, sampai pada epoch ke 10 menghasilkan nilai akhir sebesar 1.0002. Selisih nilai *loss* pada setiap model lumayan tipis, *BERT Uncased* menghasilkan nilai *loss* sebesar 0.9763, sedangkan *IndoBERT* menghasilkan nilai *loss* sebesar 1.0002

- b. Penelitian ini menggunakan model *IndoBERT* untuk dijadikan sebuah sistem. Alasan peneliti menggunakan model *IndoBERT* karena pada model tersebut berbasiskan bahasa Indonesia sedangkan *BERT Uncased* menggunakan bahasa Inggris. Walaupun tingkat akurasinya lebih besar pada model *BERT Uncased*, namun *IndoBERT* dipilih untuk menghindari adanya keambiguan dalam proses deteksi emosi pada bahasa Indonesia karena peneliti ingin menggunakan bahasa indonesia untuk dideteksi. Berikut adalah gambar sistem hasil dari model *IndoBERT*.

Peneliti membuat tampilan *website* menggunakan *streamlit*. Dengan *streamlit*, model mempunyai tampilan *website* agar pengguna dapat menggunakannya dengan mudah dan nyaman untuk dilihat. Dibawah ini adalah kode pemrograman untuk membuat tampilan *website* dengan menggunakan *library streamlit*. Pengimplementasiannya model yang sudah dibangun berbentuk sebuah tampilan *website* dari model *IndoBERT* dapat dilihat pada gambar 4.22 dan 4.23. Pengimplementasinya, model mampu untuk memprediksi suatu teks opini dengan tepat dan aktual.



Gambar 4.22 Website klasifikasi emosi



Emotion Classifier App

Home-Emotion In Text

Type Here

dari pagi lo dateng udh bikin emosi aja ajg gw lagi ulangan lo bilang gausah berlindung dibalik kata belajar jing gw diem main diatas sengaja biar gak kesel lu malah naik marah marah gajelas

Submit

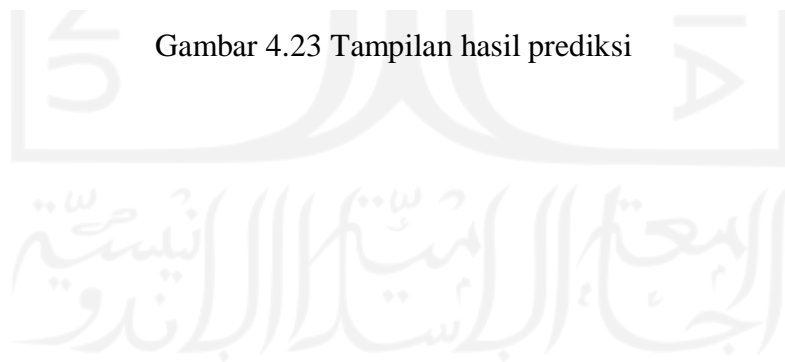
Original Text

dari pagi lo dateng udh bikin emosi aja ajg gw lagi
ulangan lo bilang gausah berlindung dibalik kata
belajar jing gw diem main diatas sengaja biar gak
kesel lu malah naik marah marah gajelas

Prediction

marah:0

Gambar 4.23 Tampilan hasil prediksi



BAB V PENUTUP

5.1 Kesimpulan

Penelitian ini dilakukan klasifikasi emosi pada teks dengan jumlah data sebanyak 2515 data yang masing-masing data memiliki sembilan label emosi (marah, sedih, takut, jijik, netral, senang, kaget, tertarik dan percaya). Dengan menggunakan *BERT uncased* dan *IndoBERT* yang mana hasilnya akan dibandingkan nilai akurasi. *BERT uncased* menghasilkan akurasi sebesar 90% sedangkan *IndoBERT* menghasilkan akurasi sebesar 81%. Perbandingan hasil yang dilakukan sedikit berbeda tetapi memiliki nilai akurasi yang cukup baik.

Berdasarkan hasil pengujian dari implementasi model *BERT* dalam melakukan klasifikasi emosi, maka diperoleh kesimpulan sebagai berikut:

- a. Klasifikasi emosi menggunakan *Bidirectional Encoder Representation from Transformer* (*BERT*) menghasilkan akurasi pada tiap model sebesar 90%, pada *BERT Uncased* dan *IndoBERT* sebesar 81%.
- b. Berdasarkan hasil pengujian dengan epoch 5, 10, dan 20, diperoleh hasil yang baik pada epoch 10. Sehingga epoch 10 digunakan untuk melakukan klasifikasi emosi.
- c. Jumlah data tiap label pada dataset menentukan keakuratan model saat dilatih dan diuji coba, dengan rata-rata 200 data tiap label emosi sudah cukup untuk melatih model.
- d. Saat dilakukan pengujian dalam sebuah kalimat opini, model *IndoBERT* dapat memprediksi emosi pada teks yang dimasukan, ini membuktikan bahwa model mampu menganalisis emosi pada bahasa indonesia.
- e. Berdasarkan hasil evaluasi model pada *IndoBERT*, model menghasilkan nilai akurasi sebesar 97,6%. Akurasi 97,6% ini dinilai bagus untuk mengidentifikasi emosi pada teks, hal ini dibuktikan pada pengujian dengan kalimat opini, model dapat mengidentifikasi dengan tepat.

5.2 Saran

Penelitian ini terdapat beberapa saran untuk pengembangan penelitian selanjutnya:

- a. Menambah jumlah dataset dan perhatikan keseimbangan jumlah data pada setiap label emosi agar nilai akurasi pada model menjadi lebih baik.
- b. Melakukan *split data* dengan percobaan perbandingan 0.1, 0.2, 0.3, dan 0.4 agar mendapatkan data latih dan uji yang lebih tepat.

- c. Memberikan tampilan yang lebih rapih pada sistem untuk mempermudah *user* menggunakan model yang sudah dibangun.
- d. Pada pembuatan web menggunakan *streamlit*, disarankan untuk meng-*install nvidia cuda driver* terlebih dahulu karena untuk penggunaannya harus memerlukan *gpu* agar prosesnya berjalan lancar.



DAFTAR PUSTAKA

- Amigo, J. M. (2021). Data mining, machine learning, deep learning, chemometrics: Definitions, common points and trends (Spoiler Alert: VALIDATE your models!). *Brazilian Journal of Analytical Chemistry*, 8(32), 22–38. <https://doi.org/10.30744/BRJAC.2179-3425.AR-38-2021>
- Chatterjee, A., Narahari, K. N., Joshi, M., & Agrawal, P. (2019). *SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text*. 39–48. <https://doi.org/10.18653/v1/s19-2005>
- Chiorrini, A., Diamantini, C., Mircoli, A., & Potena, D. (2021). Emotion and sentiment analysis of tweets using BERT. *CEUR Workshop Proceedings*, 2841.
- Chowdhary, K. R. (2020). Fundamentals of artificial intelligence. In *Fundamentals of Artificial Intelligence*. Springer India. <https://doi.org/10.1007/978-81-322-3972-7>
- Fera Fanesya, Randy Cahya Wihandika, I. (2019). Deteksi Emosi pada Twitter Menggunakan Metode Naive Bayes dan Kombinasi Fitur. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), 3.
- Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Sajjad, H., Nakov, P., Chen, D., & Winslett, M. (n.d.). *Compressing Large-Scale Transformer-Based Models: A Case Study on BERT*. <https://doi.org/10.1162/tacl>
- Huang, C., Trabelsi, A., & Zaïane, O. (2019). *ANA at SemEval-2019 Task 3: Contextual Emotion detection in Conversations through hierarchical LSTMs and BERT*. 49–53. <https://doi.org/10.18653/v1/s19-2006>
- Jacovi, A., Shalom, O. S., & Goldberg, Y. (2018). *Understanding Convolutional Neural Networks for Text Classification*. <http://arxiv.org/abs/1809.08037>
- Khan, M., Naeem, M. R., Al-Ammar, E. A., Ko, W., Vettikalladi, H., & Ahmad, I. (2022). Power Forecasting of Regional Wind Farms via Variational Auto-Encoder and Deep Hybrid Transfer Learning. *Electronics (Switzerland)*, 11(2). <https://doi.org/10.3390/electronics11020206>
- Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). *IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP*. <http://arxiv.org/abs/2011.00677>
- Luan, Y., & Lin, S. (n.d.). *Research on Text Classification Based on CNN and LSTM*.

- Luo, L., & Wang, Y. (2019). *EmotionX-HSU: Adopting Pre-trained BERT for Emotion Classification*. <http://arxiv.org/abs/1907.09669>
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning-Based Text Classification. In *ACM Computing Surveys* (Vol. 54, Issue 3). Association for Computing Machinery. <https://doi.org/10.1145/3439726>
- Mohsin, M. A., & Beltiukov, A. (2019). *Summarizing Emotions from Text Using Plutchik's Wheel of Emotions*. *166(Itids)*, 291–294. <https://doi.org/10.2991/itids-19.2019.52>
- Noviadrianti. (n.d.). *Stemming pada Preprocessing Twit Berbahasa Indonesia dengan Mengimplementasikan Algoritma Fonetik Soundex untuk Proses Klasifikasi Stemming in Indonesian Language Twit Preprocessing Implementing Phonetic Soundex Algorithm for Classification Process*.
- Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, *3*(1), 15–26. <https://doi.org/10.25139/ijair.v3i1.3827>
- Rohman, A. N., Utami, E., & Raharjo, S. (2019). Deteksi Kondisi Emosi pada Media Sosial Menggunakan Pendekatan Leksikon dan Natural Language Processing. *Eksplora Informatika*, *9*(1), 70–76. <https://doi.org/10.30864/eksplora.v9i1.277>
- Syarief, F. (2017). Pemanfaatan Media Sosial Dalam Proses Pembentukan Opini Publik (Analisa Wacana Twitter Sby). *Jurnal Komunikasi*, *3*(September), 2579–329.
- Torregrosa, J., Bello-Orgaz, G., Martínez-Cámara, E., Ser, J. del, & Camacho, D. (2022). A survey on extremism analysis using natural language processing: definitions, literature review, trends and challenges. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-021-03658-z>
- Vig, J., & Belinkov, Y. (2019). *Analyzing the Structure of Attention in a Transformer Language Model*. <http://arxiv.org/abs/1906.04284>
- W., Y. A. S., Palit, H. N., & Andjarwirawan, J. (2018). Aplikasi Pendeteksi Unsur Hinaan dalam Komentar di Media Sosial Berbahasa Indonesia. *Jurnal Infra Petra*.

LAMPIRAN

Lampiran A

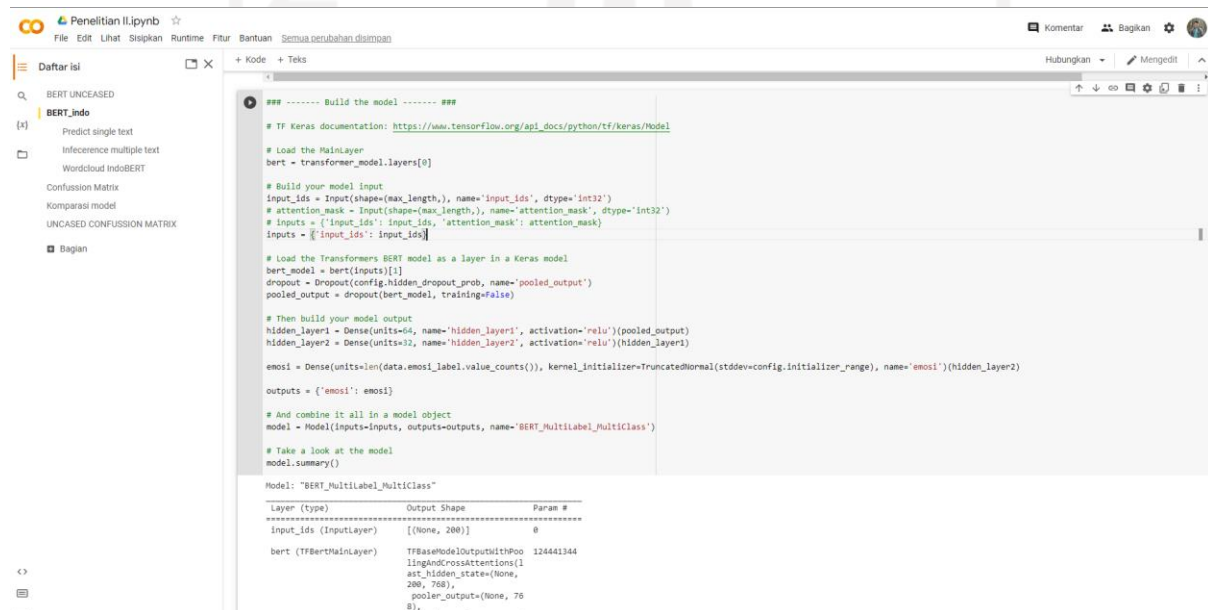
Penelitian Deep Learning

Penelitian ini mengarah pada klasifikasi emosi pada teks menggunakan *deep learning*. Mendeteksi emosi dalam dialog tekstual adalah masalah yang menantang karena tidak adanya ekspresi wajah dan modulasi suara (Chatterjee et al., 2019). Untuk mendeteksi emosi tersebut perlu dilakukan klasifikasi emosi. Klasifikasi sendiri merupakan proses pengelompokan beberapa data sesuai kriteria. Penelitian ini mengarah pada klasifikasi emosi pada teks opini yang digunakan untuk mengetahui jenis emosi dari opini yang telah diberikan. Opini tersebut diambil dari media sosial Twitter. Metode yang digunakan dalam penelitian ini adalah BERT (*Bidirectional Encoder Representations from Transformers*). Penelitian ini menggunakan label emosi sebanyak sembilan emosi, diantaranya marah, bahagia, sedih, tertarik, netral, percaya, jijik, kaget, dan takut.

Dalam pengembangan klasifikasi emosi pada teks, tidak semudah apa yang sudah direncanakan sebelumnya. Banyak sekali perubahan-perubahan yang dilakukan untuk menyesuaikan keadaan sebenarnya. Perubahan terjadi untuk membuat sistem lebih baik lagi.

Lampiran B

Kode pemrograman



```
### ----- Build the model ----- ###  
  
# TF Keras documentation: https://www.tensorflow.org/api\_docs/python/tf/keras/Model  
  
# Load the MainLayer  
bert = transformer_model.layers[0]  
  
# Build your model input  
input_ids = Input(shape=(max_length,), name='input_ids', dtype='int32')  
attention_mask = Input(shape=(max_length,), name='attention_mask', dtype='int32')  
inputs = {'input_ids': input_ids, 'attention_mask': attention_mask}  
inputs = {'input_ids': input_ids}  
  
# Load the Transformers BERT model as a layer in a Keras model  
bert_model = bert(inputs[0])  
dropout = Dropout(config.hidden_dropout_prob, name='pooled_output')  
pooled_output = dropout(bert_model, training=False)  
  
# Then build your model output  
hidden_layer1 = Dense(units=64, name='hidden_layer1', activation='relu')(pooled_output)  
hidden_layer2 = Dense(units=32, name='hidden_layer2', activation='relu')(hidden_layer1)  
  
emosi = Dense(units=len(data.emosi_label.value_counts()), kernel_initializer=TruncatedNormal(stddev=config.initializer_range), name='emosi')(hidden_layer2)  
  
outputs = {'emosi': emosi}  
  
# And combine it all in a model object  
model = Model(inputs=inputs, outputs=outputs, name='BERT_MultiLabel_MultiClass')  
  
# Take a look at the model  
model.summary()  
  
Model: "BERT_MultiLabel_MultiClass"  
-----  
Layer (type) Output Shape Param #  
-----  
input_ids (InputLayer) [(None, 200)] 0  
  
bert (TFBertMainLayer) TFBaseModelOutputWithPool- 124441344  
lingAndCrossAttentions(1  
ast_hidden_state=(None,  
200, 768),  
pooler_output=(None, 76  
8),
```