

**SKRIPSI**  
**Pengenalan Emosi dari Ulasan Pelanggan**  
**E-Commerce Menggunakan Deep Learning**  
**Berbasis Transformer**

Diajukan untuk memenuhi salah satu syarat memperoleh  
gelar Sarjana Teknik Informatika



Disusun Oleh:

Nama : Ahmad Sabil Deva Pratama  
NIM : A11.2020.13098  
Program Studi : Teknik Informatika

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS DIAN NUSWANTORO**  
**SEMARANG**  
**2024**

## ABSTRAK

Penelitian ini mengeksplorasi penerapan arsitektur deep learning berbasis Transformer untuk mengidentifikasi emosi dari ulasan pelanggan dalam e-commerce berbahasa Indonesia. Menggunakan dataset yang terdiri dari 5.400 ulasan pelanggan, model ini dirancang untuk mengklasifikasikan lima kategori emosi: Happy, Sadness, Anger, Love, dan Fear. Hasil analisis menunjukkan kecenderungan pelanggan untuk berbagi pengalaman positif dengan dominasi emosi Happy dalam ulasan. Model Transformer berhasil mencapai akurasi klasifikasi keseluruhan sebesar 77.2%, dengan efisiensi waktu pelatihan yang optimal sekitar 90 detik. Evaluasi performa menggunakan confusion matrix dan metrik presisi, recall, dan F1-score memberikan wawasan tentang keefektifan model dalam membedakan emosi kompleks. Temuan ini merefleksikan potensi pemanfaatan teknologi deep learning dalam meningkatkan pemahaman terhadap perilaku pelanggan dan mendukung pengembangan strategi bisnis yang responsif.

**Kata Kunci:** Deep Learning, Transformer, Analisis Sentimen, Ulasan Pelanggan, E-commerce, Natural Language Processing (NLP), Klasifikasi Emosi.

## PERSETUJUAN LAPORAN TUGAS AKHIR

Nama Pelaksana : Ahmad Sabil Deva Pratama  
NIM : A11.2020.13098  
Program Studi : Teknik Informatika  
Fakultas : Ilmu Komputer  
Judul Tugas Akhir : **“Pengenal Emosi Dari Ulasan Pelanggan  
E-Commerce Menggunakan Deep Learning Berbasis Transformer”**

Tugas Akhir ini telah diperiksa dan disetujui pada,  
Semarang 20 Februari 2024,

Menyetujui  
Pembimbing

Mengetahui  
Dekan Fakultas Ilmu Komputer



Dr. Ir. NOVA RIJATI, S.Si., M.Kom, IPU, ASEAN Eng  
NPP: 0686.11.1996.093

Dr. Drs. ABDUL SYUKUR, MM  
NPP: 0686.11.1992.017

## **PENGESAHAN DEWAN PENGUJI**

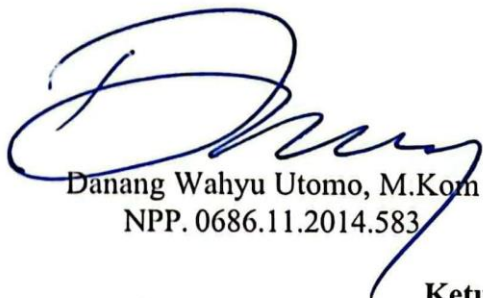
Nama Pelaksana : Ahmad Sabil Deva Pratama  
NIM : A11.2020.13098  
Program Studi : Teknik Informatika  
Fakultas : Ilmu Komputer  
Judul Tugas Akhir : **“Pengenalan Emosi Dari Ulasan Pelanggan E-Commerce Menggunakan Deep Learning Berbasis Transformer”**

Tugas Akhir ini telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang tugas akhir tanggal 20 Februari 2024. Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugerahan gelar Sarjana Komputer (S.Kom.)

Semarang, 20 Februari 2024


Dewan Penguji

**Anggota 1**




Danang Wahyu Utomo, M.Kom  
NPP. 0686.11.2014.583

**Anggota 2**



Nurul Anisa Sri Winarsih, S.Kom, M.Cs  
NPP. 0686.11.2018.747

**Ketua Penguji**



Feri Agustina, M.Kom  
NPP. 0686.11.1997.141

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan di bawah ini,

Nama : Ahmad Sabil Deva Pratama

NIM : A11.2020.13098

Menyatakan bahwa karya ilmiah saya yang berjudul :

### **Pengenalan Emosi Dari Ulasan Pelanggan E-Commerce Menggunakan Deep Learning Berbasis Transformer**

Merupakan karya asli saya, kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan darimana sumbernya dan perangkat pendukungnya. Apabila dikemudian hari, yang disertai bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut. Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Semarang, 20 Februari 2024

Yang menyatakan



(Ahmad Sabil Deva Pratama)

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat rahmat, karunia dan anugrah-Nya, penulis mampu menyelesaikan laporan akhir yang berjudul **"Pengenal Emosi Dari Ulasan Pelanggan E-Commerce Menggunakan Deep Learning Berbasis Transformer"**.

Tugas akhir ini merupakan salah satu mata kuliah wajib yang harus ditempuh oleh seluruh mahasiswa dalam rangka menyelesaikan pendidikan kesarjanaan Strata-1 (S1) di Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang.

Dalam proses pelaksanaan penyusunan tugas akhir ini, saya mendapat bantuan, dukungan, dan doa dari banyak pihak, baik secara langsung maupun tidak langsung. Untuk itu, ucapan terima kasih ini saya berikan kepada:

1. Allah SWT, atas rahmat, petunjuk dan ridho yang telah diberikan kepada saya.
2. Bapak Prof. Dr. Ir Edi Noersasongko M.Kom selaku Rektor Universitas Dian Nuswantoro
3. Bapak Dr. Drs. Abdul Syukur, MM selaku Dekan Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
4. Bapak Sri Winarno, Ph.D selaku Ketua Program Studi Teknik Informatika Universitas Dian Nuswantoro.
5. Ibu Dr. Nova Rijati, S.Si., M.Kom selaku dosen pembimbing tugas akhir yang telah memberikan pengarahan, bimbingan, dan saran.
6. Ayah, Ibu, dan keluarga besar yang tercinta yang selalu memberikan doa dan dukungan moral maupun materi.
7. Seluruh rekan dan sahabat Teknik Informatika 2020 atas segala bantuannya
8. Dan seluruh pihak yang telah memberikan dukungan moril, materil, dan doa, yang tidak dapat saya sebutkan satu persatu.

Penyusun menyadari bahwa dalam penulisan ini masih banyak

kekurangan dan jauh dari kata sempurna. Oleh karena itu segala saran dan kritik yang bersifat membangun sangat diharapkan dengan senang hati untuk penyempurnaan laporan tugas akhir ini.

Akhir kata, penyusun berharap semoga laporan ini bermanfaat bagi perkembangan penguasaan ilmu Deep Learning Berbasis Transformer di Program Studi Teknik Informatika Universitas Dian Nuswantoro.

Semarang, 20 Februari 2024

A handwritten signature in black ink, consisting of stylized, overlapping loops and lines, representing the name Ahmad Sabil Deva Pratama.

Ahmad Sabil Deva Pratama

## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>PERSETUJUAN LAPORAN TUGAS AKHIR.....</b>	<b>ii</b>
<b>PENGESAHAN DEWAN PENGUJI .....</b>	<b>iii</b>
<b>PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian .....	5
1.5.1 Bagi Penulis .....	5
1.5.2 Bagi Dunia Akademik .....	5
1.5.3 Bagi Industri E-commerce .....	5
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1 Penelitian Terkait.....	6
2.2 Dasar Teori .....	13
2.2.1 Emosi .....	13
2.2.2 Ulasan Pelanggan dalam E-commerce.....	13
2.2.3 Deep Learning .....	14
2.2.4 Word Embedding.....	14



2.2.5 Mekanisme Attention .....	15
2.2.6 Transformer .....	16
2.3 Kerangka Pemikiran.....	19
<b>BAB III METODE PENELITIAN .....</b>	<b>20</b>
3.1 Tahapan Penelitian .....	20
3.2 Tools .....	20
3.2.1 Google Colab .....	21
3.2.2 Python .....	21
3.2.3 Pandas .....	21
3.2.4 TensorFlow dan Keras.....	21
3.2.5 Matplotlib.....	22
3.2.6 Scikit-learn .....	22
3.3 Pengumpulan Data .....	22
3.4 Pra-Pemrosesan Data .....	24
3.4.1 Pemilihan dan Pembersihan Data Awal .....	24
3.4.2 Normalisasi dan Pembersihan Teks Lanjutan .....	25
3.4.3 Tokenisasi, Vektorisasi, dan Pembuatan Sekuens .....	25
3.4.4 Encoding Label dan Pembagian Dataset Pemodelan .....	26
3.5 Pembangunan Model.....	27
3.5.1 Arsitektur Transformer Block .....	27
3.5.1.1 Multi-Head Attention (MHA) .....	27
3.5.1.2 Feed-Forward Network (FFN) .....	28
3.5.1.3 Normalisasi Lapisan dan Dropout.....	29
3.5.1.4 Implementasi Transformer Block.....	29
3.5.1.5 Contoh Perhitungan MHA .....	29

3.5.2 Token dan Positional Embedding.....	33
3.5.2.1 Token Embedding .....	33
3.5.2.2 Positional Embedding .....	33
3.5.2.3 Implementasi Token dan Positional Embedding.....	34
3.5.3 Penggabungan Model.....	34
3.5.3.1 Global Average Pooling .....	34
3.5.3.2 Dropout dan Regularisasi .....	35
3.5.3.3 Dense Layer .....	35
3.5.3.4 Lapisan Output dan Softmax.....	35
3.5.3.5 Contoh Perhitungan Model .....	36
3.6 Pelatihan Model .....	38
3.6.1 Penetapan Bobot Kelas .....	38
3.6.2 Penerapan Early Stopping dalam Pelatihan Model.....	39
3.6.3 Optimizer dan Kompilasi Model.....	40
3.6.3.1 Pemilihan Optimizer .....	40
3.6.3.2 Kompilasi Model.....	41
3.6.4 Pelatihan Model .....	41
3.6.4.1 Setting Pelatihan .....	41
3.6.4.2 Proses Pelatihan .....	42
3.6.4.3 Monitoring dan Evaluasi .....	43
3.7 Evaluasi Model.....	43
3.7.1 Visualisasi Kinerja Selama Pelatihan.....	43
3.7.1.1 Visualisasi Akurasi .....	44
3.7.1.2 Visualisasi Loss .....	45
3.7.1.3 Implementasi dan Analisis .....	45

3.7.2 Evaluasi Model Dengan Data Test .....	46
3.7.2.1 Prediksi Data Test.....	46
3.7.2.2 Hasil Prediksi .....	47
3.7.3 Confusion Matrix .....	47
3.7.3.1 True Positives (TP).....	48
3.7.3.2 True Negatives (TN) .....	48
3.7.3.3 False Positives (FP).....	48
3.7.3.4 False Negatives (FN).....	48
3.7.3.5 Implementasi Confusion Matrix .....	49
3.7.4 Evaluasi Hasil Model Keseluruhan.....	49
3.7.4.1 Akurasi .....	49
3.7.4.2 Presisi .....	50
3.7.4.3 Recall.....	50
3.7.4.4 Skor F1 .....	51
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN .....</b>	<b>52</b>
4.1 Alur Kerja Model .....	52
4.2 Deskripsi Dataset .....	54
4.2.1 Karakteristik Umum Dataset.....	54
4.2.2 Distribusi Emosi dalam Dataset.....	54
4.2.2.1 Distribusi Frekuensi Emosi .....	55
4.2.2.2 Visualisasi Distribusi Emosi .....	55
4.2.2.3 Implikasi Distribusi Emosi Terhadap Model .....	57
4.2.3 Frekuensi Kata dalam Dataset.....	58
4.2.4 Contoh Ulasan.....	60
4.2.4.1 Tabel Ulasan Contoh .....	60

4.3 Pra-Pemrosesan dan Persiapan Data .....	61
4.3.1 Pembagian Dataset untuk Pengujian dan Pelatihan .....	62
4.3.1.1 Proses Pembagian Dataset .....	62
4.3.2 Pra-Pemrosesan Data .....	63
4.3.2.1 Langkah-Langkah Pra-Pemrosesan.....	63
4.3.2.2 Tabel Contoh Pra-Pemrosesan .....	65
4.3.2.3 Implikasi Pra-Pemrosesan.....	66
4.3.3 Persiapan Data.....	67
4.3.3.1 Konversi dan Tokenisasi Teks .....	68
4.3.3.2 Tokenisasi Teks .....	68
4.3.3.3 Padding Sekuens .....	69
4.3.3.4 Enkoding Label .....	69
4.3.3.5 Pembagian Data .....	70
4.4 Pembangunan dan Pelatihan Model Transformer .....	71
4.4.1 Arsitektur Model Transformer .....	71
4.4.1.1 Implementasi Transformer Block.....	73
4.4.1.2 Token dan Position Embedding.....	74
4.4.1.3 Contoh Hasil Proses Embedding.....	75
4.4.2 Pembangunan Model Transformer.....	76
4.4.2.1 Pembangunan Model.....	77
4.4.2.2 Proses Pembangunan.....	78
4.4.3 Strategi Pelatihan .....	79
4.4.3.1 Penetapan Class Weights.....	79
4.4.3.2 Callback Early Stopping .....	80
4.4.3.3 Optimizer dan Laju Pembelajaran.....	80

4.4.3.4 Kompilasi dan Pelatihan Model .....	80
4.4.4 Performa Model .....	81
4.4.4.1 Analisis Akurasi .....	81
4.4.4.2 Analisis Loss .....	82
4.5 Evaluasi Model Dengan Data Pengujian.....	83
4.5.1 Confusion Matrix .....	83
4.5.2 Evaluasi Hasil Model .....	87
4.5.2.1 Evaluasi Kelas Anger .....	90
4.5.2.2 Evaluasi Kelas Fear.....	90
4.5.2.3 Evaluasi Kelas Happy .....	91
4.5.2.4 Evaluasi Kelas Love.....	91
4.5.2.5 Evaluasi Kelas Sadness .....	92
4.5.2.6 Evaluasi Keseluruhan Model .....	92
4.5.3 Analisis Kesalahan Prediksi .....	93
<b>BAB V KESIMPULAN DAN PENELITIAN SELANJUTNYA.....</b>	<b>96</b>
5.1 Kesimpulan .....	96
5.2 Penelitian Selanjutnya.....	97
<b>DAFTAR PUSTAKA.....</b>	<b>98</b>

## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Transformer .....	17
Gambar 2. 2 Kerangka Pemikiran .....	19
Gambar 3. 1 Diagram Tahapan Penelitian .....	20
Gambar 3. 2 PRDECT-ID Dataset .....	23
Gambar 4. 1 Alur Kerja Model .....	52
Gambar 4. 2 Distribusi Emosi dalam Dataset .....	55
Gambar 4. 3 Kode Untuk Membagi Dataset.....	62
Gambar 4. 4 Kode Untuk Melakukan Pemrosesan Data .....	63
Gambar 4. 5 Kode Untuk Persiapan Data .....	67
Gambar 4. 6 Arsitektur Model Transformer.....	72
Gambar 4. 7 Kode Untuk Membangun Model Transformer.....	77
Gambar 4. 8 Strategi Pelatihan Pada Model Transformer.....	79
Gambar 4. 9 Grafik Akurasi Model Transformer.....	81
Gambar 4. 10 Grafik Loss Model Transformer.....	82
Gambar 4. 11 Confusion Matrix Model Transformer .....	84

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	9
Tabel 4. 1 Frekuensi Kata dalam Dataset.....	59
Tabel 4. 2 Contoh Ulasan .....	60
Tabel 4. 3 Contoh Pra-Pemrosesan Data.....	65
Tabel 4. 4 Jumlah Data Untuk Pelatihan dan Pengujian .....	71
Tabel 4. 5 Contoh Hasil Proses Embedding.....	75
Tabel 4. 6 Evaluasi Hasil Model Transformer .....	90
Tabel 4. 7 Sampel Contoh Kesalahan Prediksi .....	93

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era pertumbuhan e-commerce yang pesat, pelanggan semakin cenderung mengandalkan platform pemasaran online untuk melakukan pembelian produk. Setelah melakukan pembelian, mereka kemudian berbagi ulasan yang mencerminkan pengalaman pribadi, perasaan, dan respons emosional mereka terhadap produk tersebut (Alsubari et al., 2022). Kebebasan yang diberikan oleh internet telah menghapuskan batasan geografis fisik, memungkinkan pelanggan untuk menjelajahi berbagai pilihan dari berbagai merek dan penjual di seluruh dunia. Perubahan ini juga telah memengaruhi pola perilaku konsumen dalam proses pembelian, memberikan mereka akses yang lebih luas terhadap informasi yang komprehensif tentang produk atau layanan yang sedang mereka pertimbangkan, yang tersedia dalam bentuk ulasan yang dibagikan oleh sesama pelanggan (Grigoreva et al., 2021).

Emosi adalah unsur yang tak terpisahkan dalam kehidupan manusia dan memainkan peran yang signifikan dalam pengambilan keputusan individu. Dalam konteks bisnis dan konsumen, ulasan pelanggan online telah diakui sebagai faktor yang memiliki pengaruh besar terhadap keputusan pembelian (Guo et al., 2020). Dalam era e-commerce yang tumbuh pesat, ulasan pelanggan telah menjadi komponen penting dalam membentuk preferensi pembelian, dan konsumen sendiri percaya bahwa ulasan yang mencerminkan emosi yang kuat akan berdampak lebih besar pada orang lain serta memengaruhi keputusan pembelian mereka (Rocklage & Fazio, 2020). Oleh karena itu, pemahaman yang mendalam tentang bagaimana emosi tercermin dalam ulasan pelanggan dan bagaimana emosi tersebut memengaruhi perilaku pembelian adalah hal yang sangat penting dalam memahami dinamika pasar e-commerce yang kompetitif saat ini.

Natural Language Processing (NLP) muncul sebagai teknologi yang mampu mengekstraksi makna dan emosi dari teks. Melalui penggunaan algoritma yang canggih dan model pelatihan yang mendalam, NLP memiliki kapabilitas untuk mengenali pola bahasa, memahami konteks, dan bahkan menafsirkan emosi yang



terkandung dalam teks (Maulud et al., 2021). Terutama dalam konteks ulasan produk, NLP membuka peluang yang sangat menarik untuk memahami pandangan pelanggan dalam skala besar. Ini memungkinkan penyedia layanan untuk memberikan analisis sentimen yang akurat dan wawasan mendalam yang sebelumnya sulit dicapai dalam jumlah besar. Penggunaan NLP dalam menganalisis ulasan pelanggan membuka pintu untuk pemahaman yang lebih baik tentang preferensi dan opini pelanggan, yang dapat menginformasikan pengambilan keputusan bisnis yang lebih cerdas dan efektif.

Seiring dengan pesatnya perkembangan dalam bidang Natural Language Processing (NLP), model berbasis Transformer telah muncul sebagai salah satu pendekatan yang paling canggih dan berpengaruh dalam NLP. Vaswani et al (2017) dalam makalah "Attention Is All You Need" memperkenalkan model berbasis Transformer yang telah merevolusi bidang NLP dengan pendekatan inovatifnya terhadap pemrosesan sekuens. Model berbasis Transformer memperkenalkan paradigma baru dalam pemrosesan bahasa yang telah mengubah cara kita memahami dan mengolah teks. Salah satu fitur paling menonjol dari Transformer adalah struktur "attention," yang memungkinkan model untuk fokus pada bagian-bagian tertentu dari teks sumber saat menganalisis atau menghasilkan teks target (Sun et al., 2021). Kemampuan struktur attention ini memiliki dampak yang sangat besar dalam pemahaman kontekstual, yang sangat penting dalam menangkap nuansa emosi dalam teks.

Dalam beberapa tahun terakhir, teknologi NLP telah mengalami kemajuan pesat dalam mengenali emosi dari teks. Ulasan produk di e-commerce oleh konsumen telah menjadi sumber informasi yang kaya untuk mendapatkan wawasan tentang persepsi dan emosi pelanggan terhadap produk atau layanan. Analisis sentimen, khususnya, telah menjadi metode yang populer untuk mengekstrak dan memahami emosi dari ulasan tersebut. Peran analisis sentimen sangat signifikan dalam interpretasi ulasan konsumen terkait produk, melalui penerapan machine learning, serta perbandingannya dengan teknik machine learning lainnya (Purohit, 2021). Terlebih lagi, pendekatan yang berfokus pada teks dalam memprediksi emosi telah menunjukkan potensi yang signifikan. Sebagai contoh, Ab Nasir et al. (2020)

telah mengembangkan sistem prediksi emosi yang berlandaskan teks menggunakan machine learning, yang berhasil mengenali dan mengkategorikan emosi dari teks. Lebih lanjut, Bharti et al. (2022) menekankan penggunaan deep learning dalam pengenalan emosi berbasis teks, yang menunjukkan hasil yang lebih baik dalam menangkap nuansa emosi yang kompleks dalam teks dibandingkan dengan metode-metode tradisional.

Dalam beberapa penelitian terkini, model berbasis Transformer, seperti BERT, telah menunjukkan keunggulan dalam tugas-tugas NLP, termasuk dalam menganalisis ulasan pelanggan. Sebagai contoh, Durairaj & Chinnalagu (2021) mengusulkan penggunaan BERT yang telah disesuaikan untuk memprediksi sentimen pelanggan dengan menggunakan ulasan pelanggan dari platform Twitter, IMDB Movie Reviews, Yelp, dan Amazon. Hasil dari eksperimen yang mereka jalankan menunjukkan bahwa model BERT yang mereka ajukan unggul dalam kinerjanya dengan skor akurasi sebesar 90%, recall 91%, presisi 90%, dan F1 90%, dibandingkan dengan model-model lain seperti LSVM, fastText, dan BiLSTM yang skor evaluasinya lebih rendah. Selain itu, penelitian oleh Davoodi & Mezei (2022) menunjukkan bahwa penggunaan BERT dan RoBERTa dalam sentiment analisis pada customer review di e-commerce menghasilkan skor akurasi sebesar 90% untuk BERT dan skor akurasi sebesar 97,6% untuk RoBERTa. Skor tersebut melampaui skor akurasi dari Naive Bayes dan SVM yang menggunakan pembobotan bag of words (BOW).

Penelitian ini akan menggunakan arsitektur Transformer dalam upaya untuk memperdalam pemahaman tentang bagaimana emosi tercermin dalam ulasan pelanggan pada perilaku pembelian dalam konteks e-commerce yang sangat kompetitif. Diharapkan bahwa hasil penelitian ini akan memberikan wawasan berharga bagi pengambilan keputusan bisnis yang lebih cerdas dan efektif, membantu perusahaan dalam memahami preferensi dan opini pelanggan dengan lebih baik. Oleh karena itu, penelitian ini akan memberikan kontribusi berharga untuk pemahaman dan perkembangan lebih lanjut dalam domain ini, memungkinkan perusahaan untuk menghadapi persaingan sengit dalam dunia e-commerce secara lebih baik.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan, masalah-masalah yang dapat dirumuskan adalah:

1. Bagaimanakah distribusi dan karakteristik emosi yang terkandung dalam ulasan pelanggan?
2. Bagaimana arsitektur deep learning berbasis Transformer dapat diterapkan untuk mengenali emosi dari ulasan pelanggan?
3. Sejauh mana performa dan keakuratan model deep learning berbasis Transformer dalam mengklasifikasikan emosi dari ulasan pelanggan?

## **1.3 Batasan Masalah**

Untuk memberikan arah yang lebih jelas pada penelitian ini dan menjadikan hasilnya lebih relevan, peneliti menerapkan beberapa pembatasan berikut:

1. Penelitian akan dilakukan hanya pada ulasan produk dalam Bahasa Indonesia yang diperoleh dari platform e-commerce Tokopedia.
2. Emosi yang akan dikenali terbatas pada lima kategori: anger, fear, happy, love, dan sadness.
3. Model yang akan digunakan adalah arsitektur Transformer dalam bentuk dasarnya, yang sering disebut sebagai Vanilla Transformer.

## **1.4 Tujuan Penelitian**

Berdasarkan rumusan masalah di atas, penelitian ini diharapkan dapat:

1. Menganalisis distribusi serta karakteristik emosi yang ada dalam ulasan pelanggan.
2. Merancang dan melatih model deep learning berbasis Transformer untuk tugas pengenalan emosi dalam ulasan pelanggan.
3. Mengevaluasi performa dan keakuratan model deep learning berbasis Transformer dalam mengklasifikasikan emosi dari ulasan pelanggan.

## **1.5 Manfaat Penelitian**

### **1.5.1 Bagi Penulis**

1. Memperoleh pemahaman mendalam mengenai metode dan teknik Natural Language Processing khususnya pada deep learning yang berbasis Transformer dan aplikasinya dalam analisis emosi.
2. Sebagai syarat kelulusan dan pencapaian akademik yang mendukung perkembangan karier dan keberlanjutan pendidikan penulis di masa depan.

### **1.5.2 Bagi Dunia Akademik**

1. Memberikan wawasan dan informasi baru bagi dunia akademik, khususnya dalam bidang Natural Language Processing dan analisis sentimen, tentang kemampuan model berbasis Transformer dalam mengenali emosi dari teks berbahasa Indonesia.
2. Menjadi bahan referensi dan literatur untuk penelitian-penelitian selanjutnya yang memiliki fokus serupa atau relevan.

### **1.5.3 Bagi Industri E-commerce**

1. Membantu pelaku industri e-commerce untuk mendapatkan pemahaman yang lebih mendalam mengenai emosi pelanggan, sehingga dapat meningkatkan strategi pemasaran dan pengambilan keputusan bisnis yang lebih tepat.
2. Menyediakan model yang dapat diintegrasikan ke dalam sistem rekomendasi, sehingga platform e-commerce dapat menyajikan produk yang lebih sesuai dengan emosi dan kebutuhan pelanggan.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penelitian Terkait**

Dalam era digital saat ini, ulasan pelanggan e-commerce menjadi kunci dalam pengambilan keputusan bisnis dan konsumen. Analisis sentimen, sebuah segmen penting dari Natural Language Processing (NLP), memungkinkan pemahaman mendalam tentang emosi yang terkandung dalam teks ini. Dengan berkembangnya model-model Deep Learning, khususnya arsitektur berbasis Transformer, telah terjadi pergeseran paradigma dalam cara kita mengolah dan memahami data teks besar. Sejak diperkenalkannya Transformer oleh Vaswani et al. (2017), metode ini telah merevolusi pemahaman kita tentang data teks, dengan kemampuan superior dalam mengolah konteks dan nuansa bahasa.

Dalam penelitiannya yang berjudul "Text-Based Emotion Recognition Using Deep Learning Approach", Bharti et al (2022), memberikan wawasan penting dalam bidang pengenalan emosi dari teks menggunakan pendekatan Deep Learning. Penelitian ini mengatasi keterbatasan pendekatan berbasis kata kunci dan leksikon dalam analisis sentimen dengan mengusulkan model hibrida yang menggabungkan teknik Machine Learning dan Deep Learning. Model ini menggunakan Convolutional Neural Network (CNN) dan Bidirectional Gated Recurrent Unit (Bi-GRU) sebagai teknik Deep Learning, serta Support Vector Machine (SVM) sebagai pendekatan Machine Learning. Penelitian ini mengevaluasi performa model menggunakan tiga jenis dataset yang berbeda: kalimat, tweet, dan dialog, mencapai akurasi sebesar 80.11%. Hasil ini menunjukkan keunggulan model hibrida dalam mengidentifikasi emosi dari teks. Penelitian ini memberikan kontribusi signifikan dalam memahami bagaimana model Deep Learning, khususnya CNN dan Bi-GRU, dapat diterapkan secara efektif untuk tugas pengenalan emosi, yang merupakan aspek kunci dalam analisis sentimen e-commerce.

Salah satu model berbasis transformer yang terkenal adalah BERT. BERT, yang merupakan singkatan dari Bidirectional Encoder Representations from Transformers, adalah model yang menggunakan mekanisme Transformer untuk

memproses teks secara dua arah. Penelitian yang berjudul "Transformer based Contextual Model for Sentiment Analysis of Customer Reviews: A Fine-tuned BERT" oleh Durairaj & Chinnalagu (2021), mengeksplorasi penggunaan model BERT yang telah disesuaikan untuk analisis sentimen ulasan pelanggan. Penelitian ini mengatasi tantangan dalam analisis sentimen berbasis konteks dari data ulasan pelanggan yang bersifat tidak terstruktur dan berjumlah besar, yang sering ditemukan di berbagai platform media sosial. Peneliti menggunakan model BERT yang telah disesuaikan, yang merupakan model bahasa canggih berbasis Transformer, untuk memprediksi sentimen pelanggan dari ulasan di Twitter, IMDB Movie Reviews, Yelp, dan Amazon. Selain itu, mereka membandingkan hasil model BERT yang disesuaikan ini dengan model Linear Support Vector Machine (LSVM), fastText, BiLSTM, dan model hibrida fastText-BiLSTM. Hasilnya, model BERT yang mereka gunakan dalam penelitian memiliki kinerja yang unggul dengan skor akurasi sebesar 90%, recall 91%, presisi 90%, dan F1 90%, dibandingkan dengan model-model lain seperti LSVM, fastText, dan BiLSTM. Penelitian ini menunjukkan pentingnya model berbasis Transformer dalam analisis sentimen, terutama dalam konteks ulasan pelanggan, dapat digunakan untuk meningkatkan akurasi dan efisiensi dalam analisis sentimen berbasis teks.

Dalam penelitian "Text Sentiment Analysis Based on Transformer and Augmentation" oleh Gong et al (2022), tantangan analisis sentimen teks di media sosial diatasi dengan model berbasis Transformer yang menggabungkan distilasi pengetahuan dan augmentasi teks. Tujuannya adalah mengurangi jumlah parameter model untuk memangkas biaya komputasi dan waktu pelatihan, serta meningkatkan kinerja analisis sentimen pada sampel data terbatas. Eksperimen dilakukan menggunakan dua dataset: AG News Corpus dan Stanford Sentiment Treebank (SST). Penelitian ini membandingkan beberapa model termasuk BERT, ALBERT, dan MobileBert. Hasil eksperimen menunjukkan bahwa model BERT unggul dalam akurasi, yaitu sebesar 93,38% akurasi pada dataset AG News dan 92,94% akurasi pada dataset SST. Hasil ini menunjukkan bahwa model BERT yang merupakan model berbasis Transformer efektif dalam meningkatkan kinerja analisis sentimen.

Selanjutnya penelitian "A Comparative Study of Machine Learning Models for Sentiment Analysis: Customer Reviews of E-Commerce Platforms" oleh Davoodi & Mezei (2022), berfokus pada perbandingan berbagai model pembelajaran mesin, baik tradisional maupun berbasis deep learning, untuk klasifikasi sentimen ulasan pelanggan di platform e-commerce. Penelitian ini menggunakan data dari salah satu situs ulasan pelanggan terbesar, Trustpilot, yang mencakup ulasan dari lima platform e-commerce yang berbeda. Untuk model berbasis transformer, tokenizers khusus digunakan, dan BERT serta RoBERTa dioptimalkan dengan lapisan dropout dan lapisan padat untuk tugas klasifikasi. Performa model dievaluasi menggunakan metrik akurasi dan F1, dengan mempertimbangkan komponen confusion matrix seperti true positives, true negatives, false positives, dan false negatives. Hasil penelitian menunjukkan bahwa model berbasis transformer seperti BERT dan RoBERTa unggul dibandingkan model pembelajaran mesin tradisional. RoBERTa khususnya menunjukkan performa yang sangat baik dengan akurasi di atas 98% pada set uji. Meskipun model berbasis transformer membutuhkan waktu pelatihan yang lebih lama, peningkatan performa yang mereka tawarkan menjadi kompensasi yang berharga

Sementara itu penelitian "Sentiment Analysis for E-commerce Product Reviews by Deep Learning Model of Bert-BiGRU-Softmax" oleh Liu et al (2020), mengembangkan model pembelajaran mendalam untuk mengatasi masalah disambiguasi kata sentimen dan polaritas sentimen dalam ulasan produk e-commerce. Model ini, yang menggabungkan Bert-BiGRU-Softmax, dirancang untuk mengatasi tantangan khusus dalam analisis sentimen ulasan e-commerce, seperti pemetaan dimensi dan disambiguasi kata sentimen. Model ini menggunakan Bert untuk ekstraksi fitur dari ulasan e-commerce pada lapisan input, lapisan tersembunyi BiGRU dengan mekanisme attention untuk mendapatkan kode semantik termasuk probabilitas attention dari lapisan input, dan model Softmax sebagai lapisan output untuk mengklasifikasikan kecenderungan sentimen ulasan e-commerce. Penelitian ini dilakukan pada dataset ulasan skala besar dari situs web Suning dan Taobao. Hasil eksperimen menunjukkan bahwa model Bert-BiGRU-Softmax memiliki kinerja yang lebih baik dengan akurasi hingga sebesar 95,5%.

Dibandingkan dengan model RNN, BiGRU, dan Bert-BiLSTM, model Bert-BiGRU-Softmax memiliki peningkatan akurasi setidaknya 3% dalam analisis sentimen ulasan kualitas produk e-commerce. Kesimpulan penelitian ini menekankan pentingnya model Bert-BiGRU-Softmax dalam mengungkapkan opini pelanggan tentang produk yang dijual di platform e-commerce, yang sangat berguna bagi produsen untuk meningkatkan kualitas produk mereka. Penelitian ini menunjukkan bahwa model deep learning, khususnya yang berbasis Transformer dapat melakukan prediksi yang akurat untuk analisis sentimen yang penting dalam mengelola kualitas produk dan layanan pelanggan di sektor e-commerce.

Tabel 2. 1 Penelitian Terkait

No	Tahun	Peneliti	Judul	Metode	Hasil
1	2022	Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, dan Amena Mahmoud	Text-Based Emotion Recognition Using Deep Learning Approach	Hybrib Model (CNN, Bi-GRU, SVM)	Evaluasi performa model menggunakan tiga jenis dataset yang berbeda: kalimat, tweet, dan dialog, mencapai akurasi sebesar 80.11%
2	2021	Ashok Kumar D dan Anandan Chinnalagu	Transformer based Contextual Model for Sentiment Analysis of Customer Reviews: A	Fine-tuned BERT	Dalam memprediksi sentimen pelanggan dari ulasan di Twitter, IMDB Movie Reviews, Yelp, dan



			Fine-tuned BERT		Amazon, model BERT memiliki kinerja yang unggul dengan skor akurasi sebesar 90%, recall 91%, presisi 90%, dan F1 90%, dibandingkan dengan model-model lain seperti LSVM, fastText, dan BiLSTM
3	2022	Xiaokang Gong, Wenhao Ying, Shan Zhong, Shengrong Gong	Text Sentiment Analysis Based on Transformer and Augmentation	BERT, ALBERT, dan MobileBERT	Hasil penelitian menunjukkan bahwa model BERT unggul dalam akurasi, yaitu sebesar 93,38% akurasi pada dataset AG News dan 92,94% akurasi pada dataset SST
4	2022	Laleh Davoodi dan József Mezei	A Comparative Study of	BERT, RoBERTa, Naive	Hasil penelitian menunjukkan bahwa model

			Machine Learning Models for Sentiment Analysis: Customer Reviews of E-Commerce Platforms	Bayes, SVM	<p>berbasis transformer seperti BERT dan RoBERTa unggul dibandingkan model pembelajaran mesin tradisional. RoBERTa khususnya menunjukkan performa yang sangat baik dengan akurasi di atas 98% pada set uji. Meskipun model berbasis transformer membutuhkan waktu pelatihan yang lebih lama, peningkatan performa yang mereka tawarkan menjadi kompensasi yang berharga</p>
--	--	--	--	------------	---

5	2020	Yi Liu, Jiahuan Lu, Jie Yang, dan Feng Mao	Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax	Bert-BiGRU-Softmax, RNN, BiGRU, BiLSTM	<p>Hasil eksperimen menunjukkan bahwa model Bert-BiGRU-Softmax memiliki kinerja yang lebih baik dengan akurasi hingga sebesar 95,5%.</p> <p>Dibandingkan dengan model RNN, BiGRU, dan Bert-BiLSTM, model Bert-BiGRU-Softmax memiliki peningkatan akurasi setidaknya 3% dalam analisis sentimen ulasan kualitas produk e-commerce</p>
---	------	--	--	--	--

## **2.2 Dasar Teori**

### **2.2.1 Emosi**

Emosi merupakan suatu fenomena yang seringkali sulit untuk didefinisikan, namun memiliki peran penting dalam mempengaruhi perilaku dan respons individu terhadap berbagai situasi. Menurut Mulligan & Scherer (2012), emosi dapat dianggap sebagai serangkaian respons terhadap stimulus tertentu yang mencerminkan perasaan, pikiran, dan tindakan individu. Bentuk emosi dapat bervariasi, mulai dari kegembiraan, kepuasan, hingga kekecewaan dan frustrasi. Dalam konteks e-commerce, emosi pelanggan seringkali tercermin melalui ulasan yang mereka tulis setelah berinteraksi dengan produk atau layanan. Ulasan ini tidak hanya memberikan informasi mengenai kualitas produk atau layanan, tetapi juga menunjukkan emosi dan perasaan pelanggan terhadap pengalaman yang mereka alami. Sebagai contoh, ulasan positif mungkin mencerminkan kepuasan dan kegembiraan pelanggan, sedangkan ulasan negatif dapat menunjukkan kekecewaan atau frustrasi. Oleh karena itu, memahami dan mengenali emosi dari ulasan pelanggan e-commerce menjadi penting bagi perusahaan untuk meningkatkan kualitas layanan dan memenuhi harapan pelanggan.

### **2.2.2 Ulasan Pelanggan dalam E-commerce**

Ulasan pelanggan telah menjadi salah satu aspek krusial dalam dunia e-commerce. Ulasan ini tidak hanya memberikan gambaran tentang kualitas produk atau layanan, tetapi juga mencerminkan pengalaman dan emosi pelanggan saat berbelanja. Menurut sebuah studi oleh Katole (2022), ulasan produk online telah berkembang menjadi sumber informasi yang berharga yang mempengaruhi keputusan pembelian dengan meningkatnya media digital dan sosial. Ulasan positif tentunya menguntungkan bagi toko, namun ulasan negatif dapat mempengaruhi keputusan konsumen dan berpotensi menurunkan penjualan. Oleh karena itu, respons yang tepat dan persuasif terhadap ulasan negatif menjadi penting untuk meminimalkan efek buruknya.

Selanjutnya, penting untuk membedakan antara analisis sentimen dan pengenalan emosi. Analisis sentimen biasanya mengkategorikan ulasan ke dalam tiga kategori: positif, negatif, dan netral. Sementara pengenalan emosi mencoba untuk mengidentifikasi emosi yang lebih spesifik yang mungkin terkandung dalam ulasan, seperti senang, sedih, atau marah. Sebagai contoh, sebuah ulasan mungkin memiliki sentimen positif tetapi mengungkapkan emosi seperti kegembiraan atau kelegaan. Sebaliknya, ulasan dengan sentimen negatif mungkin mencerminkan emosi seperti frustrasi, kekecewaan, atau kemarahan (Changchit & Klaus, 2020).

### **2.2.3 Deep Learning**

Deep learning, sebagai salah satu cabang dari machine learning, memanfaatkan neural networks dengan banyak lapisan untuk memahami dan memodelkan data dengan kompleksitas tinggi. Dalam konteks pengenalan emosi dari teks, deep learning menawarkan kemampuan untuk memahami nuansa dan konteks yang mungkin sulit diidentifikasi dengan metode tradisional. Sebuah penelitian oleh Atmaja et al (2019) menunjukkan bahwa pendekatan berbasis deep learning dapat digunakan untuk mengenali emosi dari teks tertulis dan teks lisan. Hasil penelitian menunjukkan bahwa sistem yang mereka kembangkan mampu mengenali emosi dengan akurasi yang lebih baik, terutama ketika menggunakan dataset yang lebih besar. Dengan demikian, pendekatan deep learning menawarkan potensi besar dalam meningkatkan akurasi dan efektivitas pengenalan emosi dari teks.

### **2.2.4 Word Embedding**

Word embedding adalah salah satu teknik canggih dalam bidang NLP yang bertujuan untuk merepresentasikan kata-kata sebagai vektor dalam ruang berdimensi rendah. Teknik ini berdasarkan prinsip bahwa kata-kata dengan makna atau fungsi yang serupa cenderung memiliki distribusi yang serupa dalam teks (Jiao & Zhang, 2021). Dengan kata lain, jika dua kata sering muncul bersamaan dalam

konteks yang sama, maka representasi vektor dari kedua kata tersebut akan dekat satu sama lain dalam ruang vektor.

Cara kerja word embedding dimulai dengan inisialisasi acak dari vektor untuk setiap kata. Selama proses pelatihan, model berusaha untuk memperbarui vektor ini berdasarkan konteks di mana kata-kata muncul. Dalam konteks pengenalan emosi dari ulasan pelanggan e-commerce, word embedding sangat bermanfaat. Misalnya, ulasan seperti "Produk ini luar biasa" dan "Saya sangat puas dengan pembelian ini" keduanya mengungkapkan sentimen positif. Dengan word embedding, model dapat mengenali bahwa "luar biasa" dan "puas" memiliki konotasi emosi yang serupa, meskipun kata-kata tersebut berbeda. Hal ini memungkinkan model deep learning, seperti arsitektur berbasis Transformer, untuk mengidentifikasi dan mengklasifikasikan emosi dalam ulasan dengan lebih akurat.

### 2.2.5 Mekanisme Attention

Attention merupakan salah satu komponen kunci dalam arsitektur Transformer yang memungkinkan model untuk memfokuskan pada bagian-bagian tertentu dari data masukan saat melakukan prediksi (Vaswani et al., 2017). Mekanisme ini memetakan sebuah query dan sekumpulan pasangan key-value ke sebuah output, di mana query, keys, values, dan output semuanya adalah vektor. Output dihitung sebagai jumlah terbobot dari values, di mana bobot yang diberikan ke setiap value dihitung oleh fungsi kompatibilitas dari query dengan key yang sesuai.

Salah satu jenis attention yang populer adalah “Scaled Dot-Product Attention” yang juga digunakan pada arsitektur Transformer yang diperkenalkan oleh Vaswani et al (2017) dalam makalah berjudul “Attention Is All You Need”. Dalam mekanisme ini, dot products dari query dengan semua keys dihitung, dibagi dengan akar dari dimensi  $d_k$ , dan fungsi softmax diterapkan untuk mendapatkan bobot pada values. Secara matematis, ini dapat direpresentasikan sebagai:

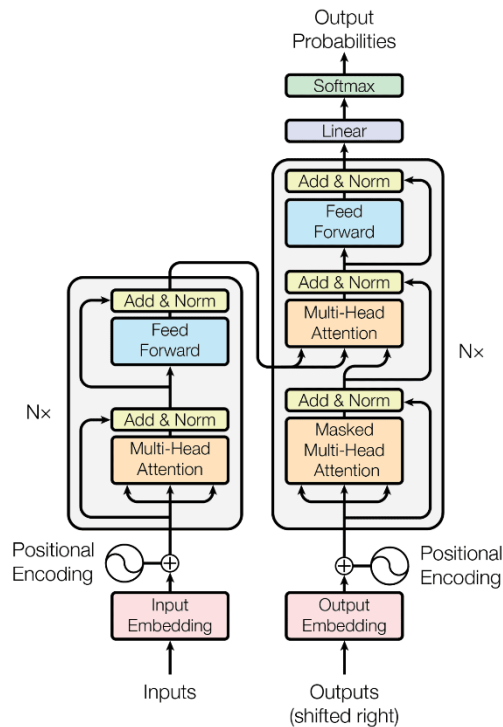
$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Dalam "Scaled Dot-Product Attention", terdapat tiga komponen utama: Query (Q), Key (K), dan Value (V). Semua komponen ini direpresentasikan sebagai matriks. Proses attention dimulai dengan menghitung dot product antara Query dan Key. Hasil dari dot product ini menghasilkan sebuah matriks yang menunjukkan seberapa relevan setiap elemen dalam Query terhadap setiap elemen dalam Key. Namun, untuk memastikan bahwa dot product tidak tumbuh sangat besar (yang dapat menyebabkan gradien yang sangat kecil saat backpropagation), diperlukan penskalaan dengan membagi hasil dot product dengan akar kuadrat dari dimensi  $dk$  (dimensi dari Query atau Key). Kemudian digunakan fungsi softmax pada hasil penskalaan ini untuk mengubah angka-angka dalam matriks menjadi distribusi probabilitas. Distribusi probabilitas ini menunjukkan bobot atau "perhatian" yang harus diberikan model pada setiap elemen dalam Value.

Dalam konteks ulasan pelanggan, misalkan kita memiliki kalimat "Pengiriman cepat tetapi produk rusak". Kata "cepat" mungkin memiliki bobot attention tinggi saat model mencoba memahami aspek pengiriman, sedangkan "rusak" memiliki bobot attention tinggi saat memahami kualitas produk. Dengan demikian, model dapat memfokuskan perhatiannya pada kata-kata yang paling relevan tergantung pada konteks yang sedang dianalisis.

### 2.2.6 Transformer

Transformer adalah arsitektur yang diperkenalkan oleh Vaswani et al pada tahun 2017 dalam makalah berjudul "Attention Is All You Need". Arsitektur ini dirancang untuk mengatasi keterbatasan dari model-model sekuensial seperti RNN dan LSTM dalam pemrosesan sekuensial panjang. Transformer mengandalkan mekanisme "attention" untuk menangkap hubungan antar kata dalam sekuensial tanpa harus memproses kata-kata secara berurutan. Berikut merupakan gambar dari arsitektur Transformer:



Gambar 2. 1 Arsitektur Transformer

Sumber: (Vaswani et al., 2017)

Pada gambar 2.1 di atas, terdapat dua blok utama, yaitu blok encoder (kiri) dan blok decoder (kanan). Blok encoder bertugas mengolah input teks dan mengubahnya menjadi representasi vektor yang kaya informasi. Sementara itu, blok decoder bertugas mengambil output dari encoder dan menghasilkan teks keluaran.

Berikut adalah penjelasan mengenai komponen-komponen utama dari arsitektur Transformer:

1. **Multi-Head Attention:** Mekanisme ini memungkinkan model untuk memperhatikan informasi dari berbagai posisi dalam sekuensial secara bersamaan. Ini terdiri dari beberapa "heads" attention, di mana setiap head mempelajari hubungan antar kata dari perspektif yang berbeda. Multi-head attention menggunakan "scaled dot-product attention", yang menghitung skor attention berdasarkan perkalian titik antara query, key, dan value, yang kemudian di-skalakan dan diaktifkan dengan fungsi softmax.
2. **Positional Encoding:** Karena Transformer tidak memiliki konsep urutan



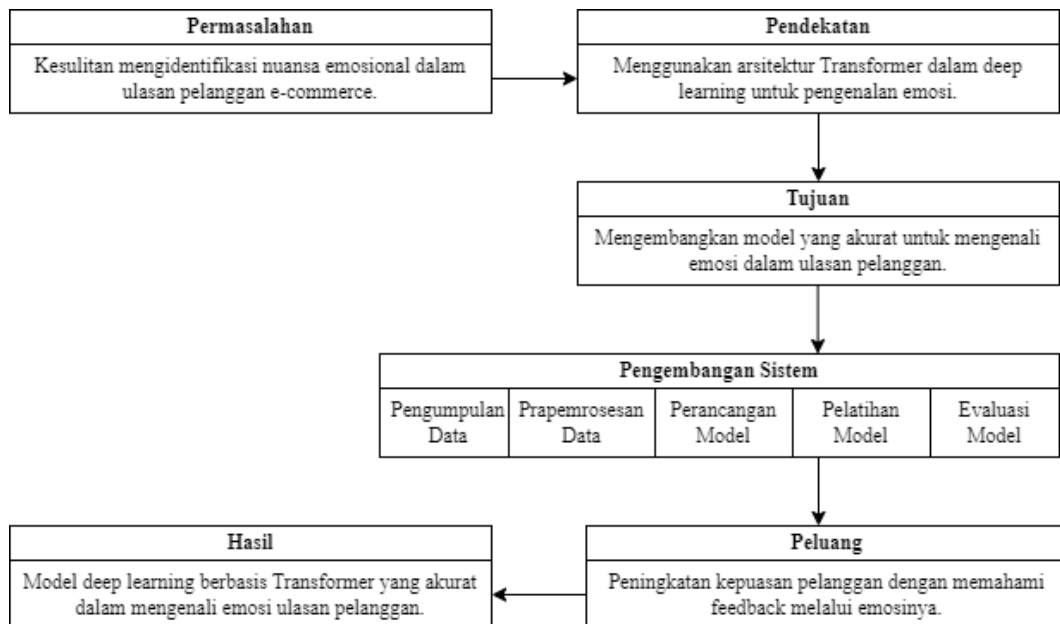
atau posisi kata bawaan, positional encoding ditambahkan ke vektor embedding kata untuk memberikan informasi tentang posisi relatif atau absolut kata dalam sekuensial.

3. Feed Forward Neural Networks: Setiap layer dalam Transformer mengandung feed forward neural network yang beroperasi secara independen pada setiap posisi. Ini membantu dalam transformasi representasi kata pada setiap posisi.
4. Normalization and Residual Connections: Layer normalisasi dan residual connections digunakan untuk membantu pelatihan model yang dalam, memastikan bahwa sinyal gradien dapat mengalir melalui jaringan dengan baik.

Untuk tugas klasifikasi teks seperti pengenalan emosi, biasanya hanya bagian encoder dari arsitektur Transformer yang digunakan. Ini karena tugas klasifikasi teks tidak memerlukan pembangkitan teks baru, yang merupakan peran utama dari bagian decoder dalam Transformer. Encoder cukup untuk memahami konteks dan makna dari teks masukan, dan menghasilkan representasi yang kaya yang kemudian dapat digunakan untuk klasifikasi.

Dalam konteks "Pengenalan Emosi dari Ulasan Pelanggan E-commerce Menggunakan Deep Learning Berbasis Transformer", Transformer dapat digunakan untuk memahami nuansa dan konteks emosi dalam ulasan pelanggan. Mekanisme attention memungkinkan model untuk memperhatikan kata-kata kunci yang mungkin menunjukkan emosi tertentu, sementara positional encoding membantu model memahami struktur kalimat dan hubungan antar kata. Dengan demikian, model dapat menghasilkan representasi yang kaya dari teks ulasan, yang kemudian dapat digunakan untuk klasifikasi emosi.

### 2.3 Kerangka Pemikiran

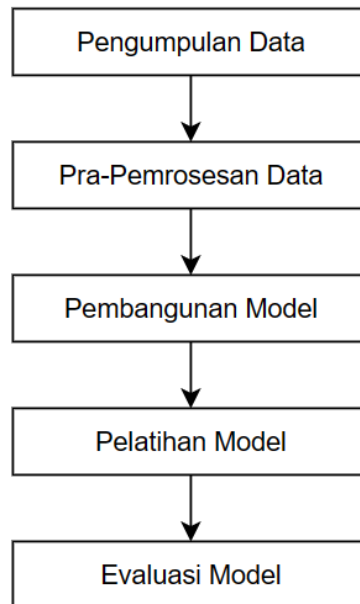


Gambar 2. 2 Kerangka Pemikiran

Gambar 2.2 merupakan Kerangka Pemikiran yang dimulai dengan identifikasi permasalahan, yaitu kesulitan dalam mengidentifikasi nuansa emosional pada ulasan pelanggan e-commerce. Untuk mengatasi hal ini, pendekatan yang diusulkan adalah pemanfaatan arsitektur Transformer dalam deep learning. Tujuannya adalah menciptakan model yang dapat mengenali emosi dengan akurasi tinggi dari ulasan pelanggan. Selanjutnya, dalam tahap pengembangan sistem, terdapat empat fase penting: pengumpulan data, pra-pemrosesan data, perancangan model, serta pelatihan dan evaluasi model. Hasil yang diharapkan adalah terciptanya model deep learning berbasis Transformer yang akurat dalam mengenali emosi dari ulasan pelanggan. Selain itu, peluang yang dapat diperoleh dari penelitian ini adalah meningkatkan kepuasan pelanggan e-commerce dengan memahami feedback mereka melalui emosi yang tersirat dalam ulasan.

## **BAB III METODE PENELITIAN**

### **3.1 Tahapan Penelitian**



Gambar 3. 1 Diagram Tahapan Penelitian

Gambar 3.1 Diagram Tahapan Penelitian merupakan tahapan metodologis yang menyajikan kerangka kerja penelitian. Tahapan ini dimulai dengan pengumpulan data yang dipilih secara spesifik untuk memenuhi kebutuhan penelitian ini. Diikuti dengan pra-pemrosesan, di mana data diolah untuk mendapatkan format yang sesuai untuk analisis berikutnya.

Pembangunan model merupakan langkah inti di mana teknik deep learning berbasis Transformer dikembangkan. Model ini kemudian dilatih dengan dataset yang telah diproses, dengan fokus khusus pada penyesuaian dan optimasi berdasarkan karakteristik data. Akhirnya, model dievaluasi untuk menentukan efektivitasnya dalam mengklasifikasikan emosi, yang merupakan langkah penting untuk mengukur pencapaian tujuan penelitian.

### **3.2 Tools**

Dalam pelaksanaan penelitian "Pengenalan Emosi dari Ulasan Pelanggan E-commerce Menggunakan Deep Learning Berbasis Transformer", pemilihan dan

pemanfaatan perangkat lunak yang tepat adalah penting untuk mendukung efisiensi dan keakuratan proses penelitian. Penelitian ini mengandalkan sejumlah alat dan teknologi yang telah terbukti dapat mendukung pekerjaan yang kompleks dan komputasi intensif yang terkait dengan deep learning dan pemrosesan bahasa alami.

### **3.2.1 Google Colab**

Sebagai platform utama, Google Colab dipilih karena kemudahan akses dan penggunaan sumber daya komputasi yang tinggi tanpa perlu investasi infrastruktur yang signifikan. Google Colab menyediakan lingkungan pengembangan yang berbasis cloud dengan akses ke GPU, memungkinkan pelatihan model yang cepat dan iterasi eksperimen tanpa hambatan.

### **3.2.2 Python**

Python, dengan sintaksnya yang bersih dan komprehensif, menjadi bahasa pemrograman utama yang digunakan dalam penelitian ini. Keputusan ini didukung oleh ekosistem Python yang kaya, yang menyediakan berbagai pustaka dan kerangka kerja untuk NLP dan pembelajaran mesin.

### **3.2.3 Pandas**

Pandas adalah pustaka pengolahan dan analisis data yang kuat dalam Python, yang digunakan untuk manipulasi dataset. Ini memungkinkan penelitian ini untuk melakukan pembersihan data, seperti menghilangkan nilai yang hilang dan duplikat, serta memudahkan seleksi dan transformasi data yang diperlukan untuk analisis lebih lanjut.

### **3.2.4 TensorFlow dan Keras**

TensorFlow dan Keras, dipilih sebagai kerangka kerja untuk membangun dan melatih model deep learning. Keras menyederhanakan proses pembuatan model Transformer dengan menyediakan blok bangunan abstrak yang

mempermudah implementasi lapisan seperti Multi-Head Attention, Layer Normalization, dan Dropout.

### 3.2.5 Matplotlib

Matplotlib, pustaka visualisasi Python, digunakan untuk menggambarkan grafik yang mengilustrasikan akurasi dan loss selama proses pelatihan dan validasi model. Visualisasi ini memberikan insight visual yang berharga terkait dengan performa model dan membantu mengidentifikasi masalah seperti overfitting atau underfitting dengan lebih cepat.

### 3.2.6 Scikit-learn

Scikit-learn adalah pustaka pembelajaran mesin yang memberikan fungsi untuk membagi dataset, menghitung class weights, dan mengukur kinerja model. Di dalam penelitian ini, scikit-learn memfasilitasi proses pembagian data menjadi set pelatihan dan validasi, encoding label emosi, serta menghitung confusion matrix dan skor akurasi.

## 3.3 Pengumpulan Data

Pengumpulan data merupakan langkah awal dan fundamental dalam penelitian yang bertujuan untuk "Pengenalan Emosi dari Ulasan Pelanggan E-commerce Menggunakan Deep Learning Berbasis Transformer". Proses ini memastikan bahwa data yang diperoleh sesuai untuk mendukung analisis penelitian. Penelitian ini menggunakan "Product Reviews Dataset for Emotions Classification Tasks - Indonesian (PRDECT-ID) Dataset", (Sutoyo, et al., 2022) yang diterbitkan pada 20 Mei 2022. Untuk keperluan penelitian ini, dataset telah diunduh pada tanggal 21 November 2023, dan dapat diakses melalui tautan berikut: <https://data.mendeley.com/datasets/574v66hf2v/1>.

Dataset ini dihimpun oleh sekelompok peneliti dan akademisi dari Bina Nusantara University yang terdiri atas Rhio Sutoyo, Andry Chowanda, Said Achmad, dan rekan-rekan yang lain. Dataset ini diambil dari platform e-commerce

terkemuka di Indonesia, Tokopedia, dan mengandung ulasan produk yang dianotasi dengan label emosi dan sentimen dalam bahasa Indonesia, yang mencakup 29 kategori produk.

Annotasi dilakukan oleh sekelompok ahli yang mengikuti kriteria anotasi emosi yang dirancang oleh seorang ahli psikologi klinis. Setiap ulasan produk diberi label emosi tunggal, yaitu anger, fear, happy, love, dan sadness yang memungkinkan penelitian ini untuk fokus pada pengenalan emosi yang spesifik dari ulasan pelanggan. Selain ulasan dan label emosi, dataset juga meliputi atribut lain seperti Lokasi, Harga, Rating Keseluruhan, Jumlah Terjual, Total Ulasan, dan Rating Pelanggan. Atribut-atribut ini memberikan konteks tambahan yang mendukung penelitian lebih lanjut dan memungkinkan analisis yang lebih mendalam terkait dengan aspek-aspek lain dari ulasan pelanggan yang dapat mempengaruhi emosi yang dinyatakan.

Category	Product Name	Location	Price	Overall Rating	Number Sold	Total Review	Customer Rating	Customer Review	Sentiment	Emotion
Party Supplies and Craft	Gantungan kait lengket transparan mudah dipasa...	Jakarta Utara	499	4.8	9969	794	5	Barang ori, harga murah, seller fast response,...	Positive	Happy
Camera	2 in 1 Portable Mini Folding Hand Monopod Stan...	Jakarta Utara	15000	4.7	3304	1028	1	kecewa barang tdk lengkap tdk ada holder U moh...	Negative	Sadness
Computers and Laptops	RAM KINGSTON HYPERX FURY DDR4 8GB 2666MHz 2130...	Jakarta Selatan	425000	4.9	5145	1970	5	Respon cepat, barang original terbaik recomend...	Positive	Love
Mother and Baby	Merries Pants Positive Skin L 37	Kota Bekasi	50800	5.0	5700	1696	1	ampuuuuunn dibungkus pake plastik bening.. jdi ...	Negative	Fear
Mother and Baby	MOOIMOM Breathable Hipseat Carrier Gendongan B...	Jakarta Utara	590000	4.9	1501	810	2	paket datang kardus penyok & robek. malu klo b...	Negative	Fear
Phones and Tablets	XIAOMI 1more Omthing E-Joy Smart Watch - Hitam	Jakarta Barat	279000	4.8	249	104	4	Respon penjualnya top.	Positive	Happy
Movies and Music	Wireless Lavalier Microphone Mini Mic for iPho...	Jakarta Pusat	250000	4.8	496	264	4	Bagus mantab jernih	Positive	Happy
Beauty	Madame Gie Madame Perfect Brow - Pensil alis -...	Jakarta Barat	20000	4.8	12300	5500	2	Kualitas produk bgs sebenarnya, tapi sayangnya...	Negative	Fear
Automotive	SEPASANG Lampu Tembak Sorot 6 Mata LED Spion M...	Jakarta Barat	37500	4.7	571	347	5	Barang mantap, sesuai deskripsi, tapi belum di...	Positive	Happy
Automotive	Little Trees Parfum Mobil Car Perfume Air Fres...	Jakarta Utara	16000	4.9	12700	1515	1	yg dipesan lain yg dikirim lain !!!	Negative	Anger

Gambar 3. 2 PRDECT-ID Dataset

PRDECT-ID Dataset terdiri dari 5400 baris dan 11 kolom yang mencakup beragam kategori produk, yang menjamin adanya variasi dalam ekspresi emosi dan memberikan representasi yang lebih luas dari pola pembelian dan ulasan pelanggan dalam lingkup e-commerce Indonesia.

### 3.4 Pra-Pemrosesan Data

Pra-pemrosesan data adalah tahap yang penting dalam pipeline pembelajaran mesin, terutama dalam ranah NLP. Tahapan ini berfungsi untuk mengkonversi teks asli yang kaya akan variasi linguistik menjadi format yang lebih terstruktur dan seragam, memungkinkan algoritma pembelajaran mesin untuk menggali pola dan hubungan dengan lebih efisien. Tahapan ini juga membantu dalam mengurangi kompleksitas komputasional dan meningkatkan kecepatan serta kinerja algoritma yang digunakan.

#### 3.4.1 Pemilihan dan Pembersihan Data Awal

Langkah awal yang diambil adalah memuat dataset yang berisi ulasan produk ke dalam lingkungan pemrograman Python menggunakan pustaka `pandas`, yang merupakan alat standar industri untuk manipulasi data dalam bentuk `DataFrame`. Langkah selanjutnya adalah melakukan pembersihan data dan memisahkan data untuk keperluan pemodelan dan pengujian. Pengelompokan data pengujian dilakukan berdasarkan label `'Emotion'` untuk menentukan distribusi sampel yang akan digunakan dalam pengujian.

Dengan menggunakan metode `'.sample()'` yang dilengkapi dengan penanda `'random_state'`, dipastikan bahwa pemilihan sampel bersifat deterministik, memungkinkan reproducibility dalam pengujian yang dilakukan oleh peneliti lain di masa mendatang. Sebanyak 100 sampel data pengujian diambil dari setiap kelas, yaitu `anger`, `fear`, `happy`, `love`, dan `sadness` sehingga terdapat total 500 data pengujian. Proses ini bertujuan untuk memastikan bahwa evaluasi model nantinya dilakukan pada data yang belum pernah "dilihat" oleh model selama fase pelatihan, sehingga memberikan indikasi yang lebih akurat mengenai kinerja model di dunia nyata.

Sebelum pemisahan data pemodelan dan pengujian, dilakukan proses pembersihan data pemodelan yang dimulai dengan mengeliminasi baris yang mengandung nilai yang hilang menggunakan metode `'dropna()'`. Ini dilakukan untuk mencegah kemungkinan distorsi pada model yang dapat disebabkan oleh

ketidaklengkapan informasi. Selanjutnya, untuk menghindari redundansi dalam dataset yang dapat mempengaruhi distribusi kelas dan performa model, data duplikat diidentifikasi menggunakan fungsi `'drop_duplicates()'` dan dihapus. Ini memastikan bahwa setiap entri dalam dataset mewakili sebuah sampel unik, yang memberikan diversitas dan variabilitas yang diperlukan untuk pembelajaran mesin yang robust.

### **3.4.2 Normalisasi dan Pembersihan Teks Lanjutan**

Setelah pembersihan data awal, langkah selanjutnya adalah pemilihan kolom yang relevan, yang dalam hal ini adalah 'Customer Review' dan 'Emotion'. Langkah selanjutnya adalah normalisasi teks, yang melibatkan pengubahan semua teks ke huruf kecil menggunakan fungsi `'lower()'`, yang menjadi langkah penting dalam mengeliminasi varian yang tidak perlu yang dapat muncul dari penggunaan huruf kapital.

Selanjutnya, karakter-karakter yang bukan alfabet diganti dengan spasi menggunakan ekspresi regular `'[^a-z\s]'` untuk menciptakan keseragaman dan untuk menghindari kebingungan model dalam menginterpretasikan simbol-simbol tersebut sebagai bagian dari teks yang relevan. Sementara itu, penghilangan spasi dengan menggunakan ekspresi regular `'\s+'` berlebih juga dilakukan untuk menstandarkan teks ulasan, sehingga setiap sampel teks memiliki struktur yang konsisten dan mudah diproses oleh algoritma tokenisasi yang akan diaplikasikan berikutnya. Setelah semua proses dilakukan, selanjutnya adalah memisahkan dataset testing. Semua proses tersebut dilakukan dengan menggunakan bahasa pemrograman Python.

### **3.4.3 Tokenisasi, Vektorisasi, dan Pembuatan Sekuens**

Setelah teks dibersihkan, langkah selanjutnya adalah tokenisasi teks. Tokenisasi adalah proses mengubah teks menjadi token atau unit-unit terkecil yang bermakna. Penulis menggunakan `'Tokenizer'` dari pustaka Keras, yang diinisialisasi dengan ukuran kosakata yang ditentukan dalam eksperimen. Penulis menetapkan



eksperimen dengan variasi ukuran kosakata dari 5000 hingga 10000 untuk menemukan ukuran yang optimal. Keras 'Tokenizer' kemudian dijalankan pada teks ulasan yang telah dinormalisasi untuk mengubah teks ulasan menjadi urutan token atau kata. Selain itu juga akan diproses vektorisasi dengan mengonversi urutan token menjadi urutan angka yang mewakili indeks kata dalam kamus yang telah ditentukan. Proses ini menghasilkan representasi numerik dari teks yang siap untuk diproses oleh algoritma pembelajaran mesin.

Langkah berikutnya adalah padding atau penambahan padding pada sekuens untuk mencapai panjang sekuens yang seragam, yang diperlukan untuk pemodelan nantinya. Padding dilakukan dengan menentukan nilai melalui eksperimen dengan rentang 20 hingga 150 token untuk menemukan panjang sekuens yang ideal. Panjang ini sangat penting karena terlalu pendek dapat menghilangkan informasi penting, sedangkan terlalu panjang dapat meningkatkan kompleksitas komputasi tanpa manfaat yang proporsional.

#### **3.4.4 Encoding Label dan Pembagian Dataset Pemodelan**

Label emosi yang merupakan variabel target dalam penelitian ini diubah dari format teks ke numerik menggunakan LabelEncoder. Proses ini menghasilkan vektor numerik yang mewakili kategori emosi, yang diperlukan untuk proses pembelajaran mesin. Langkah selanjutnya adalah dataset yang akan digunakan untuk pemodelan dibagi menjadi set pelatihan dan validasi.

Pembagian ini dilakukan dengan menggunakan fungsi 'train\_test\_split' dari scikit-learn, yang secara acak membagi data menjadi set pelatihan (80%) dan set validasi (20%) yang menghasilkan sebanyak 3920 baris data untuk pelatihan model dan 980 baris data untuk validasi model. Selain itu 'random\_state' juga ditetapkan untuk memastikan konsistensi pemisahan data di berbagai eksperimen. Proporsi pembagian yang dipilih bertujuan untuk memberikan jumlah data yang cukup untuk pelatihan model sambil mempertahankan jumlah yang signifikan untuk validasi kinerja model.

### 3.5 Pembangunan Model

Dalam pengembangan model yang bertujuan untuk klasifikasi emosi dari ulasan pelanggan e-commerce, pemilihan arsitektur Transformer sebagai dasar pembangunan model bukan tanpa alasan. Model Transformer, yang dirancang oleh Vaswani et al. (2017), telah merevolusi pemahaman tentang pengolahan bahasa dengan menghadirkan konsep 'attention mechanism' yang memungkinkan pemodelan dependensi tanpa menghiraukan jarak dalam urutan kata.

#### 3.5.1 Arsitektur Transformer Block

Dalam pembangunan model klasifikasi emosi dari ulasan pelanggan e-commerce, elemen kunci yang membedakan arsitektur ini adalah penerapan Transformer Block. Transformer Block, yang merupakan terobosan dalam NLP yang memanfaatkan mekanisme 'attention' untuk meningkatkan kemampuan model dalam memahami bahasa.

##### 3.5.1.1 Multi-Head Attention (MHA)

MHA merupakan komponen inti dalam Transformer Block, dirancang untuk memberikan model kemampuan dalam menangkap konteks dari input teks dengan cara yang lebih efisien. Esensinya, MHA memungkinkan model untuk memproses bagian input secara serentak dari berbagai sudut pandang atau 'heads'. Core dari MHA adalah fungsi attention yang dijelaskan oleh rumus:

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

Dimana  $Q$ ,  $K$ , dan  $V$  adalah matriks query, key, dan value. Mekanisme ini menghasilkan skor attention dengan menghitung produk dot antara query dan key, yang kemudian dinormalisasi dengan faktor  $\sqrt{d_k}$  untuk menghindari nilai yang terlalu besar yang bisa menyebabkan gradien yang sangat kecil. Skor ini kemudian

digunakan untuk menimbang value, sehingga outputnya adalah representasi terbobot dari input berdasarkan skor attention tersebut.

Dalam Implementasi MHA:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Setiap head dalam MHA memiliki matriks bobot proyeksi yang unik, yaitu  $W_i^Q$ ,  $W_i^K$ , dan  $W_i^V$ . Ini memungkinkan setiap head untuk memfokuskan perhatian pada aspek yang berbeda dari input. Output dari setiap head kemudian digabungkan dan diproyeksikan melalui matriks bobot  $W^O$ , menghasilkan representasi akhir yang menggabungkan informasi dari semua heads.

MHA diimplementasikan dengan menggunakan layer MultiHeadAttention yang tersedia dalam library TensorFlow. Dalam konteks ulasan pelanggan, hal ini memungkinkan model untuk mengidentifikasi berbagai aspek dari emosi, seperti intensitas, konteks, dan nuansa emosi, yang mungkin tersebar di seluruh ulasan.

### 3.5.1.2 Feed-Forward Network (FFN)

Setelah proses attention, output dari MHA dilewatkan melalui jaringan Feed-Forward. Setiap blok Transformer dilengkapi dengan FFN yang terdiri dari dua lapisan dense, dengan lapisan pertama menggunakan aktivasi ReLU untuk menambahkan non-linearitas pada model. FFN pada setiap blok Transformer dijelaskan oleh rumus:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Dimana  $W_1$  dan  $W_2$  adalah matriks bobot yang dipelajari dan  $b_1$  dan  $b_2$  adalah bias. Fungsi aktivasi ReLU,  $\max(0, z)$ , digunakan di sini untuk menambahkan non-linearitas pada model. Ini memberikan model kemampuan

untuk memetakan representasi attention yang rumit ke ruang yang lebih tinggi (dimensi yang lebih tinggi), yang memungkinkan model untuk mempelajari hubungan yang lebih abstrak dalam data. Lapisan dense kedua bertindak sebagai pengubah linear yang mengembalikan output ke dimensi yang sama seperti input awal, memastikan bahwa output dari FFN dapat diintegrasikan kembali ke dalam aliran utama jaringan.

### **3.5.1.3 Normalisasi Lapisan dan Dropout**

Komponen penting lainnya dari blok ini adalah penggunaan layer normalization, yang membantu stabilisasi pembelajaran dengan memastikan bahwa distribusi output dari setiap sub-lapisan memiliki mean dan varians yang tetap. Penggunaan dropout setelah layer attention dan FFN merupakan langkah preventif untuk menghindari overfitting, dengan cara secara acak mematikan sebagian dari output neuron selama fase pelatihan.

### **3.5.1.4 Implementasi Transformer Block**

Implementasi Transformer Block ini dalam kode Python menggunakan TensorFlow ditandai dengan pembuatan kelas `TransformerBlock` yang mewarisi dari `layers.Layer`. Ini menyediakan kerangka kerja yang dibutuhkan untuk menentukan arsitektur custom dan perilaku saat melakukan forward pass, yang didefinisikan dalam metode `call`. Dalam metode ini, input pertama-tama melewati MHA, diikuti oleh operasi dropout dan normalization. Kemudian, output dari langkah pertama ini diteruskan ke FFN yang juga diikuti oleh dropout dan normalization. Proses ini menghasilkan output yang kaya informasi dan siap untuk diintegrasikan ke dalam layer berikutnya dari model.

### **3.5.1.5 Contoh Perhitungan MHA**

Contoh perhitungan Multi-Head Attention (MHA) dalam konteks ulasan "pelayanan cepat", akan melalui seluruh proses dari inisialisasi matriks, perhitungan attention scores, hingga akhirnya mendapatkan output akhir dari MHA.

Hyperparameter yang akan digunakan adalah 3 dimensi embedding (embed\_dim = 3) dan 2 heads attention (num\_heads = 2).

Misalkan terdapat ulasan "pelayanan cepat" yang telah di-tokenize dan di-encode menjadi vektor embedding sebagai berikut:

- "pelayanan": [1, 0, 2]
- "cepat": [0, 1, 1]

Kedua vektor ini digabung menjadi matriks input  $X$  dengan ukuran 2 x 3 (2 kata dengan 3 dimensi embedding):

$$X = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix}$$

Untuk setiap head dalam MHA, diperlukan inisialisasi matriks bobot  $W^Q, W^K, W^V$  untuk query, key, dan value, serta  $W^O$  untuk output. Nilai tersebut biasanya diinisialisasi secara acak dan kemudian dilatih menggunakan backpropagation. Berikut merupakan asumsi dari nilai matriks bobot:

$$\begin{aligned} W_1^Q &= \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, W_1^K = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, W_1^V = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \\ W_2^Q &= \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, W_2^K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, W_2^V = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ W^O &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Berikutnya adalah menghitung Attention dari head pertama. Langkah pertama adalah menghitung Q, K, dan V:

$$\begin{aligned} Q_1 &= XW_1^Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \\ K_1 &= XW_1^K = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \\ V_1 &= XW_1^V = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \end{aligned}$$

Selanjutnya adalah menghitung skor Attention dengan mengalikan  $Q$  dengan transpos dari  $K$  dan membaginya dengan akar kuadrat dari dimensi key (dalam kasus ini adalah  $\sqrt{3}$ ):

$$AttentionScores_1 = \frac{Q_1 K_1^T}{\sqrt{3}} = \frac{\begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}}{\sqrt{3}} = \frac{\begin{bmatrix} 11 & 5 \\ 5 & 3 \end{bmatrix}}{\sqrt{3}}$$

Kemudian menerapkan fungsi softmax pada skor ini:

$$AttentionWeights_1 = softmax(AttentionScores_1)$$

$$AttentionWeights_1 = softmax\left(\frac{\begin{bmatrix} 11 & 5 \\ 5 & 3 \end{bmatrix}}{\sqrt{3}}\right)$$

$$AttentionWeights_1 = \begin{bmatrix} 0.88 & 0.12 \\ 0.62 & 0.38 \end{bmatrix}$$

Nilai di atas menunjukkan probabilitas atau bobot yang diberikan pada setiap kata dalam konteks kata yang lain. Sebagai contoh, nilai 0.88 pada baris pertama, kolom pertama menunjukkan bahwa kata "pelayanan" memiliki korelasi yang kuat dengan dirinya sendiri dalam konteks ini.

Selanjutnya adalah mengalikan bobot attention di atas dengan  $V_1$  untuk mendapatkan output dari head pertama:

$$HeadOutput_1 = AttentionWeights_1 \cdot V_1$$

$$HeadOutput_1 = \begin{bmatrix} 0.88 & 0.12 \\ 0.62 & 0.38 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$HeadOutput_1 = \begin{bmatrix} 1.12 & 1.88 & 2.64 \\ 1 & 1.38 & 1.76 \end{bmatrix}$$

Berikutnya adalah menghitung untuk output dari head kedua. Asumsikan hasil softmax untuk head kedua adalah:

$$AttentionWeights_2 = softmax(AttentionScores_2)$$

$$AttentionWeights_2 = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

Ini mengindikasikan distribusi bobot yang berbeda, memberi penekanan yang berbeda pada hubungan antar kata. Dengan  $V_2$  yang dihitung menggunakan  $W_2^V$ , asumsikan didapatkan:

$$V_2 = XW_2^V = \begin{bmatrix} 0.5 & 1.5 & 2 \\ 1 & 1 & 0.5 \end{bmatrix}$$

Maka output head kedua adalah:

$$\begin{aligned} HeadOutput_2 &= AttentionWeights_2 \cdot V_2 \\ HeadOutput_2 &= \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 1.5 & 2 \\ 1 & 1 & 0.5 \end{bmatrix} \\ HeadOutput_2 &= \begin{bmatrix} 0.85 & 1.2 & 1.05 \\ 0.7 & 1.2 & 0.8 \end{bmatrix} \end{aligned}$$

Setelah mendapatkan output dari head pertama dan kedua, langkah selanjutnya adalah menggabungkan dan memproyeksikan outputnya. Output dari kedua heads digabungkan, dalam kasus ini 2 heads dengan 3 dimensi embedding, didapatkan matriks dengan ukuran  $2 \times 6$ :

$$\begin{aligned} CombinedOutput &= Concat(HeadOutput_1, HeadOutput_2) \\ CombinedOutput &= \begin{bmatrix} 1.12 & 1.88 & 2.64 & 0.85 & 1.2 & 1.05 \\ 1 & 1.38 & 1.76 & 0.7 & 1.2 & 0.8 \end{bmatrix} \end{aligned}$$

Terakhir, output gabungan ini diproyeksikan kembali ke ruang asli dengan menggunakan  $W^O$ :

$$\begin{aligned} MHAOutput &= CombinedOutput \cdot W^O \\ MHAOutput &= \begin{bmatrix} 1.12 & 1.88 & 2.64 & 0.85 & 1.2 & 1.05 \\ 1 & 1.38 & 1.76 & 0.7 & 1.2 & 0.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \\ MHAOutput &= \begin{bmatrix} 4.07 & 5.57 \\ 3.08 & 4.36 \end{bmatrix} \end{aligned}$$

Output dari MHA di atas menunjukkan bahwa model telah berhasil mengintegrasikan informasi dari kedua kata ("pelayanan" dan "cepat") untuk menghasilkan representasi yang lebih kaya. Angka-angka ini merefleksikan bagaimana konteks "cepat" mempengaruhi pemahaman model tentang "pelayanan", dan sebaliknya. Nilai-nilai dalam output menunjukkan distribusi "perhatian" atau bobot yang diberikan model kepada setiap kata dalam konteks kata lainnya. Misalnya, nilai yang lebih tinggi (5.57 dan 4.36) menunjukkan bahwa ada aspek tertentu dari "pelayanan" yang sangat relevan atau diperkuat oleh konteks "cepat", menandakan bahwa kecepatan pelayanan adalah aspek kunci dalam ulasan ini.

### 3.5.2 Token dan Positional Embedding

Dalam konteks model berbasis Transformer untuk analisis emosi pada ulasan e-commerce, proses embedding merupakan langkah awal yang krusial. Embedding mengubah kata-kata dalam teks menjadi vektor numerik yang kaya informasi, yang selanjutnya dapat diproses oleh model deep learning. Dua jenis embedding yang digunakan dalam model ini adalah token embedding dan positional embedding, yang masing-masing memainkan peran penting dalam menghasilkan representasi teks yang efektif untuk analisis lebih lanjut.

#### 3.5.2.1 Token Embedding

Token embedding bertujuan untuk mengonversi setiap token kata dalam teks menjadi vektor berdimensi tetap. Proses ini sangat penting karena model deep learning tidak dapat memproses teks mentah; sebaliknya, mereka memerlukan input numerik. Token embedding dilakukan dengan cara setiap token diubah menjadi vektor menggunakan matriks embedding yang dipelajari selama proses pelatihan. Dalam konteks model ini, matriks embedding memiliki dimensi yang telah ditetapkan, yang ditentukan oleh parameter `'embed_dim'`. Matriks embedding token tidak ditentukan secara manual tetapi dipelajari dari data selama proses pelatihan. Ini memungkinkan model untuk secara otomatis menangkap dan menyandikan informasi semantik dari kata-kata berdasarkan konteksnya dalam data pelatihan.

#### 3.5.2.2 Positional Embedding

Positional embedding menambahkan informasi tentang posisi relatif dari setiap token dalam teks. Berbeda dengan token embedding, positional embedding bertujuan untuk menyediakan model dengan pemahaman tentang urutan kata dalam teks, yang penting untuk banyak tugas NLP, termasuk analisis emosi. Setiap posisi dalam urutan teks diberi representasi vektor unik, yang memungkinkan model untuk membedakan antara kata yang sama yang muncul di posisi yang berbeda dalam teks. Output dari token dan positional embedding kemudian dijumlahkan



untuk menghasilkan representasi vektor gabungan untuk setiap token. Gabungan ini menghasilkan vektor yang mencakup informasi semantik dari kata itu sendiri dan konteks urutannya dalam teks.

### 3.5.2.3 Implementasi Token dan Positional Embedding

Token dan Positional Embedding ini diimplementasikan dalam kelas `'TokenAndPositionEmbedding'`, yang menggunakan layer `'Embedding'` dari TensorFlow untuk masing-masing jenis embedding. Dalam konteks ini, positional embedding sangat penting karena Transformer tidak memiliki rekurensi atau konvolusi yang secara implisit menangkap informasi posisi, sehingga penambahan positional embedding secara eksplisit adalah krusial. Ukuran vektor embedding (baik untuk token maupun posisi) ditentukan oleh parameter `'embed_dim'`, yang telah diatur sesuai dengan kebutuhan model dan karakteristik dataset.

Pengaturan ukuran vector embedding mempengaruhi jumlah parameter yang dipelajari dalam lapisan embedding dan berkontribusi langsung pada kemampuan model untuk mempelajari representasi teks yang kaya. Pemilihan dimensi embedding juga mencerminkan trade-off antara kemampuan representasi dan efisiensi komputasi. Dimensi yang lebih besar memungkinkan model untuk menangkap lebih banyak informasi tentang setiap token dan posisinya tetapi juga meningkatkan kompleksitas model dan waktu yang diperlukan untuk pelatihan. Dalam hal ini, `'embed_dim'` yang diuji di model berkisar antara 15 hingga 150.

### 3.5.3 Penggabungan Model

Penggabungan dan pelatihan model merupakan tahapan penting dalam pembangunan sistem klasifikasi emosi. Tahapan ini melibatkan eksperimen yang cermat untuk menentukan arsitektur dan hyperparameter yang optimal.

#### 3.5.3.1 Global Average Pooling

Setelah melewati Transformer Block, output diolah melalui Global Average Pooling 1D. Tujuan dari lapisan ini adalah mereduksi dimensi sekaligus

mengekstrak fitur penting. Global Average Pooling menghitung rata-rata dari setiap fitur di seluruh token, menghasilkan satu vektor representatif. Proses ini efektif dalam mengurangi risiko overfitting dibandingkan dengan Fully Connected Layers karena mengurangi jumlah parameter yang harus dipelajari.

### **3.5.3.2 Dropout dan Regularisasi**

Dalam model, diterapkan lapisan Dropout dengan nilai yang ditentukan melalui eksperimen, dengan variasi dari 0.1 hingga 0.5. Tujuan dari Dropout adalah untuk mencegah overfitting dengan mematikan sejumlah neuron secara acak selama proses pelatihan. Hal ini mendorong model untuk mempelajari representasi yang lebih robust.

Regularisasi L2 juga diterapkan pada lapisan Dense dengan koefisien yang ditentukan melalui eksperimen dengan rentang dari 0.0001 hingga 0.01. Regularisasi L2 menambahkan penalti pada besar bobot, mendorong model untuk mempelajari bobot yang lebih kecil, yang sering menghasilkan model yang lebih general dan tahan terhadap noise.

### **3.5.3.3 Dense Layer**

Lapisan Dense diterapkan dengan jumlah unit yang disesuaikan. Eksperimen dilakukan dengan jumlah unit dari 8 hingga 256. Fungsi aktivasi ReLU digunakan pada lapisan pertama karena efektivitasnya dalam memodelkan non-linearitas.

### **3.5.3.4 Lapisan Output dan Softmax**

Lapisan output model merupakan lapisan kunci yang mengklasifikasikan input ke dalam kategori emosi yang berbeda, dalam hal ini adalah 5 output emosi, yaitu anger, fear, happy, love, dan sadness. Untuk lapisan ini, digunakan fungsi aktivasi softmax, yang mengubah logit dari lapisan sebelumnya menjadi probabilitas yang terstandarisasi untuk setiap kelas. Output dari fungsi softmax

adalah vektor probabilitas yang elemennya mewakili probabilitas bahwa input termasuk dalam kelas emosi tertentu.

### 3.5.3.5 Contoh Perhitungan Model

Setelah melewati langkah-langkah embedding dan Transformer Block, sekarang didapatkan MHAOutput. Selanjutnya adalah menggunakan MHAOutput untuk perhitungan selanjutnya hingga mendapatkan probabilitas dari setiap kelas emosi, yaitu happy, fear, anger, love, dan sadness. Langkah selanjutnya adalah menghitung GlobalAveragePooling1D, Dense Layer, dan Output Layer.

MHAOutput yang didapatkan dari hasil perhitungan MHA sebelumnya adalah:

$$MHAOutput = \begin{bmatrix} 4.07 & 5.57 \\ 3.08 & 4.36 \end{bmatrix}$$

Selanjutnya adalah menghitung Global Average Pooling 1D (GAP1D):

$$v = \frac{1}{2} \left( \begin{bmatrix} 4.07 \\ 3.08 \end{bmatrix} + \begin{bmatrix} 5.57 \\ 4.36 \end{bmatrix} \right) = \begin{bmatrix} 3.575 \\ 4.965 \end{bmatrix}$$

Berikutnya adalah menghitung Dense Layer. Asumsikan  $W_{dense}$  dan  $b_{dense}$  adalah sebagai berikut:

$$W_{dense} = \begin{bmatrix} 0.5 & 1.0 \\ 1.5 & -0.5 \end{bmatrix}, \quad b_{dense} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

Maka,

$$\begin{aligned} z_{dense} &= ReLU(W_{dense}v + b_{dense}) \\ z_{dense} &= ReLU \left( \begin{bmatrix} 0.5 & 1.0 \\ 1.5 & -0.5 \end{bmatrix} \begin{bmatrix} 3.575 \\ 4.965 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \right) \\ z_{dense} &= ReLU \left( \begin{bmatrix} 5.5075 \\ 4.9625 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \right) \\ z_{dense} &= ReLU \left( \begin{bmatrix} 5.6075 \\ 5.1625 \end{bmatrix} \right) \\ z_{dense} &= \begin{bmatrix} 5.6075 \\ 5.1625 \end{bmatrix} \end{aligned}$$

Berikutnya adalah menghitung output layer. Asumsikan  $W_{dense}$  dan  $b_{dense}$  adalah sebagai berikut:

$$W_{output} = \begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 1 & 1 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}, b_{output} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}$$

Maka,

$$z_{output} = W_{output}z_{dense} + b_{output}$$

$$z_{output} = \begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 1 & 1 \\ -1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 5.6075 \\ 5.1625 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}$$

$$z_{output} = \begin{bmatrix} 16.0325 \\ -4.9625 \\ 10.97 \\ 0.445 \\ 5.6625 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}$$

$$z_{output} = \begin{bmatrix} 16.1325 \\ -4.7625 \\ 11.27 \\ 0.845 \\ 6.1625 \end{bmatrix}$$

Selanjutnya adalah menerapkan fungsi aktivasi softmax pada  $z_{output}$  untuk mendapatkan probabilitas:

$$Softmax(z_{output})_i = \frac{e^{z_{output}_i}}{\sum_{j=1}^5 e^{z_{output}_j}}$$

Untuk  $i$  dari 1 hingga 5, hitung nilai eksponensial dari setiap elemen dalam  $z_{output}$ , dan bagi dengan jumlah dari semua nilai eksponensial tersebut untuk mendapatkan probabilitas untuk setiap kelas emosi.

- Happy = 99.23%
- Fear = 0%
- Anger = 0.77%
- Love = 0%
- Sadness = 0%

Dari hasil ini model sangat yakin (dengan probabilitas sekitar 99.23%) bahwa emosi yang diberikan dari ulasan ‘Pelayanan Cepat’ adalah Happy. Probabilitas untuk emosi lainnya sangat rendah, menunjukkan bahwa model memiliki keyakinan yang sangat tinggi bahwa input tersebut tidak termasuk dalam kategori emosi tersebut.

### 3.6 Pelatihan Model

Proses pelatihan merupakan tahap kritis dalam pengembangan model machine learning, di mana model belajar dari data dan menyesuaikan parameter internalnya. Pada proses pelatihan model ini, sebanyak 3920 baris data (80%) digunakan untuk pelatihan model dan 980 baris data (20%) untuk validasi model

#### 3.6.1 Penetapan Bobot Kelas

Dalam pembangunan model klasifikasi emosi dari ulasan e-commerce, salah satu tantangan utama adalah ketidakseimbangan dalam distribusi kelas. Kondisi ini sering terjadi dalam kumpulan data nyata, di mana beberapa kelas memiliki lebih banyak sampel daripada kelas lain. Untuk mengatasi masalah ini dan meningkatkan kinerja model pada kelas yang kurang terwakili, diterapkan teknik penetapan bobot kelas.

Teknik yang digunakan untuk menentukan bobot kelas didasarkan pada proporsi invers dari frekuensi kelas dalam data. Metode ini bertujuan untuk memberikan bobot yang lebih tinggi pada kelas yang kurang terwakili, sehingga meningkatkan dampak sampel dari kelas tersebut dalam proses pembelajaran. Bobot kelas dihitung dengan fungsi `compute_class_weight` dari library `scikit-learn`. Rumus yang digunakan untuk menghitung bobot kelas adalah:

$$\omega_j = \frac{N}{k \cdot n_j}$$

Di mana  $\omega_j$  adalah bobot untuk kelas  $j$ ,  $N$  adalah jumlah total sampel dalam dataset,  $k$  adalah jumlah kelas, dan  $n_j$  adalah jumlah sampel dalam kelas  $j$ . Bobot ini kemudian dinormalisasi sehingga rata-rata bobot untuk semua kelas sama dengan 1.

Setelah bobot kelas dihitung, bobot tersebut diterapkan dalam proses pelatihan model. Dalam TensorFlow/Keras, ini diimplementasikan dengan menyediakan bobot kelas sebagai argumen dalam fungsi `'fit'` selama pelatihan. Implementasi ini memastikan bahwa setiap sampel dari kelas minoritas memberikan kontribusi yang lebih signifikan ke fungsi loss selama pelatihan, yang membantu mengurangi bias terhadap kelas yang lebih dominan.

### 3.6.2 Penerapan Early Stopping dalam Pelatihan Model

Early stopping merupakan teknik penting dalam pelatihan model machine learning, khususnya dalam konteks deep learning, untuk mencegah fenomena overfitting. Overfitting terjadi ketika model terlalu menyesuaikan diri dengan data pelatihan hingga kehilangan kemampuannya untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Early stopping bertindak sebagai penjaga yang memonitor performa model pada set validasi dan menghentikan pelatihan ketika model tidak lagi menunjukkan peningkatan.

Dalam implementasi early stopping, salah satu metrik kinerja model (seperti loss atau akurasi) pada set validasi diamati sepanjang proses pelatihan. Jika metrik tersebut tidak membaik (atau bahkan memburuk) selama jumlah epoch yang ditentukan, pelatihan akan dihentikan. Ini dilakukan untuk menghindari pemborosan waktu dan sumber daya komputasi pada pelatihan yang tidak efektif dan untuk menyelamatkan model dari overfitting.

Dalam konteks pelatihan model klasifikasi emosi ini, early stopping dikonfigurasi untuk mengamati `'val_loss'`, yaitu loss pada set validasi. Pengamatan dilakukan dengan parameter sebagai berikut:

1. Monitor: `'val_loss'`, menentukan bahwa pelatihan harus dihentikan berdasarkan peningkatan atau penurunan loss pada set validasi.

2. Patience: Diatur pada 15 epoch. Ini berarti pelatihan akan berlanjut selama 15 epoch setelah deteksi terakhir dari peningkatan performa pada set validasi. Nilai ini dipilih berdasarkan eksperimen yang menunjukkan bahwa ini adalah jangka waktu yang cukup untuk memastikan bahwa model memiliki kesempatan yang adil untuk membaik sebelum dihentikan.
3. Restore Best Weights: Parameter ini diatur ke True, yang memungkinkan model untuk mengembalikan bobot terbaik yang diperoleh selama pelatihan, yaitu bobot yang menghasilkan nilai 'val\_loss' terendah.

### **3.6.3 Optimizer dan Kompilasi Model**

Setelah menetapkan mekanisme early stopping, langkah selanjutnya dalam proses pelatihan model adalah pemilihan dan konfigurasi optimizer. Optimizer berperan penting dalam proses pembelajaran model dengan menyesuaikan bobot jaringan berdasarkan loss yang dihasilkan selama pelatihan.

#### **3.6.3.1 Pemilihan Optimizer**

Untuk model ini, optimizer Adam dipilih karena efisiensinya dalam menangani pembelajaran dalam jaringan saraf dalam. Adam, singkatan dari "Adaptive Moment Estimation", menggabungkan keuntungan dari dua teknik optimisasi lainnya: AdaGrad dan RMSProp. Adam menyesuaikan laju pembelajaran untuk setiap parameter secara individual berdasarkan estimasi pertama dan kedua dari momen gradien.

Hal ini membuat Adam sangat efektif dalam menangani masalah yang melibatkan data besar dan parameter yang banyak. Laju pembelajaran untuk Adam diatur dari eksperimen yaitu dari 0.00001 hingga 0.001 diuji untuk menemukan keseimbangan yang tepat dan memastikan bahwa proses pelatihan tidak berhenti pada minimum lokal yang tidak optimal.

### 3.6.3.2 Kompilasi Model

Setelah menentukan optimizer, langkah selanjutnya adalah mengompilasi model. Kompilasi model adalah proses dimana konfigurasi pembelajaran (seperti optimizer, loss function, dan metrik) didefinisikan. Untuk tugas klasifikasi ini, 'sparse\_categorical\_crossentropy' digunakan sebagai loss function. Loss function ini cocok untuk masalah klasifikasi multi-kelas di mana label kelas disediakan sebagai bilangan bulat.

Fungsi ini menghitung loss antara label kelas aktual dan prediksi yang dihasilkan oleh model, memberikan ukuran kinerja yang dapat digunakan untuk mengoptimalkan bobot jaringan. Metrik 'accuracy' dipilih untuk memonitor kinerja model selama pelatihan dan validasi. Akurasi diukur sebagai proporsi sampel yang diklasifikasikan dengan benar oleh model. Metrik ini memberikan gambaran intuitif tentang seberapa baik model melakukan tugas klasifikasi.

### 3.6.4 Pelatihan Model

Setelah menyelesaikan tahapan-tahapan penting seperti penetapan bobot kelas, konfigurasi early stopping, dan kompilasi model, langkah selanjutnya adalah pelatihan model itu sendiri. Proses pelatihan adalah di mana model belajar dari data dengan menyesuaikan bobot internalnya untuk meminimalkan loss dan meningkatkan akurasi. Pelatihan model dilakukan menggunakan metode 'fit' yang disediakan oleh Keras, sebuah library deep learning populer.

#### 3.6.4.1 Setting Pelatihan

Pelatihan model dilakukan dengan parameter berikut:

1. Data Pelatihan ('x\_train', 'y\_train'): Ini adalah data dan label yang digunakan untuk pelatihan model. Data ini telah melalui proses pra-pemrosesan dan pembagian sehingga siap untuk digunakan dalam pelatihan.
2. Batch Size: Ukuran batch dipilih berdasarkan eksperimen yang mempertimbangkan trade-off antara efisiensi memori dan kecepatan komputasi. Eksperimen dengan batch size bervariasi dari 8 hingga 64



dilakukan untuk menemukan nilai yang menghasilkan konvergensi yang cepat dan stabil.

3. Epochs: Epoch adalah jumlah iterasi yang dilakukan model atas keseluruhan data pelatihan. Jumlah epoch dipilih berdasarkan eksperimen yang berkisar dari 50 hingga 300, untuk menentukan jumlah iterasi optimal yang diperlukan model untuk belajar tanpa overfitting.
4. Validation Data (`x\_val`, `y\_val`): Data validasi digunakan untuk menguji model terhadap dataset yang tidak digunakan selama pelatihan. Hal ini memungkinkan penilaian objektif tentang kinerja model dan pencegahan overfitting.
5. Callbacks ([early\_stopping]): Callback early stopping digunakan untuk menghentikan pelatihan model ketika tidak ada peningkatan pada loss validasi. Ini membantu mencegah overfitting dan memastikan bahwa model telah berlatih secara optimal.
6. Class Weight (`class\_weight\_dict`): Bobot kelas diterapkan untuk mengimbangi distribusi kelas yang tidak seimbang dalam data. Ini membantu memastikan bahwa model memperhatikan secara setara pada semua kelas, termasuk kelas minoritas.

#### **3.6.4.2 Proses Pelatihan**

Selama pelatihan, model menyesuaikan bobotnya berdasarkan loss dan akurasi dari data pelatihan dan validasi. Proses ini melibatkan optimisasi berkelanjutan di mana model berupaya meminimalkan loss dan meningkatkan akurasi. Fungsi `fit` secara otomatis mengelola forward propagation, loss computation, backpropagation, dan update bobot.

Peran dari early stopping dalam proses ini sangat penting. Callback ini mengamati loss pada set validasi dan jika model tidak menunjukkan peningkatan dalam jangka waktu yang ditetapkan (15 epochs dalam kasus ini), pelatihan akan dihentikan. Ini menghindarkan model dari penghabisan waktu pada fase pelatihan yang tidak lagi memberikan peningkatan yang signifikan.

### 3.6.4.3 Monitoring dan Evaluasi

Selama pelatihan, kinerja model dipantau berdasarkan metrik yang ditentukan selama kompilasi (dalam hal ini, 'accuracy'). History dari proses pelatihan, yang mencakup loss dan akurasi untuk setiap epoch, disimpan dalam variabel `history`. Data ini sangat berharga untuk analisis kinerja model sepanjang pelatihan dan digunakan untuk menentukan apakah model telah belajar secara efektif atau apakah ada kebutuhan untuk penyesuaian lebih lanjut dalam arsitektur atau hyperparameter.

Pelatihan model merupakan proses iteratif dan eksperimental. Pemilihan parameter seperti batch size, jumlah epochs, dan strategi early stopping adalah hasil dari eksperimen yang teliti dan evaluasi kinerja model. Eksperimen ini penting untuk menemukan keseimbangan yang tepat yang memungkinkan model untuk belajar dari data secara efisien sambil menghindari risiko overfitting atau underfitting.

## 3.7 Evaluasi Model

Evaluasi merupakan langkah penting untuk menilai kinerja model dan memahami seberapa baik model bekerja pada data yang belum pernah dilihat sebelumnya. Berbagai teknik dan metrik evaluasi digunakan untuk mengevaluasi model secara menyeluruh, termasuk visualisasi kinerja selama pelatihan, penggunaan confusion matrix, dan pengukuran performa dan keakuratan model. Evaluasi model ini menggunakan data latih yang disiapkan sebanyak 100 sampel dari setiap kelas, yaitu anger, fear, happy, love, dan sadness sehingga terdapat total 500 data pengujian.

### 3.7.1 Visualisasi Kinerja Selama Pelatihan

Dalam proses pengembangan model klasifikasi emosi, visualisasi kinerja selama pelatihan adalah langkah penting yang memberikan wawasan mendalam tentang bagaimana model belajar dan berkembang seiring waktu. Visualisasi ini

melibatkan penggunaan grafik untuk menggambarkan perubahan dalam akurasi dan loss, baik pada data pelatihan maupun validasi, sepanjang periode pelatihan.

Visualisasi kinerja memegang peranan penting dalam proses pengembangan model. Dengan menggunakan grafik akurasi dan loss, kita bisa langsung melihat seberapa efektif pelatihan itu. Pertama, kita bisa melacak bagaimana model belajar dari data seiring berjalannya waktu. Kedua, kita bisa mengidentifikasi overfitting, yang terjadi ketika model terlalu spesifik pada data pelatihan hingga performanya menurun saat dihadapkan pada data validasi. Terakhir, grafik tersebut membantu kita dalam mengevaluasi dan menyesuaikan hyperparameter pelatihan, seperti laju pembelajaran dan jumlah epochs. Visualisasi ini tidak hanya membantu dalam mengoptimalkan model, tapi juga memberikan wawasan yang berharga untuk pengembangan lebih lanjut.

#### **3.7.1.1 Visualisasi Akurasi**

Grafik akurasi menunjukkan perubahan dalam proporsi prediksi yang benar (akurasi) model sepanjang pelatihan dan validasi. Penyajian visual ini memungkinkan kita untuk memantau seberapa baik model mengklasifikasikan data dengan benar pada setiap epoch.

Ketika melihat akurasi pelatihan yang terus meningkat, ini memberitahu kita bahwa model kita belajar dengan efektif dari data pelatihan. Ini adalah tanda positif. Selanjutnya, jika akurasi validasi mengikuti tren yang sama dengan akurasi pelatihan, ini menandakan bahwa model kita tidak hanya belajar dari data pelatihan, tapi juga mampu menggeneralisasi pembelajarannya ke data baru yang belum pernah dilihat sebelumnya.

Namun, kita perlu berhati-hati jika akurasi pelatihan terus meningkat tetapi akurasi validasi stagnan atau bahkan menurun. Kondisi ini menandakan kemungkinan overfitting, di mana model terlalu spesifik pada data pelatihan dan gagal menggeneralisasi ke data lain.

### 3.7.1.2 Visualisasi Loss

Grafik loss menunjukkan bagaimana nilai kerugian (loss) model berubah selama proses pelatihan. Loss adalah ukuran seberapa jauh prediksi model menyimpang dari label sebenarnya, dengan loss yang lebih rendah menunjukkan prediksi yang lebih akurat.

Interpretasi grafik loss memberikan pandangan penting tentang bagaimana model kita mengatasi kesalahan dalam prediksinya. Jika kita melihat loss pelatihan yang terus menurun, ini menunjukkan bahwa model kita semakin efektif dalam meminimalisir kesalahan selama proses pelatihan. Ini adalah sinyal baik yang menandakan bahwa model sedang belajar dengan tepat.

Selanjutnya, jika loss validasi mengikuti pola yang sama dengan loss pelatihan, ini menunjukkan bahwa model tidak hanya berhasil memperbaiki kesalahannya pada data pelatihan, tapi juga efektif ketika dihadapkan pada data baru yang belum pernah dilihat. Ini menandakan bahwa model dapat menggeneralisasi pembelajarannya dengan baik.

Namun, kita harus waspada jika terjadi divergensi antara loss pelatihan dan validasi. Jika loss validasi meningkat atau tidak menunjukkan penurunan yang signifikan meskipun loss pelatihan menurun, ini bisa menjadi tanda overfitting. Dalam situasi seperti ini, model mungkin terlalu menyesuaikan diri dengan data pelatihan dan gagal dalam menggeneralisasi ke data lain. Memahami tren ini sangat penting untuk menghindari overfitting dan memastikan bahwa model kita dapat berkinerja baik secara umum.

### 3.7.1.3 Implementasi dan Analisis

Dalam penelitian ini, untuk memvisualisasikan kinerja model, peneliti menggunakan matplotlib, sebuah library visualisasi yang populer di Python. Proses ini dimulai dengan mengumpulkan data yang diperlukan. Data ini diambil dari objek 'history', yang merupakan hasil pengembalian metode 'fit' dari Keras. Objek 'history' ini berharga karena ia menyimpan nilai-nilai akurasi dan loss untuk setiap epoch, mencakup baik data pelatihan maupun validasi.

Grafik dibuat dengan dua subplot: satu untuk akurasi dan satu lagi untuk loss. Pada subplot akurasi, nilai `history.history['accuracy']` dan `history.history['val_accuracy']` digunakan untuk menggambar garis akurasi pelatihan dan validasi. Pada subplot loss, nilai `history.history['loss']` dan `history.history['val_loss']` digunakan untuk menggambar garis loss pelatihan dan validasi. Judul, label sumbu, dan legenda ditambahkan untuk kejelasan interpretasi. Ini membantu peneliti dan pembaca dalam memahami dan menginterpretasikan data dengan lebih mudah.

### 3.7.2 Evaluasi Model Dengan Data Test

Dalam proses evaluasi model klasifikasi emosi, data latih yang digunakan telah disiapkan dengan sangat teliti untuk memastikan representasi yang seimbang dari setiap kelas emosi. Total terdapat 500 sampel dalam set pengujian, dengan masing-masing 100 sampel untuk lima kelas emosi yang berbeda: anger (kemarahan), fear (ketakutan), happy (kebahagiaan), love (cinta), dan sadness (kesedihan). Setiap emosi memiliki representasi yang adil dalam set data, memungkinkan evaluasi yang lebih akurat dan objektif tentang bagaimana model mampu mengenali dan membedakan berbagai jenis emosi. Distribusi yang seimbang ini penting untuk menguji kemampuan model dalam menangkap nuansa dan variasi emosi yang kompleks, serta dalam menguji kekuatan dan kelemahan model dalam mengklasifikasikan emosi yang berbeda.

#### 3.7.2.1 Prediksi Data Test

Proses selanjutnya adalah membuat prediksi emosi dari ulasan. Langkah pertama dalam proses ini adalah tokenisasi teks ulasan. Tokenisasi dilakukan menggunakan tokenizer yang sama dengan yang digunakan dalam pelatihan model, mengubah teks ulasan menjadi urutan token numerik. Tokenizer ini memetakan setiap kata ke indeks tertentu dan mengubah teks menjadi format yang dapat diproses oleh model.

Urutan token kemudian di-pad untuk memastikan bahwa semua input memiliki panjang yang sama. Panjang maksimum yang diatur oleh maxlen digunakan untuk menentukan panjang sequence yang diperlukan. Padding dilakukan dengan fungsi pad\_sequences dari Keras, menambahkan nol pada urutan yang lebih pendek untuk mencapai panjang yang diinginkan.

Model yang telah dilatih sebelumnya kemudian digunakan untuk memprediksi emosi dari sequence yang telah di-pad. Model mengeluarkan prediksi dalam bentuk probabilitas untuk setiap kelas emosi. Prediksi ini kemudian diinterpretasikan dengan mengambil indeks kelas dengan probabilitas tertinggi sebagai kelas emosi yang diprediksi untuk ulasan tersebut.

### **3.7.2.2 Hasil Prediksi**

Hasil prediksi disusun dalam sebuah DataFrame yang mencakup kolom untuk ulasan asli, label emosi sebenarnya, dan emosi yang diprediksi oleh model. Struktur DataFrame ini dirancang untuk memudahkan perbandingan antara emosi yang diprediksi oleh model dan emosi sebenarnya, memberikan wawasan langsung tentang keakuratan model. Dalam DataFrame ini, setiap baris mewakili satu ulasan dengan label emosinya yang sebenarnya dan prediksi model. Confusion matrix dan akurasi dihitung berdasarkan perbandingan antara label sebenarnya dan label yang diprediksi.

### **3.7.3 Confusion Matrix**

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi performa model klasifikasi pada set data uji. Matrix ini menunjukkan hubungan antara label yang diprediksi oleh model dan label sebenarnya. Melalui confusion matrix, kita dapat mengidentifikasi kasus-kasus di mana model membuat prediksi yang benar (true positives dan true negatives) serta kasus-kasus di mana model salah dalam membuat prediksi (false positives dan false negatives).

### **3.7.3.1 True Positives (TP)**

True Positives (TP) terjadi ketika model secara benar memprediksi kelas positif. Dalam konteks klasifikasi emosi, TP merupakan kasus di mana model dengan benar mengidentifikasi ulasan dengan emosi tertentu (misalnya, kebahagiaan) sebagai emosi tersebut. Jumlah TP memberikan informasi tentang seberapa efektif model dalam mengenali dan mengklasifikasikan kasus positif secara akurat. Jumlah TP yang tinggi umumnya mengindikasikan bahwa model sangat baik dalam mengenali kelas positif.

### **3.7.3.2 True Negatives (TN)**

True Negatives (TN) adalah kasus di mana model dengan benar memprediksi kelas negatif. Dalam konteks klasifikasi emosi, ini berarti model secara akurat mengidentifikasi ulasan yang tidak mengandung emosi tertentu (misalnya, bukan kebahagiaan). TN penting untuk memahami seberapa baik model dapat mengidentifikasi ketiadaan fitur atau kelas yang diuji. TN yang tinggi menunjukkan bahwa model efektif dalam membedakan ketiadaan kelas positif.

### **3.7.3.3 False Positives (FP)**

False Positives (FP) terjadi ketika model salah memprediksi kelas negatif sebagai positif. Misalnya, model mengklasifikasikan ulasan netral atau negatif sebagai positif. Jumlah FP memberikan wawasan tentang seberapa sering model membuat kesalahan tipe I, yaitu salah mengklasifikasikan non-kejadian sebagai kejadian. FP yang tinggi dapat menunjukkan bahwa model terlalu liberal dalam memprediksi kelas positif dan mungkin memerlukan penyesuaian.

### **3.7.3.4 False Negatives (FN)**

False Negatives (FN) terjadi ketika model gagal mengidentifikasi kelas positif, salah mengklasifikasikannya sebagai negatif. Misalnya, model tidak mengenali ulasan yang sebenarnya mengandung emosi positif. Jumlah FN memberikan wawasan tentang seberapa sering model membuat kesalahan tipe II.

FN yang tinggi bisa berarti model terlalu konservatif atau gagal mengenali ciri-ciri penting dari kelas positif.

### **3.7.3.5 Implementasi Confusion Matrix**

Dalam implementasi confusion matrix untuk model klasifikasi emosi, pertama-tama, model yang telah dilatih digunakan untuk menghasilkan prediksi pada set data uji. Setiap ulasan dalam set uji diproses, dan model menghasilkan label prediksi berdasarkan pembelajaran yang telah dilakukan. Kemudian, hasil prediksi ini dibandingkan dengan label sebenarnya dari set uji untuk membentuk confusion matrix. Fungsi `'confusion_matrix'` dari library scikit-learn digunakan untuk melakukan perhitungan ini secara efisien. Matrix yang dihasilkan menggambarkan jumlah True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN) untuk setiap kelas emosi. Hasil dari confusion matrix divisualisasikan menggunakan library Matplotlib dan Seaborn agar lebih mudah untuk dianalisis. Melalui visualisasi dan analisis confusion matrix ini, kita dapat memperoleh pemahaman yang lebih dalam tentang keakuratan dan kelemahan khusus model dalam mengklasifikasikan berbagai emosi.

### **3.7.4 Evaluasi Hasil Model Keseluruhan**

Evaluasi hasil model secara keseluruhan sering kali diukur dengan akurasi. Namun, untuk pemahaman yang lebih dalam, penulis juga mempertimbangkan metrik lain seperti presisi, recall, dan skor F1. Setiap metrik ini memberikan wawasan yang berbeda tentang kinerja model.

#### **3.7.4.1 Akurasi**

Akurasi adalah ukuran paling umum yang digunakan untuk mengevaluasi model klasifikasi. Ini mengukur proporsi prediksi yang benar dibandingkan dengan total jumlah kasus. Rumus akurasi adalah:



$$Akurasi = \frac{True\ Positives\ (TP)}{Total\ Prediksi}$$

Dalam konteks model ini, akurasi akan menunjukkan seberapa sering model mengklasifikasikan emosi secara benar. Nilai akurasi yang tinggi menunjukkan bahwa model secara umum melakukan prediksi yang tepat, tetapi nilai ini mungkin tidak sepenuhnya mencerminkan performa model pada data yang tidak seimbang.

#### 3.7.4.2 Presisi

Presisi mengukur proporsi prediksi positif yang benar-benar positif. Ini penting untuk kasus di mana konsekuensi dari false positive tinggi. Rumusnya adalah:

$$Presisi = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

Presisi yang tinggi menunjukkan bahwa ketika model memprediksi kelas positif, itu cenderung benar. Nilai presisi yang rendah bisa menunjukkan banyak false positives.

#### 3.7.4.3 Recall

Recall, atau sensitivitas, mengukur seberapa baik model mengidentifikasi semua kasus positif. Ini penting di mana penting untuk menangkap semua kasus positif. Rumusnya adalah:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

Recall yang tinggi menunjukkan bahwa model efisien dalam menangkap kasus positif. Recall yang rendah menunjukkan bahwa model melewatkan sejumlah besar kasus positif.

#### 3.7.4.4 Skor F1

Skor F1 adalah rata-rata harmonis dari presisi dan recall, menawarkan keseimbangan antara kedua metrik tersebut. Ini sangat berguna dalam kasus di mana penting untuk memiliki keseimbangan antara mengurangi false positives dan menangkap semua kasus positif. Rumusnya adalah:

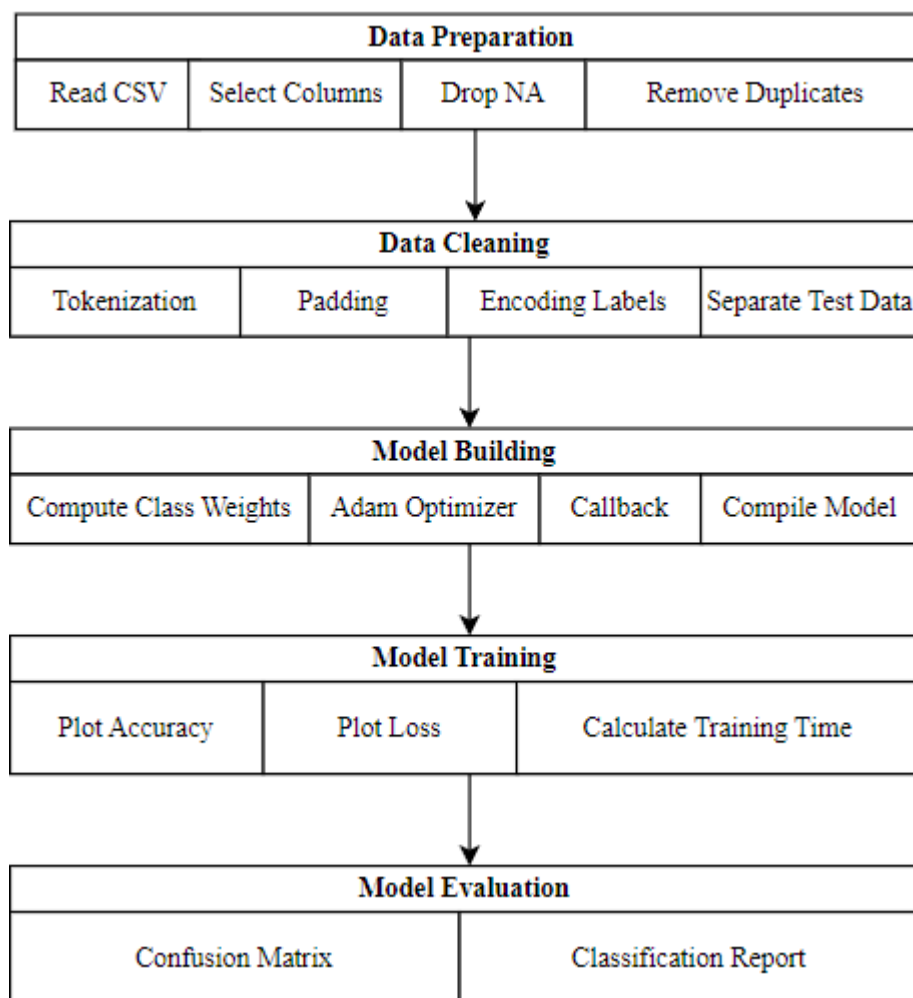
$$F1 = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Skor F1 yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Skor yang rendah mungkin menunjukkan bahwa model memiliki kelemahan pada salah satu atau kedua aspek tersebut.

## BAB IV HASIL PENELITIAN DAN PEMBAHASAN

### 4.1 Alur Kerja Model

Alur kerja model melibatkan berbagai tahapan, mulai dari persiapan data awal hingga evaluasi model, masing-masing dengan langkah-langkah sub yang penting untuk memastikan keberhasilan dan efektivitas model. Berikut merupakan gambar dari alur kerja model:



Gambar 4. 1 Alur Kerja Model

Dalam penelitian ini, berdasarkan gambar 4.1 proses dimulai dengan pengumpulan dan pembersihan data yang ekstensif, menggunakan dataset yang diperoleh dari ulasan produk di platform e-commerce Tokopedia. Data awal dimuat dari file CSV, di mana dipilih kolom yang relevan, menghilangkan baris dengan

nilai yang hilang, dan menghapus duplikat untuk memastikan integritas data. Pembersihan data melibatkan konversi teks ulasan menjadi huruf kecil dan mengganti karakter non-alfabetik dengan spasi, serta menghilangkan spasi berlebih. Kami kemudian mengekstrak subset dari data untuk dijadikan data uji, memisahkan 100 sampel acak dari setiap kategori emosi, memastikan bahwa setiap kategori diwakili secara setara dalam pengujian.

Proses persiapan data lebih lanjut melibatkan tokenisasi teks, di mana teks dipecah menjadi unit-unit yang lebih kecil yang dapat diproses oleh model. Setelah tokenisasi, urutan token dipad untuk memastikan konsistensi panjang, dan label emosi dikodekan untuk persiapan pelatihan. Data kemudian dibagi menjadi set pelatihan dan validasi, dengan 80% data untuk pelatihan dan 20% untuk validasi, menggunakan teknik splitting yang mempertimbangkan keacakan dan distribusi kelas yang seimbang.

Model klasifikasi emosi dibangun berdasarkan arsitektur Transformer yang menggunakan teknik Multi Head Attention. Selanjutnya parameter model ditentukan, seperti dimensi embedding, jumlah attention heads, dan ukuran lapisan tersembunyi dalam jaringan feed-forward. Regularisasi L2 juga diterapkan untuk mengurangi overfitting. Dengan arsitektur ini, model dikompilasi menggunakan optimizer Adam dengan learning rate yang sangat rendah untuk memfasilitasi konvergensi yang stabil.

Pelatihan model melibatkan penggunaan bobot kelas yang dihitung untuk menangani ketidakseimbangan distribusi kelas dalam data. Proses pelatihan dipantau menggunakan grafik yang memplot akurasi dan loss dari set pelatihan dan validasi, memberikan visualisasi yang jelas dari kemajuan dan kinerja model seiring waktu. Waktu pelatihan juga dipantau untuk mengevaluasi efisiensi komputasional dari model. Setelah pelatihan selesai, model dievaluasi menggunakan data uji yang sebelumnya telah dipisahkan. Prediksi model dianalisis menggunakan confusion matrix dan laporan klasifikasi yang menyediakan metrik kinerja seperti presisi, recall, dan skor F1 untuk setiap kelas.

## **4.2 Deskripsi Dataset**

Sebelum mendalami karakteristik individu dari data yang dianalisis, esensial untuk memahami gambaran besar dari dataset yang akan dijadikan pondasi penelitian ini. Sebuah dataset yang komprehensif dan representatif adalah kunci utama untuk mendapatkan insight yang akurat dan valid. Maka dari itu, perhatian pertama diarahkan pada pengenalan dataset secara menyeluruh, memastikan bahwa dasar penelitian ini telah terbangun dengan kuat dan jelas.

### **4.2.1 Karakteristik Umum Dataset**

Dataset yang digunakan dalam penelitian ini, dikumpulkan dari ulasan produk di platform e-commerce Tokopedia, merupakan representasi kaya dari beragam ekspresi emosi pelanggan terhadap berbagai produk. Dengan total 5.400 entri, dataset ini mencakup ulasan dalam Bahasa Indonesia, yang memberikan wawasan unik tentang pengalaman dan persepsi pelanggan. Setiap entri dalam dataset ini terdiri dari 11 kolom, namun fokus penelitian ini hanya pada kolom 'Customer Review' dan 'Emotion', yang masing-masing menyediakan teks ulasan dan label emosi yang terkait.

Dari perspektif konten, ulasan-ulasan ini bervariasi dalam panjang dan detail, menawarkan gambaran lengkap tentang berbagai aspek produk yang dibahas oleh pelanggan. Dengan memfokuskan pada teks ulasan dan emosi yang dikaitkan, penelitian ini mengeksplorasi hubungan antara ekspresi linguistik dan emosi yang termanifestasi, yang merupakan kunci untuk memahami dinamika interaksi pelanggan dalam lingkungan e-commerce.

### **4.2.2 Distribusi Emosi dalam Dataset**

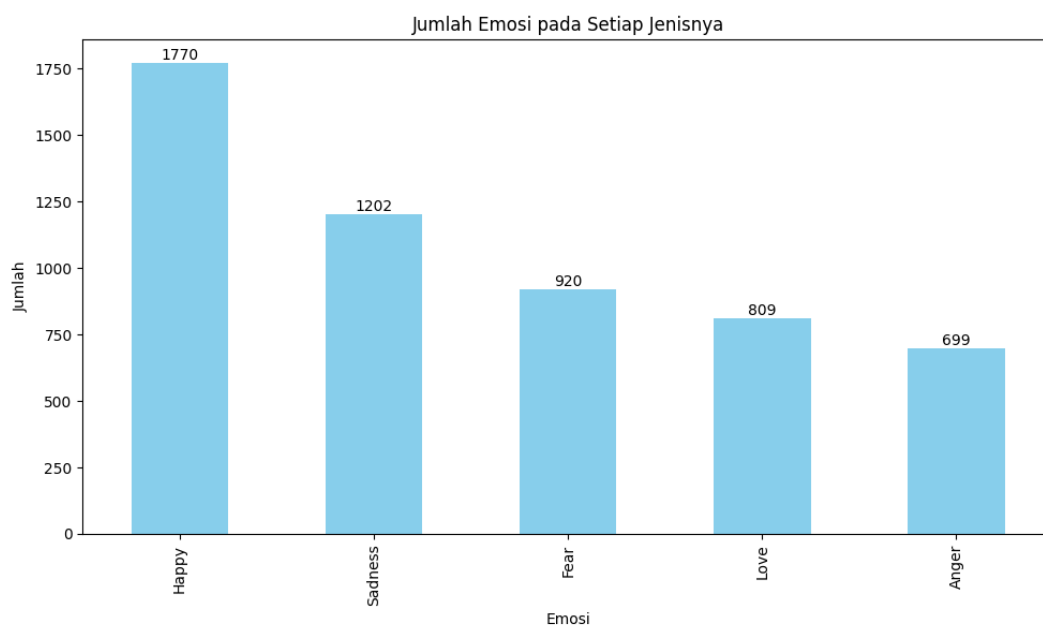
Dalam memahami dinamika interaksi pelanggan dan produk, penting untuk mengeksplorasi distribusi emosi yang diekspresikan dalam ulasan. Dataset yang digunakan dalam penelitian ini memberikan representasi yang beragam dari respons emosional pelanggan, yang tercermin dalam frekuensi kemunculan masing-masing kategori emosi.

#### 4.2.2.1 Distribusi Frekuensi Emosi

Berdasarkan analisis dataset, terdapat variasi signifikan dalam distribusi frekuensi lima kategori emosi. Dengan 'Happy' menjadi emosi yang paling sering muncul, diikuti oleh 'Sadness', 'Fear', 'Love', dan 'Anger'. Secara khusus, frekuensi kemunculan untuk 'Happy' tercatat paling tinggi, menandakan bahwa pengalaman positif dalam e-commerce cenderung lebih sering dilaporkan atau diungkapkan dalam ulasan. Sebaliknya, 'Anger' memiliki frekuensi kemunculan terendah, yang dapat mengindikasikan bahwa pelanggan mungkin lebih segan untuk menyatakan ketidakpuasan yang ekstrem atau bahwa momen-momen positif lebih banyak dibagi.

#### 4.2.2.2 Visualisasi Distribusi Emosi

Untuk memberikan perspektif visual tentang data ini, berikut sebuah grafik batang yang menunjukkan jumlah ulasan untuk setiap emosi.



Gambar 4. 2 Distribusi Emosi dalam Dataset

Grafik batang yang disajikan pada Gambar 4.1 menawarkan perspektif visual yang penting mengenai distribusi emosi dalam dataset yang digunakan. Visualisasi ini tidak hanya mengkomunikasikan data secara intuitif tetapi juga

mengungkap pola distribusi yang penting dalam konteks penelitian pengenalan emosi.

1. Happy: Kategori 'Happy' menunjukkan jumlah yang dominan dalam dataset dengan total 1770 ulasan yang mencerminkan emosi senang. Kategori ini mencakup ulasan yang mengekspresikan kepuasan, kesenangan, atau kegembiraan pelanggan terhadap produk atau layanan yang mereka terima. Dominasi emosi senang dapat mencerminkan kecenderungan pelanggan untuk berbagi pengalaman positif sebagai bentuk apresiasi atau rekomendasi kepada pengguna lain. Ini juga bisa menandakan bahwa pelanggan yang puas lebih termotivasi untuk meninggalkan ulasan dibandingkan dengan emosi lainnya.
2. Sadness: 'Sadness' sebagai emosi kedua terbanyak dengan jumlah 1202 ulasan, memperlihatkan bahwa pelanggan juga aktif menyampaikan kekecewaan mereka. Kategori ini biasanya mencakup ulasan yang berhubungan dengan harapan yang tidak terpenuhi atau pengalaman negatif dengan produk atau layanan. Frekuensi yang signifikan dari ulasan sedih ini mengindikasikan pentingnya memperhatikan feedback negatif sebagai sumber informasi yang berharga untuk perbaikan produk dan layanan.
3. Fear: Dengan 920 ulasan, emosi 'Fear' tergambar lebih sedikit dibandingkan dengan 'Happy' dan 'Sadness', namun masih merupakan bagian penting dari dataset. Ulasan yang mengungkapkan ketakutan atau kekhawatiran dapat berhubungan dengan kecemasan atas kualitas produk, potensi risiko, atau ketidakpastian dalam transaksi e-commerce. Kehadiran emosi ini menekankan pada pentingnya menjaga kepercayaan pelanggan dengan memastikan transparansi dan keamanan dalam proses pembelian.
4. Love: Emosi 'Love' dengan total 809 ulasan mengindikasikan bahwa pelanggan merasa sangat terhubung dan memiliki kesetiaan terhadap merek atau produk. Ulasan dengan emosi cinta sering kali lebih dari sekadar kepuasan; pelanggan mungkin merasa bahwa produk telah sangat memenuhi kebutuhan atau melebihi ekspektasi mereka sehingga memicu

ekspresi afektif yang kuat. Frekuensi yang relatif tinggi dari ulasan positif ini juga bisa menandakan bahwa pelanggan yang memiliki hubungan emosional dengan sebuah merek cenderung lebih termotivasi untuk memberikan ulasan. Kehadiran emosi cinta dalam dataset ini adalah indikator penting bahwa pengalaman pelanggan yang sangat positif memiliki potensi untuk dibagi dan dapat mempengaruhi pembeli potensial lainnya.

5. Anger: 'Anger', dengan jumlah 699 ulasan, merupakan emosi dengan frekuensi terendah dalam dataset ini. Meskipun jumlahnya lebih kecil dibandingkan kategori emosi lainnya, keberadaan ulasan yang mengungkapkan kemarahan tidak bisa diabaikan. Ulasan yang mengandung kemarahan seringkali menunjukkan tingkat kekecewaan yang tinggi dari pelanggan terhadap aspek tertentu dari produk atau layanan, seperti kualitas yang buruk, ketidaksesuaian dengan deskripsi, atau layanan pelanggan yang tidak memuaskan. Jumlah yang lebih rendah dari ulasan marah ini bisa mengindikasikan bahwa pelanggan mungkin lebih memilih untuk langsung menghubungi penyedia layanan untuk menyelesaikan masalah daripada meninggalkan ulasan negatif secara publik. Di sisi lain, ini juga bisa mencerminkan keengganan pelanggan untuk secara terbuka mengekspresikan ketidakpuasan ekstrem dalam forum publik seperti ulasan online.

#### **4.2.2.3 Implikasi Distribusi Emosi Terhadap Model**

Distribusi emosi yang diungkapkan dalam dataset ini memiliki implikasi penting terhadap proses pelatihan dan validasi model yang digunakan dalam penelitian ini. Implikasinya adalah sebagai berikut:

1. Keseimbangan Data: Dengan 'Happy' sebagai emosi yang paling dominan dan 'Anger' sebagai yang paling jarang, terdapat ketidakseimbangan kelas yang dapat mempengaruhi kinerja model. Model yang dilatih pada data tidak seimbang cenderung memiliki bias terhadap kelas mayoritas dan mungkin tidak mampu mengenali kelas minoritas dengan efektif.



2. **Pengaturan Bobot Kelas:** Untuk mengatasi masalah ketidakseimbangan kelas, teknik seperti penyesuaian bobot kelas atau resampling data mungkin perlu digunakan. Hal ini akan memastikan bahwa model tidak hanya memprioritaskan kelas mayoritas dan meningkatkan kemampuannya untuk mendeteksi emosi yang kurang umum.
3. **Validasi Model:** Saat mengevaluasi model, penting untuk menggunakan metrik yang mempertimbangkan ketidakseimbangan kelas, seperti skor F1, presisi, dan recall, bukan hanya akurasi. Metrik-metrik tersebut memberikan gambaran yang lebih lengkap tentang bagaimana model bekerja pada semua kelas emosi.
4. **Generalisasi Model:** Distribusi yang tidak seimbang dalam dataset pelatihan mungkin tidak mencerminkan distribusi emosi dalam data dunia nyata. Oleh karena itu, validasi model pada dataset yang terpisah dengan distribusi emosi yang berbeda akan penting untuk menguji generalisasi model.
5. **Analisis Kesalahan:** Mengingat pentingnya distribusi emosi yang seimbang, analisis kesalahan yang mendalam pada prediksi model untuk kelas emosi yang kurang terwakili akan sangat penting. Analisis ini dapat membantu mengidentifikasi pola kesalahan dan mengarahkan perbaikan pada model atau proses pengumpulan data di masa mendatang.

Dalam konteks penelitian ini, memahami bagaimana distribusi emosi dalam dataset dapat mempengaruhi pembelajaran dan kinerja model adalah kunci untuk mengembangkan sistem pengenalan emosi yang akurat dan dapat diandalkan. Penyesuaian terhadap model berdasarkan distribusi ini akan memungkinkan peneliti untuk mengembangkan solusi yang lebih robust yang dapat memproses berbagai ekspresi emosi dengan lebih adil dan efisien.

#### **4.2.3 Frekuensi Kata dalam Dataset**

Analisis frekuensi kata dalam konteks emosi pada ulasan pelanggan e-commerce memberikan wawasan tentang ekspresi emosional dan terminologi yang sering digunakan oleh konsumen. Berikut merupakan kata-kata kunci yang

dominan dalam masing-masing kategori emosi, yakni Happy, Sadness, Anger, Love, dan Fear.

Tabel 4. 1 Frekuensi Kata dalam Dataset

Emosi	10 Kata Teratas
Happy	dan, sesuai, cepat, bagus, barang, pengiriman, sangat, harga, beli, kualitas
Sadness	di, tidak, yg, kecewa, barang, dan, saya, nya, yang, ini
Anger	di, yg, barang, tidak, ga, nya, saya, ada, dan, yang
Love	dan, banget, bagus, cepat, seller, sesuai, barang, sangat, pengiriman, terima kasih
Fear	di, tidak, yg, barang, dan, saya, ada, ga, nya, yang

Berikut merupakan pembahasan dari Tabel 4.1 Frekuensi Kata dalam Dataset:

1. Emosi Happy: Kata-kata seperti "dan", "sesuai", "cepat", "bagus", dan "barang" mendominasi. Ini menunjukkan bahwa pelanggan cenderung menggunakan kata-kata yang menggambarkan kepuasan dan kecepatan layanan saat mereka merasa senang. Kata "sesuai" mungkin menunjukkan kepuasan pelanggan ketika ekspektasi mereka terpenuhi.
2. Emosi Sadness: Kata-kata seperti "tidak", "kecewa", "barang", dan "saya" sering muncul, menandakan ekspresi ketidakpuasan dan kekecewaan. Frekuensi tinggi kata "tidak" dan "kecewa" menunjukkan bahwa ulasan negatif sering kali terfokus pada ketidaksesuaian produk atau layanan dengan ekspektasi pelanggan.
3. Emosi Anger: Kata-kata seperti "tidak", "barang", "yg", dan "saya" menunjukkan ketidakpuasan dan frustrasi. Kemunculan kata "tidak" yang sering mungkin mengindikasikan penolakan atau ketidaksetujuan pelanggan terhadap aspek tertentu dari produk atau layanan.
4. Love: Kata "banget", "bagus", "dan", "cepat" menonjol, mengindikasikan apresiasi tinggi dan kepuasan mendalam. Pemilihan kata "banget" dan "bagus" secara konsisten menunjukkan kekaguman atau kecintaan yang

kuat terhadap produk atau layanan.

5. Fear: Kata-kata seperti "tidak", "barang", "yg", menandakan kekhawatiran atau kecemasan. Frekuensi tinggi kata "tidak" dapat berhubungan dengan ekspresi ketakutan atau kekhawatiran tentang ketidakpastian atau kemungkinan hasil negatif dari produk atau layanan.

#### 4.2.4 Contoh Ulasan

Sebagai representasi konkret dari data yang dianalisis, beberapa ulasan khusus dipilih untuk masing-masing kategori emosi. Ulasan ini dipilih secara acak untuk memastikan keaslian dan keberagaman sampel, memberikan gambaran yang nyata tentang bagaimana emosi diekspresikan oleh pelanggan dalam konteks e-commerce.

##### 4.2.4.1 Tabel Ulasan Contoh

Berikut ini adalah tabel yang menunjukkan ulasan pelanggan untuk setiap kategori emosi:

Tabel 4. 2 Contoh Ulasan

Emosi	Ulasan Contoh
Happy	“bahannya adem banget ini yang saya butuhkan...”
Sadness	“Niat mau main di televisi tapi sama sekali gak bisa digunakan”
Anger	“Buku palsu”
Love	“Selamat Siang Bos, pesanan sudah saya terima...”
Fear	“barang tidak sesuai dengan jumlah pesanan”

Berikut merupakan pembahasan dari Tabel 4.2 Contoh Ulasan:

1. Happy: Ulasan yang ditandai dengan kebahagiaan sering kali mencerminkan kepuasan terhadap kualitas produk, seperti dalam contoh di atas, di mana pelanggan merasa puas dengan bahan produk yang "adem banget".

2. Sadness: Kesedihan dalam ulasan biasanya berkaitan dengan kekecewaan, seperti dalam contoh yang menunjukkan ketidakpuasan terhadap produk yang "sama sekali gak bisa digunakan".
3. Anger: Emosi kemarahan biasanya terkait dengan respons negatif yang kuat terhadap produk yang tidak memenuhi ekspektasi atau dianggap menyesatkan, seperti yang terlihat dari ulasan singkat namun kuat: "Buku palsu".
4. Love: Ekspresi cinta sering kali merupakan indikasi dari kepuasan yang tinggi dan koneksi emosional dengan produk, seperti yang terlihat dari ucapan terima kasih dan penghargaan dalam ulasan "Selamat Siang Bos, pesanan sudah saya terima...".
5. Fear: Rasa takut atau kekhawatiran dalam ulasan biasanya menyoroti ketidakpastian atau potensi masalah, seperti yang diungkapkan dalam kekhawatiran bahwa "barang tidak sesuai dengan jumlah pesanan".

Ulasan ini memberikan wawasan yang berharga ke dalam kompleksitas emosi manusia dan bagaimana mereka diungkapkan dalam konteks ulasan e-commerce. Menganalisis ulasan ini tidak hanya penting untuk mengembangkan model deep learning yang efektif tetapi juga untuk memahami lebih dalam tentang perilaku dan harapan pelanggan. Ini, pada gilirannya, bisa sangat berharga dalam membantu bisnis untuk meningkatkan produk dan layanan mereka, serta dalam membangun hubungan pelanggan yang lebih kuat.

#### **4.3 Pra-Pemrosesan dan Persiapan Data**

Memasuki ranah pra-pemrosesan dan persiapan data, merupakan langkah yang tidak kalah pentingnya dalam penelitian NLP. Proses ini tidak hanya melibatkan penghilangan variabilitas yang tidak perlu tetapi juga peningkatan kualitas data yang akan diolah. Pra-pemrosesan dan persiapan data yang teliti dan terkontrol ini adalah dasar yang memungkinkan algoritma pembelajaran mesin untuk bekerja dengan efektivitas maksimal.

### 4.3.1 Pembagian Dataset untuk Pengujian dan Pelatihan

Dalam penelitian yang melibatkan pengenalan pola atau klasifikasi dengan pembelajaran mesin, seperti studi ini yang berfokus pada pengenalan emosi dari ulasan pelanggan e-commerce, pendekatan standar sebelum memulai pelatihan model adalah dengan membagi dataset secara strategis. Tujuan utama dari pembagian ini adalah untuk mengevaluasi model dengan cara yang objektif, memastikan bahwa model yang dilatih dapat berperforma baik pada data yang belum pernah dilihat sebelumnya, dan bukan hanya pada data yang telah digunakan selama fase pelatihan.

#### 4.3.1.1 Proses Pembagian Dataset

```
test_data = dataset.groupby('Emotion').apply(lambda x: x.sample(n=100,
random_state=42)).reset_index(drop=True)

dataset = dataset.drop(test_data.index)
```

Gambar 4. 3 Kode Untuk Membagi Dataset

Pembagian dataset ini dilakukan melalui dua tahap utama. Tahap pertama adalah ekstraksi 'testing set', yang akan digunakan nantinya untuk mengevaluasi performa akhir dari model. 'Testing set' ini harus cukup representatif dan mencakup semua kategori emosi yang ada dalam dataset. Berdasarkan kode yang diberikan, proses pembagian ini dilakukan sebagai berikut:

1. Stratifikasi Data Pengujian: Dataset dikelompokkan berdasarkan label 'Emotion', dan dari setiap kelompok, sejumlah sampel diambil secara acak, dalam hal ini 100 sampel dari setiap kategori emosi, untuk membentuk 'testing set'. Stratifikasi ini memastikan bahwa semua kategori emosi terwakili secara proporsional, memungkinkan evaluasi yang lebih seimbang dan akurat dari model.
2. Pembuatan Dataset Pelatihan: Setelah 'testing set' dikeluarkan, sisa dataset yang berjumlah 4900 baris dijadikan 'training set'. Langkah ini penting untuk memisahkan data yang akan digunakan dalam proses pelatihan dan validasi dari data yang akan digunakan dalam pengujian akhir model.

Memisahkan 'testing set' sebelum melakukan pra-pemrosesan dan pelatihan lebih lanjut menjamin bahwa model akan dievaluasi pada data yang benar-benar belum pernah dilihat sebelumnya, menghindari kontaminasi informasi yang dapat memberi keuntungan tidak adil kepada model. Proses ini, yang diilustrasikan dalam kode yang disediakan, menggunakan fungsi 'groupby' dan 'apply' dari library pandas untuk mengorganisir dan sampel data. Penggunaan 'lambda' function memungkinkan pemilihan sampel yang tepat dan fleksibel dari setiap grup. Parameter 'random\_state=42' menjamin bahwa proses sampling ini konsisten setiap kali kode dijalankan, yang memastikan bahwa hasil pembagian data ini dapat direplikasi dan divalidasi oleh peneliti lain atau dalam eksperimen ulang.

#### 4.3.2 Pra-Pemrosesan Data

Sebelum data dapat digunakan untuk pelatihan model deep learning, penting untuk memastikan bahwa data tersebut bersih dan konsisten. Pra-pemrosesan adalah langkah krusial dalam pipeline pembelajaran mesin, yang bertujuan untuk mengkonversi teks mentah menjadi format yang lebih terstruktur dan seragam. Proses ini mencakup beberapa langkah penting yang dirancang untuk menyederhanakan kompleksitas teks dan mengurangi variasi dalam dataset.

```
# 1. Mengubah ke huruf kecil
data['Customer Review'] = data['Customer Review'].str.lower()

# 2. Mengganti karakter yang bukan huruf dengan spasi
data['Customer Review'] = data['Customer Review'].str.replace(r'^a-z\s', ' ', regex=True)

# 3. Menghilangkan spasi berlebih
data['Customer Review'] = data['Customer Review'].apply(lambda x: re.sub(r'\s+', ' ', x).strip())
```

Gambar 4. 4 Kode Untuk Melakukan Pemrosesan Data

##### 4.3.2.1 Langkah-Langkah Pra-Pemrosesan

Proses pra-pemrosesan yang dilakukan pada dataset ulasan pelanggan meliputi:

1. Seleksi Kolom: Sebagai langkah awal, dataset yang kompleks dengan banyak kolom disederhanakan menjadi dua kolom yang relevan yaitu

'Customer Review' dan 'Emotion'. Kolom ini dipilih karena langsung berkaitan dengan tujuan penelitian, yaitu untuk menganalisis teks ulasan dan mengidentifikasi emosi yang terkait.

2. Pembersihan Awal: Langkah penting selanjutnya adalah pembersihan awal data. Pada tahap ini, data yang tidak lengkap (null values) dihapus menggunakan metode ``dropna()``. Data yang redundan atau duplikat juga dihilangkan dengan ``drop_duplicates()``, meninggalkan dataset yang lebih bersih dan akurat untuk analisis.
3. Normalisasi Teks: Untuk memastikan konsistensi dalam teks, semua ulasan dikonversi ke huruf kecil menggunakan metode ``str.lower()``. Normalisasi ini penting karena mengurangi kompleksitas model dengan menghilangkan perbedaan antara huruf besar dan kecil yang tidak mempengaruhi makna teks.
4. Pembersihan Karakter Non-Alfabetik: Karakter yang bukan huruf, seperti angka dan simbol, dianggap tidak relevan untuk analisis emosi dan oleh karena itu diganti dengan spasi. Hal ini dilakukan dengan metode ``str.replace()`` dan ekspresi reguler ``[^a-z\s]``, yang mencari karakter non-alfabetik dan menggantinya.
5. Penghilangan Spasi Berlebih: Teks ulasan sering kali mengandung spasi ekstra yang tidak perlu, yang jika tidak dihilangkan, dapat menyebabkan inkonsistensi dalam pemrosesan data lebih lanjut. Oleh karena itu, penghilangan spasi berlebih dilakukan dengan menerapkan fungsi lambda yang memanfaatkan ``re.sub()`` untuk menggantikan satu atau lebih spasi (``\s+``) dengan spasi tunggal dan kemudian memangkas spasi di awal dan akhir string.

Langkah-langkah tersebut diimplementasikan melalui kode python pada gambar 4.3, menghasilkan dataset yang telah dinormalisasi dan dibersihkan, siap untuk langkah selanjutnya dalam pipeline pembelajaran mesin. Dengan melakukan pra-pemrosesan ini, kita mengurangi beban komputasi yang tidak perlu pada model dan meningkatkan akurasi pembelajaran dengan fokus pada ciri-ciri teks yang paling relevan untuk analisis emosi.

#### 4.3.2.2 Tabel Contoh Pra-Pemrosesan

Berikut ini adalah tabel yang menunjukkan perbandingan teks ulasan sebelum dan setelah langkah-langkah pra-pemrosesan:

Tabel 4. 3 Contoh Pra-Pemrosesan Data

Sebelum Pra-Pemrosesan	Setelah Pra-Pemrosesan
Bulu mata jadi lentik. Packaging rapi. Packaging aman ???. Proses pesanan cepat. Aplikator mudah dipakai.	bulu mata jadi lentik packaging rapi packaging aman proses pesanan cepat aplikator mudah dipakai
Material Produk Bagus. Ukuran Produk Sesuai. Warna Produk Sesuai.tapi sayang karetnya kurang baik krn masih bocor saat test simpan air	material produk bagus ukuran produk sesuai warna produk sesuai tapi sayang karetnya kurang baik krn masih bocor saat test simpan air
suka banget sama strap maskernya???. lucu + bagus banget. recommended bgt. bakalan jdi langganan sih	suka banget sama strap maskernya lucu bagus banget recommended bgt bakalan jdi langganan sih
Ternyata tidak sesuai. Tidak dpat dipakai	ternyata tidak sesuai tidak dpat dipakai
pengemasan lambat banget, pengiriman lamaaaaaa!!!	pengemasan lambat banget pengiriman lamaaaaaa

Berikut merupakan penjelasan dari tabel 4.3 Contoh Pra-Pemrosesan Data:

1. Sebelum Pra-Pemrosesan: Teks ulasan asli menampilkan berbagai bentuk kapitalisasi dan penggunaan karakter non-alfabetik, yang menciptakan variasi yang tidak diperlukan dalam data.
2. Setelah Pra-Pemrosesan: Teks yang telah dibersihkan menunjukkan konsistensi dalam penggunaan huruf kecil dan eliminasi karakter non-alfabetik. Penghilangan spasi berlebih juga memastikan bahwa teks lebih rapi dan siap untuk proses analisis selanjutnya.



#### 4.3.2.3 Implikasi Pra-Pemrosesan

Pra-pemrosesan data adalah fondasi yang menentukan keberhasilan model pembelajaran mesin dalam Natural Language Processing (NLP). Proses ini membawa implikasi yang mendalam terhadap efektivitas model dalam mengenali dan menginterpretasikan emosi dari teks. Melalui normalisasi teks, yaitu mengkonversi semua karakter ke bentuk huruf kecil, kita menghilangkan salah satu sumber variabilitas terbesar dalam data teks. Ini memastikan bahwa model tidak membedakan antara varian yang sama dari sebuah kata yang mungkin muncul di awal kalimat (dan oleh karenanya ditulis dengan huruf kapital) dengan varian yang muncul di tengah kalimat. Dengan demikian, model dapat fokus pada pembelajaran pola yang relevan tanpa terganggu oleh perbedaan yang secara linguistik tidak signifikan.

Selanjutnya, menghilangkan karakter non-alfabetik dan spasi ekstra membantu dalam mengurangi 'noise' dalam data. Karakter seperti tanda baca dan angka seringkali tidak memberikan kontribusi yang signifikan terhadap analisis emosi dalam konteks ini. Oleh karena itu, penghapusan karakter-karakter tersebut mengarah pada dataset yang lebih 'bersih', yang memudahkan model untuk mengidentifikasi fitur penting yang berkaitan dengan emosi. Langkah ini juga menghindarkan model dari pembelajaran pola yang mungkin mengakibatkan overfitting, yaitu model menjadi terlalu spesifik pada dataset pelatihan dan kehilangan kemampuannya untuk menggeneralisasi pada data baru.

Proses pra-pemrosesan juga memberikan dampak pada efisiensi pelatihan. Data yang telah dibersihkan memungkinkan model untuk melakukan operasi pada representasi teks yang lebih padat dan terfokus. Ini berarti bahwa waktu pelatihan dapat diminimalkan, memungkinkan penggunaan sumber daya komputasi yang lebih efisien, yang merupakan pertimbangan kritis, terutama ketika bekerja dengan dataset besar.

### 4.3.3 Persiapan Data

Persiapan data adalah tahapan penting yang memfasilitasi transisi dari data mentah ke format yang dapat diolah oleh model pembelajaran mesin. Langkah ini harus dilakukan dengan teliti untuk memastikan bahwa input yang diberikan ke model mampu merefleksikan informasi penting dari teks asli sambil menghilangkan aspek yang tidak penting atau berpotensi mengaburkan pembelajaran yang efektif. Proses ini melibatkan konversi teks ulasan menjadi sekuens angka yang kemudian di-padding untuk memastikan bahwa semua input memiliki panjang yang sama sebelum dilatih oleh model.

```

texts = data['Customer Review'].astype(str).tolist()
labels = data['Emotion'].tolist()

# Parameters
vocab_size = 7000
maxlen = 40

# Tokenize the text
tokenizer = Tokenizer(num_words=vocab_size)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

# Pad the sequences
x_data = pad_sequences(sequences, maxlen=maxlen, padding='post')

# Encode labels
labels = data['Emotion'].tolist()
label_encoder = LabelEncoder()
y_data = label_encoder.fit_transform(labels)
y_data = y_data.reshape(-1, 1)

# Split the dataset into training and validation sets
x_train, x_val, y_train, y_val = train_test_split(x_data,
                                                    y_data,
                                                    test_size=0.2,
                                                    random_state=42)

```

Gambar 4. 5 Kode Untuk Persiapan Data

#### 4.3.3.1 Konversi dan Tokenisasi Teks

Proses ini dimulai dengan ekstraksi teks ulasan dan label emosi dari dataset. Konversi ini mengubah kolom 'Customer Review' menjadi list string, yang merupakan format dasar yang diperlukan untuk tahap pemrosesan selanjutnya. Sejalan dengan ini, label emosi juga dikonversi menjadi list untuk memudahkan akses dan manipulasi selanjutnya. Kedua konversi ini merupakan langkah awal yang memungkinkan untuk mengoperasikan data teks dan label dengan lebih mudah dalam pipeline pemrosesan NLP.

#### 4.3.3.2 Tokenisasi Teks

Setelah konversi, langkah berikutnya adalah tokenisasi. Tokenisasi adalah langkah di mana teks dipecah menjadi potongan-potongan yang lebih kecil, atau "token". Dalam konteks ini, tokenizer dari Keras diinisialisasi dengan ukuran vocabular tertentu (`'vocab_size=7000'`), yang berarti bahwa tokenizer akan mempertimbangkan 7000 kata paling umum dari dataset. Setiap kata yang unik diberi indeks yang berurutan dari 1 hingga 7000, berdasarkan frekuensi kemunculannya dalam dataset. Kata-kata yang tidak termasuk dalam 7000 kata tersebut akan diabaikan. Pemilihan jumlah kata ini didasarkan pada trade-off antara diversitas linguistik dan kompleksitas model; dengan membatasi vokabular, model dapat dilatih lebih efisien tanpa kehilangan kemampuan untuk memahami konteks yang paling umum dari ulasan.

Tokenizer kemudian diaplikasikan pada teks ulasan untuk menghasilkan sekuens angka, di mana setiap angka merepresentasikan token tertentu. Langkah ini mengubah ulasan teks menjadi data yang bisa diproses oleh model neural network. Sebagai contoh teks ulasan "Pelayanan sangat memuaskan!" setelah tokenisasi teks tersebut mungkin diwakili sebagai [23, 48, 120], di mana setiap angka mewakili indeks kata yang sesuai dalam vocabular model. Sekuens yang dihasilkan tersebut merupakan representasi numerik dari teks, yang esensial untuk proses pembelajaran yang akan datang.

#### 4.3.3.3 Padding Sekuens

Setelah tokenisasi, penting untuk memastikan bahwa semua sekuens memiliki panjang yang sama agar dapat diproses dalam batch oleh model. Langkah ini disebut padding. Dalam konteks ini, `'maxlen=40'` menunjukkan bahwa semua sekuens akan di-pad atau dipotong sehingga panjangnya tepat 40 token. Jika sekuens lebih pendek dari 40 token, padding akan ditambahkan di akhir (disebabkan oleh `'padding='post'`) dalam bentuk token nol (`'0'`) hingga mencapai panjang yang diperlukan. Untuk sekuens yang lebih panjang, pemotongan akan dilakukan untuk mengurangnya menjadi 40 token. Menggunakan contoh sebelumnya, setelah padding, representasi numerik dari ulasan akan menjadi [23, 48, 120, 0, 0, ..., 0], dengan 37 nol ditambahkan ke akhir sekuens untuk mencapai panjang 40 token.

Penambahan 'post' dipilih karena, dalam banyak kasus, informasi penting dalam teks biasanya terletak di awal, sehingga padding di akhir sekuens cenderung mengurangi potensi gangguan terhadap konteks kalimat yang relevan. Ini mengoptimalkan model untuk fokus pada informasi utama tanpa terganggu oleh nilai padding selama proses pelatihan.

#### 4.3.3.4 Enkoding Label

Selain mempersiapkan teks, label emosi juga harus di-encode sebelum digunakan dalam pelatihan. Dalam kasus ini, `'LabelEncoder'` dari `sklearn` digunakan untuk mengonversi label emosi yang berupa string menjadi angka yang sesuai. Ini penting karena model machine learning bekerja dengan angka dan memerlukan representasi numerik dari label kategori. `'LabelEncoder'` secara efektif memetakan setiap label emosi unik ke sebuah nilai integer. Misalnya, jika kita memiliki lima emosi: 'anger', 'fear', 'happy', 'love', dan 'sadness', `'LabelEncoder'` mungkin akan memetakan mereka ke angka 0 hingga 4 secara berurutan.

Setelah proses encoding, label-label ini kemudian diubah bentuknya untuk memenuhi persyaratan format target yang diharapkan oleh model keras saat pelatihan. Ini dilakukan melalui metode `'.reshape(-1, 1)'`, yang mengubah array

label menjadi matriks dengan satu kolom, di mana setiap baris sesuai dengan satu sampel pelatihan. Label diubah bentuknya agar sesuai dengan kebutuhan output model yang akan menggunakan `sparse_categorical_crossentropy` sebagai fungsi loss.

#### 4.3.3.5 Pembagian Data

Dengan teks yang telah di-tokenisasi dan di-pad, serta label emosi yang telah di-encode, langkah selanjutnya adalah membagi dataset menjadi set pelatihan dan validasi. Pembagian ini sangat penting untuk memvalidasi kinerja model dan untuk mencegah overfitting. Dalam kode yang diberikan, `train_test_split` digunakan dengan parameter `test_size=0.2` dan `random_state=42`. Ini menentukan bahwa 20% dari data akan digunakan sebagai set validasi sementara sisanya akan menjadi set pelatihan. `random_state=42` adalah seed untuk generator angka acak yang menjamin bahwa hasil split sama setiap kali kode dijalankan, yang penting untuk reproduktibilitas eksperimen.

Fungsi `train_test_split` juga menerima parameter `random_state=42` yang memastikan bahwa pembagian ini konsisten di setiap eksekusi. Ini membantu dalam keperluan reproduktivitas eksperimen, memungkinkan peneliti lain untuk mengulangi proses pelatihan dengan set data yang identik untuk validasi hasil.

Berikut merupakan tabel jumlah data dari hasil pra-pemrosesan data dan persiapan data yang telah dilakukan:

Tabel 4. 4 Jumlah Data Untuk Pelatihan dan Pengujian

<b>Tipe Data</b>	<b>Jumlah Data</b>	<b>Deskripsi</b>
Total Data Awal	5400	Jumlah total sampel dalam dataset sebelum pra-pemrosesan data dan persiapan data.
Pelatihan	3876	80% dari dataset yang disiapkan untuk pelatihan model.
Validasi	969	20% dari dataset yang disiapkan untuk validasi model.
Pengujian	500	Sampel yang dipisahkan dari awal untuk pengujian model.

Dengan langkah-langkah persiapan data yang telah dilakukan dengan teliti ini, dataset sekarang siap untuk digunakan dalam pelatihan model.

#### **4.4 Pembangunan dan Pelatihan Model Transformer**

Dalam ranah pengenalan emosi dari ulasan pelanggan e-commerce, arsitektur Transformer memegang peranan penting dalam memproses dan memahami bahasa alami. Model ini dibangun dengan prinsip-prinsip yang menggabungkan pemahaman kontekstual yang mendalam dengan kemampuan untuk menangani urutan kata yang panjang, yang keduanya adalah kunci dalam menangkap nuansa emosional yang kaya dalam teks ulasan. Proses pembangunan dan pelatihan model ini dirancang untuk tidak hanya menggali makna langsung dari kata-kata yang digunakan dalam ulasan tetapi juga untuk menginterpretasikan konteks di mana kata-kata tersebut disampaikan, yang secara signifikan dapat mempengaruhi emosi yang terkandung dalam ulasan tersebut.

##### **4.4.1 Arsitektur Model Transformer**

Pengenalan emosi dari ulasan pelanggan e-commerce melalui Deep Learning memerlukan pemahaman yang mendalam tentang teks berbasis urutan.

Model Transformer, yang merupakan fokus dari penelitian ini, mewakili evolusi terbaru dalam pemrosesan bahasa alami (NLP) untuk menginterpretasikan dan memproses informasi teks dengan efisiensi yang tinggi.

Model Transformer yang dirancang dalam penelitian ini terdiri dari beberapa blok dan lapisan yang saling terintegrasi untuk mengolah dan memahami data input. Komponen-komponen ini dirancang dengan tujuan untuk mengidentifikasi dan mengklasifikasikan emosi yang terkandung dalam ulasan dengan presisi yang tinggi.

```
# Define the Transformer Block
class TransformerBlock(layers.Layer):
    def __init__(self, embed_dim, num_heads, ff_dim, rate=0.1):
        super(TransformerBlock, self).__init__()
        self.att = layers.MultiHeadAttention(num_heads=num_heads,
                                              key_dim=embed_dim)

        self.ffn = keras.Sequential(
            [layers.Dense(ff_dim, activation="relu"), layers.Dense(embed_dim),]
        )
        self.layernorm1 = layers.LayerNormalization(epsilon=1e-6)
        self.layernorm2 = layers.LayerNormalization(epsilon=1e-6)
        self.dropout1 = layers.Dropout(rate)
        self.dropout2 = layers.Dropout(rate)

    def call(self, inputs, training):
        attn_output = self.att(inputs, inputs)
        attn_output = self.dropout1(attn_output, training=training)
        out1 = self.layernorm1(inputs + attn_output)
        ffn_output = self.ffn(out1)
        ffn_output = self.dropout2(ffn_output, training=training)
        return self.layernorm2(out1 + ffn_output)

# Define the Token and Position Embedding Layer
class TokenAndPositionEmbedding(layers.Layer):
    def __init__(self, maxlen, vocab_size, embed_dim):
        super(TokenAndPositionEmbedding, self).__init__()
        self.token_emb = layers.Embedding(input_dim=vocab_size,
                                           output_dim=embed_dim)

        self.pos_emb = layers.Embedding(input_dim=maxlen,
                                         output_dim=embed_dim)

    def call(self, x):
        maxlen = tf.shape(x)[-1]
        positions = tf.range(start=0, limit=maxlen, delta=1)
        positions = self.pos_emb(positions)
        x = self.token_emb(x)
        return x + positions
```

Gambar 4. 6 Arsitektur Model Transformer

#### 4.4.1.1 Implementasi Transformer Block

`TransformerBlock` merupakan fondasi arsitektur ini, mewakili kelas yang dibangun di atas `layers.Layer` dari Keras. Blok ini dikonfigurasi untuk menangani berbagai aspek dari masukan teks dengan cara yang kompleks. Konstruktor kelas `TransformerBlock` diinisialisasi dengan empat parameter utama yang menentukan karakteristik inti dari blok ini:

1. `embed_dim`: Parameter ini menetapkan ukuran dimensi embedding untuk setiap token. Dalam konteks ini, `embed\_dim` diatur ke nilai spesifik yang akan menentukan ukuran vektor embedding yang digunakan dalam model. Ini adalah ukuran dari ruang fitur di mana kata-kata akan diwakili dan mempengaruhi kemampuan model untuk menangkap informasi semantik.
2. `num_heads`: Ini menentukan jumlah kepala attention dalam mekanisme Multi-Head Attention. Setiap 'kepala' mengamati data input dari perspektif yang berbeda, dan dengan demikian, `num\_heads` memungkinkan model untuk menangkap berbagai aspek dari informasi yang terkandung dalam data. Nilai yang lebih tinggi untuk `num\_heads` menawarkan model kemampuan untuk memperhatikan lebih banyak detail sekaligus, tetapi juga meningkatkan kompleksitas komputasi.
3. `ff_dim`: Parameter ini menunjukkan ukuran dimensi dari lapisan internal di jaringan feed-forward yang merupakan bagian dari blok Transformer. Ini menentukan jumlah neuron di lapisan tersembunyi ini, yang berfungsi sebagai ruang untuk memproses dan menggabungkan informasi yang telah dikumpulkan oleh mekanisme attention.
4. `rate`: Ini adalah tingkat dropout yang akan diterapkan di dalam blok Transformer, yang merupakan persentase dari output yang dihilangkan secara acak selama proses pelatihan untuk mencegah overfitting. Dengan menetapkan rate sebesar 0.1, model secara acak menonaktifkan 10% dari nilai output sebagai bagian dari upaya regularisasi.

Dalam definisi `call` yang merupakan bagian dari kelas `TransformerBlock`, model melakukan beberapa operasi utama:



1. Attention: Menggunakan lapisan `'MultiHeadAttention'`, blok ini menerapkan attention pada inputnya sendiri, memungkinkan model untuk menentukan bagian dari input yang paling relevan terhadap bagian lainnya.
2. Normalisasi dan Dropout: Setelah mekanisme attention, hasilnya dinormalisasi dan dropout diterapkan. `LayerNormalization` diimplementasikan dengan epsilon sebesar  $1e-6$  untuk stabilitas numerik. Dropout membantu dalam mengurangi kemungkinan overfitting dengan secara acak menghilangkan sebagian dari output neuron selama pelatihan.
3. Feed-Forward Network: Output dari attention kemudian diteruskan melalui jaringan feed-forward yang terdiri dari dua lapisan dense. Lapisan pertama menggunakan fungsi aktivasi ReLU, yang merupakan pilihan standar untuk menambahkan non-linearitas dan membantu jaringan mempelajari representasi yang lebih kompleks.

#### 4.4.1.2 Token dan Position Embedding

Selanjutnya, model memanfaatkan `'TokenAndPositionEmbedding'`, kelas yang penting untuk menggabungkan informasi tentang identitas dan posisi token dalam sebuah sekuens:

1. Token Embedding: Dalam komponen ini, setiap token atau kata dalam ulasan diubah menjadi vektor embedding melalui lapisan `'Embedding'`. Parameter `'vocab_size'` menentukan jumlah kata unik yang diakui oleh model, sementara `'embed_dim'` menentukan ukuran vektor embedding yang digunakan. Ini memungkinkan model untuk memproses kata-kata sebagai entitas numerik yang kaya informasi.
2. Positional Embedding: Model juga membutuhkan informasi tentang posisi token dalam sekuens untuk memahami struktur dan makna teks. `'Positional Embedding'` menggunakan lagi lapisan `'Embedding'`, kali ini dengan `'input_dim'` sebesar `'maxlen'`, yang merepresentasikan panjang maksimum dari sekuens yang diolah model. Output dari kedua embedding, token dan posisi, kemudian dijumlahkan untuk menghasilkan representasi akhir dari setiap kata dalam konteksnya.

#### 4.4.1.3 Contoh Hasil Proses Embedding

Misalkan terdapat sekuens ulasan berikut: "Pelayanan cepat" dengan panjang token maksimal 4. Setelah tokenisasi dan padding, sekuens ini diwakili dalam bentuk numerik sebagai [23, 45, 0, 0], di mana '23' mewakili token untuk "Pelayanan", '45' untuk "cepat", dan '0' untuk padding. Kemudian, setiap token diubah menjadi vektor embedding dan ditambahkan dengan positional embedding yang sesuai. Hasil akhir adalah representasi vektor yang lebih kaya yang mempertimbangkan baik makna semantik dari kata-kata maupun posisi relatif mereka dalam sekuens.

Tabel berikut menunjukkan contoh hasil dari token dan positional embedding untuk sekuens di atas:

Tabel 4. 5 Contoh Hasil Proses Embedding

Token	Embedding Token	Position	Embedding Position	Hasil Embedding Gabungan
Pelayanan	[0.32, -0.24, 0.11, ...]	1	[0.01, 0.00, 0.02, ...]	[0.33, -0.24, 0.13, ...]
cepat	[-0.47, 0.15, -0.35, ...]	2	[0.00, 0.01, -0.02, ...]	[-0.47, 0.16, -0.37, ...]
(padding)	[0.00, 0.00, 0.00, ...]	3	[-0.01, 0.00, 0.01, ...]	[-0.01, 0.00, 0.01, ...]
(padding)	[0.00, 0.00, 0.00, ...]	4	[0.00, -0.01, 0.01, ...]	[0.00, -0.01, 0.01, ...]

Berikut merupakan penjelasan dari proses token embedding, positional embedding, dan embedding gabungan:

1. Token Embedding: Setiap token (kata) memiliki embedding-nya sendiri, yang diperoleh dari lapisan embedding token. Ini adalah representasi vektor yang telah dipelajari selama proses pelatihan model yang menangkap makna semantik kata tersebut. Misalnya, kata "layanan" mungkin diwakili oleh

angka token 23, dan kata "cepat" oleh angka token 45.

2. Positional Embedding: Setiap posisi dalam sekuens memiliki positional embedding sendiri. Ini adalah vektor yang diberikan kepada setiap posisi dalam sekuens untuk memberikan informasi tentang urutan kata dalam sekuens. Positional embedding memungkinkan model untuk membedakan antara "layanan cepat" dan "cepat layanan", meskipun terdiri dari kata-kata yang sama.
3. Gabungan Token dan Positional Embedding: Dalam model Transformer, setiap token dalam sekuens ditambahkan dengan positional embedding-nya yang sesuai. Ini berarti bahwa untuk sekuens tertentu, kata "layanan" di posisi pertama akan dijumlahkan dengan positional embedding untuk posisi pertama, kata "cepat" di posisi kedua akan dijumlahkan dengan positional embedding untuk posisi kedua, dan seterusnya. Proses penjumlahan ini menghasilkan vektor embedding gabungan yang digunakan oleh model untuk memproses teks.

#### **4.4.2 Pembangunan Model Transformer**

Proses pembangunan model Transformer untuk tugas pengenalan emosi dari ulasan pelanggan e-commerce menggabungkan teknik-teknik pemrosesan bahasa terkini dengan arsitektur jaringan saraf yang inovatif. Berikut merupakan uraian proses pembangunan model, dengan fokus pada setiap komponen dan parameter yang digunakan untuk menyusun arsitektur Transformer yang efektif dan adaptif.

```

# Define the model parameters
embed_dim = 32
num_heads = 2
ff_dim = 32

regularizer = keras.regularizers.l2(0.001)

# Build the model
inputs = layers.Input(shape=(maxlen,))
embedding_layer = TokenAndPositionEmbedding(maxlen, vocab_size, embed_dim)
x = embedding_layer(inputs)
transformer_block = TransformerBlock(embed_dim, num_heads, ff_dim)
x = transformer_block(x)
x = layers.GlobalAveragePooling1D()(x)
x = layers.Dropout(0.2)(x)
x = layers.Dense(128, activation="relu")(x)
outputs = layers.Dense(5, activation="softmax")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

```

Gambar 4. 7 Kode Untuk Membangun Model Transformer

#### 4.4.2.1 Pembangunan Model

Pembangunan model dimulai dengan penentuan parameter utama yang akan menentukan ukuran dan kapasitas model. Parameter ini termasuk `embed\_dim`, `num\_heads`, dan `ff\_dim`, yang masing-masing memiliki peran krusial dalam arsitektur Transformer:

1. Ukuran Embedding (`embed_dim`): Setiap token dalam teks diwakili oleh sebuah vektor dengan dimensi yang ditentukan oleh `embed\_dim`. Dalam kasus ini, `embed\_dim` ditetapkan ke 32, yang berarti setiap kata dalam ulasan akan diwakili oleh vektor 32-dimensi. Ukuran ini merupakan trade-off antara kemampuan model untuk menangkap informasi semantik kata-kata dan keefisienan komputasi.
2. Jumlah Attention Heads (`num_heads`): Mekanisme attention di dalam Transformer memungkinkan model untuk fokus pada bagian-bagian tertentu dari input sekuens. `num\_heads` menentukan berapa banyak 'head' attention yang digunakan, dengan setiap 'head' mempelajari aspek yang berbeda dari data. Dua head attention menunjukkan bahwa model akan mempelajari dua set hubungan yang berbeda secara simultan dalam setiap

blok Transformer.

3. Dimensi Feed-Forward Network (ff\_dim): Di dalam setiap blok Transformer, terdapat jaringan feed-forward yang menambahkan lapisan komputasi tambahan. Di sini, 'ff\_dim' juga diatur ke 32, yang berarti lapisan tersembunyi dari jaringan feed-forward ini akan memiliki 32 unit. Ini bertujuan untuk memperluas kapasitas model dalam mempelajari representasi non-linear dari data.

Regularisasi dilakukan menggunakan regularizer L2 dengan koefisien 0.001 untuk mengurangi overfitting, memberikan penalti pada bobot yang lebih besar di dalam model dan mendorong pembelajaran representasi yang lebih general.

#### **4.4.2.2 Proses Pembangunan**

Proses pembangunan model dilakukan melalui beberapa tahap. Pertama, input model diinisialisasi untuk menerima sekuens token dengan panjang maksimal yang ditetapkan ('maxlen'). Lapisan embedding gabungan, yang menggabungkan token dan positional embeddings, diaplikasikan pada input untuk menghasilkan representasi vektor yang mendalam dan berposisi. Blok Transformer yang mendefinisikan model kemudian menambahkan mekanisme self-attention, yang memungkinkan model untuk memahami konteks dan hubungan dalam data.

Setelah melewati blok Transformer, output dimasukkan ke dalam Global Average Pooling untuk mereduksi dimensi dan mengekstrak fitur penting, diikuti oleh lapisan Dropout untuk menghindari overfitting. Lapisan Dense berikutnya dengan 128 unit dan aktivasi ReLU menambahkan kapasitas tambahan untuk pemodelan non-linearitas. Lapisan Dropout kedua memperkuat upaya untuk mencegah model dari terlalu menyesuaikan dengan data pelatihan.

Akhirnya, lapisan output dengan aktivasi softmax mengklasifikasikan teks ke dalam lima kategori emosi yang berbeda. Model ini dikompilasi dengan optimizer Adam dan fungsi loss `sparse_categorical_crossentropy` yang ideal untuk tugas klasifikasi multi-kelas dengan label kelas yang disediakan sebagai bilangan bulat.

#### 4.4.3 Strategi Pelatihan

Strategi pelatihan merupakan fondasi penting dalam memastikan efektivitas model Transformer dalam mengklasifikasikan emosi dari ulasan pelanggan e-commerce. Berikut merupakan uraian strategi pelatihan yang digunakan dalam penelitian ini, termasuk pengaturan class weights, inisialisasi callback untuk early stopping, penyesuaian optimizer, dan proses fitting model.

```
# Calculate class weights
class_weights = compute_class_weight(
    class_weight='balanced',
    classes=np.unique(y_train),
    y=y_train.flatten()
)
class_weight_dict = {i: weight for i, weight in enumerate(class_weights)}

from tensorflow.keras.callbacks import EarlyStopping
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=15,
    restore_best_weights=True
)

optimizer = keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer,
               loss="sparse_categorical_crossentropy",
               metrics=["accuracy"])

history = model.fit(
    x_train, y_train,
    batch_size=32,
    epochs=200,
    validation_data=(x_val, y_val),
    callbacks=[early_stopping],
    class_weight=class_weight_dict # Add class weights here
)
```

Gambar 4. 8 Strategi Pelatihan Pada Model Transformer

##### 4.4.3.1 Penetapan Class Weights

Karena distribusi kelas dalam dataset tidak seimbang, penetapan class weights menjadi langkah penting untuk memastikan bahwa model tidak bias

terhadap kelas yang lebih dominan. Dengan menggunakan fungsi `'compute_class_weight'` dari scikit-learn, class weights dihitung berdasarkan distribusi frekuensi kelas dalam data pelatihan. Strategi ini menyeimbangkan kelas dengan memberikan bobot lebih kepada kelas yang kurang terwakili, memastikan bahwa setiap kelas memiliki pengaruh yang setara selama proses pelatihan. Class weights ini kemudian disimpan dalam `'class_weight_dict'`, memungkinkan mereka untuk diterapkan langsung ke model selama proses fitting.

#### **4.4.3.2 Callback Early Stopping**

Untuk menghindari overfitting dan memastikan bahwa model tidak terlalu menyesuaikan diri dengan data pelatihan, callback early stopping diterapkan. Ini diinisialisasi dengan parameter `'patience=15'`, yang berarti pelatihan akan dihentikan jika tidak ada peningkatan pada validation loss selama 15 epoch berturut-turut. Callback ini juga dikonfigurasi untuk mengembalikan bobot model terbaik yang tercapai sebelum terjadinya peningkatan, berkat parameter `'restore_best_weights=True'`.

#### **4.4.3.3 Optimizer dan Laju Pembelajaran**

Optimizer Adam dipilih karena kemampuannya yang telah terbukti dalam menangani berbagai tugas optimasi dalam deep learning. Dengan laju pembelajaran yang diatur ke 0.0001, optimizer ini bertujuan untuk membuat perubahan bertahap dan konsisten pada bobot model, mencari titik optimal dalam ruang parameter model dengan cara yang efisien dan efektif.

#### **4.4.3.4 Kompilasi dan Pelatihan Model**

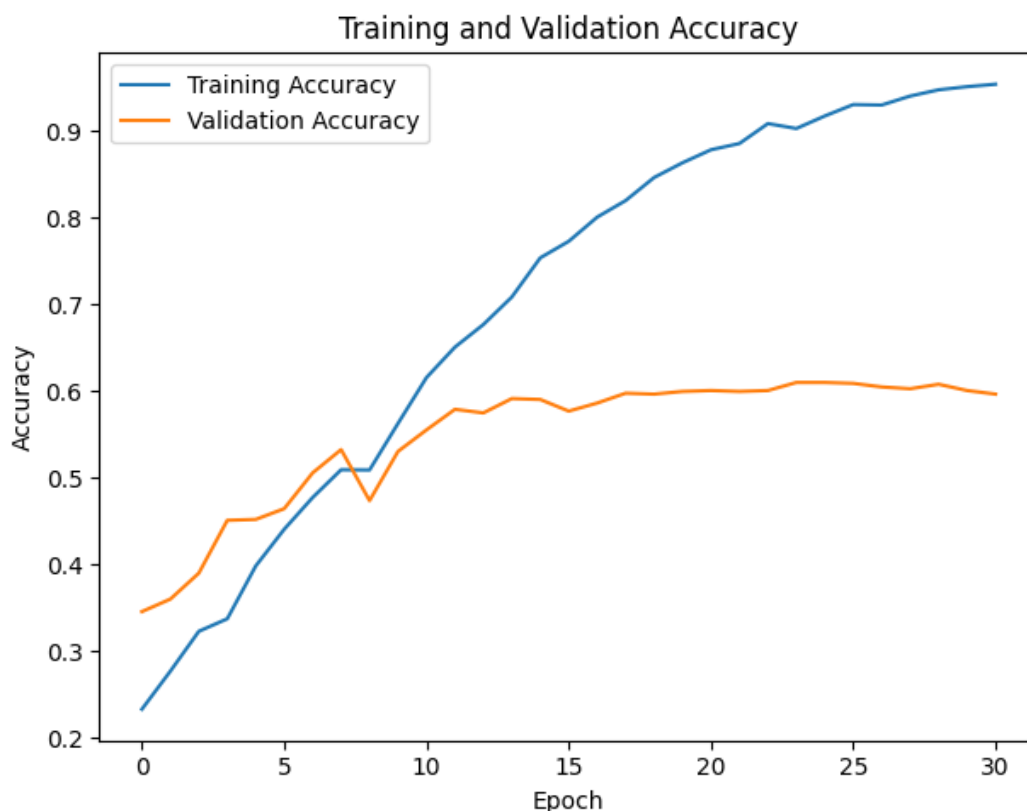
Model dikompilasi menggunakan loss function `'sparse_categorical_crossentropy'`, yang cocok untuk masalah klasifikasi dengan label kelas sebagai bilangan bulat, dan metrik `'accuracy'` untuk memantau kinerja model. Model kemudian dilatih menggunakan metode `'fit'`, dengan data pelatihan dan validasi yang telah dipersiapkan sebelumnya, batch size sebesar 32, dan jumlah

epoch sebanyak 200. Selama pelatihan, class weights diterapkan untuk menyesuaikan pengaruh sample dari setiap kelas, dan callback early stopping dimasukkan untuk memonitor kinerja model dan menghentikan pelatihan jika diperlukan.

#### 4.4.4 Performa Model

Analisis performa model merupakan tahapan krusial untuk mengukur efektivitas algoritma Transformer dalam klasifikasi emosi dari ulasan pelanggan e-commerce. Berikut merupakan uraian evaluasi akurasi dan loss selama pelatihan dan validasi yang tercermin dalam grafik.

##### 4.4.4.1 Analisis Akurasi



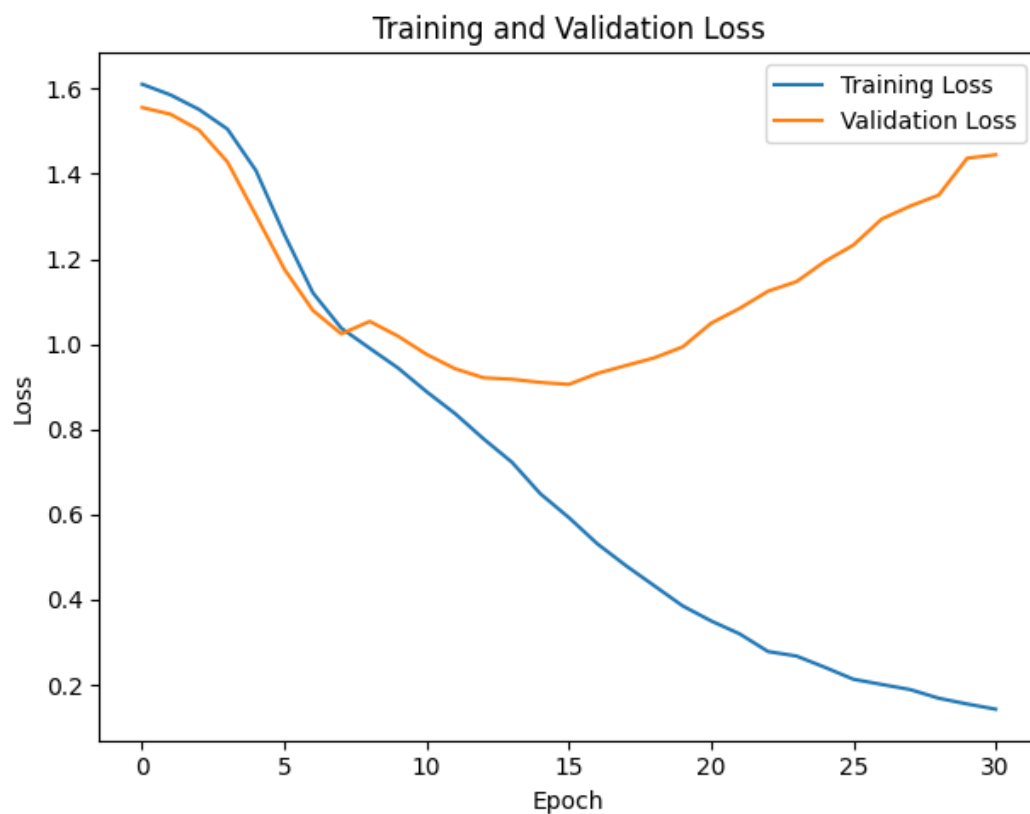
Gambar 4. 9 Grafik Akurasi Model Transformer

Grafik akurasi pada gambar 4.8 menampilkan tren yang meningkat selama pelatihan, mengindikasikan bahwa model secara konsisten memperbaiki



kemampuan prediksinya setiap epoch. Namun, validasi akurasi mengalami fluktuasi dan menunjukkan peningkatan yang lebih sedang, menggambarkan tantangan model dalam menggeneralisasi pembelajaran ke data yang belum pernah dilihat. Selisih antara akurasi pelatihan dan validasi mengindikasikan adanya overfitting yang mungkin terjadi, meskipun ini diatasi sebagian dengan penggunaan callback early stopping.

#### 4.4.4.2 Analisis Loss



Gambar 4. 10 Grafik Loss Model Transformer

Grafik loss pada gambar 4.9 memberikan perspektif lain dari pembelajaran model. Loss pelatihan yang terus menurun mengindikasikan bahwa model secara efektif meminimalkan kesalahan prediksi pada data pelatihan. Namun, loss validasi tidak menunjukkan penurunan yang konsisten, melainkan cenderung stabil setelah penurunan awal. Ini dapat menunjukkan bahwa model mungkin telah mencapai batas kemampuannya dalam menangkap pola dari data validasi, atau mungkin

memerlukan penyesuaian lebih lanjut pada arsitektur atau hyperparameter untuk meningkatkan generalisasi.

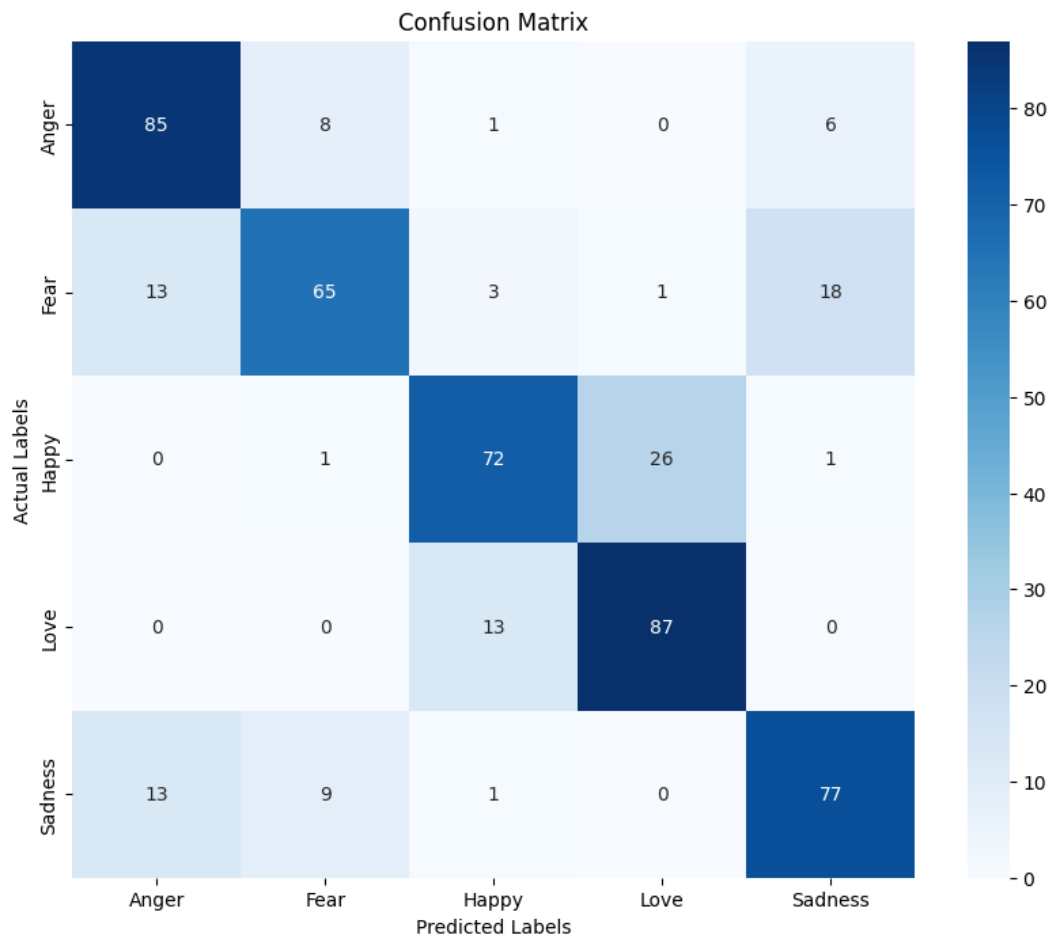
Dari hasil yang diamati, model menunjukkan kemampuan yang kuat dalam memahami data pelatihan, tetapi menghadapi tantangan dalam mengaplikasikan pembelajaran tersebut secara luas. Ini menggarisbawahi pentingnya teknik pelatihan yang tepat, termasuk penyetelan hyperparameter, regularisasi yang tepat, dan strategi pelatihan untuk menghindari overfitting.

#### **4.5 Evaluasi Model Dengan Data Pengujian**

Dalam proses evaluasi model dengan data pengujian, fokus utama adalah untuk mengukur sejauh mana model Transformer yang telah dilatih mampu mengenali dan mengklasifikasikan emosi yang terkandung dalam ulasan pelanggan secara akurat. Evaluasi ini penting untuk memahami bagaimana model akan beroperasi dalam lingkungan nyata, di mana ketepatan dalam mengidentifikasi emosi dari teks adalah kunci untuk memperoleh insight yang berharga dari data pelanggan. Evaluasi model dengan data pengujian ini dilakukan menggunakan dataset testing sebanyak 500 data dengan masing-masing kelasnya 100 data.

##### **4.5.1 Confusion Matrix**

Confusion matrix adalah alat yang sangat penting dalam machine learning untuk mengukur kinerja model klasifikasi. Berikut confusion matrix yang dihasilkan oleh model klasifikasi emosi dari ulasan pelanggan e-commerce:



Gambar 4. 11 Confusion Matrix Model Transformer

Dari Confusion Matrix yang ditampilkan pada gambar 4.10, dapat diamati bahwa:

1. Kelas Anger:

- True Positives (TP): Terdapat 85 ulasan yang secara akurat diklasifikasikan sebagai 'Anger', menunjukkan model sangat efektif dalam mengidentifikasi emosi kemarahan.
- False Positives (FP): Sebanyak 8 ulasan dari kelas lain keliru diklasifikasikan sebagai 'Anger', menandakan adanya kemungkinan overlap ekspresi dengan emosi lain.
- False Negatives (FN): Ada 15 ulasan dengan emosi 'Anger' yang model tidak bisa deteksi, yang bisa mengindikasikan bahwa beberapa nuansa emosi kemarahan mungkin belum dipelajari

dengan baik oleh model.

- True Negatives (TN): TN untuk 'Anger' adalah jumlah kasus di mana model dengan benar mengidentifikasi bahwa ulasan tidak mengandung emosi 'Anger'. Dalam hal ini, TN diperoleh dengan menghitung semua prediksi yang benar untuk kategori emosi selain 'Anger', yaitu, jumlah semua nilai yang tidak berada di baris atau kolom kelas tersebut yang totalnya adalah 374. Ini berarti model tidak salah mengklasifikasikan 374 ulasan sebagai 'Anger', menunjukkan keberhasilan model dalam membedakan ulasan yang tidak marah dari yang marah.

## 2. Kelas Fear:

- True Positives (TP): Model mengenali 65 ulasan sebagai 'Fear', mencerminkan kemampuan model untuk mendeteksi ekspresi kekhawatiran atau ansietas.
- False Positives (FP): Terdapat 13 kesalahan positif dimana emosi 'Fear' diprediksi pada ulasan yang sebenarnya tidak mengandungnya, mungkin karena adanya kata-kata atau konteks yang mirip dengan emosi lain.
- False Negatives (FN): Model gagal mengidentifikasi 35 ulasan yang seharusnya dikategorikan sebagai 'Fear', menunjukkan bahwa model mungkin tidak peka terhadap beberapa indikator kekhawatiran yang lebih halus.
- True Negatives (TN): TN untuk 'Fear' menunjukkan 387 ulasan yang dengan tepat tidak diidentifikasi sebagai 'Fear' oleh model. Ini mencerminkan jumlah kesempatan di mana model dengan tepat tidak mendeteksi kekhawatiran atau ketakutan, menunjukkan keefektifan model dalam memisahkan ekspresi 'Fear' dari ekspresi emosi lain.

## 3. Kelas Happy:

- True Positives (TP): Terdapat 72 kasus dimana ulasan 'Happy' diidentifikasi dengan tepat, menandakan model mampu mengenali

ekspresi kegembiraan atau kepuasan.

- False Positives (FP): Ada 26 kesalahan positif di mana model salah menganggap emosi lain sebagai 'Happy', yang bisa jadi karena penggunaan kata-kata positif yang serupa dalam konteks yang berbeda.
- False Negatives (FN): Model tidak mengenali 28 ulasan yang sebenarnya mengandung 'Happy', menunjukkan bahwa model perlu lebih sensitif terhadap variasi ekspresi kebahagiaan.
- True Negatives (TN): TN untuk 'Happy' adalah 374, menandakan jumlah ulasan yang model klasifikasikan dengan benar sebagai tidak mengandung kebahagiaan. Nilai ini mengindikasikan bahwa dalam 374 kasus, model berhasil mengenali bahwa ulasan tidak menyampaikan emosi 'Happy', menegaskan kemampuan model dalam membedakan antara kebahagiaan dan emosi lain.

#### 4. Kelas Love:

- True Positives (TP): Dengan 87 ulasan yang diklasifikasikan dengan benar sebagai 'Love', model ini menunjukkan keefektifan dalam mengenali ekspresi kasih sayang dan kehangatan.
- False Positives (FP): Terdapat 13 kesalahan positif, yang mungkin mengindikasikan adanya kesamaan linguistik antara 'Love' dan emosi positif lain seperti 'Happy'.
- False Negatives (FN): Tidak ada kesalahan negatif terdeteksi untuk 'Love', menandakan bahwa setiap ulasan 'Love' dalam sampel berhasil diidentifikasi.
- True Negatives (TN): TN untuk 'Love' tercatat sebanyak 387, menunjukkan jumlah ulasan yang tidak diklasifikasikan sebagai 'Love' oleh model. Angka ini menunjukkan bahwa model dapat mengenali dan memisahkan 387 ulasan dari ekspresi kasih sayang, memverifikasi bahwa model mampu membedakan 'Love' dari emosi lain.

#### 5. Kelas Sadness:

- True Positives (TP): Sebanyak 77 ulasan 'Sadness' diidentifikasi dengan benar, menunjukkan bahwa model cukup sensitif terhadap ekspresi kesedihan.
- False Positives (FP): Model membuat 22 kesalahan positif dengan mengklasifikasikan ulasan non-'Sadness' sebagai 'Sadness'. Ini menunjukkan bahwa ekspresi kekecewaan atau penyesalan dalam ulasan mungkin memiliki ciri linguistik yang serupa dengan emosi negatif lainnya.
- False Negatives (FN): Ada 23 kesalahan negatif di mana model gagal mendeteksi 'Sadness', yang bisa menandakan bahwa model perlu lebih memperhatikan konteks atau kata-kata kunci yang menandakan kesedihan.
- True Negatives (TN): TN untuk 'Sadness' sebanyak 378 menunjukkan bahwa model dengan tepat tidak menandai 378 ulasan sebagai menyatakan kesedihan. Ini mengonfirmasi bahwa model dapat mengidentifikasi ulasan yang tidak sedih dan membedakannya dari mereka yang menyatakan kesedihan.

Dari confusion matrix, model transformer untuk klasifikasi emosi memiliki kemampuan yang kuat dalam mengidentifikasi emosi tertentu, namun masih menghadapi tantangan dalam membedakan antara emosi-emosi yang serupa. Keakuratan dalam mengidentifikasi 'Anger' dan 'Love' cukup tinggi, namun model perlu peningkatan dalam memahami 'Fear', 'Happy', dan 'Sadness', terutama dalam mengenali perbedaan antara emosi-emosi yang sering kali memiliki tanda-tanda yang mirip. Hal ini menunjukkan pentingnya meningkatkan pra-pemrosesan data dan mungkin mempertimbangkan teknik ekstraksi fitur yang lebih canggih atau pemberian pelatihan dengan dataset yang lebih beragam untuk meningkatkan pemahaman model terhadap nuansa emosi yang halus dalam teks.

#### **4.5.2 Evaluasi Hasil Model**

Setelah evaluasi menggunakan confusion matrix, selanjutnya adalah evaluasi hasil model untuk mengetahui mengenai performa model klasifikasi di

setiap kelas. Evaluasi ini membandingkan output prediksi model dengan label sebenarnya dan menghasilkan metrik kunci: presisi, recall, dan skor F1 untuk setiap kelas emosi. Dari confusion matrix pada gambar 4.10 didapatkan nilai-nilai berikut untuk setiap kelas:

1. Kelas Anger: TP=85, TN=374, FP=26, FN=15
2. Kelas Fear: TP=65, TN=382, FP=18, FN=35
3. Kelas Happy: TP=72, TN=382, FP=18, FN=28
4. Kelas Love: TP=87, TN=373, FP=27, FN=13
5. Kelas Sadness: TP=77, TN=375, FP=25, FN=23

Menggunakan nilai-nilai tersebut, berikut perhitungan metrik untuk masing-masing kelas:

1. Kelas Anger:

- $Akurasi = \frac{TP}{Total\ Prediksi} = \frac{85}{100} = 0.85$
- $Presisi = \frac{TP}{(TP+FP)} = \frac{85}{(85+26)} = \frac{85}{111} = 0.766$
- $Recall = \frac{TP}{(TP+FN)} = \frac{85}{(85+15)} = \frac{85}{100} = 0.85$
- $F1\ Score = 2 \times \frac{(Prec \times Rec)}{(Prec + Rec)} = 2 \times \frac{(0.766 \times 0.85)}{(0.766 + 0.85)} = 0.805$

2. Kelas Fear:

- $Akurasi = \frac{TP}{Total\ Prediksi} = \frac{65}{100} = 0.65$
- $Presisi = \frac{TP}{(TP+FP)} = \frac{65}{(65+18)} = \frac{65}{83} = 0.783$
- $Recall = \frac{TP}{(TP+FN)} = \frac{65}{(65+35)} = \frac{65}{100} = 0.65$
- $F1\ Score = 2 \times \frac{(Prec \times Rec)}{(Prec + Rec)} = 2 \times \frac{(0.783 \times 0.65)}{(0.783 + 0.65)} = 0.711$

3. Kelas Happy:

- $Akurasi = \frac{TP}{Total\ Prediksi} = \frac{72}{100} = 0.72$
- $Presisi = \frac{TP}{(TP+FP)} = \frac{72}{(72+18)} = \frac{72}{90} = 0.8$
- $Recall = \frac{TP}{(TP+FN)} = \frac{72}{(72+28)} = \frac{72}{100} = 0.72$
- $F1\ Score = 2 \times \frac{(Prec \times Rec)}{(Prec + Rec)} = 2 \times \frac{(0.8 \times 0.72)}{(0.8 + 0.72)} = 0.758$

## 4. Kelas Love:

- $Akurasi = \frac{TP}{Total\ Prediksi} = \frac{87}{100} = 0.87$
- $Presisi = \frac{TP}{(TP+FP)} = \frac{87}{(87+27)} = \frac{87}{114} = 0.763$
- $Recall = \frac{TP}{(TP+FN)} = \frac{87}{(87+13)} = \frac{87}{100} = 0.87$
- $F1\ Score = 2 \times \frac{(Prec \times Rec)}{(Prec + Rec)} = 2 \times \frac{(0.763 \times 0.87)}{(0.763 + 0.87)} = 0.812$

## 5. Kelas Sadness:

- $Akurasi = \frac{TP}{Total\ Prediksi} = \frac{77}{100} = 0.77$
- $Presisi = \frac{TP}{(TP+FP)} = \frac{77}{(77+25)} = \frac{77}{102} = 0.755$
- $Recall = \frac{TP}{(TP+FN)} = \frac{77}{(77+23)} = \frac{77}{100} = 0.77$
- $F1\ Score = 2 \times \frac{(Prec \times Rec)}{(Prec + Rec)} = 2 \times \frac{(0.755 \times 0.77)}{(0.755 + 0.77)} = 0.763$

Selanjutnya adalah menghitung metrik keseluruhan model. Berikut adalah total nilai TP, FP, TN, dan FN dari semua kelas:

- Total TP = TP\_Anger + TP\_Fear + TP\_Happy + TP\_Love + TP\_Sadness =  
85 + 65 + 72 + 87 + 77 = 386
- Total FP = FP\_Anger + FP\_Fear + FP\_Happy + FP\_Love + FP\_Sadness =  
26 + 18 + 18 + 27 + 25 = 114
- Total TN = TN\_Anger + TN\_Fear + TN\_Happy + TN\_Love + TN\_Sadness  
= 374 + 382 + 382 + 373 + 375 = 1886
- Total FN = FN\_Anger + FN\_Fear + FN\_Happy + FN\_Love + FN\_Sadness  
= 15 + 35 + 28 + 13 + 23 = 114

Berikut merupakan perhitungan metrik evaluasi dari keseluruhan model:

- $Akurasi\ Total = \frac{Total\ TP}{Total\ Prediksi} = \frac{386}{500} = 0.772$
- $Presisi\ Total = \frac{Total\ TP}{Total\ TP + Total\ FP} = \frac{386}{386 + 114} = 0.772$
- $Recall\ Total = \frac{Total\ TP}{Total\ TP + Total\ FN} = \frac{386}{386 + 114} = 0.772$



$$\bullet \quad F1 \text{ Score Total} = 2 \times \frac{Prec \text{ Total} \times Rec \text{ Total}}{Prec \text{ Total} + Rec \text{ Total}} = 2 \times \frac{0.722}{0.722} = 0.722$$

Tabel berikut ini menampilkan hasil dari perhitungan metrik evaluasi dari setiap kelas dan keseluruhan model:

Tabel 4. 6 Evaluasi Hasil Model Transformer

Emosi	Akurasi	Precision	Recall	F1-Score
Anger	0.850	0.766	0.850	0.805
Fear	0.650	0.783	0.650	0.711
Happy	0.720	0.800	0.720	0.758
Love	0.870	0.763	0.870	0.812
Sadness	0.770	0.755	0.770	0.763
Total	0.772	0.772	0.772	0.772

#### 4.5.2.1 Evaluasi Kelas Anger

1. Akurasi 85% menandakan bahwa model cukup handal dalam mengidentifikasi ulasan yang mengandung emosi kemarahan, dengan sebagian besar ulasan yang benar-benar marah berhasil diidentifikasi.
2. Precision sebesar 76.6% mengindikasikan bahwa ketika model memprediksi ulasan sebagai marah, sekitar tiga perempat dari prediksi tersebut akurat. Ini menunjukkan adanya margin kesalahan di mana ulasan tidak marah dapat dianggap sebagai marah.
3. Recall juga 85% menunjukkan bahwa model sangat efektif dalam menangkap ulasan yang seharusnya dikategorikan sebagai marah. Ini merupakan pertanda baik bahwa sedikit ulasan marah yang sebenarnya terlewatkan oleh model.
4. F1-Score 80.5% adalah rata-rata harmonis dari precision dan recall, menunjukkan bahwa model memiliki keseimbangan yang baik antara sensitivitas dan spesifisitas dalam mengenali emosi kemarahan.

#### 4.5.2.2 Evaluasi Kelas Fear

1. Akurasi yang lebih rendah pada 65% menandakan bahwa ada beberapa

tantangan dalam mengenali ulasan yang mengandung ketakutan, dengan sejumlah signifikan ulasan takut tidak terdeteksi atau salah diklasifikasikan.

2. Precision yang tinggi pada 78.3% mengimplikasikan bahwa ketika model memprediksi ulasan sebagai takut, prediksi tersebut cenderung akurat, tetapi kesalahan dalam mengidentifikasi ulasan lain sebagai takut juga cukup tinggi.
3. Recall pada 65% menunjukkan bahwa model memiliki ruang untuk peningkatan dalam mendeteksi semua kasus ketakutan, dengan sekitar sepertiga dari ulasan takut tidak teridentifikasi.
4. F1-Score pada 71.1% menunjukkan bahwa model memiliki kinerja yang lebih rendah dalam keseimbangan antara precision dan recall untuk emosi ketakutan dibandingkan dengan kemarahan.

#### **4.5.2.3 Evaluasi Kelas Happy**

1. Akurasi sebesar 72% menunjukkan bahwa model cukup efektif namun tidak sempurna dalam mengenali ulasan bahagia. Ini menandakan adanya potensi untuk perbaikan dalam mengenali ekspresi kebahagiaan.
2. Precision pada 80% menandakan bahwa model cukup akurat ketika memprediksi ulasan sebagai bahagia, tetapi masih ada ruang untuk meningkatkan kemampuan model dalam membedakan antara kebahagiaan dan emosi lain.
3. Recall pada 72% menandakan bahwa model cukup handal dalam mengenali ulasan bahagia tetapi masih melewatkan beberapa kasus.
4. F1-Score 75.8% mencerminkan keseimbangan yang cukup baik antara kemampuan model dalam mengidentifikasi kebahagiaan dan meminimalkan kesalahan positif.

#### **4.5.2.4 Evaluasi Kelas Love**

1. Akurasi yang tinggi sebesar 87% menunjukkan bahwa model sangat kompeten dalam mengenali ulasan yang mengungkapkan cinta atau afeksi,

yang berarti bahwa hampir semua ulasan yang sebenarnya mengungkapkan cinta dapat diidentifikasi dengan benar.

2. Precision pada 76.3% menunjukkan bahwa sebagian besar prediksi model tentang emosi cinta adalah akurat, namun masih ada kasus di mana emosi positif lainnya seperti kebahagiaan mungkin salah dikenali sebagai cinta.
3. Recall juga tinggi pada 87%, yang menandakan bahwa model dapat mengidentifikasi sebagian besar ulasan yang mengungkapkan cinta tanpa melewatkan banyak kasus.
4. F1-Score yang tinggi pada 81.2% menegaskan bahwa model menunjukkan keseimbangan yang baik antara precision dan recall, menggambarkan efektivitas model dalam mengenali cinta dengan tingkat kesalahan yang relatif rendah.

#### **4.5.2.5 Evaluasi Kelas Sadness**

1. Akurasi 77% menunjukkan bahwa model cukup efektif dalam mengenali ulasan yang mengungkapkan kesedihan, meskipun masih ada ruang untuk peningkatan, terutama dalam menangkap semua nuansa ekspresi kesedihan.
2. Precision pada 75.5% mengindikasikan bahwa ada kemungkinan kecil bahwa ulasan dengan emosi lain dapat keliru diklasifikasikan sebagai sedih, tetapi sebagian besar prediksi kesedihan akurat.
3. Recall pada 77% menunjukkan bahwa model cukup andal dalam mengidentifikasi ulasan sedih, namun ada beberapa kasus yang terlewatkan.
4. F1-Score pada 76.3% menggambarkan keseimbangan yang relatif baik antara kemampuan model untuk mengenali kesedihan dan menghindari kesalahan dalam prediksi.

#### **4.5.2.6 Evaluasi Keseluruhan Model**

Model menunjukkan kinerja yang kuat secara keseluruhan dengan akurasi rata-rata sekitar 77.2%, menandakan bahwa dalam lebih dari tiga perempat kasus, model dapat dengan tepat mengklasifikasikan emosi yang terkandung dalam ulasan.

1. Dari semua kategori emosi, model paling handal dalam mengidentifikasi 'Love' dan paling kurang dalam mengidentifikasi 'Fear' dan 'Happy'.
2. F1-Score yang relatif tinggi di seluruh emosi menunjukkan keseimbangan yang baik antara precision dan recall, menandakan bahwa model tidak hanya mengejar kemampuan untuk mengidentifikasi emosi yang benar, tetapi juga meminimalkan kesalahan baik dalam hal positif maupun negatif.

Dari analisis di atas, dapat dilihat bahwa model klasifikasi emosi yang dikembangkan telah menunjukkan performa yang kuat dengan beberapa area yang dapat ditingkatkan. Meskipun model telah berhasil dalam mengenali emosi tertentu dengan akurasi yang tinggi, ada peluang untuk meningkatkan pemahaman model terhadap nuansa dalam ekspresi emosi, terutama dalam membedakan antara emosi yang sering kali beririsan dalam ekspresi bahasa.

#### 4.5.3 Analisis Kesalahan Prediksi

Analisis kesalahan prediksi memberikan wawasan tentang kinerja model dan bagaimana ia bisa ditingkatkan.

Tabel 4. 7 Sampel Contoh Kesalahan Prediksi

Customer Review	Emotion	Predicted Emotion
kartunya kw semua,katanya ori gimana sih	Anger	Happy
Baru pake sekali per nya lepas2 terus ampas	Anger	Sadness
Barang tidak sesuai pesanan ... Mohon pendataan barang lebih cermat lagi	Fear	Sadness
lamaaaa bangettt respons nya	Fear	Anger
Sesuai deskripsi seller gercep & interaktif rekomended	Happy	Love
Mantap ori	Happy	Love

Nyaman dipakai, kemasan bagus, pengiriman super !	Love	Happy
sudah diterima dengan baik mantaffffffffffffff	Love	Happy
bahannya jelek kualitasnya sangat jelek baru brp menit sudah rusak	Sadness	Anger
barang jelek, belom d pake udah pada patah	Sadness	Anger

Berikut merupakan analisis kesalahan prediksi yang dihasilkan oleh model klasifikasi emosi dari ulasan pelanggan e-commerce.

1. Anger: Ketika model mengklasifikasikan ulasan dengan ekspresi "kartunya kw semua, katanya ori gimana sih" sebagai 'Happy' bukan 'Anger', ini menunjukkan bahwa model gagal mengenali tanda-tanda frustrasi dan kekecewaan yang terkandung dalam ulasan tersebut. Ini bisa jadi karena adanya kata-kata atau frasa yang biasanya dihubungkan dengan kesenangan atau kegembiraan yang digunakan dalam konteks yang ironis atau sarkastik. Kesalahan prediksi lain terjadi ketika model mengklasifikasikan "Baru pake sekali per nya lepas2 terus ampas" sebagai 'Sadness'. Dalam hal ini, model mungkin tertipu oleh kata-kata yang seringkali dikaitkan dengan kesedihan, seperti 'lepas' atau 'ampas', tanpa mempertimbangkan konteks keseluruhan yang menunjukkan ketidaksenangan atau marah.
2. Fear: Kesalahan prediksi dalam kategori 'Fear' terlihat ketika model salah mengklasifikasikan "Barang tidak sesuai pesanan ... Mohon pendataan barang lebih cermat lagi" sebagai 'Sadness' dan "lamaaa banget respons nya" sebagai 'Anger'. Dalam kedua kasus, meskipun ulasan tersebut mengungkapkan kekhawatiran atau rasa takut akan sesuatu yang tidak memuaskan atau respons yang lambat, model tidak berhasil menangkap rasa urgensi atau kekhawatiran tersebut dan malah membingungkannya dengan emosi lain yang lebih dominan dalam konteks tersebut.
3. Happy: Analisis kesalahan prediksi untuk emosi 'Happy' menunjukkan

kecenderungan model untuk mengklasifikasikan ulasan positif yang mengandung kata-kata seperti "Sesuai deskripsi seller gercep & interaktif rekomended" dan "Mantap ori" sebagai 'Love'. Ini menandakan bahwa model cenderung mengasosiasikan kata-kata yang menunjukkan kepuasan atau pujian dengan emosi cinta atau kekaguman, yang mengindikasikan bahwa pemahaman model terhadap nuansa bahasa positif perlu lebih ditingkatkan.

4. Love: Dalam kasus 'Love', kesalahan prediksi terjadi ketika ulasan dengan nada apresiatif dan pujian seperti "Nyaman dipakai, kemasan bagus, pengiriman super!" dan "sudah diterima dengan baik mantafffffffffffff" diklasifikasikan sebagai 'Happy'. Meskipun kedua emosi ini saling terkait dan seringkali sulit dibedakan, penting bagi model untuk membedakan antara kepuasan umum dan ekspresi kasih sayang yang mungkin lebih spesifik atau intens.
5. Sadness: Akhirnya, dalam kategori 'Sadness', kesalahan terjadi ketika model mengklasifikasikan "bahannya jelek kualitasnya sangat jelek baru brp menit sudah rusak" dan "barang jelek, belom d pake udah pada patah" sebagai 'Anger'. Walaupun kedua ulasan tersebut jelas menunjukkan kekecewaan, model tampaknya lebih fokus pada kata-kata yang biasanya dikaitkan dengan kemarahan. Kesalahan ini menyoroti perlunya model untuk memperbaiki cara membedakan antara ungkapan kekecewaan yang mendalam dengan ungkapan kemarahan yang lebih tegas.

Analisis kasus-kasus ini menunjukkan beberapa alasan mengapa model dapat membuat kesalahan prediksi, termasuk kesulitan dalam menangkap nuansa bahasa dan emosi, terbatasnya konteks, dan kesamaan ekspresi antara kategori emosi yang berbeda. Untuk mengatasi masalah ini, peningkatan dalam pra-pemrosesan data, penyesuaian pada arsitektur model, atau pelatihan dengan dataset yang lebih besar dan lebih beragam mungkin diperlukan untuk meningkatkan kemampuan model dalam memahami dan mengklasifikasikan emosi dengan lebih akurat.

## **BAB V**

### **KESIMPULAN DAN PENELITIAN SELANJUTNYA**

#### **5.1 Kesimpulan**

Penelitian ini telah menunjukkan kemampuan model deep learning berbasis Transformer dalam mengklasifikasikan emosi dari ulasan pelanggan e-commerce berbahasa Indonesia, dengan beberapa temuan kunci sebagai berikut:

1. Dari analisis dataset yang digunakan, emosi 'Happy' paling dominan dalam ulasan dengan 1770 kasus, diikuti oleh 'Sadness' (1100 ulasan) dan 'Anger' (699 ulasan), menunjukkan kecenderungan pelanggan untuk berbagi pengalaman positif dan negatif.
2. Model berbasis Transformer yang diimplementasikan menunjukkan efisiensi Model dengan waktu pelatihan 90 detik dan akurasi keseluruhan 0.772. Spesifik untuk emosi, 'Love' mencatat akurasi tertinggi 0.870 (F1-score 0.812), 'Anger' 0.850 (F1-score 0.805), 'Happy' 0.720 (F1-score 0.758), 'Sadness' 0.770 (F1-score 0.763), dan 'Fear' menunjukkan ruang peningkatan dengan 0.650 (F1-score 0.711).
3. Model berbasis Transformer menunjukkan potensi yang signifikan dalam analisis emosi ulasan pelanggan, dengan kemampuan untuk mengklasifikasikan emosi secara akurat dan efisien. Meskipun terdapat variasi dalam performa antar kategori emosi, dengan skor total mencapai akurasi 0.772, precision 0.772, recall 0.772, dan F1-score 0.772, temuan ini menegaskan efektivitas model dalam memahami nuansa emosional dalam teks.

Kesimpulan dari penelitian ini adalah bahwa model berbasis Transformer menawarkan pendekatan yang menjanjikan untuk analisis emosi dalam ulasan pelanggan di sektor e-commerce. Meskipun terdapat keterbatasan, seperti keakuratan yang belum sepenuhnya optimal dan kurangnya pembandingan langsung dalam beberapa aspek metodologi, model ini menawarkan landasan yang kokoh untuk pengembangan lebih lanjut, baik dalam penelitian akademik maupun aplikasi industri.

## 5.2 Penelitian Selanjutnya

Penelitian ini membuka berbagai kemungkinan untuk eksplorasi lebih lanjut dalam bidang pengenalan emosi menggunakan deep learning. Berdasarkan temuan dan pengalaman yang diperoleh, terdapat beberapa rekomendasi untuk penelitian selanjutnya:

1. Pengembangan Model: Model yang ada dapat ditingkatkan dengan mengintegrasikan dataset yang lebih besar dan beragam untuk melatih model, termasuk data dari berbagai domain e-commerce dan ulasan dalam bahasa dan dialek Indonesia yang berbeda. Hal ini diharapkan dapat meningkatkan keakuratan dan robustness model dalam mengenali spektrum emosi yang lebih luas.
2. Optimisasi Hyperparameter: Eksperimen lebih lanjut dengan penyetelan hyperparameter dapat memberikan insight berharga tentang konfigurasi yang optimal untuk model Transformer dalam konteks NLP. Penelitian tentang dampak variasi ukuran embedding, jumlah attention heads, dan parameter lainnya terhadap performa model sangat direkomendasikan.
3. Multimodal Emotion Recognition: Menggabungkan data teks dengan data non-teksual, seperti gambar atau metadata produk, dapat memberikan konteks tambahan yang berguna dalam pengenalan emosi. Penelitian ini dapat meluas ke pengembangan model multimodal yang dapat memanfaatkan berbagai jenis input untuk meningkatkan pemahaman model terhadap emosi.

Melalui rekomendasi ini, diharapkan penelitian selanjutnya dapat melanjutkan dan memperluas temuan ini untuk mencapai pemahaman yang lebih mendalam tentang dinamika emosi dalam interaksi manusia dan mesin serta aplikasinya dalam dunia nyata.



## DAFTAR PUSTAKA

- Ab Nasir, A. F., Seok Nee, E., Sern Choong, C., Shahrizan Abdul Ghani, A., Abdul Majeed, A. P. P., Adam, A., & Furqan, M. (2020). Text-based emotion prediction system using machine learning approach. *IOP Conference Series: Materials Science and Engineering*, 769(1). <https://doi.org/10.1088/1757-899X/769/1/012022>
- Alsubari, S. N., Deshmukh, S. N., Alqarni, A. A., Alsharif, N., Aldhyani, T. H. H., Alsaade, F. W., & Khalaf, O. I. (2022). Data analytics for the identification of fake reviews using supervised learning. *Computers, Materials and Continua*, 70(2), 3189–3204. <https://doi.org/10.32604/cmc.2022.019625>
- Atmaja, B. T., Shirai, K., & Akagi, M. (2019). Deep Learning-based Categorical and Dimensional Emotion Recognition for Written and Spoken Text. *IPTEK Journal of Proceedings Series*, 1–10.
- Bharti, S. K., Varadhaganapathy, S., Gupta, R. K., Shukla, P. K., Bouye, M., Hingaa, S. K., & Mahmoud, A. (2022). Text-Based Emotion Recognition Using Deep Learning Approach. *Computational Intelligence and Neuroscience*, 2022, 8. <https://doi.org/10.1155/2022/2645381>
- Changchit, C., & Klaus, T. (2020). Determinants and Impact of Online Reviews on Product Satisfaction. *Journal of Internet Commerce*, 19(1), 82–102. <https://doi.org/10.1080/15332861.2019.1672135>
- Davoodi, L., & Mezei, J. (2022). a Comparative Study of Machine Learning Models for Sentiment Analysis: Customer Reviews of E-Commerce Platforms. *35th Bled EConference: Digital Restructuring and Human (Re)Action, BLED 2022 - Proceedings*, 217–231. <https://doi.org/10.18690/um.fov.4.2022.13>
- Durairaj, A. K., & Chinnalagu, A. (2021). Transformer based Contextual Model for Sentiment Analysis of Customer Reviews: A Fine-tuned BERT A Sequence Learning BERT Model for Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, 12(11), 474–480. <https://doi.org/10.14569/IJACSA.2021.0121153>
- Gong, X., Ying, W., Zhong, S., & Gong, S. (2022). Text Sentiment Analysis Based on Transformer and Augmentation. *Frontiers in Psychology*, 13(May).

<https://doi.org/10.3389/fpsyg.2022.906061>

- Grigoreva, E. A., Garifova, L. F., & Polovkina, E. A. (2021). Consumer Behavior in the Information Economy: Generation Z. *International Journal of Financial Research*, 12(2), 164. <https://doi.org/10.5430/ijfr.v12n2p164>
- Guo, J., Wang, X., & Wu, Y. (2020). Positive emotion bias: Role of emotional content from online customer reviews in purchase decisions. *Journal of Retailing and Consumer Services*, 52(July 2019). <https://doi.org/10.1016/j.jretconser.2019.101891>
- Jiao, Q., & Zhang, S. (2021). A Brief Survey of Word Embedding and Its Recent Development. *IAEAC 2021 - IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, 2021*, 1697–1701. <https://doi.org/10.1109/IAEAC50856.2021.9390956>
- Katole, H. J. (2022). A research article on Importance of Online Customer Reviews on Customer Purchase. *The Business and Management Review*, 13(03), 10–11. <https://doi.org/10.24052/bmr/v13nu03/art-04>
- Liu, Y., Lu, J., Yang, J., & Mao, F. (2020). Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax. *Mathematical Biosciences and Engineering*, 17(6), 7819–7837. <https://doi.org/10.3934/MBE.2020398>
- Maulud, D. H., Zeebaree, S. R. M., Jacksi, K., Sadeeq, M. A. M., & Sharif, K. H. (2021). A State of Art for Semantic Analysis of Natural Language Processing. *Qubahan Academic Journal*, 1(2), 21–28. <https://doi.org/10.48161/qaj.v1n2a44>
- Mulligan, K., & Scherer, K. R. (2012). Toward a working definition of emotion. *Emotion Review*, 4(4), 345–357. <https://doi.org/10.1177/1754073912445818>
- Purohit, A. (2021). Sentiment Analysis of Customer Product Reviews using deep Learning and Compare with other Machine Learning Techniques. *International Journal for Research in Applied Science and Engineering Technology*, 9(VII), 233–239. <https://doi.org/10.22214/ijraset.2021.36202>
- Rocklage, M. D., & Fazio, R. H. (2020). The Enhancing Versus Backfiring Effects of Positive Emotion in Consumer Reviews. *Journal of Marketing Research*,

57(2), 332–352. <https://doi.org/10.1177/0022243719892594>

Sun, J., Xie, J., & Zhou, H. (2021). EEG classification with transformer-based models. *LifeTech 2021 - 2021 IEEE 3rd Global Conference on Life Sciences and Technologies, LifeTech*, 92–93. <https://doi.org/10.1109/LifeTech52111.2021.9391844>

Sutoyo, R., Chowanda, A., Achmad, S., Andangsari, E. W., M. Isa, S., Soetandar, J. P., . . . Pirdaus, F. P. (2022). *Product Reviews Dataset for Emotions Classification Tasks - Indonesian (PRDECT-ID) Dataset*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 5999–6009.