ANALISIS SENTIMEN PERSEPSI KONSUMEN TERHADAP KUALITAS LAYANAN APLIKASI BSI MOBILE MENGGUNAKAN MODEL INDOBERT

TUGAS AKHIR

Diajukan Oleh:

IRVAN ANDIKA
NIM. 200705065
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2023 M / 1445 H

LEMBAR PERSETUJUAN

ANALISIS SENTIMEN PERSEPSI KONSUMEN TERHADAP KUALITAS LAYANAN APLIKASI BSI MOBILE MENGGUNAKAN MODEL INDOBERT

TUGAS AKHIR

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

Oleh:
IRVAN ANDIKA
NIM. 200705065
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Hendri Ahmadian, S.Si., M.I.M	**Malahayati, M.T**
NIP. 198301042014031002	**NIP. 198301272015032003**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Ima Dwitawati, MBA
NIP. 198210132014032002

LEMBAR PENGESAHAN

ANALISIS SENTIMEN PERSEPSI KONSUMEN TERHADAP KUALITAS LAYANAN APLIKASI BSI MOBILE MENGGUNAKAN MODEL INDOBERT

TUGAS AKHIR

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

Pada Hari/Tanggal: Rabu, 20 Desember 2023
7 Jumadil Akhir 1445 H
di Darussalam, Banda Aceh

Panitia Ujian Munaqasah Tugas Akhir:

Ketua,	Sekretaris,
Hendri Ahmadian, S.Si., M.I.M	**Malahayati, M.T**
NIP. 198301042014031002	**NIP. 198301272015032003**
Penguji I,	Penguji II,
Khairan AR, M.Kom	**Baihaqi, M.T**
NIP. 198607042014031001	**NIP. 198802212022031001**

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

Dr. Ir. M. Dirhamsyah, M.T.,IPU
NIP.196210021988111001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Irvan Andika

NIM : 200705065

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul Tugas Akhir : Analisis Sentimen Persepsi Konsumen Terhadap Kualitas Layanan Aplikasi BSI Mobile Menggunakan Model IndoBERT

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

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

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

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

Banda Aceh, 20 Desember 2023

Yang Menyatakan

(Irvan Andika)

ABSTRAK

Nama	: Irvan Andika
NIM	: 200705065
Program Studi	: Teknologi Informasi
Judul	: Analisis Sentimen Persepsi Konsumen Terhadap Kualitas Layanan Aplikasi BSI Mobile Menggunakan Model IndoBERT
Tanggal Sidang	: 20 Desember 2023
Jumlah Halaman	: 95 Halaman
Pembimbing I	: Hendri Ahmadian, S.Si., M.I.M
Pembimbing II	: Malahayati, M.T
Kata Kunci	: BSI Mobile, Analisis Sentimen, IndoBERT, *Confusion Matrix*

Perkembangan internet memberikan inovasi baru pada layanan perbankan digital. Salah satu inovasi layanan perbankan digital adalah aplikasi *mobile banking*. Bank yang telah merilis aplikasi *mobile banking* adalah Bank Syariah Indonesia (BSI) dengan nama BSI Mobile. Penilaian dari persepsi konsumen diperlukan untuk mengetahui kelebihan dan kekurangan dari layanan aplikasi BSI Mobile. Penilaian terhadap suatu layanan aplikasi dapat diukur menggunakan data ulasan dan komentar. Upaya yang dilakukan untuk mengolah data dan menganalisis ulasan yang ada untuk mengklasifikasikannya menjadi ulasan positif, netral dan negatif melalui proses analisis sentimen. Penelitian ini menggunakan model *Bidirecitional Encoder Representations from Transformers* dalam bentuk bahasa Indonesia (IndoBERT) dengan model arsitektur *Transformer* untuk analisis sentimen. Dataset Penelitian ini berjumlah 23931 data. Dataset terdiri dari data komentar dan label sentimen. Pelabelan sentimen pada dataset menggunakan 2 cara, sesuai *rating* dan menggunakan TextBlob. Data di-*split* menjadi 2 bagian, 80% data *training* dan 20% data *validation.* Model IndoBERT dengan arsitektur *Transformer* pada penelitian ini menerapkan 8 skenario percobaan model dengan *hyperparameters* yang bervariasi. Model dibangun menggunakan bahasa pemograman *python*. Evaluasi terhadap performa model IndoBERT dalam menganalisis sentimen menggunakan *confusion matrix* dengan menghitung nilai *accuracy*, *precision*, *recall* dan *f*1-*score*. Hasil Performa model IndoBERT didapatkan hasil terbaik pada model dengan *batch size* 16 dan *epoch* 5 yang menggunakan dataset pelabelan sesuai *rating* dengan *accuracy* 0.903, *precision* 0.900, *recall* 0.903 dan *f*1-*score* 0.901. Hasil analisis sentimen menggunakan *confusion matrix* menunjukkan model dengan *batchsize* 16, *epoch* 5 dan *batch size* 32, *epoch* 10 pada dataset pelabelan sesuai TextBlob mendapatkan hasil terbaik dengan nilai yang didapatkan sama besar yaitu 0.94 untuk label positif, 0.64 netral dan 0.91 negatif.

ABSTRACT

Name	: Irvan Andika
Student Number	: 200705065
Department	: Teknologi Informasi
Title	: *Sentiment Analysis of Consumer Perceptions Quality of BSI Mobile Application Service Using IndoBERT Model*
Date	: 20 *December* 2023
Number of Pages	: 95 *Pages*
Supervisor I	: Hendri Ahmadian, S.Si., M.I.M
Supervisor II	: Malahayati, M.T
Keywords	: *BSI Mobile, Sentiment Analysis, IndoBERT, Confusion Matrix*

Internet developments provide new innovations in digital banking services. One of the digital banking service innovations is the mobile banking application. The bank that has released a mobile banking application is Bank Syariah Indonesia (BSI) with the name BSI Mobile. An assessment of consumer perceptions is needed to determine the advantages and disadvantages of the BSI Mobile application service. Assessment of an application service can be measured using review and comment data. Efforts are made to process data and analyze existing reviews to classify them into positive, neutral and negative reviews through a sentiment analysis process. This research uses the Bidirectional Encoder Representations from Transformers model in Indonesian (IndoBERT) with the Transformer architectural model for sentiment analysis. This research dataset consists of 23931 data. The dataset consists of comment data and sentiment labels. Sentiment labeling on the dataset uses 2 ways, according to ratings and using TextBlob. The data was split into 2 parts, 80% training data and 20% validation data. The IndoBERT model with Transformer architecture in this research applies 8 model experiment scenarios with varying hyperparameters. The model was built using the Python programming language. Evaluation of the performance of the IndoBERT model in analyzing sentiment using a confusion matrix by calculating accuracy, precision, recall and f1-score values. IndoBERT model performance results obtained the best results in the model with batch size 16 and epoch 5 which used a labeling dataset according to the rating with accuracy 0.903, precision 0.900, recall 0.903 and f1-score 0.901. The results of sentiment analysis using the confusion matrix show that the model with batch size 16, epoch 5 and batch size 32, epoch 10 on the labeling dataset according to TextBlob got the best results with the same values obtained, namely 0.94 for positive labels, 0.64 neutral and 0.91 negative.

KATA PENGANTAR

Bismillahirrahmanirrahim.

Puji syukur penulis panjatkan atas kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul "**Analisis Sentimen Persepsi Konsumen Terhadap Kualitas Layanan Aplikasi BSI Mobile Menggunakan Model IndoBERT**". Shalawat beserta salam semoga tersampaikan kepada Rasulullah SAW berserta keluarga dan sahabat beliau sekalian yang telah memperjuangkan umat Islam kepada jalan kebenaran dengan dibekali ilmu yang bermanfaat untuk dunia dan akhirat.

Penyusunan tugas akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan tugas akhir pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry. Dalam penulisan tugas akhir ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Cinta pertama dan pintu surga saya, Ibunda Roza Nofriza, yang tidak henti-hentinya memberikan kasih sayang dengan penuh cinta dan selalu memberikan dukungan kepada penulis disaat masa-masa tersulit dalam hidup. Terima kasih bunda, semoga Allah selalu memberikan kesehatan untukmu hingga bisa melihat putra sulungmu ini mencapai di titik tertinggi di dalam doamu.
2. Cinta kedua dan panutan saya, Ayahanda Samsul Bahri AM, terima kasih selalu berjuang untuk kehidupan penulis, beliau memang tidak sempat merasakan pendidikan sampai bangku perkuliahan. Namun, beliau mampu mendidik penulis, memotivasi dan memberikan dukungan hingga penulis mampu menyelesaikan studinya sampai sarjana.
3. Kepada cinta kasih kedua saudara kandung saya, Rava Dwi Putra dan Muhammad Farhan. Terima kasih atas segala doa, usaha, motivasi yang telah diberikan kepada abang pertama ini.

4. Bapak Hendri Ahmadian, S.Si., M.I.M selaku Pembimbing I dan Ibu Malahayati, M.T selaku Pembimbing II, sungguh besar rasa terima kasih penulis ucapkan atas bimbingan, kritik, saran dan selalu meluangkan waktunya di sela kesibukan. Menjadi salah satu dari anak bimbingan kalian merupakan nikmat yang selalu saya syukurkan. Semoga jerih payah kalian terbayarkan dan selalu dilimpahkan kesehatan.
5. Ibu Ima Dwitawati, MBA dan Bapak Khairan AR, M.Kom selaku Ketua dan Sekretaris Prodi Teknologi Informasi, yang telah memberikan bimbingan dan arahan dalam menyelesaikan tugas akhir ini.
6. Bapak Nazaruddin Ahmad, M.T sebagai Penasehat Akademik yang senantiasa memberikan arahan, motivasi dan nasehat kepada penulis dari semester 1 hingga semester 7. Semoga semua yang bapak berikan kepada saya dibalaskan di akhirat kelak.
7. Bapak Bustami, M.Sc yang telah memberikan ide menarik dan membuka wawasan kepada penulis terkait penelitian ini hingga dapat terselesaikan penulisan tugas akhir ini.
8. Ibu Cut Ida Rahmadiana, S.Si selaku *Staff* Prodi Teknologi Informasi, yang senantiasa membantu dan selalu sabar menjelaskan kepada penulis dalam pemberkasan administrasi.
9. Bapak Dr. Ir. M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.
10. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah membekali penulis dengan ilmu pengetahuan di bidang Teknologi Informasi.
11. Sahabat-sahabat yang selalu menghibur dan memberikan dukungan moral kepada penulis. Semoga persahabatan ini akan terus terjalin hingga kita bisa melihat titik sukses satu sama lain di masa depan nantinya.
12. Kepada teman-teman bisnis joki game yang selalu memberikan kata motivasi dan semangat kepada penulis.
13. Pihak-pihak terkait yang lainnya yang membantu penulis dalam penyusunan tugas akhir ini.

Penulis menyadari sepenuhnya bahwa dalam penyusunan tugas akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.
Amiin Ya Rabbal A'lamin.

Banda Aceh, 20 Desember 2023
Penulis

Irvan Andika

DAFTAR ISI

LEMBAR PERSETUJUAN ..... **i**
LEMBAR PENGESAHAN ..... **ii**
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR ..... **iii**
ABSTRAK ..... **iv**
ABSTRACT ..... **v**
KATA PENGANTAR ..... **vi**
DAFTAR ISI ..... **ix**
DAFTAR GAMBAR ..... **xii**
DAFTAR TABEL ..... **xv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Manfaat 3
1.6 Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA **5**
2.1 Penelitian Terdahulu 5
2.2 Persepsi Konsumen 8
2.3 *Mobile Banking* 8
2.3.1 BSI Mobile 9
2.4 Google Play Store 9
2.5 Analisis Sentimen 10
2.6 *Bidirectional Encoder Representations from Transformers* (BERT) 10
2.6.1 *Masked Language Modelling* (MLM) 13
2.6.2 *Next Sentence Prediction* 14
2.7 *Fine-tuning Bidirectional Encoder Representations from Transformers* (BERT) 15
2.8 IndoBERT 16
2.9 Python 16

2.10 Google Colaboratory 18
2.11 *Confusion Matrix* 18
2.12 Kerangka Pemikiran Penelitian 21
BAB III METODOLOGI PENELITIAN 22
3.1 Tahapan Penelitian 22
3.2 Studi Literatur 23
3.3 Pengumpulan Data 23
3.4 Pelabelan Data 23
3.5 *Pre-Processing* Data 24
3.6 *Split* Data 24
3.7 *Encode* 25
3.8 Implementasi Model IndoBERT 25
3.9 Evaluasi Model 25
3.10 Grafik dan Diagram 26
3.11 Dokumentasi 26
3.12 Alat Bantu Penelitian 26
BAB IV HASIL DAN PEMBAHASAN 27
4.1 Pengumpulan Data 27
4.2 Pelabelan Data 29
4.2.1 Pelabelan Data Sesuai *Rating* 29
4.2.2 Pelabelan Data Menggunakan TextBlob 30
4.2.3 Pelabelan Tahap Akhir 32
4.3 *Pre-Processing* Data 34
4.4 *Split* Data 35
4.5 *Encode* 37
4.6 Implementasi IndoBERT 41
4.7 Evaluasi Model 43
4.7.1 Evaluasi Performa Model Pelabelan Sesuai *Rating* 43
4.7.2 Evaluasi Performa Model Pelabelan Menggunakan TextBlob 47
4.8 Grafik dan Diagram 50
4.8.1 Grafik dan Diagram Model dari Pelabelan Sesuai *Rating* 51
4.8.1.1 Grafik Model dari Pelabelan Sesuai *Rating* 51

4.8.1.2 Diagram *Confusion Matrix* dari Pelabelan Sesuai *Rating* 56
4.8.2 Grafik dan Diagram Model dari Pelabelan Menggunakan TextBlob 60
4.8.2.1 Grafik Model dari Pelabelan Menggunakan TextBlob 60
4.8.2.2 Diagram *Confusion Matrix* dari Pelabelan Menggunakan TextBlob 65
4.9 Perbandingan Hasil Pelabelan Sesuai *Rating* dan TextBlob 69
BAB V KESIMPULAN DAN SARAN 73
5.1 Kesimpulan 73
5.2 Saran 73
DAFTAR PUSTAKA 74
LAMPIRAN 79
RIWAYAT HIDUP 87

DAFTAR GAMBAR

Gambar 2.1 Arsitektur *Transformer* (Vaswani et al., 2017)..... 11
Gambar 2.2 Representasi Input BERT (Devlin et al., 2018) 12
Gambar 2.3 Mekanisme *Pre-training* dan *Fine-tuning* (Devlin et al., 2018). 13
Gambar 2.4 Mekanisme *Masked Language Modelling* (Rahmatullah, 2021) 14
Gambar 2.5 *Fine-tuning* BERT (Devlin et al., 2018) 15
Gambar 2.6 Kerangka Pemikiran Penelitian..... 21
Gambar 3.1 Diagram Alir Tahapan Penelitian 22
Gambar 3.2 Proses *Encoder* (Vaswani et al., 2017) 25
Gambar 4.1 Ulasan Konsumen Terhadap Aplikasi BSI Mobile di *Website* Google Play Store 27
Gambar 4.2 *Split* Data *Train* dan *Validation* pada Dataset Pelabelan Sesuai *Rating* 36
Gambar 4.3 *Split* Data *Train* dan *Validation* pada Dataset Pelabelan Menggunakan TextBlob..... 37
Gambar 4.4 Contoh Kata pada *Vocabulary* IndoBERT 38
Gambar 4.5 Ilustrasi Proses *Fine-tuning* IndoBERT untuk Klasifikasi 42
Gambar 4.6 Performa Model dengan *Batch Size* 16 dan *Epoch* 5..... 43
Gambar 4.7 Performa Model dengan *Batch Size* 16 dan *Epoch* 10..... 44
Gambar 4.8 Performa Model dengan *Batch Size* 32 dan *Epoch* 5..... 45
Gambar 4.9 Performa Model dengan *Batch Size* 32 dan *Epoch* 10..... 46
Gambar 4.10 Performa Model dengan *Batch Size* 16 dan *Epoch* 5 47
Gambar 4.11 Performa Model dengan *Batch Size* 16 dan *Epoch* 10 48
Gambar 4.12 Performa Model dengan *Batch Size* 32 dan *Epoch* 5 49
Gambar 4.13 Performa Model dengan *Batch Size* 32 dan *Epoch* 10 50
Gambar 4.14 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 5..... 51
Gambar 4.15 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 10... 52
Gambar 4.16 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 5..... 52
Gambar 4.17 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 10... 53

Gambar 4.18 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 5 54
Gambar 4.19 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 10 54
Gambar 4.20 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 5 55
Gambar 4.21 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 10 56
Gambar 4.22 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 5 57
Gambar 4.23 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 10 58
Gambar 4.24 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 5 58
Gambar 4.25 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 10 59
Gambar 4.26 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 5 60
Gambar 4.27 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 10 ... 61
Gambar 4.28 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 5 61
Gambar 4.29 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 10 ... 62
Gambar 4.30 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 5 63
Gambar 4.31 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 10 63
Gambar 4.32 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 5 64
Gambar 4.33 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 10 65
Gambar 4.34 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch 5* 66

Gambar 4.35 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 10 67
Gambar 4.36 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 5 67
Gambar 4.37 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 10 68

جامعة الرانري
AR-RANIRY

DAFTAR TABEL

Tabel 2.1 *State of the Art* Penelitian 6
Tabel 2.2 *Confusion Matrix* untuk *multi-class classification* 18
Tabel 4.1 Contoh dari Hasil Pengumpulan Data..... 28
Tabel 4.2 Pelabelan Data Sesuai *Rating* 29
Tabel 4.3 Pelabelan Data Menggunakan TextBlob 30
Tabel 4.4 Pelabelan Akhir dari Pelabelan Sesuai *Rating*..... 32
Tabel 4.5 Pelabelan Akhir dari Pelabelan Menggunakan TextBlob..... 33
Tabel 4.6 Tahapan Pembersihan Data..... 34
Tabel 4.7 Proses Penyesuaian Input yang Dapat Diterima Oleh IndoBERT. 39
Tabel 4.8 *Hyperparameter* IndoBERT 43
Tabel 4.9 Perbandingan Nilai Evaluasi Performa Model..... 69
Tabel 4.10 Perbandingan Hasil Analisis Sentimen 71

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan internet telah membuka banyak inovasi baru pada layanan perbankan digital, salah satu inovasi layanan perbankan digital adalah munculnya aplikasi *mobile banking* (Sari et al., 2021). Salah satu bank yang telah merilis aplikasi *mobile banking* adalah Bank Syariah Indonesia (BSI) dengan nama aplikasi BSI Mobile. Sebagai aplikasi yang masih tergolong baru, tentunya memerlukan penilaian dari persepsi konsumen untuk mengetahui kelebihan dan kekurangan dari layanan aplikasi BSI Mobile. Sehingga dari hasil penilaian tersebut BSI Mobile dapat meningkatkan dan mengembangkan layanan yang sesuai dengan kebutuhan konsumen (Arifiyanti et al., 2022). Penilaian terhadap suatu layanan aplikasi dapat diukur menggunakan data ulasan, komentar dan emosi pada suatu aplikasi (Kavabilla et al., 2023). Namun, banyaknya data ulasan menyebabkan perusahaan sulit untuk melakukan analisis. Oleh karena itu, analisis sentimen diperlukan untuk mengolah data dan menganalisis ulasan yang ada untuk mengklasifikasikannya menjadi ulasan positif, netral, atau negatif secara otomatis (Akhmad, 2023).

Analisis sentimen menjadi salah satu metode yang digunakan untuk mengolah data ulasan yang diberikan oleh pengguna melalui berbagai media mengenai suatu produk, layanan ataupun sebuah lembaga (Evasaria M. Sipayung et al., 2016). Ulasan tersebut mampu mengidentifikasi kategori aspek di dalam aplikasi BSI Mobile mendapatkan penilaian baik atau buruk dari para pengguna. Ulasan perlu diekstrak terlebih dahulu agar didapatkan informasi mengenai opini pengguna terhadap aplikasi (Utami et al., 2023).

Beberapa model arsitektur yang dapat digunakan untuk analisis sentimen seperti, *Long Short-Term Memory* (LSTM), *Convolutional Neural Network* (CNN) dan *Transformer*. Model Arsitektur LSTM memiliki beberapa kelebihan antara lain memiliki kemampuan memahami urutan dan memproses data berurutan secara real-time. Kekurangan metode LSTM adalah keterbatasan pemahaman konteks jangka

panjang dan komputasi yang lambat (Wang et al., 2018). Model arsitektur menggunakan CNN memiliki kelebihan mampu mengekstraksi fitur otomatis dan melakukan pendekatan paralel. Kekurangan metode CNN adalah ketergantungan pada data berlabel dan keterbatasan dalam memahami konteks jangka panjang (Elngar et al., 2021). Kelebihan model arsitektur menggunakan *Transformer* adalah memiliki kemampuan pemahaman konteks jangka panjang dan menggunakan *pre-trained models*. Kekurangan *Transformer* adalah keterbatasan interpretabilitas dan rentan terhadap *overfitting* (Zhu et al., 2023). *Bidirectional Encoder Representations from Transformers* (BERT) adalah *stack encoder* yang dibangun dengan menerapkan arsitektur *Transformer* (Sudhir & Suresh, 2021).

BERT dibangun sebagai model *pre-trained* yang telah dilatih secara *deep bidirectional* pada dataset yang tidak diberi label dengan menyatukan kondisi sisi kanan dan kiri lapisan. Model BERT dapat dimodifikasi untuk melakukan tugas *deep learning* lainnya, seperti klasifikasi teks dan *question-answering* dengan hanya menambahkan satu lapisan saja (Devlin et al., 2018). BERT awalnya dikembangkan oleh J. Devlin dkk pada tahun 2018 yang dipakai oleh Google pada *Search Engine Optimization* untuk menyesuaikan hasil eksplorasi dengan kata kunci yang dimasukkan pengguna. Saat ini, sudah terdapat beragam bahasa untuk Model *pre-trained* BERT termasuk bahasa Indonesia. IndoBERT merupakan model *pre-trained* BERT dalam bentuk bahasa Indonesia (Wilie et al., 2020).

Penelitian mengenai analisis sentimen berbahasa Indonesia menggunakan IndoBERT saat ini masih sangat minim (Rahmatullah, 2021). Pada penelitian ini akan dilakukan analisis sentimen persepsi konsumen terhadap kualitas layanan aplikasi BSI Mobile yang diekstrak dari ulasan pada kolom komentar *website* Google Play Store menggunakan IndoBERT. Dari hasil penelitian ini diharapkan dapat mengetahui proses, performa dan keakuratan IndoBERT dalam melakukan analisis sentimen persepsi konsumen terhadap kualitas layanan aplikasi BSI Mobile serta mengklasifikasikannya menjadi sentimen positif, netral atau negatif. Sehingga dapat menjadi nota untuk pihak BSI Mobile untuk lebih mengembangkan dan meningkatkan layanan untuk konsumen atau nasabah.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah penelitian ini adalah:

1. Bagaimana proses IndoBERT dalam melakukan analisis sentimen?
2. Bagaimana performa dan keakuratan IndoBERT dalam melakukan analisis sentimen?

1.3 Tujuan Penelitian

Penelitian ini bertujuan sebagai berikut:

1. Mengetahui proses IndoBERT dalam melakukan analisis sentimen.
2. Mengetahui performa dan keakuratan IndoBERT dalam melakukan analisis sentimen.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Pembuatan sistem dibangun menggunakan bahasa Python dan dieksekusi menggunakan Google Colab.
2. Data yang digunakan adalah data ulasan pengguna aplikasi BSI Mobile yang berbahasa Indonesia di *website* Google Play Store.
3. Menggunakan model IndoBERT-Large phase 1 dari platform Hugging Face untuk model *pre-trained*.

1.5 Manfaat

Manfaat dari penelitian ini sebagai berikut:

1. Bagi *mobile app developer* dapat dimanfaatkan sebagai pertimbangan dalam menciptakan dan mengembangkan layanan aplikasi.
2. Bagi industri dapat dimanfaatkan sebagai peninjauan dan penilaian dalam upaya meningkatkan kualitas layanan kepada pengguna aplikasi.
3. Bagi pembaca, dapat dimanfaatkan sebagai referensi untuk mengetahui proses, performa dan keakuratan IndoBERT dalam melakukan analisis sentimen.

1.6 Sistematika Penulisan

Untuk dapat mengetahui isi penelitian ini, maka secara singkat akan disusun dalam sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

   Bab ini berisi gambaran umum tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

   Bab ini memuat materi-materi yang menunjang penelitian ini.

3. BAB III METODE PENELITIAN

   Bab ini menjelaskan berbagai metode yang digunakan untuk penelitian ini dapat selesai.

4. BAB IV HASIL DAN EVALUASI

   Bab ini berisi tentang hasil dari penerapan model. Kemudian dilakukan evaluasi model menggunakan *confusion matrix*.

5. BAB V PENUTUP

   Bab ini memuat tentang kesimpulan serta saran jika penelitian ini ingin dikembangkan.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian analisis sentimen persepsi konsumen terhadap kualitas layanan Gojek menggunakan IndoBERT (Mahfudiyah & Alamsyah, 2022). *Crawling* data dilakukan pada media sosial Twitter yang dikumpulkan dalam rentang waktu 3 bulan, terhitung mulai tanggal 1 Oktober 2021 sampai 31 Desember 2021. *Keywords* yang digunakan adalah "gojek" dan "gojekindonesia". Data yang dikumpulkan sebanyak 32895 data. Didapatkan hasil *accuracy* tertinggi dengan varian pembagian data 80:10:10 sebesar 96% untuk *accuracy*, *precision* 93% dan *recall* 95%.

Kuncahyo Setyo N,dkk (Nugroho et al., 2021) melakukan analisis sentimen menggunakan IndoBERT. Dalam penelitian ini dilakukan analisis sentimen terhadap data ulasan 10 aplikasi terbaik di Google Play Store di tahun 2020. Data diambil dengan teknik *scraping* sebanyak 10.615 data. Pelabelan data dilakukan dengan 2 cara yaitu, cara pertama dengan *rating* yang diberikan oleh ulasan reviewer. *Rating* 1 dan 2 dilabel negatif, *rating* 3 dilabel netral, *rating* 4 dan 5 dilabel positif. Cara kedua dilakukan pelabelan dengan metode *lexicon-based.* Kemudian dilakukan komparasi peforma analisis sentimen antara BERT-*base-multilingual* dan IndoBERT-*base*. Hasil menampilkan bahwa *accuracy* yang cukup baik pada model IndoBERT-*base* menggunakan teknik pelabelan *lexicon-based* yaitu 84% dengan 25 *epoch* dan menghabiskan waktu *training* selama 24 menit.

Penelitian analisis sentimen ulasan pada Google Play Store menggunakan IndoBERT juga pernah dilakukan oleh Ekka Pujo Ariesanto Akhmad (Akhmad, 2023). Data diambil menggunakan teknik *scraping* pada kolom komentar pada aplikasi DLU Ferry di Google Play Store. Proporsi pembagian dataset adalah 70% data *training*, 20% data *validation*, 10% data *test.* Peneliti menggunakan teknik *fine-tuning* dengan model IndoBERT-base-p1. Dengan menggunakan *batch size* 32, *learning rate* 3e-6, dan *epoch 5* menghasilkan *f*1-*score* sebesar 86%.

Fransiscus dan Abba Suganda Girsang melakukan penelitian tentang analisis sentimen terhadap dampak PPKM pada periode COVID-19 menggunakan IndoBERT (Fransiscus & Girsang, 2022). Peneliti menggunakan Twitter dev API untuk mengumpulkan data dengan *keyword* "PPKM" dan "Jakarta". Total *record data* yang dikumpulkan adalah 50.000 data, yang dikumpulkan dengan rentang waktu dari tanggal 1 Januari sampai 30 Maret 2022. Pelabelan data dilakukan secara manual oleh 2 orang. Jumlah data yang sudah berlabel dan bersih dari data duplikat adalah 5315 data. Kemudian dilakukan pembagian data menjadi 4877 data t*raining*, 293 data *validation* dan 145 data *testing*. Hasil uji model yang didapatkan dengan 10 *epoch* adalah *precision* 86%, *recall* 84% dan *f*1-*score* 84%.

Penelitian mengenai analisis sentimen smartphone Xiaomi menggunakan IndoBERT dilakukan oleh Priyan Fadhil Supriyadi dan Yuliant Sibaroni (Supriyadi & Sibaroni, 2023). Data diambil pada media Twitter dengan *keyword* "Xiaomi" dan "redmi" menggunakan teknik *scraping* sebanyak 3801 data. Data dibagi menjadi 2 yaitu data *traning* dan data *test* dengan proporsi 80:20. Kemudian dilakukan komparasi hasil sentimen analisis menggunakan BERT dan IndoBERT. Hasil akhir dari *accuracy* IndoBERT mempunyai skor *accuracy* tertinggi sampai 90% dalam *epoch* 4 pada setiap lipatan.

Tabel 2.1 *State of the Art* Penelitian

PENELITIAN	MODEL	JUDUL	HASIL	CELAH PENELITIAN
(Mahfudiyah & Alamsyah, 2022)	IndoBERT (Wilie et al., 2020)	Analisis Persepsi Konsumen Terhadap Kualitas Layanan Gojek Menggunakan Sentiment Analysis dan *Topic Modeling* Berdasarkan *Deep*	*Accuracy* 96%, *precision* 93% dan *recall* 95%	Dataset yang digunakan tidak seimbang tetapi tidak menggunakan matriks evaluasi *f*1-*score* untuk mengevaluasi

		Learning IndoBERT		performa model.
(Nugroho et al., 2021)	IndoBERT (Wilie et al., 2020)	*BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews*	*Accuracy* 84%	Hanya menggunakan a*ccuracy* sebagai matriks evaluasi performa model.
(Akhmad, 2023)	IndoBERT (Wilie et al., 2020)	Analisis Sentimen Ulasan Aplikasi DLU Ferry Pada Google Play Store Menggunakan *Bidirectional Encoder Representations from Transformers*	*Accuracy* 86%, *Precision* 86%, *recall* 94%, dan *f*1-*score* 89%	Jumlah dataset yang tidak besar menyebabkan model kesulitan untuk klasifikasi sentimen netral
(Fransiscus & Girsang, 2022)	IndoBERT (Koto et al., 2020)	*Sentiment Analysis of COVID-19 Public Activity Restriction (PPKM) Impact using BERT Method*	*Precision* 86%, *recall* 84% dan *f*1-*score* 84%	Tidak menggunakan *accuracy* sebagai matriks evaluasi performa model

(Supriyadi & Sibaroni, 2023)	IndoBERT (Koto et al., 2020)	*Xiaomi Smartphone Sentiment Analysis on Twitter Social Media Using IndoBERT*	*Accuracy* 90%	Hanya menggunakan *accuracy* sebagai matriks evaluasi performa model

2.2 Persepsi Konsumen

Persepsi adalah adalah proses yang menjadikan kita sadar akan ransangan stimulus terhadap indra kita. Stimulus dipengaruhi oleh respon yang didapatkan dari perhatian individu terhadap suatu objek. Hasil persepsi mempengaruhi setiap pesan apa yang kita tangkap serta apa makna setiap kita menilai informasi yang kita dapatkan (Jayanti & Arista, 2019).

Persepsi konsumen adalah proses di mana konsumen memahami dan menginterpretasikan informasi yang berkaitan dengan produk, layanan, merek, atau pengalaman pembelian dalam lingkungan pemasaran. Ini adalah proses kognitif dan emosional yang kompleks yang mempengaruhi bagaimana konsumen bertindak dan membuat keputusan pembelian. Persepsi konsumen dipengaruhi oleh berbagai faktor, seperti: pengalaman sebelumnya, pengetahuan dan informasi, sikap dan nilai-nilai, pengaruh sosial, kualitas produk dan citra merek.

Penting bagi pemasar untuk memahami persepsi konsumen karena dapat mempengaruhi keputusan konsumen membeli. Persepsi yang positif tentang produk atau merek dapat mendorong konsumen untuk membeli dan menjadi pelanggan setia, sementara persepsi negatif dapat menyebabkan penolakan atau memilih produk atau merek pesaing (Sardanto & Ratnanto, 2016).

2.3 *Mobile Banking*

Mobile banking adalah aplikasi, perangkat lunak atau program yang dirancang khusus untuk layanan perbankan. Sehingga memungkinkan nasabah

untuk melakukan berbagai aktivitas perbankan melalui perangkat seluler. *Mobile banking* memberdayakan nasabah untuk melakukan layanan keuangan dengan menggunakan *smartphone* yang menghubungkan pelanggan ke server untuk melakukan pembayaran, transaksi, dan melakukan layanan lainnya (Zhou et al., 2021).

2.3.1 BSI Mobile

BSI Mobile adalah aplikasi *mobile* yang berperan sebagai fasilitator *mobile banking* Bank Syariah Indonesia bagi para nasabah. BSI Mobile adalah *mobile banking* yang beroperasi berdasarkan prinsip-prinsip Bank Syariah Islam. Prinsip ini mencakup larangan riba (bunga), maysir (perjudian), gharar (ketidakpastian) dan aktivitas yang bertentangan dengan prinsip etika Islam (Rantemangiling et al., 2022). BSI Mobile memiliki beberapa fitur layanan yang sangat memudahkan para nasabahnya, seperti: buka rekening secara online, mengakses rekening tabungan nasabah untuk melakukan transaksi secara online, info rekening, transfer uang, melakukan pembelian dan pembayaran suatu produk, dan *top-up* dompet digital. Sebagai aplikasi *mobile banking* yang islami, BSI Mobile juga memberikan fitur layanan islami, berbagi ziwaf dan notifikasi ketika masuknya waktu sholat.

2.4 Google Play Store

Google Play Store adalah salah satu platform distribusi aplikasi *mobile* terbesar dan terpopuler di dunia. Jutaan pengguna Android di seluruh dunia menggunakan layanan ini untuk mendapatkan akses ke berbagai aplikasi yang mereka butuhkan. Selain itu, Google Play Store juga menyediakan fitur tambahan seperti pembaruan aplikasi otomatis, rekomendasi aplikasi berdasarkan preferensi pengguna, penilaian dan ulasan pengguna.

Kolom komentar pada aplikasi yang berisikan *rating* dan masukkan dari pengguna bisa menjadi sumber informasi untuk membangun reputasi sebagai peningkatan layanan di bidang tertentu, mempromosikan bisnis, produk atau konten, dan dapat dimanfaatkan sebagai sumber mendapatkan berbagai data opini konsumen untuk dilakukan analisi sentimen (Firmansyah & Ahsan, 2023).

2.5 Analisis Sentimen

Analisis sentimen adalah proses komputasi untuk mengidentifikasi dan mengkategorikan pendapat yang diungkapkan dalam sebuah kalimat, terutama untuk menentukan sikap dan persepsi penulis terhadap topik tertentu, produk, layanan dan isu faktual tergolong ke dalam sentimen positif, negatif atau netral (Bonta et al., 2019). Analisis sentimen yang diterapkan pada Google Play Store berfungsi untuk mencari informasi, dan opini pengguna serta makna data postingan, data tersebut dapat berupa *rating*, emoji, *likes* dan komentar.

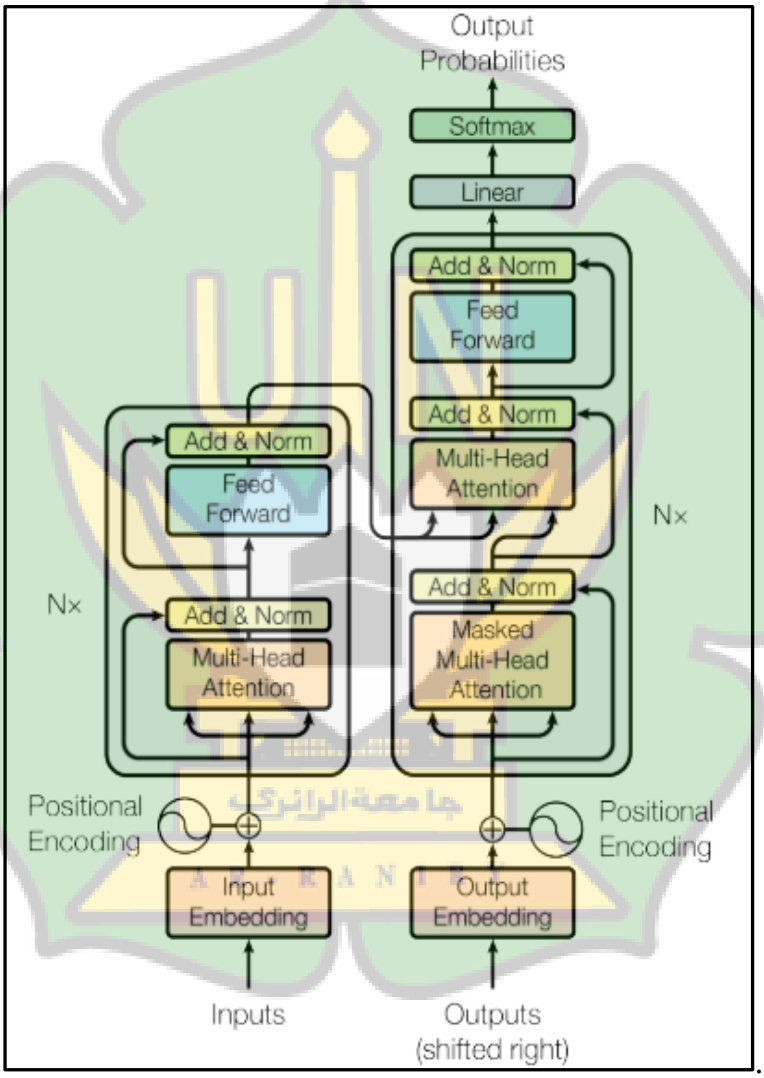
Dalam dunia usaha analisis sentimen bisa membantu dalam menganalisis trend, perasaan pelanggan sehingga dan kualitas layanan sehingga dapat digunakan untuk mengembangkan produk. Analisis sentimen juga dapat membantu perusahaan untuk mengetahui opini yang diberikan oleh konsumen terhadap kualitas layanan aplikasi yang sudah diberikan. Tujuan dari analisis sentimen adalah untuk mengekstraksi attribut dalam dokumen atau teks yang mengandung komentar agar dapat mengetahui respon di dalamnya, sehingga dapat digolongkan menjadi respon positif, negatif dan netral (Herwinsyah & Witanti, 2022).

2.6 *Bidirectional Encoder Representations from Transformers* (BERT)

Bidirectional Encoder Representations from Transformers (BERT) merupakan model *Natural Language Processing* (NLP) yang dipopulerkan oleh Google pada tahun 2018. Model ini dirancang untuk menghasilkan representasi vektor yang informatif dari teks dengan memanfaatkan arsitektur *Transformer* (Devlin et al., 2018).

Multi-layer bidirectional Transformer encoder merupakan kerangka utama dalam arsitektur BERT. Dalam rangkaian lapisan-lapisan tersebut, BERT memanfaatkan konsep *Transformer,* yaitu arsitektur jaringan neural yang fokus pada perhitungan *self-attention* (Vaswani et al., 2017). *Self-attention* memungkinkan model untuk memahami konteks global dari seluruh kalimat atau dokumen, sehingga dapat menghasilkan representasi vektor yang lebih baik untuk setiap kata atau token dalam teks. *Transformer* menerapkan teknik *self-attention* untuk mempelajari representasi input dan output. *Transformer* terdiri dari 2

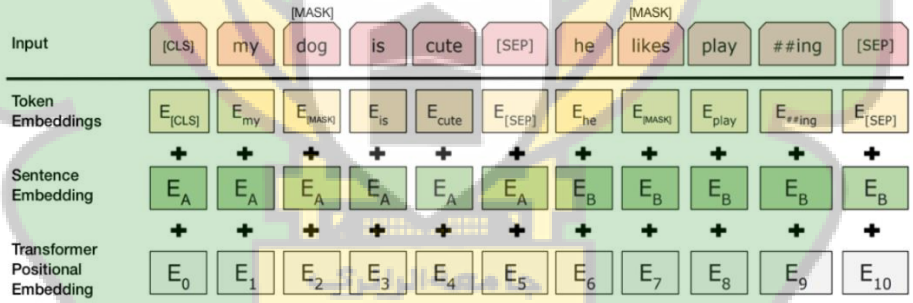
mekanisme terpisah, yaitu *encoder* bertugas untuk membaca teks input dan decoder yang menentukan prediksi. *Encoder* menghasilkan representasi *anttention-based* dengan kemampuan menemukan potongan informasi dari konteks yang memiliki potensi besar. *Decoder* bisa mengambil dari representasi yang di-*encode* (Devlin et al., 2018). **Gambar 2.1** menunjukkan arsitektur *Transformer.*



Gambar 2.1 Arsitektur *Transformer* (Vaswani et al., 2017)

BERT menggunakan *WordPiece embeddings* sebagai salah satu komponen penting dalam representasi kata atau token. *Wordpiece embeddins* adalah teknik *word embeddings* yang digunakan dalam pemrosesan BERT dengan 30.000 token

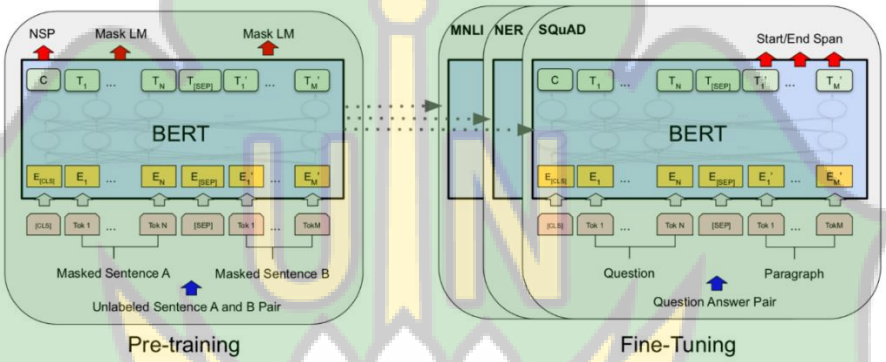
pada vocabulary. Teknik ini adalah pengembangan dari teknik *subword embedding* yang pertama kali diperkenalkan oleh Google dalam proyek pengembangan bahasa alami. *Wordpiece embeddings* mengambil pendekatan *bottom-up* untuk membangun kata representasi vektor. Dengan menggunakan *WordPiece*, kata-kata dalam kalimat dipisahkan menjadi potongan-potongan yang lebih kecil yang disebut *subword* token. *Subword* token adalah unit token terkecil yang digunakan oleh BERT yang bertujuan untuk menjadikan BERT lebih fleksibel dalam mengatasi kata-kata yang tidak dikenal atau *Out-of-Vocabulary* (OOV) (Chen et al., 2019). Proses *tokenization WordPiece* biasanya melibatkan pembentukan kamus yang terdiri dari kata-kata yang paling sering muncul dalam data pelatihan. Namun jika ada kata-kata yang tidak ada dalam kamus atau OOV, kata-kata tersebut akan dipecah menjadi subword token yang lebih kecil sehingga model tetap dapat memproses dan memahaminya. **Gambar 2.2** merupakan Representasi dari Input BERT.



Gambar 2.2 Representasi Input BERT (Devlin et al., 2018)

Input yang diterima oleh *Encoder* adalah adalah total dari *Token Embedding*, *Sentence Embedding*, dan *Transformer Positional Embedding*. Token *special class classification* [CLS] dipasang di awal sequence karena bertindak sebagai pengganti kelas atau label output dari *sequence* tersebut. Token dari setiap kalimat dalam dokumen akan ditempatkan secara berurutan dan dipisahkan oleh token [SEP]. Untuk menandakan posisi token pada kalimat yang ditambahkan *positional embedding*.

Dalam kinerja *framework* BERT, terdapat 2 tahapan yaitu *pre-training* dan *fine-tuning*. Pada tahapan *pre-training*, model dilatih pada dataset besar yang tidak berlabel. Model belajar tentang bahasa melalui proses gabungan *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). MLM mengubah kata acak dalam kalimat menjadi token [MASK] dan tugas model memprediksi kata-kata tersebut berdasarkan konteks kiri dan kanan. NSP membantu model memahami hubungan antara dua kalimat dalam pasangan teks (Devlin et al., 2018).

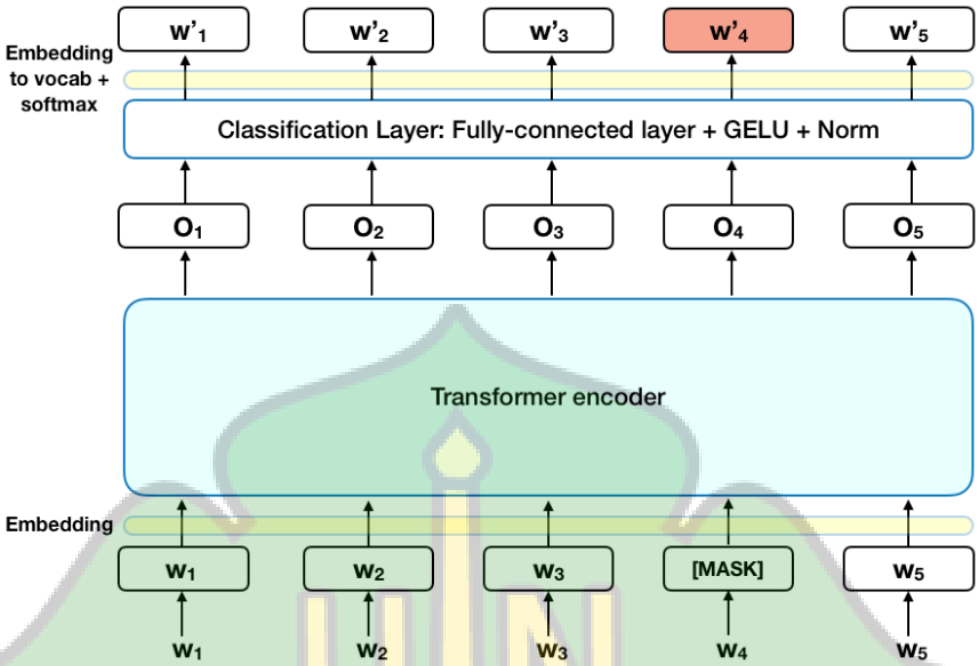


Gambar 2.3 Mekanisme *Pre-training* dan *Fine-tuning* (Devlin et al., 2018)

2.6.1 *Masked Language Modelling* (MLM)

Masked Language Modeling (MLM) adalah teknik pembelajaran mesin yang digunakan dalam pemrosesan bahasa alami. Teknik ini bertujuan untuk memprediksi kata yang dihilangkan dalam teks yang diberikan. MLM biasanya digunakan dalam model arsitektur pemrosesan bahasa alami yang berbasis *transformer* seperti BERT.

Teknik MLM bekerja dengan cara mengambil dan menghilangkan beberapa kata secara acak dalam setiap kalimat. Model tersebut kemudian memprediksi kata sebenarnya sesuai konteks kalimat pada *vocabulary*. Supaya model berhasil memprediksikan kata, model harus ditambahkan lapisan klasifikasi di atas output *encoder*. Setelah itu, vektor output dikalikan dengan matriks *embedding* untuk mengubahnya ke dalam dimensi *vocabulary*. Selanjutnya, dilakukan perhitungan probabilitas dari setiap kata dalam *vocabulary* menggunakan *softmax.*



Gambar 2.4 Mekanisme *Masked Language Modelling* (Rahmatullah, 2021)

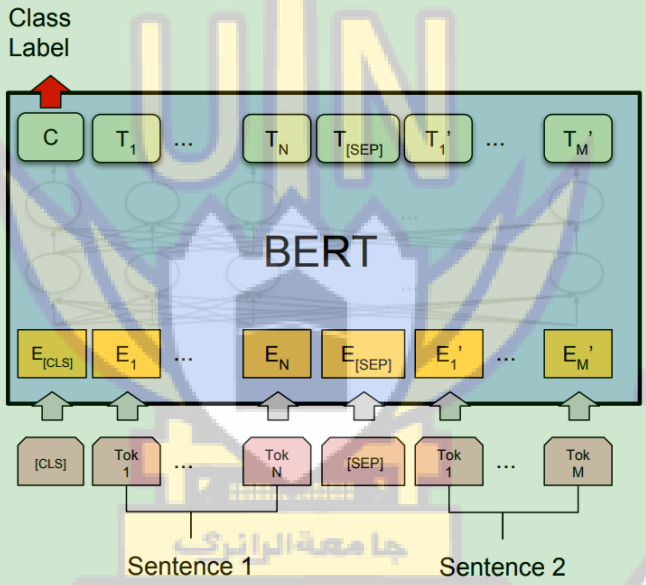
2.6.2 *Next Sentence Prediction*

Dalam *Next Sentence Prediction* (NSP) model diberikan dua kalimat dan harus memprediksi apakah kalimat kedua mengikuti kalimat pertama. Tujuan dari tugas ini adalah untuk melatih model dalam memahami hubungan antara dua kalimat dan memperoleh pemahaman yang lebih baik tentang konteks kalimat pada dataset. Selama *training*, sebagian input menjadi pasangan kalimat pertama, sementara sisanya adalah kalimat acak yang dipilih dari korpus teks menjadi kalimat kedua (Devlin et al., 2018).

Hasil dari *pre-training* menghasilkan model BERT dengan bobot (*weight*) dan representasi vektor dari kata-kata dalam korpus teks yang digunakan untuk pelatihan. Dalam tahap *pre-training*, BERT belajar untuk memahami hubungan antar kata dalam konteks bahasa dengan tujuan akhir mengembangkan representasi kata yang kaya dan kontekstual. Representasi vektor ini kemudian dapat digunakan untuk tugas-tugas pemrosesan bahasa alami seperti klasifikasi, question-answering, dan lainnya melalui tahap fine-tuning (Rahmatullah, 2021).

2.7 *Fine-tuning Bidirectional Encoder Representations from Transformers* (BERT)

Fine-tuning Bidirectional Encoder Representations from Transformers (BERT) adalah teknik yang digunakan untuk memodifikasi model pemrosesan bahasa alami BERT dengan melatih model pada tugas tertentu yang spesifik, seperti klasifikasi teks atau pengenalan entitas bernama. Proses ini dilakukan dengan cara mengubah sedikit parameter atau struktur model yang ada sehingga dapat melakukan tugas yang baru. **Gambar 2.5** menunjukkan proses dari *Fine-tuning* BERT.



Gambar 2.5 *Fine-tuning* BERT (Devlin et al., 2018)

Dalam fase *fine-tuning*, semua *hyperparameter* yang digunakan dalam proses *pre-training* tidak bervariasi kecuali *learning rate*, *batch size* dan *epoch*. Lapisan terakhir dari model BERT yang disebut "*classifier layer*", biasanya diubah untuk mengakomodasi tugas yang spesifik. Kemudian, model diterapkan pada tugas spesifik data tersebut dan bobotnya ditingkatkan secara iteratif melalui algoritma pembelajaran mesin.

2.8 IndoBERT

IndoBERT adalah *pre-trained model* untuk memproses teks dalam berbahasa Indonesia. IndoBERT mengadopsi prinsip mekanisme perubahan yang memungkinkannya untuk memahami relasi antar kata dalam sebuah kalimat (Hidayat & Maharani, 2022). IndoBERT memanfaatkan dataset Indo4B yang mencakup lebih dari 23 GB data teks bahasa Indonesia, terdiri dari 3.6 Miliar kata yang meliputi bentuk formal dan *colloquial* (bahasa sehari-hari) yang diambil dari sosial media, *website*, blog dan berita (Wilie et al., 2020). IndoBERT menggunakan SentencePiece dengan Byte Pair Encoding (BPE) tokenizer sebagai metode untuk membentuk *vocabulary* atau daftar *subword* token. SentecePiece adalah tokenizer yang berfungsi untuk membagi teks menjadi *subword* token yang lebih kecil, dan BPE adalah salah satu algoritma pembentuk *subword* token yang efektif.

IndoBERT memiliki empat varian model, yakni IndoBERT-Base, IndoBERT-Large, IndoBERT-liteBase, IndoBERT-liteLarge. Semua model IndoBERT dilatih menggunakan TPU v3-8 dalam dua tahapan. Tahapan pertama model dilatih dengan *maximum sequence length* 128, dan tahapan kedua model dilatih dengan *maximum sequence length* 512 (Wilie et al., 2020). Dari semua model IndoBERT, IndoBERT-Large memiliki keakuratan paling tinggi, namun memerlukan kapasitas memori yang besar (Rahmatullah, 2021). IndoBERT-Large merupakan modifikasi dari BERT-large yang sudah mengikuti kofigurasi dari BERT-Large (uncased) yang memiliki 24 *hidden layers*, 1024 *hidden size*, 16 *attention heads*, 4096 dimensi *feed-forward hidden layers* dan *languange type* Monolingual (Wilie et al., 2020).

2.9 Python

Python merupakan bahasa pemograman interpretatif yang dirancang untuk membaca kode dan sintaks dengan tujuan mempermudah programmer membuat berbagai skema rancangan yang diimplementasikan ke dalam baris kode serta bersifat *multiplatform* (Wati & Ernawati, 2021).

Python dikenal dengan sintaksisnya yang mudah dipahami dan bahasa yang fleksibel serta dapat diaplikasikan dalam berbagai macam lingkungan. Python juga

mendukung berbagai macam pemrograman paradigma, termasuk pemrograman berorientasi objek, pemrograman fungsional, dan pemrograman procedural. Terdapat berbagai macam *librabry* Python yang dapat digunakan untuk melakukan analisis data (Nongthombam & Sharma, 2021), yaitu:

- Scrapy

Scrapy adalah *framework open-source* yang digunakan untuk mengekstraksi data dari situs web. Scrapy dapat membuat program yang bisa mengekstraksi informasi dari berbagai situs web dan menyimpannya dalam format yang terstruktur, seperti JSON atau CSV.

- Panda

Panda atau sering disebut pandas adalah sebuah *library* Python yang populer digunakan dalam analisis data dan kecerdasan buatan. Pandas menyediakan struktur data yang fleksibel dan efisien untuk memanipulasi dan menganalisis data, terutama tabular data seperti *spreadsheet* atau database.

- NumPy (*Numerical Python*)

NumPy adalah *library* Python yang berfokus pada komputasi numerik, numpy menyediakan struktur data dan fungsi untuk mendukung operasi matematika yang efisien dan cepat, terutama untuk array dan matriks multidimensi

- Scikit-Learn

Scikit-Learn atau sederhananya sklearn adalah *library* Python *open source* yang digunakan untuk *machine learning*. *Library* ini dapat digunakan untuk membagi data menjadi subset pelatihan dan pengujian.

- Matplotlib

Matplotlib adalah *library* Python yang digunakan untuk membuat visualisasi data dengan berbagai jenis plot, grafik dan gambar.

- TextBlob

TextBlob adalah sebuah *library* Python yang digunakan untuk memproses data teks secara sederhana. *Library* ini menyediakan antarmuka yang mudah digunakan untuk melakukan berbagai tugas pemrosesan teks, seperti analisis sentimen, pengklasifikasian teks, ekstraksi frasa kata kunci, pemecahan frasa, penerjemahan, dan lain sebagainya.

2.10 Google Colaboratory

Google Colaboratory atau disebut juga Google Colab adalah sebuah *Integrated Development Environtment* (IDE) atau layanan komputasi awan yang disediakan oleh Google untuk melakukan pemograman dan analisis data secara online melalui *notebook* berbasis Python (Guntara, 2023). Google Colab menggunakan Jupyter Notebook yang memungkinkan pengguna untuk menulis dan mengeksekusi kode Python.

2.11 *Confusion Matrix*

Confusion matrix adalah sebuah tabel yang digunakan untuk mengevaluasi performa model klasifikasi atau prediksi. Tabel ini menggambarkan jumlah data yang diprediksi dengan benar atau salah oleh model dan dibagi menjadi empat bagian yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative. Confusion Matrix Multiclass* adalah sebuah tabel yang sering digunakan untuk mendeskripsikan performa dan kinerja dari model klasifikasi dalam menganalisis sentimen pada kumpulan data set *training* dan *validation* dengan nilai aktual yang sudah diketahui (Grandini et al., 2020). **Tabel 2.2** menunjukkan *confusion matrix* untuk *multi-class classification.*

Tabel 2.2 *Confusion Matrix* untuk *multi-class classification*

Classes		*Predicted Classification*		
		X	Y	Z
Actual Classification	X	TN	FN	FP
	Y	FN	TN	FP
	Z	FN	FP	TP

Evaluasi sistem dilakukan dengan menggunakan *confusion matrix multiclass* dan menghitung nilai *accuracy, precision, recall* dan *f*1-score dari setiap kelas. Dengan menggunakan tabel *confusion matrix*, berbagai nilai yang digunakan untuk evaluasi performa model seperti *accuracy*, *precision*, *recall*, dan *f*1-*score* dapat dihitung untuk mengetahui performa model dalam melakukan analisis sentimen.

Accuracy merupakan salah satu matriks evaluasi performa model yang mengukur seberapa banyak prediksi yang benar dari total jumlah data yang diprediksi. Secara matematis, penghitungan dihitung dengan cara membagi jumlah prediksi yang benar dengan total jumlah data yang diprediksi. Untuk menghitung nilai *accuracy* menggunakan persamaan **(1)**: (Grandini et al., 2020).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad \textbf{(1)}$$

Precision adalah salah satu matriks evaluasi performa model yang mengukur seberapa banyak prediksi positif yang benar dari total jumlah data yang diprediksi positif. *Precision* didefinisikan sebagai rasio antara jumlah data positif yang diprediksi benar dengan jumlah total data yang diprediksi positif. Untuk menghitung nilai *precision* menggunakan persamaan **(2)**: (Grandini et al., 2020).

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(2)}$$

Recall adalah salah satu matriks evaluasi performa model yang mengukur seberapa banyak data positif yang berhasil diidentifikasi dari total jumlah data positif yang sebenarnya. *Recall* didefinisikan sebagai rasio antara jumlah data positif yang diprediksi benar dengan jumlah total data positif yang sebenarnya. Untuk menghitung nilai *recall* menggunakan persamaan **(3)**: (Grandini et al., 2020).

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(3)}$$

*F*1-*score* adalah salah satu matriks evaluasi performa model yang menggabungkan *precision* dan *recall* menjadi satu skor yang menggambarkan keseimbangan antara kedua matriks tersebut. *F*1-*score* dapat dianggap sebagai rata-rata harmonik dari *precision* dan *recall*, skor tertinggi diperoleh ketika *precision* dan *recall* sama-sama tinggi. Untuk menghitung nilai *f*1-*score* menggunakan persamaan **(4)**: (Grandini et al., 2020).

$$F1\ Score\ = \frac{2.Precision.Recall}{Precision + Recall} \quad \textbf{(4)}$$

Perhitungan nilai *macro f*1-*score* dan *micro f*1-*score* harus diterapkan dalam perhitungan *f*1-*score* dalam *multi-class classification* (Grandini et al., 2020). Kedua matriks evaluasi performa model ini memiliki kegunaan yang berbeda tergantung pada tujuan evaluasi. *Macro f*1-*score* mengidentifikasi sejauh mana model baik dalam mengklasifikasikan semua kelas secara merata. Sedangkan *micro f*1-*score* untuk mengetahui kinerja model secara umum di seluruh dataset. Perhitungan nilai *macro f*1-*score* menggunakan persamaan **(5)**: (Grandini et al., 2020)

$$Macro\ F1\text{-}Score = 2 * \left(\frac{Macro\ Average\ Precision * Macro\ Average\ Recall}{Macro\ Average\ Precision^{-1} + Macro\ Average\ Recall^{-1}}\right) \quad \textbf{(5)}$$

Nilai dari *macro average precision* dan *macro average recall* dapat dihitung menggunakan persamaan **(6)** dan **(7)**. *K* yang terdapat dalam Rumus **(6)** dan **(7)** adalah jumlah kelas atau label yang ada dalam evaluasi. (Grandini et al., 2020).

$$Macro\ Average\ Precision\ = \frac{\sum_K^K Precision_K}{K} \quad \textbf{(6)}$$

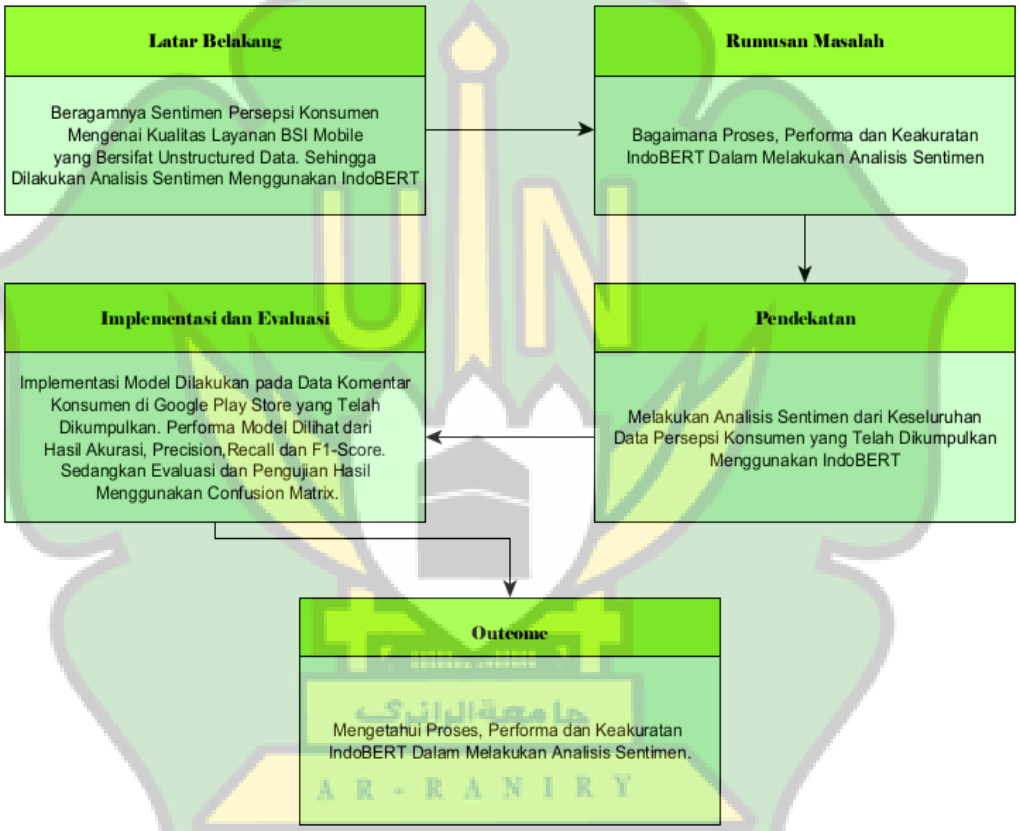
$$Macro\ Average\ Recall\ = \frac{\sum_K^K Recall_K}{K} \quad \textbf{(7)}$$

Untuk menghitung nilai *micro f*1-*score* menggunakan persamaan **(8)**. *Grand Total* yang terdapat dalam rumus **(8)** adalah jumlah seluruh kasus posisif yang diidentifikasi dengan benar dalam sebuah kelas. (Grandini et al., 2020).

$$Micro\ Average\ F1 - Score\ = \frac{\sum_K^K TP_K}{Grand\ Total} \quad \textbf{(8)}$$

2.12 Kerangka Pemikiran Penelitian

Kerangka pemikiran menggambarkan struktur atau paradigma berpikir yang digunakan sebagai landasan oleh penulis untuk memahami, menganalisis dan memecahkan masalah. Kerangka pemikiran membantu mengorganisir informasi, konsep menjadi satu rangkain logis. **Gambar 2.6** menunjukkan kerangka pemikiran penelitian.

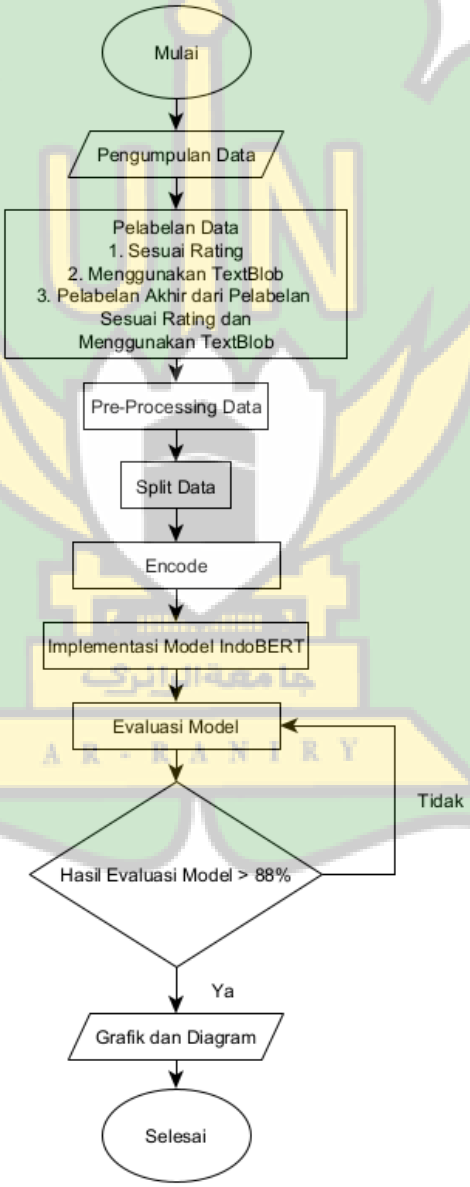


Gambar 2.6 Kerangka Pemikiran Penelitian

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian merujuk pada serangkaian langkah dan proses yang perlu diikuti oleh peneliti untuk merancang, melaksanakan, dan mengevaluasi sebuah penelitian. **Gambar 3.1** menunjukkan tahapan penelitian.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.2 Studi Literatur

Studi literatur merupakan metode yang digunakan penulis untuk mendapatkan segala informasi tentang permasalahan dan cara menyelesaikan permasalahan tersebut. Dengan melakukan peninjaun dan kajian pustaka dari beberapa buku, jurnal internasional dan penelitian sebelumnya yang relevan dengan penelitian yang sedang diteliti.

3.3 Pengumpulan Data

Data yang dikumpulkan pada penelitian ini adalah data ulasan konsumen yang berbahasa Indonesia terhadap aplikasi BSI Mobile di *website* Google Play Store. Pengumpulan data dilakukan dengan cara *scraping* pada kolom komentar pada link https://play.google.com/store/apps/details?id=com.bsm.activity2&hl=id-ID. Data ulasan yang diambil untuk dianalisis adalah *username*, tanggal, *rating* dan komentar yang berbahasa Indonesia dalam rentang waktu 6 bulan, terhitung mulai tanggal 1 Februari 2023 sampai 31 Juli 2023.

3.4 Pelabelan Data

Pelabelan data merupakan proses pemberian label atau kategori tertentu pada data. Sehingga dapat digunakan sebagai data pelatihan atau data validasi dalam pemodelan dan analisis data. Pelabelan data bertujuan agar model dapat belajar dalam menganalisis sentimen. Label dalam penelitian ini menggunakan positif, netral dan negatif. Label 0 digunakan untuk kalimat bernilai negatif, label 1 digunakan untuk kalimat bernilai netral dan label 2 digunakan untuk kalimat bernilai positif. Pada penelitian ini pelabelan data yang memuat kalimat bersentimen positif, netral atau negatif dilakukan dengan 2 cara. Pertama, sesuai dengan pemberian *rating* oleh pengguna. *Rating* 5 dan 4 dikategorikan positif, *rating* 3 dikategorikan netral serta *rating* 2 dan 1 dikategorikan negatif. Kedua, pelabelan dilakukan menggunakan TextBlob dengan melihat nilai polaritas dari sebuah teks. Kemudian akan dilakukan komparasi hasil antara pelabelan sesuai pemberian *rating* oleh pengguna dengan pelabelan sesuai nilai polaritas menggunakan TextBlob.

3.5 *Pre-Processing* Data

Tahapan *pre-processing* data sangat penting dalam analisis data, tahapan ini bertujuan untuk menyiapkan data sebelum dilakukan pemodelan atau analisis lebih lanjut. *Pre-processing* data melibatkan serangkaian langkah yang ditujukan untuk membersihkan, mengatur, dan menyiapkan data supaya bisa digunakan secara optimal sebagai data *training*. Tahapan *pre-processing* data yang harus dilakukan adalah:

- *Case folding*

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil yang seragam. Tujuan dari case folding adalah untuk mempermudah pengolahan teks dan meminimalisir kesalahan yang disebabkan oleh perbedaan karakter huruf pada teks.

- *Filtering*

Filtering adalah proses menghilangkan tanda baca dan kata yang bukan huruf. Tujuan dari pemfilteran sentimen adalah untuk mengurangi jumlah data yang perlu dianalisis dan mempermudah proses bantuan sentimen pada teks atau dokumen berukuran besar.

- *Tokenizing*

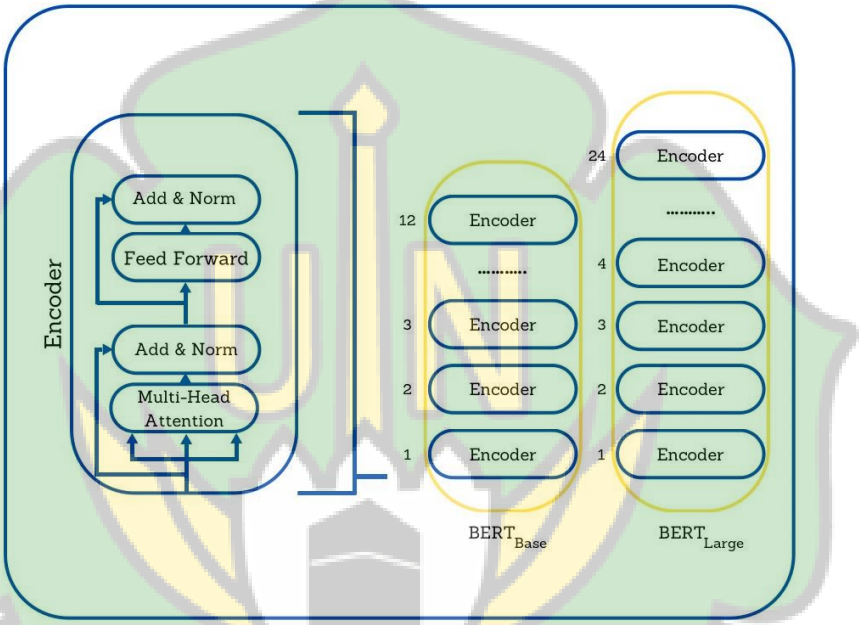
Tokenizing adalah proses memecah teks atau dokumen menjadi unit-unit kecil yang disebut token. Token dapat berupa kata, frasa, angka, simbol, atau tanda baca yang memiliki arti atau makna tertentu. Pada penelitian ini, token diambil dari BERT *Tokenizer* pada model BERT.

3.6 *Split* Data

Data yang sudah melewati tahap *pre-processing* akan masuk ke tahap *split* data. Tahapan ini mengacu pada proses memisahkan dataset menjadi dua atau lebih subset yang berbeda. Pemisahan data ini bertujuan untuk melatih, menguji, dan memvalidasi model atau algoritma dengan benar serta untuk menghindari terjadinya *overfitting*.

3.7 *Encode*

Encode adalah proses mengubah atau mengonversikan informasi dari satu bentuk ke bentuk yang lain. Dalam konteks NLP, *encode* merujuk pada proses mengonversikan teks atau kata-kata menjadi representasi numerik atau vektor sehingga dapat dipahami dan diolah oleh komputer. **Gambar 3.2** menunjukkan proses *encoder*.



Gambar 3.2 Proses *Encoder* (Vaswani et al., 2017)

3.8 Implementasi Model IndoBERT

Pada tahapan ini, BERT Tokenizer akan mengubah dataset menjadi input yang dapat diterima oleh model IndoBERT. Dataset diubah ke dalam bentuk vektor representasi kata. Lalu dilakukan proses *fine-tuning* untuk model analisis sentimen.

3.9 Evaluasi Model

Pada tahapan selanjutnya, performa model dilihat dari nilai *accuracy*, *precision*, *recall*, dan *f*1-*score*. Hasil dari klasifikasi sentimen dievaluasi dengan *confusion matrix* untuk melihat keakuratan dan performa IndoBERT dalam menganalisis sentimen.

3.10 Grafik dan Diagram

Output akhir dari penelitian ini adalah grafik dan diagram. Pada gambar grafik akan menunjukkan *training loss*, *validation loss* dan tingkat *accuracy* dari performa model yang dibangun sesuai dengan jumlah *batch size*, *epoch* dan *learning rate* yang dipilih. Pada gambar diagram akan menunjukkan hasil akhir dari *confusion matrix untuk* menguji keakuratan model dalam melakukan analisis sentimen pada label positif, netral dan negatif.

3.11 Dokumentasi

Pada fase terakhir dari penelitian ini, dokumentasi dilakukan melalui penyusunan buku tugas akhir dengan menerapkan aturan-aturan yang sesuai pada buku pedoman. Hasil dari dokumentasi penelitian ini diharapkan menjadi referensi baru serta dapat menambahkan ide dan konsep jika penelitian ini ingin dikembangkan.

3.12 Alat Bantu Penelitian

Alat bantu yang digunakan pada penelitian ini adalah perangkat keras dan perangkat lunak. Perangkat keras yang digunakan adalah satu unit laptop ROG Strix G15 (2022) G513RM-R736D6G-O dengan spesifikasi sebagai berikut:

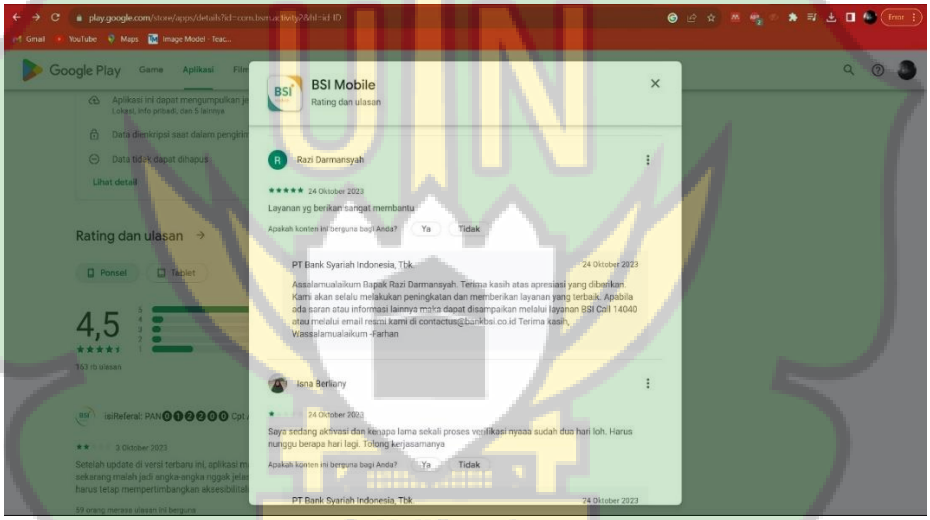
- *Processor:* AMD Ryzen™ 7 6800H *Mobile Processor* (8-*core*/16-*thread*, 20MB *cache*, up to 4.7 GHz *max boost*).
- *Display*: 15.6″ FHD (1920 x 1080) 16:9, *Refresh Rate* 300Hz, *Value IPS-level*, sRGB 100%, 300 nits.
- RAM: 16GB (8GB DDR5-4800 SO-DIMM *2).
- *Storage*: 512GB M.2 NVMe™ PCIe® 4.0 SSD.

Perangkat lunak yang digunakan adalah sistem operasi *Micorosoft Windows Windows 11 Home+Office Home Student 2021*, bahasa pemrograman Python dan dieksekusi di Google Colab.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Tahap pertama yang dilakukan ialah mengumpulkan data yang ingin diproses. Data yang diambil merupakan data ulasan konsumen yang berbahasa Indonesia terhadap aplikasi BSI Mobile di *website* Google Play Store. **Gambar 4.1** menunjukkan gambar ulasan konsumen terhadap aplikasi BSI Mobile di *website* Google Play Store.



Gambar 4.1 Ulasan Konsumen Terhadap Aplikasi BSI Mobile di *Website* Google Play Store

Proses pengambilan data menggunakan *library google play scrapper* dengan antribut *username*, *rating*, *at* dan *review*. Sortir dalam pengambilan data komentar menggunakan perintah sortir *sort=Sort.Newest*, perintah ini untuk mengambil data komentar berdasarkan komentar terbaru. Hasil pengumpulan data yang dilakukan dalam rentang waktu 6 bulan, terhitung mulai tanggal 1 Februari 2023 sampai 31 Juli 2023 didapatkan sebanyak 23931 data. Untuk mempermudah proses pengolahan data, dataset disimpan dalam format *file csv*.

Untuk memaksimalkan pemanfaatan data, maka langkah yang diambil adalah mengurangi atribut dengan menghapus atribut *username* dan *at* karena tidak relevan dalam analisis sentimen. **Tabel 4.1** menunjukkan contoh dari hasil pengumpulan data.

Tabel 4.1 Contoh dari Hasil Pengumpulan Data

Username	*Rating*	*At*	Komentar
Ratu Bintang	5	07/31/2023 19:13	aplikasi nya bener bener bagus banget untuk transaksi nya mudah banget!!! Dan juga fitur fitur baru nya mantap 💗 💗
Yusuf Prasetya	1	05/25/2023 7:23	Semenjak ada kasus lockbit randomware, bsi makin jelek, mulai cs ga bisa dihubungi payment, selalu error malah makin ragu untuk nerusin nabung disini
saefudin malik	3	06/08/2023 14:47	Tolong aplikasinya di perbaiki lagi agar pengguna jauh lebih nyaman untuk menggunakannya
Tim Tam	2	05/09/2023 12:37	Bsi kenapa 2 hari trobel terus😭 lagi butuh tapi eror lama banget. Mohon segera diperbaiki😉
Abdel Gadelhaq	4	07/11/2023 4:17	Baru-baru ini, saya install aplikasi BSI Mobile ini, tapi belum sempat saya gunakan layanannya karena setiap dibuka force close terus-menerus, mohon ditindaklanjuti ya kak.

4.2 Pelabelan Data

Setelah data komentar konsumen yang berbahasa Indonesia terhadap BSI Mobile di *website* Google Play Store dikumpulkan, kemudian dilakukan pelabelan data agar model dapat belajar dalam menganalisis sentimen. Label yang digunakan untuk pelabelan data dalam penelitian ini ada 3 label, yaitu: positif, netral dan negatif. Pelabelan data dilakukan dengan 2 cara, sesuai *rating* yang diberikan oleh pengguna dan menggunakan TextBlob.

4.2.1 Pelabelan Data Sesuai *Rating*

Data dilabel sesuai dengan *rating* yang diberikan oleh pengguna yang berkomentar terhadap aplikasi BSI Mobile di *website* Google Play Store. *Rating* 1 dan 2 dilabel menjadi sentimen negatif, *rating* 3 dilabel menjadi sentimen netral, *rating* 4 dan 5 dilabel menjadi sentimen positif. Dari hasil pelabelan data sesuai *rating* didapatkan 16647 data berlabel positif, 1023 data berlabel netral dan 6260 data berlabel negatif. **Tabel 4.2** menunjukkan hasil pelabelan data sesuai *rating*.

Tabel 4.2 Pelabelan Data Sesuai *Rating*

Komentar	*Rating*	Label Sentimen
aplikasi nya bener bener bagus banget untuk transaksi nya mudah banget!!! Dan juga fitur fitur baru nya mantap 💗 💗	5	Positif
Semenjak ada kasus lockbit randomware, bsi makin jelek, mulai cs ga bisa dihubungi payment, selalu error malah makin ragu untuk nerusin nabung disini	1	Negatif
Tolong aplikasinya di perbaiki lagi agar pengguna jauh lebih nyaman untuk menggunakannya	3	Netral

Bsi kenapa 2 hari trobel terus😭 lagi butuh tapi eror lama banget. Mohon segera diperbaiki😊	2	Negatif
Baru-baru ini, saya install aplikasi BSI Mobile ini, tapi belum sempat saya gunakan layanannya karena setiap dibuka force close terus-menerus, mohon ditindaklanjuti ya kak.	4	Positif

4.2.2 Pelabelan Data Menggunakan TextBlob

TextBlob melabel data sesuai nilai polaritas yang terkandung dalam sebuah teks. Tetapi data harus diterjemahkan ke dalam Bahasa Inggris terlebih dahulu, karena TextBlob tidak bisa beroperasi pada data teks berbahasa Indonesia. Namun, Setelah dilakukan pelabelan, dataset akan diterjemahkan balik ke dalam Bahasa Indonesia untuk melanjutkan ke tahapan berikutnya. Nilai polaritas suatu teks menjadi acuan untuk dilakukan pelabelan data. Jika nilai polaritas sebuah teks lebih besar dari 0.25 (p > 0.25), maka teks akan dikategorikan ke dalam sentimen positif. Jika nilai polaritas sebuah teks dibawah sama dengan 0 (p <= 0), maka teks akan dkategorikan ke dalam sentimen negatif. Jika nilai sebuah teks lebih besar dari nol dan lebih kecil dari 0.25 (p > 0 && <0.25), maka akan dikategorikan ke dalam sentimen netral. Dari hasil pelabelan data menggunakan TextBlob didapatkan 12863 data berlabel positif, 3565 data berlabel netral dan 7502 data berlabel negatif. **Tabel 4.3** menunjukkan pelabelan data menggunakan TextBlob.

Tabel 4.3 Pelabelan Data Menggunakan TextBlob

Komentar Bahasa Indonesia	Komentar Bahasa Inggris	*Polarity*	Label Sentimen
aplikasi nya bener bener bagus banget untuk transaksi nya mudah banget!!! Dan juga fitur	*The application is really really good for the transaction is really easy!!! And also*	0.62	Positif

fitur baru nya mantap 💚 💚	*the new features are great* 💚 💚		
Semenjak ada kasus lockbit randomware, bsi makin jelek, mulai cs ga bisa dihubungi payment, selalu error malah makin ragu untuk nerusin nabung disini	*Since there is a case of BSI randomware lockbit, the worse, starting from CS, it can't be contacted by Payment, always the error is even more hesitant to keep saving here here*	0.033	Netral
Tolong aplikasinya di perbaiki lagi agar pengguna jauh lebih nyaman untuk menggunakannya	*Please repair the application again so that users are much more comfortable to use it*	0.45	Positif
Bsi kenapa 2 hari trobel terus😭 lagi butuh tapi eror lama banget. Mohon segera diperbaiki☺	*Bsi why 2 days of Trobel continue* 😭 *to need it again but the error is so long. Please fix it immediately*☺	-0.05	Negatif
Baru-baru ini, saya install aplikasi BSI Mobile ini, tapi belum sempat saya gunakan layanannya karena setiap dibuka force close terus-menerus, mohon ditindaklanjuti ya kak.	*Recently, I installed this BSI Mobile application, but I haven't had the chance to use the service because every time it is opened for force close, please follow up, sis.*	0.0	Negatif

4.2.3 Pelabelan Tahap Akhir

Setelah dilakukan pelabelan menggunakan 2 cara, sesuai *rating* dan menggunakan TextBlob, data yang sudah dilabel sebagai sentimen positif, netral dan negatif akan dilakukan pelabelan tahap akhir sesuai dengan label sentimen yang didapatkan. Label 0 untuk sentimen negatif, label 1 untuk sentimen netral dan label 2 untuk sentimen positif. Hal ini bertujuan agar data yang mengandung sentimen positif, netral dan negatif dapat dianalisa oleh model IndoBERT. Kemudian akan dilakukan komparasi hasil untuk pelabelan sesuai *rating* dan pelabelan menggunakan TextBlob. **Tabel 4.4** dan **Tabel 4.5** menunjukkan tahap pelabelan akhir dari pelabelan sesuai *rating* dan menggunakan TextBlob.

Tabel 4.4 Pelabelan Akhir dari Pelabelan Sesuai *Rating*

Komentar	*Rating*	Label Sentimen	Label Akhir
aplikasi nya bener bener bagus banget untuk transaksi nya mudah banget!!! Dan juga fitur fitur baru nya mantap 💚 💚	5	Positif	2
Semenjak ada kasus lockbit randomware, bsi makin jelek, mulai cs ga bisa dihubungi payment, selalu error malah makin ragu untuk nerusin nabung disini	1	Negatif	0
Tolong aplikasinya di perbaiki lagi agar pengguna jauh lebih nyaman untuk menggunakannya	3	Netral	1
Bsi kenapa 2 hari trobel terus😭 lagi butuh tapi eror lama banget. Mohon segera diperbaiki😊	2	Negatif	0

Baru-baru ini, saya install aplikasi BSI Mobile ini, tapi belum sempat saya gunakan layanannya karena setiap dibuka force close terus-menerus, mohon ditindaklanjuti ya kak.	4	Positif	2

Tabel 4.5 Pelabelan Akhir dari Pelabelan Menggunakan TextBlob

Komentar Bahasa Indonesia	Komentar Bahasa Inggris	*Polarity*	Label Sentimen	Label Akhir
aplikasi nya bener bener bagus banget untuk transaksi nya mudah banget!!! Dan juga fitur fitur baru nya mantap 💚 💚	*The application is really really good for the transaction is really easy!!! And also the new features are great* 💚 💚	0.62	Positif	2
Semenjak ada kasus lockbit randomware, bsi makin jelek, mulai cs ga bisa dihubungi payment, selalu error malah makin ragu untuk nerusin nabung disini	*Since there is a case of BSI randomware lockbit, the worse, starting from CS, it can't be contacted by Payment, always the error is even more hesitant to keep saving here here*	0.033	Netral	1
Tolong aplikasinya di perbaiki lagi agar pengguna jauh lebih nyaman untuk menggunakannya	*Please repair the application again so that users are much more comfortable to use it*	0.45	Positif	2

Bsi kenapa 2 hari trobel terus😫 lagi butuh tapi eror lama banget. Mohon segera diperbaiki😊	*Bsi why 2 days of Trobel continue* 😫 *to need it again but the error is so long. Please fix it immediately*😊	-0.05	Negatif	0
Baru-baru ini, saya install aplikasi BSI Mobile ini, tapi belum sempat saya gunakan layanannya karena setiap dibuka force close terus-menerus, mohon ditindaklanjuti ya kak.	*Recently, I installed this BSI Mobile application, but I haven't had the chance to use the service because every time it is opened for force close, please follow up, sis.*	0.0	Negatif	0

4.3 *Pre-Processing* Data

Data komentar yang diperoleh dari *crawling* perlu dibersihkan terlebih dahulu. *Pre-processing* data dilakukan untuk menjadikan data lebih efektif dan efisien saat pemrosesan. Pada dataset mentah masih terdapat tanda baca, simbol, angka, emoji, dan karakter tertentu yang tidak memuat opini. Maka data harus melewati tahap *filtering* dengan menghilangkan karakter yang tidak dibutuhkan, dengan tujuan mengurangi ukuran data agar proses pelatihan menjadi lebih efisien. **Table 4.6** menunjukkan Tahapan Pembersihan Data.

Tabel 4.6 Tahapan Pembersihan Data

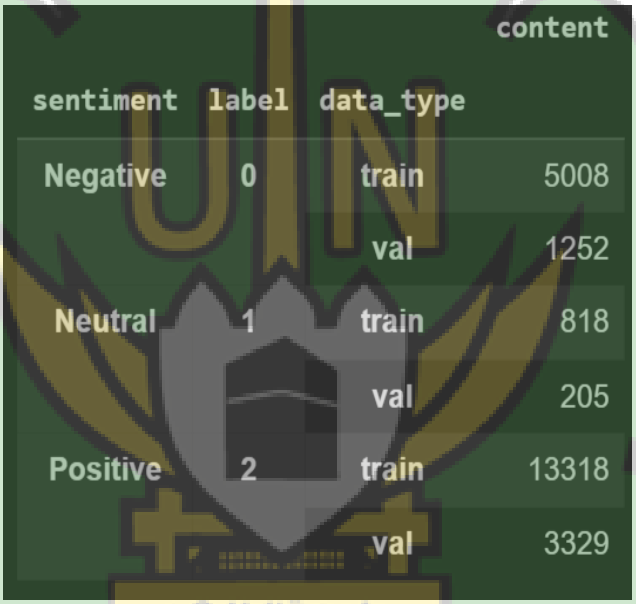
Tahapan	Hasil
Komentar Awal	Mantap saya kasih bintang 5, makin nyaman pakai aplikasi BSI sekarang! Tentu sangat membantu untuk kaum mager kepengen makan gak perlu keluar rumah hanya buka aplikasi

	makanan lalu QRIS saja lewat BSI Mobile. Mantap BSI jaya Terus!
Case Folding	mantap saya kasih bintang 5, makin nyaman pakai aplikasi bsi sekarang!tentu sangat membantu untuk kaum mager kepengen makan gak perlu keluar rumah hanya buka aplikasi makanan lalu qris saja lewat bsi mobile. mantap bsi jaya terus!🔥
Menghapus Angka	mantap saya kasih bintang, makin nyaman pakai aplikasi bsi sekarang!tentu sangat membantu untuk kaum mager kepengen makan gak perlu keluar rumah hanya buka aplikasi makanan lalu qris saja lewat bsi mobile. mantap bsi jaya terus!🔥
Menghapus simbol dan tanda baca	mantap saya kasih bintang makin nyaman pakai aplikasi bsi sekarang tentu sangat membantu untuk kaum mager kepengen makan gak perlu keluar rumah hanya buka aplikasi makanan lalu qris saja lewat bsi mobile mantap bsi jaya terus 🔥
Menghapus emoji	mantap saya kasih bintang makin nyaman pakai aplikasi bsi sekarang tentu sangat membantu untuk kaum mager kepengen makan gak perlu keluar rumah hanya buka aplikasi makanan lalu qris saja lewat bsi mobile mantap bsi jaya terus

4.4 *Split* Data

Data yang sudah melewati tahap pelabelan dan *pre-processing* akan dibagi menjadi 2 bagian yaitu: data *train* dan data *validation*. Data *training* digunakan untuk melatih model pembelajaran mesin. Dengan memberikan data yang berkualitas dan relevan kepada model. Data *train* dapat mengajarkan model untuk mengenali pola, membuat prediksi, mengklasifikasikan data. Sedangkan Data *validation* digunakan untuk mengukur kinerja model dalam melakukan suatu prediksi dari data yang diberikan oleh data *training* serta meminimalisir *overfitting* yang dapat terjadi. Data *validation* harus memuat data yang berbeda dengan data yang terdapat dalam data *train*.

Dengan menggunakan data *validation* yang belum pernah diproses sebelumnya oleh model, dapat membangun model yang lebih kaya akan teks dan dapat memprediksi data lebih akurat serta relevan. Pembagian data ke dalam data *training* dan data *validation* adalah sebanyak 80% data *training* dan 20% data *validation*. **Gambar 4.2** menunjukkan *split* data *train* dan data *validation* pada dataset yang sudah dilakukan pelabelan akhir dari pelabelan sesuai *rating*. **Gambar 4.3** menunjukkan *split* data *train* dan data *validation* pada dataset yang sudah dilakukan pelabelan akhir dari pelabelan menggunakan TextBlob.



			content
sentiment	label	data_type	
Negative	0	train	5008
		val	1252
Neutral	1	train	818
		val	205
Positive	2	train	13318
		val	3329

Gambar 4.2 *Split* Data *Train* dan *Validation* pada Dataset Pelabelan Sesuai *Rating*

Dari hasil *split* data pada dataset yang dilabel sesuai *rating* didapatkan 19144 data *train* dan 4786 data *validation*. Untuk sentimen negatif didapatkan 5008 data *train* dan 1252 data *validation*. Untuk sentimen netral didapatkan 818 data *train* dan 205 data *validation*. Pada sentimen positif didapatkan 13318 data *train* dan 3329 data *validation*.

			content
sentiment	label	data_type	
Negative	0	train	6002
		val	1500
Neutral	1	train	2852
		val	713
Positive	2	train	10290
		val	2573

Gambar 4.3 *Split* Data *Train* dan *Validation* pada Dataset Pelabelan Menggunakan TextBlob

Dari hasil *split* data pada dataset yang dilabel menggunakan TextBlob didapatkan 19144 data *train* dan 4786 data *validation*. Untuk sentimen negatif didapatkan 6002 data *train* dan 1500 data *validation*. Untuk sentimen netral didapatkan 2852 data *train* dan 713 data *validation*. Pada sentimen positif didapatkan 10290 data *train* dan 2573 data *validation.*

4.5 *Encode*

Dataset yang akan digunakan harus sesuai dengan format input yang diterima oleh IndoBERT. Setelah didapatkan dataset yang sudah diolah melalui proses *pre-processing*, tahapan selanjutnya menambahkan token khusus yaitu [CLS] di awal setiap kalimat, dan [SEP] di akhir setiap kalimat yang berfungsi sebagai pemisah antar kalimat. Setelah itu, proses pengkodean (*encoding*) akan dilakukan menggunakan tokenizer yang sesuai dengan indeks *vocabulary* yang telah didefinisikan oleh IndoBERT yang telah dilatih sebelumnya. Vocabulary ini akan digunakan dalam proses *transfer learning*. **Gambar 4.4** menunjukkan contoh dari beberapa kata yang terdapat dalam *vocabulary* IndoBERT.

Gambar 4.4 Contoh Kata pada *Vocabulary* IndoBERT

Model BERT dapat menerima input teks dengan panjang maksimal 512 kata. Jika kalimat terlalu pendek, maka akan ditambahkan token [PAD] untuk memenuhi panjang maksimal. Sebaliknya, jika kalimat terlalu panjang, maka akan dipangkas atau dipecah menjadi beberapa kata (*truncate)*. Kata yang tidak terdapat dalam *vocabulary* (*out-of vocabulary*) akan dipisah menjadi sub kata (*subword*) dengan menggunakan simbol ##. **Table 4.7** menunjukkan proses penyesuaian input yang dapat diterima oleh IndoBERT.

Tabel 4.7 Proses Penyesuaian Input yang Dapat Diterima Oleh IndoBERT

Kalimat	alhamdulillah bsi mobile setelah di update makin canggih makin mempermudah saya dengan fitur fitur yang disediakan mulai dari fitur jadwal sholat arah kiblat hingga fitur yang mempermudah saya untuk bersedekah bsi mobile makin keren dan makin nyaman good job bsi mobile
Hasil tokenisasi	'alhamdulillah', 'bs', '##i', 'mobile', 'setelah', 'di', 'update', 'makin', 'canggih', 'makin', 'mempermudah', 'saya', 'dengan', 'fitur', 'fitur', 'yang', 'disediakan', 'mulai', 'dari', 'fitur', 'jadwal', 'sholat', 'arah', 'kiblat', 'hingga', 'fitur', 'yang', 'mempermudah', 'saya', 'untuk', 'bersedekah', 'bs', ##i', 'mobile', 'makin', 'keren', 'dan', 'makin', 'nyaman', 'good', 'job', 'bs', '##i', 'mobile'
	'[CLS]','alhamdulillah', 'bs', '##i', 'mobile', 'setelah', 'di', 'update', 'makin', 'canggih', 'makin', 'mempermudah', 'saya', 'dengan', 'fitur', 'fitur', 'yang', 'disediakan',

Penambahan token khusus	‘mulai’, ‘dari’, ‘fitur’, ‘jadwal’, ‘sholat’, ‘arah’, ‘kiblat’, ‘hingga’, ‘fitur’, ‘yang’, ‘mempermudah’, ‘saya’, ‘untuk’, ‘bersedekah’, ‘bs’, ##i’, ‘mobile’, ‘makin’, ‘keren’, ‘dan’, ‘makin’, ‘nyaman’, ‘good’, ‘job’, ‘bs’, ‘##i’, ‘mobile’,’[SEP]’
Encoding	2, 5742, 5562, 30356, 5872, 450, 26, 4398, 2855, 6149, 2855, 7373, 209, 79, 2631, 2631, 34, 4790, 673, 98, 2631, 2864, 5458, 2069, 21171, 733, 2631, 34, 7373, 209, 90, 26055, 5562, 30356, 5872, 2855, 4076, 41, 2855, 2382, 7250, 4812, 5562, 30356, 5872, 3
Attention mask	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Berdasarkan informasi dalam tabel, pada kalimat ‘alhamdulillah bsi mobile setelah di update makin canggih makin mempermudah saya dengan fitur fitur yang disediakan mulai dari fitur jadwal sholat arah kiblat hingga fitur yang mempermudah saya untuk bersedekah bsi mobile makin keren dan makin nyaman good job bsi mobile’ dilakukan proses tokenisasi yang sesuai dengan *vocabulary* yang digunakan oleh IndoBERT. Kata 'bsi' adalah kata kunci dalam topik penelitian ini. Kata 'bsi' dipecah menjadi dua sub kata (*subword*), yaitu 'bs', '##i', karena kata 'bsi' tidak terdapat pada *vocabulary* IndoBERT. Kata yang ditambahkan ke dalam

vocabulary dapat mengakibatkan peningkatan ukuran *vocabulary*. Selain itu, ukuran matriks embedding juga harus disesuaikan karena adanya penambahan token lalu dilakukan proses *pre-training*. Oleh karena itu, dalam penelitian ini, hanya digunakan *vocabulary* dan model *pre-trained* yang telah tersedia tanpa perlu mengubahnya.

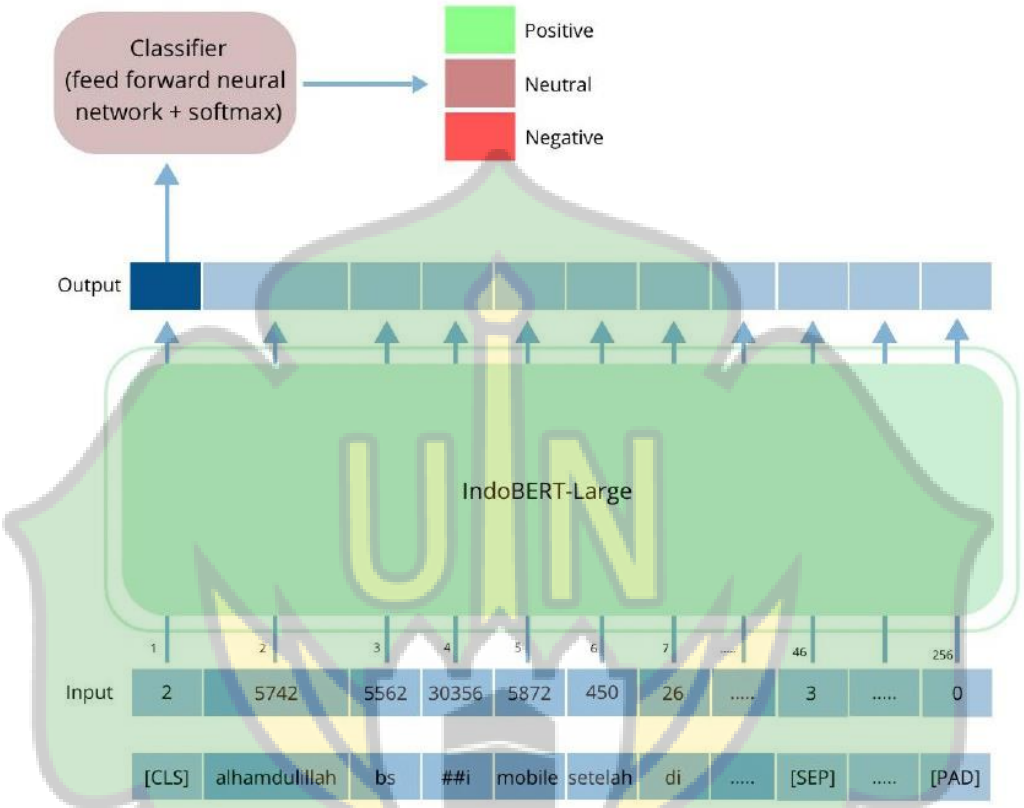
Setelah proses tokenisasi selesai, langkah selanjutnya adalah menambahkan dua token khusus yaitu token [CLS] di bagian awal kalimat dan token [SEP] di bagian akhir kalimat. Kemudian setiap token akan melewati tahap *encode* sesuai dengan indeks dalam *vocabulary* IndoBERT. Sebagai contoh, token [CLS] memiliki id 2, token ‘alhamdulillah’ memiliki id 5742 'bs' memiliki id 5562, token ‘##i’ memilki id 30356 dan seterusnya. Attention mask digunakan untuk membedakan token kata dengan token padding. Token padding [PAD] memiliki nilai 0, sementara token kata memiliki nilai 1. Kemudian, data loader dibuat untuk masing-masing dataset, yaitu data loader untuk *training* dan data loader untuk *validation*. Hal ini bertujuan untuk data yang sudah di-tokenisasi dan dikodekan dapat diproses lebih efisien dalam berbagai tahapan dalam model IndoBERT.

4.6 Implementasi IndoBERT

Input yang sudah melewati tahap penyesuaian akan disalurkan ke dalam jaringan IndoBERT-Large yang terdiri dari 24 lapisan *Transformers* Encoder. Setiap lapisan *encoder* mempunyai 2 sub-lapisan, yaitu *multi-head self-attention mechanism* sebagai sub-lapisan pertama, dan *fully connected feed forward network* sebagai sub-lapisan kedua. Setelah melalui proses pengkodean, diperoleh vektor output dari setiap token. Namun untuk proses klasifikasi hanya menggunakan vektor output dari token [CLS].

Selanjutnya IndoBERT disesuaikan untuk tugas klasifikasi melalui tahap *fine-tuning* dengan penambahan layer klasifikasi. *Library Transformers* menyediakan kelas BertForSequenceClassification yang dirancang khusus untuk tugas klasifikasi. Setelah proses *fine-tuning*, model ini dapat menghasilkan prediksi kelas dari teks input. BertForSequenceClassification menerima input berupa urutan token dan menghasilkan output yang dapat digunakan untuk memperoleh nilai

prediksi melalui fungsi softmax. **Gambar 4.5** menunjukkan ilustrasi *fine-tuning* IndoBERT untuk tugas klasifikasi.



Gambar 4.5 Ilustrasi Proses *Fine-tuning* IndoBERT untuk Klasifikasi

Pada penelitian ini, penulis menggunakan model IndoBERT-Large phase 1 yang merupakan modifikasi dari BERT-large yang sudah mengikuti kofigurasi dari BERT-Large (uncased) yang memiliki 24 *hidden layers*, 1024 *hidden size*, 16 *attention heads*, 4096 dimensi *feed-forward hidden layers*, 335.2 juta parameters dan *languange type* Monolingual (Wilie et al., 2020). *Hyperparameter* yang digunakan agar memperoleh performa yang optimal adalah menggunakan *batch size* 16 atau 32, *learning rate* dengan optimizer adam sebesar 5e-5, 3e-5, 2e-5 dengan *epoch* 2, 3 dan 4 (Devlin et al., 2018). **Tabel 4.5** menunjukkan *hyperparameter* yang digunakan pada penelitian Tugas Akhir ini.

Tabel 4.8 *Hyperparameter* IndoBERT

Hyperparameter	Skala
Batch size	16 dan 32
Learning rate	2e-5
Epoch	5 dan 10

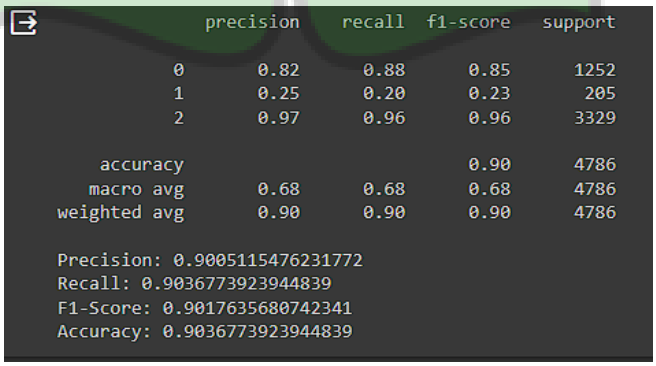
Batch size merujuk pada jumlah sampel data yang diproses oleh model dalam satu iterasi. *Learning rate* dapat menentukan seberapa besar langkah yang diambil oleh optimizer dalam mencari nilai minimum dari fungsi kerugian (*loss function*). Epoch adalah satu kali iterasi melalui seluruh dataset pelatihan. Pemilihan *hyperparameters* dapat mempengaruhi kecepatan pelatihan dan penggunaan sumber daya.

4.7 Evaluasi Model

Setelah dilakukan proses *fine-tuning* IndoBERT dengan menggunakan *hyperparameters* yang telah ditentukan, kemudian akan dilakukan perbandingan performa model klasifikasi teks antara menggunakan dataset pelabelan sesuai *rating* dan pelabelan menggunakan TextBlob. Evaluasi model akan dinilai dengan menghitung nilai *accuracy*, *precision*, *recall* dan *f*1-score dari setiap kelas.

4.7.1 Evaluasi Performa Model Pelabelan Sesuai *Rating*

Pada tahapan ini dilakukan evaluasi performa model terhadap dataset yang dilabel sesuai *rating*. **Gambar 4.6** menunjukkan performa model dengan *batch size* 16 dan *epoch* 5.



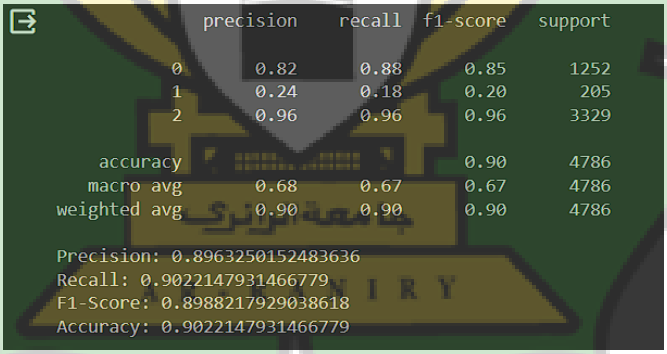
	precision	recall	f1-score	support
0	0.82	0.88	0.85	1252
1	0.25	0.20	0.23	205
2	0.97	0.96	0.96	3329
accuracy			0.90	4786
macro avg	0.68	0.68	0.68	4786
weighted avg	0.90	0.90	0.90	4786

Precision: 0.9005115476231772
Recall: 0.9036773923944839
F1-Score: 0.9017635680742341
Accuracy: 0.9036773923944839

Gambar 4.6 Performa Model dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.6** menunjukkan nilai performa model yang sangat tinggi dari empat matriks evaluasi performa model. Didapatkan nilai *accuracy* 0.903, *precision* 0.900, *recall* 0.903 dan *f*1-*score* 0.901. Dengan nilai rata-rata 0.901 menunjukkan model sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.82, *recall* 0.88 dan *f*1-*score* 0.85, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.25, *recall* 0.20 dan *f*1-*score* 0.23, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.97, *recall* 0.96 dan *f*1-*score* 0.96, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.68, *macro average recall* 0.68 dan *macro average f*1-*score* 0.68. Dari nilai *support*, model mengklasifikasikan 1252 data sebagai sentimen negatif, 205 netral dan 3329 positif. Kemudian pada **Gambar 4.7** menunjukkan performa model menggunakan *batch size* 16 dan *epoch* 10.

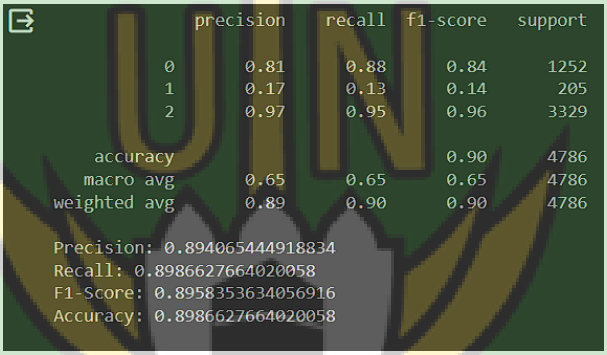


Gambar 4.7 Performa Model dengan *Batch Size* 16 dan *Epoch* 10

Pada **Gambar 4.7** menunjukkan nilai performa model yang sedikit menurun. Didapatkan nilai *accuracy* 0.902, *precision* 0.896, *recall* 0.902 dan *f*1-*score* 0.898. Dengan nilai rata-rata 0.899 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.82, *recall* 0.88 dan *f*1-*score* 0.85, nilai ini menunjukkan model sangat bagus

dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.24, *recall* 0.18 dan *f*1-*score* 0.20, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.96, *recall* 0.96 dan *f*1-*score* 0.96, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

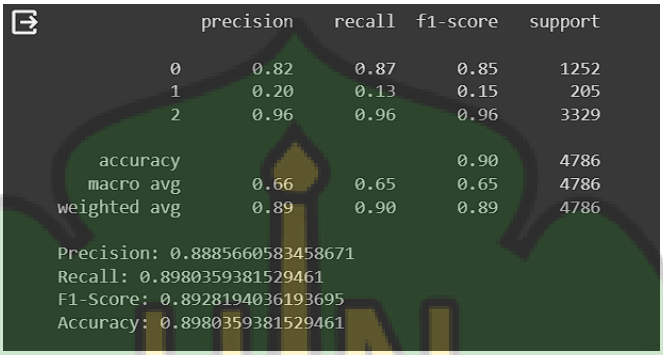
Macro average precision yang didapatkan sebesar 0.68, *macro average recall* 0.67 dan *macro average f*1-*score* 0.67. Dari nilai *support*, model mengklasifikasikan 1252 data sebagai sentimen negatif, 205 netral dan 3329 positif. Selanjutnya pada **Gambar 4.8** menunjukkan performa model menggunakan *batch size* 32 dan *epoch* 5.



Gambar 4.8 Performa Model dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.8** menunjukkan nilai performa model yang semakin menurun. Didapatkan nilai *accuracy* 0.898, *precision* 0.894, *recall* 0.898 dan *f*1-*score* 0.895. Dengan nilai rata-rata 0.896 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.81, *recall* 0.88 dan *f*1-*score* 0.84, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.17, *recall* 0.13 dan *f*1-*score* 0.14, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.97, *recall* 0.95 dan *f*1-*score* 0.96, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.65, *macro average recall* 0.65 dan *macro average f*1-*score* 0.65. Dari nilai *support*, model mengklasifikasikan 1252 data sebagai sentimen negatif, 205 netral dan 3329 positif. Pada **Gambar 4.9** menunjukkan performa model menggunakan *batch size* 32 dan *epoch* 10.



Gambar 4.9 Performa Model dengan *Batch Size* 32 dan *Epoch* 10

Pada **Gambar 4.9** menunjukkan nilai performa model yang semakin menurun. Didapatkan nilai *accuracy* 0.898, *precision* 0.888, *recall* 0.898 dan *f*1-*score* 0.892. Dengan nilai rata-rata 0.894 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.82, *recall* 0.87 dan *f*1-*score* 0.85, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.20, *recall* 0.13 dan *f*1-*score* 0.15, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.96, *recall* 0.96 dan *f*1-*score* 0.96, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

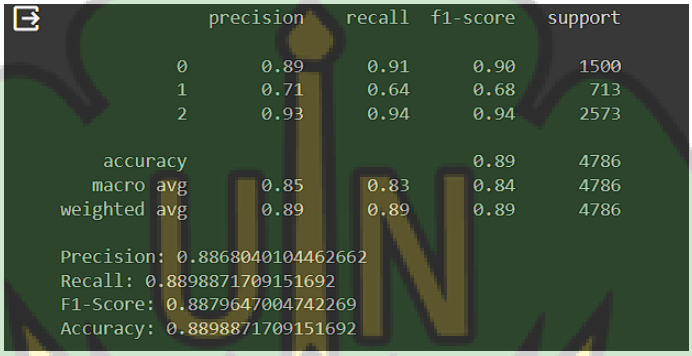
Macro average precision yang didapatkan sebesar 0.66, *macro average recall* 0.65 dan *macro average f*1-*score* 0.65. Dari nilai *support*, model mengklasifikasikan 1252 data sebagai sentimen negatif, 205 netral dan 3329 positif.

Dari hasil analisis evaluasi performa model dengan menggunakan dataset pelabelan sesuai *rating*, didapatkan model dengan performa terbaik pada model

dengan *batchsize* 16 dan *epoch* 5. Semakin besar *batchsize* dan *epoch* yang digunakan, performa model semakin menurun. Ini disebabkan oleh model yang mengalami *overfitting*.

4.7.2 Evaluasi Performa Model Pelabelan Menggunakan TextBlob

Pada tahapan ini dilakukan evaluasi performa model terhadap dataset yang dilabel menggunakan TextBlob. **Gambar 4.10** menunjukkan performa model menggunakan *batch size* 16 dan *epoch* 5.



	precision	recall	f1-score	support
0	0.89	0.91	0.90	1500
1	0.71	0.64	0.68	713
2	0.93	0.94	0.94	2573
accuracy			0.89	4786
macro avg	0.85	0.83	0.84	4786
weighted avg	0.89	0.89	0.89	4786

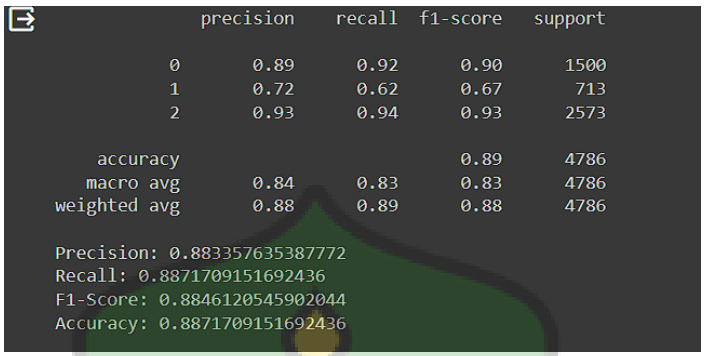
Precision: 0.8868040104462662
Recall: 0.8898871709151692
F1-Score: 0.8879647004742269
Accuracy: 0.8898871709151692

Gambar 4.10 Performa Model dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.10** didapatkan nilai *accuracy* 0.889, *precision* 0.886, *recall* 0.889 dan *f*1-*score* 0.887. Dengan nilai rata-rata 0.888 menunjukkan model sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.89, *recall* 0.91 dan *f*1-*score* 0.90, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.71, *recall* 0.64 dan *f*1-*score* 0.68, nilai ini menunjukkan model lumayan baik dalam melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.93, *recall* 0.94 dan *f*1-*score* 0.94, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.85, *macro average recall* 0.83 dan *macro average f*1-*score* 0.84. Dari nilai *support*, model mengklasifikasikan 1500 data sebagai sentimen negatif, 713 netral dan 2573 positif.

Selanjutnya pada **Gambar 4.11** menunjukkan performa model menggunakan *batch size* 16 dan *epoch* 10.



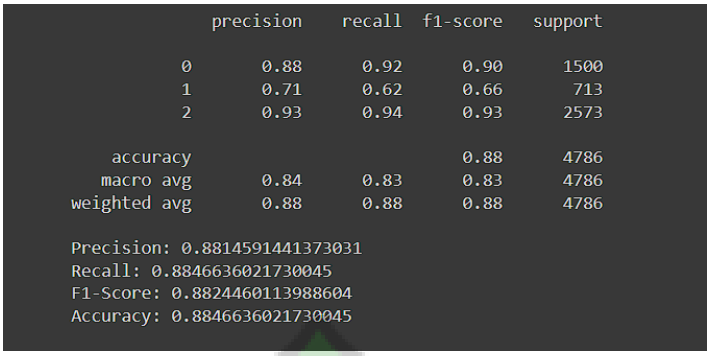
	precision	recall	f1-score	support
0	0.89	0.92	0.90	1500
1	0.72	0.62	0.67	713
2	0.93	0.94	0.93	2573
accuracy			0.89	4786
macro avg	0.84	0.83	0.83	4786
weighted avg	0.88	0.89	0.88	4786

Precision: 0.883357635387772
Recall: 0.8871709151692436
F1-Score: 0.8846120545902044
Accuracy: 0.8871709151692436

Gambar 4.11 Performa Model dengan *Batch Size* 16 dan *Epoch* 10

Pada **Gambar 4.11** menunjukkan nilai performa model yang sedikit menurun dari model sebelumnya. Didapatkan nilai *accuracy* 0.887, *precision* 0.883, *recall* 0.887 dan *f*1-*score* 0.884. Dengan nilai rata-rata 0.885 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.89, *recall* 0.92 dan *f*1-*score* 0.90, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.72, *recall* 0.62 dan *f*1-*score* 0.67, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.93, *recall* 0.94 dan *f*1-*score* 0.93, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.84, *macro average recall* 0.83 dan *macro average f*1-*score* 0.83. Dari nilai *support*, model mengklasifikasikan 1500 data sebagai sentimen negatif, 713 netral dan 2573 positif. Pada **gambar 4.12** menunjukkan performa model menggunakan *batch size* 32 dan *epoch* 5.



	precision	recall	f1-score	support
0	0.88	0.92	0.90	1500
1	0.71	0.62	0.66	713
2	0.93	0.94	0.93	2573
accuracy			0.88	4786
macro avg	0.84	0.83	0.83	4786
weighted avg	0.88	0.88	0.88	4786

Precision: 0.8814591441373031
Recall: 0.8846636021730045
F1-Score: 0.8824460113988604
Accuracy: 0.8846636021730045

Gambar 4.12 Performa Model dengan *Batch Size* 32 dan *Epoch* 5

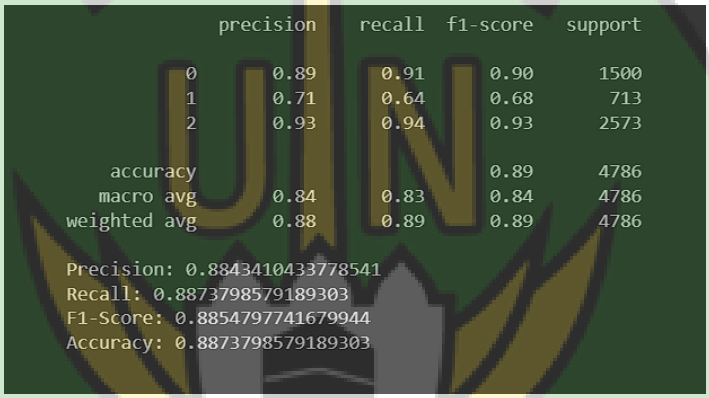
Pada **Gambar 4.12** menunjukkan nilai performa model yang semakin menurun dari model sebelumnya. Didapatkan nilai *accuracy* 0.884, *precision* 0.881, *recall* 0.884 dan *f*1-*score* 0.882. Dengan nilai rata-rata 0.883 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.88, *recall* 0.92 dan *f*1-*score* 0.90, nilai ini menunjukkan model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.71, *recall* 0.62 dan *f*1-*score* 0.66, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.93, *recall* 0.94 dan *f*1-*score* 0.93, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.84, *macro average recall* 0.83 dan *macro average f*1-*score* 0.83. Dari nilai *support*, model mengklasifikasikan 1500 data sebagai sentimen negatif, 713 netral dan 2573 positif. Pada **Gambar 4.13** menunjukkan performa model menggunakan *batch size* 32 dan *epoch* 10.

Pada **Gambar 4.13** menunjukkan nilai performa model yang sedikit meningkat dari model sebelumnya. Didapatkan nilai *accuracy* 0.887, *precision* 0.884, *recall* 0.887 dan *f*1-*score* 0.885. Dengan nilai rata-rata 0.886 menunjukkan model masih sangat akurat dalam melakukan tugas klasifikasi teks. Pada label 0 didapatkan nilai *precision* 0.89, *recall* 0.91 dan *f*1-*score* 0.90, nilai ini menunjukkan

model sangat bagus dalam melakukan klasifikasi teks pada sentimen negatif. Pada label 1 didapatkan nilai *precision* 0.71, *recall* 0.64 dan *f*1-*score* 0.68, nilai ini menunjukkan model sangat kesulitan melakukan klasifikasi teks pada sentimen netral. Pada label 2 didapatkan nilai *precision* 0.93, *recall* 0.94 dan *f*1-*score* 0.93, nilai ini menunjukkan model sangat baik dalam melakukan klasifikasi teks pada sentimen positif.

Macro average precision yang didapatkan sebesar 0.84, *macro average recall* 0.83 dan *macro average f*1-*score* 0.84. Dari nilai *support*, model mengklasifikasikan 1500 data sebagai sentimen negatif, 713 netral dan 2573 positif.



	precision	recall	f1-score	support
0	0.89	0.91	0.90	1500
1	0.71	0.64	0.68	713
2	0.93	0.94	0.93	2573
accuracy			0.89	4786
macro avg	0.84	0.83	0.84	4786
weighted avg	0.88	0.89	0.89	4786

Precision: 0.8843410433778541
Recall: 0.8873798579189303
F1-Score: 0.8854797741679944
Accuracy: 0.8873798579189303

Gambar 4.13 Performa Model dengan *Batch Size* 32 dan *Epoch* 10

Dari hasil analisis evaluasi performa model dengan memakai dataset pelabelan menggunakan TextBlob, didapatkan model dengan performa terbaik pada model dengan *batchsize* 16 dan *epoch* 5. Semakin besar *batchsize* dan *epoch* yang digunakan, performa model semakin menurun. Ini disebabkan oleh model yang mengalami *overfitting*.

4.8 Grafik dan Diagram

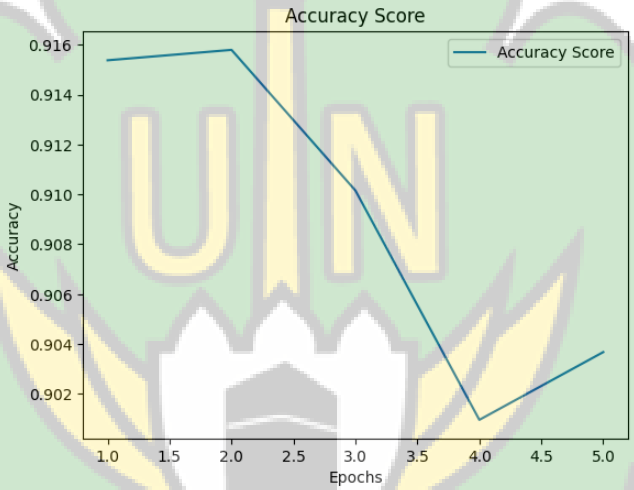
Setelah dilakukan evaluasi model menggunakan 4 metriks evaluasi nilai model, maka akan dilakukan plot menjadi grafik dan diagram. Pada plot grafik akan ditampilkan nilai *accuracy*, *training loss* dan *validation loss* pada setiap *epoch* saat proses *training* menggunakan *hyperparameters* yang sudah ditentukan. Pada plot

diagram akan ditampilkan hasil dari *confusion matrix* sebagai perhitungan nilai *validation* untuk menguji keakuratan model dalam melakukan analisis sentimen pada label positif, netral dan negatif.

4.8.1 Grafik dan Diagram Model dari Pelabelan Sesuai *Rating*

4.8.1.1 Grafik Model dari Pelabelan Sesuai *Rating*

Pada tahapan ini dilakukan plot menjadi grafik dari hasil model dengan pelabelan sesuai *rating*. **Gambar 4.14** menunjukkan grafik *accuracy* dari model dengan *batch size* 16 dan *epoch* 5.

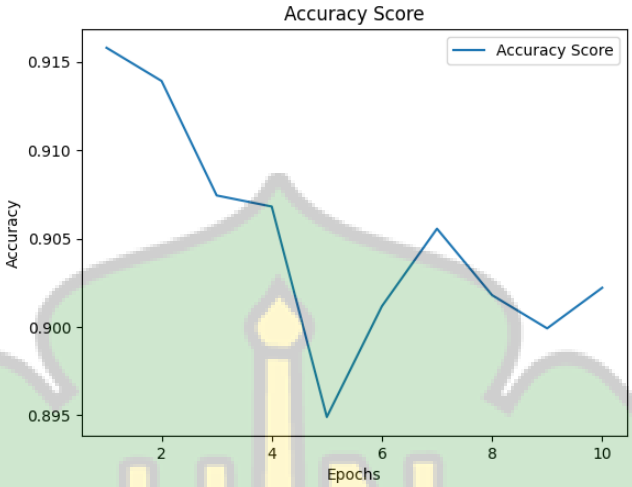


Gambar 4.14 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.14** nilai *accuracy* awal 0.915 mengalami penaikan 0.01 dari *epoch* 1 ke *epoch* 2 tetapi mengalami penurunan pada saat menuju *epoch* 3 dan *epoch* 4 sebesar 0.014, kemudian naik 0.002 pada *epoch* 5 yang berakhir pada nilai 0.904. Grafik yang naik turun disebabkan oleh model yang mengalami *overfitting*. **Gambar 4.15** menunjukkan grafik *accuracy* dari model dengan *batch size* 16 dan *epoch* 10.

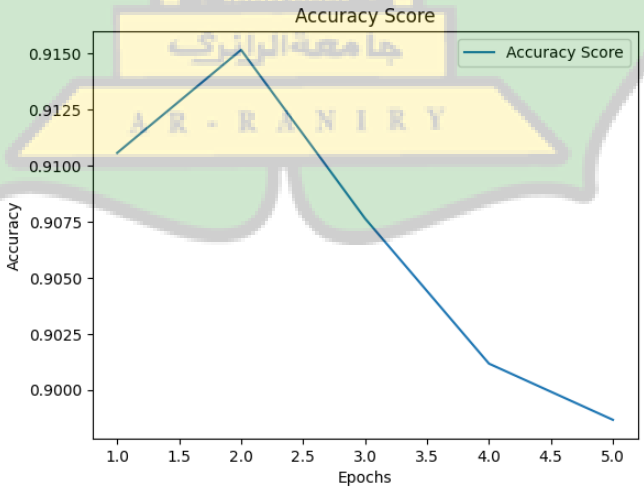
Pada **Gambar 4.15** nilai *accuracy* awal 0.915 mengalami penurunan yang signifikan dari *epoch* 1 menuju *epoch* 5 sebesar 0.020 kemudian naik 0.010 pada *epoch* 7. *Accuracy* turun kembali 0.005 pada *epoch* 9 kemudian naik 0.002 pada *epoch* 10 yang berakhir pada nilai 0.902. Grafik yang naik turun disebabkan oleh

model yang mengalami *overfitting*. **Gambar 4.16** menunjukkan grafik *accuracy* dari model dengan *batch size* 32 dan *epoch* 5.



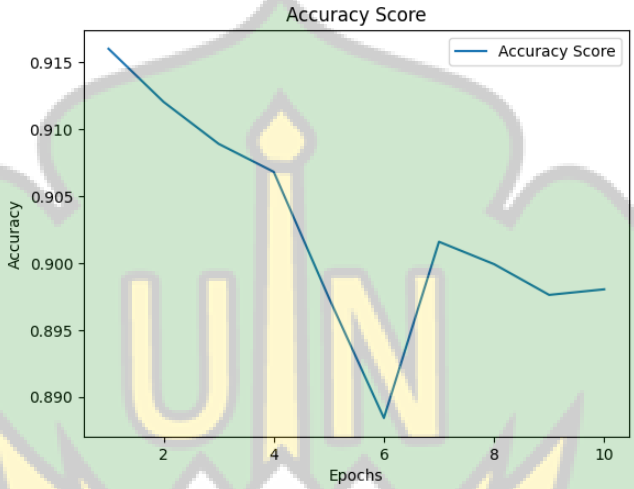
Gambar 4.15 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 10

Pada **Gambar 4.16** nilai *accuracy* awal 0.910 naik 0.005 dari *epoch* 1 ke *epoch* 2, lalu mengalami penurunan saat menuju *epoch* 5 sebesar 0.015 yang berakhir dengan nilai 0.900. Grafik yang awalnya naik lalu turun secara drastis disebabkan oleh model yang mengalami *overfitting*. **Gambar 4.17** menunjukkan grafik *accuracy* dari model dengan *batch size* 32 dan *epoch* 10.



Gambar 4.16 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 5

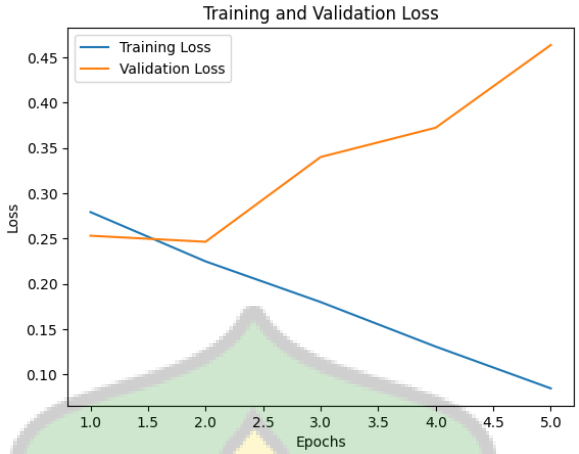
Pada **Gambar 4.17** nilai *accuracy* awal 0.915 turun drastis 0.025 dari *epoch* 1 ke *epoch* 6, lalu naik 0.012 pada *epoch* 7. Kemudian *accuracy* turun kembali 0.005 pada *epoch* 9 dan naik 0.001 pada epoch 10 yang berakhir pada nilai 0.898Grafik yang naik turun disebabkan oleh model yang mengalami *overfitting*.



Gambar 4.17 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 10

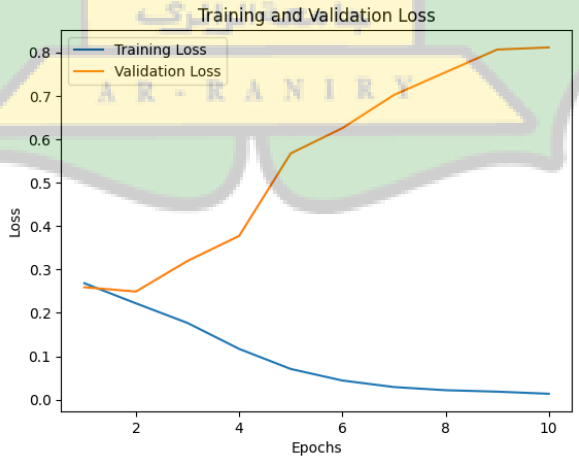
Dari hasil analisis grafik *accuracy* model dengan memakai dataset pelabelan sesuai *rating*, didapatkan model dengan *accuracy* terbaik pada model dengan *batchsize* 16 dan *epoch* 5 dengan nilai akhir 0.904. Semakin besar *batchsize* dan *epoch* yang digunakan, *accuracy* model semakin menurun. Ini disebabkan oleh model yang mengalami *overfitting*.

Setelah mendapatkan hasil grafik dari proses plot nilai *accuracy* pada model dari pelabelan sesuai *rating*, lalu dilakukan plot pada *training loss* dan *validation loss* dalam bentuk grafik pada setiap model pelabelan sesuai *rating*. **Gambar 4.18** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 16 dan *epoch* 5.



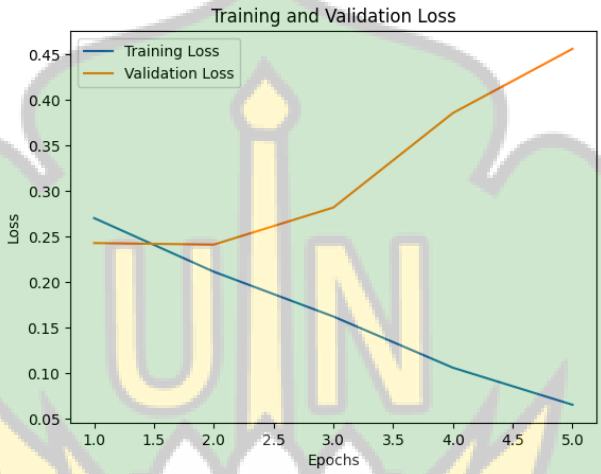
Gambar 4.18 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.18** *training loss* mengalami penurunan secara drastis dari angka 0.30 menjadi <0.10 menandakan model semakin baik dalam menyesuaikan diri dengan data pelatihan dan melakukan prediksi yang lebih akurat. Tetapi *validation loss* mengalami peningkatan dari angka 0.25 menjadi 0.45 pada *epoch* 5, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.19** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 16 dan *epoch* 10.



Gambar 4.19 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 10

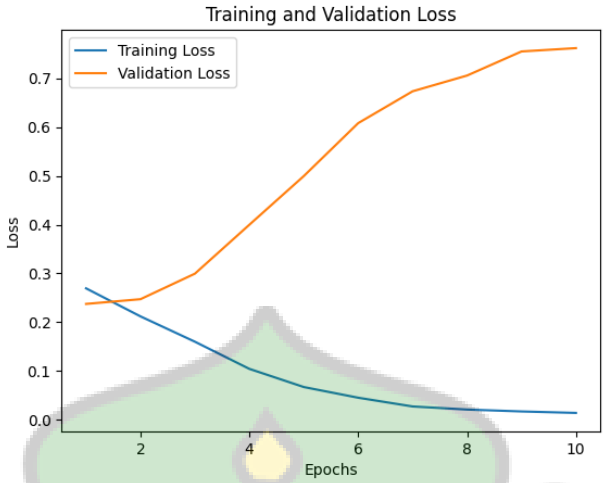
Pada **Gambar 4.19** *training loss* mengalami penurunan dari angka 0.3 menjadi <0.1 pada *epoch* 5 dan bertahan dengan nilai yang sama sampai *epoch* 10. Tetapi *validation loss* mengalami peningkatan dari angka 0.25 menjadi 0.8 pada *epoch* 10, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.20** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 32 dan *epoch* 5.



Gambar 4.20 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.20** *training loss* menurun dari angka 0.27 menjadi <0.05 pada *epoch* 5. Tetapi *validation loss* meningkat dari angka 0.25 menjadi 0.45 pada *epoch* 5, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.21** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 32 dan *epoch* 10.

Pada **Gambar 4.21** *training loss* mengalami penurunan dari angka 0.28 menjadi <.0.05 pada *epoch* 7 dan bertahan dengan nilai yang sama sampai *epoch* 10. Tetapi *validation loss* mengalami peningkatan dari angka 0.23 menjadi 0.70 pada *epoch* 10, menunjukkan bahwa terjadi *overfitting* pada model.

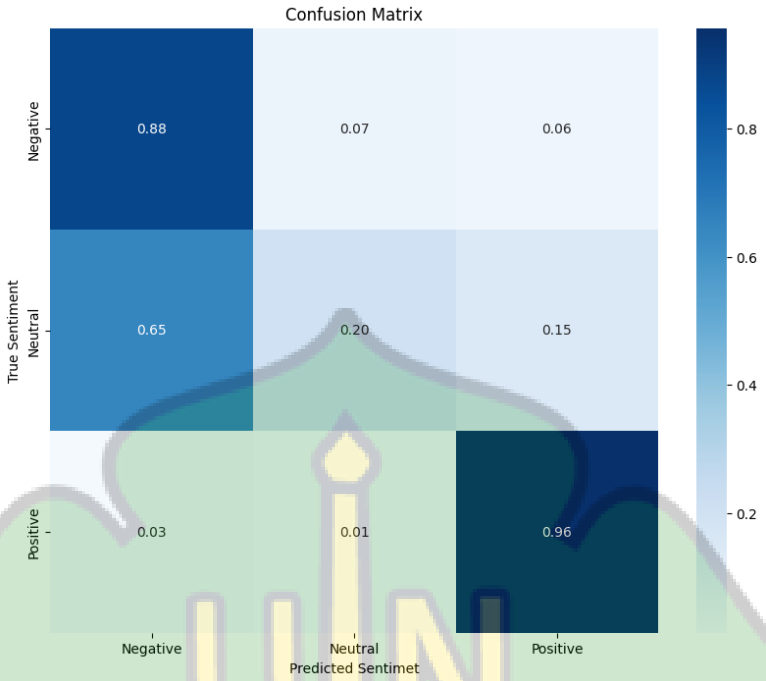


Gambar 4.21 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 10

Dari hasil *training* dan *validation loss* yang telah diplotkan ke dalam bentuk grafik menunjukkan bahwa, *training loss* pada model dengan menggunakan dataset pelabelan *rating* mengalami penurunan yang drastis. Hal ini memperlihatkan model semakin baik dalam menyesuaikan diri dengan data pelatihan dan melakukan prediksi yang lebih akurat. Namun, *validation loss* pada semua model dengan menggunakan dataset pelabelan sesuai *rating* mengalami kenaikan secara signifikan yang menandakan terjadinya *overfitting*. *Overfitting* terjadi ketika model terlalu kompleks dan telah "menghafal" data *training* hingga tingkat di mana tidak dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

4.8.1.2 Diagram *Confusion Matrix* dari Pelabelan Sesuai *Rating*

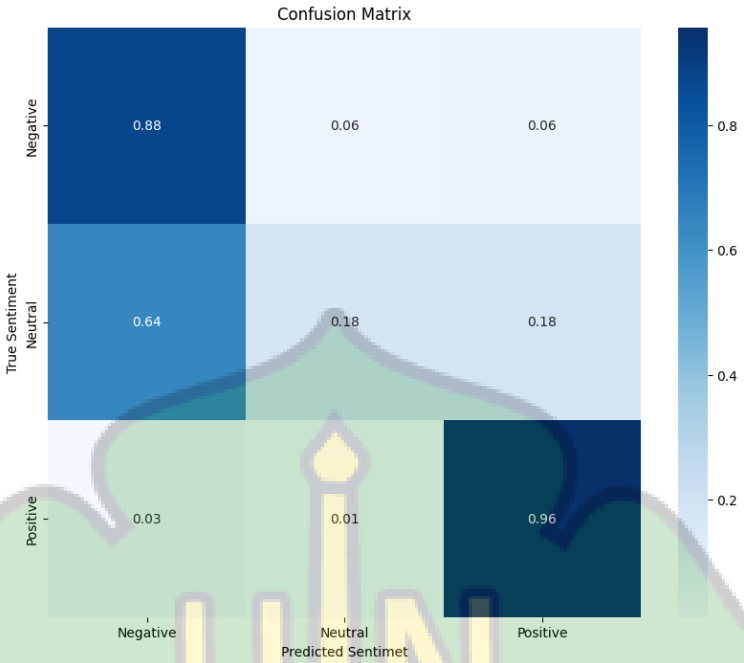
Setelah memplot *training loss* dan *validation loss* dalam bentuk grafik, langkah selanjutnya melakukan plot dari hasil model dalam melakukan analisis sentimen pada label positif, netral dan negatif dalam bentuk diagram *confusion matrix* pada model pelabelan sesuai *rating*. **Gambar 4.22** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 16 dan *epoch* 5.



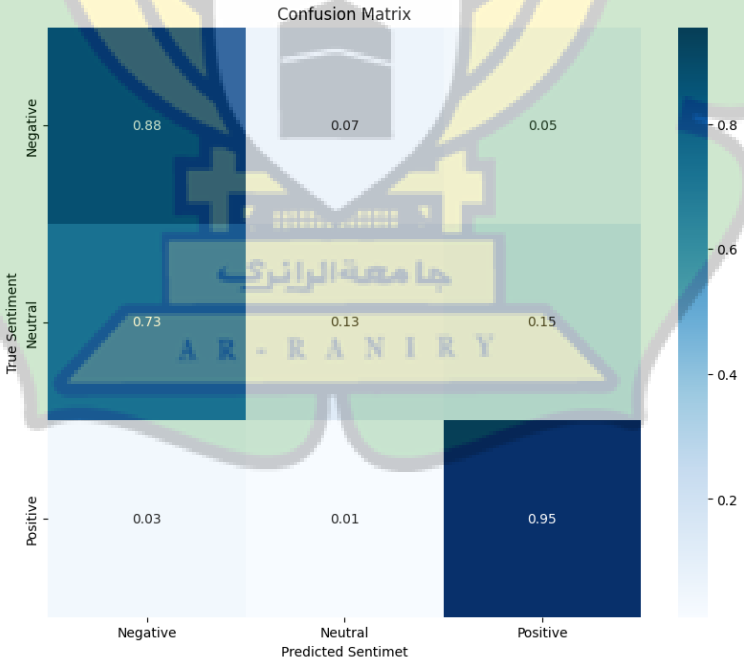
Gambar 4.22 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.22** sentimen positif yang dapat diprediksi dengan benar sebesar 0.96, sentimen netral 0.20 dan sentimen negatif 0.88. Model mengalami kesusahan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.65 dan positif 0.15. **Gambar 4.23** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 16 dan *epoch* 10.

Pada **Gambar 4.23** sentimen positif yang dapat diprediksi dengan benar sebesar 0.96, sentimen netral 0.18 dan sentimen negatif 0.88. Model mengalami kesusahan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.64 dan positif 0.18 **Gambar 4.24** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 32 dan *epoch* 5.

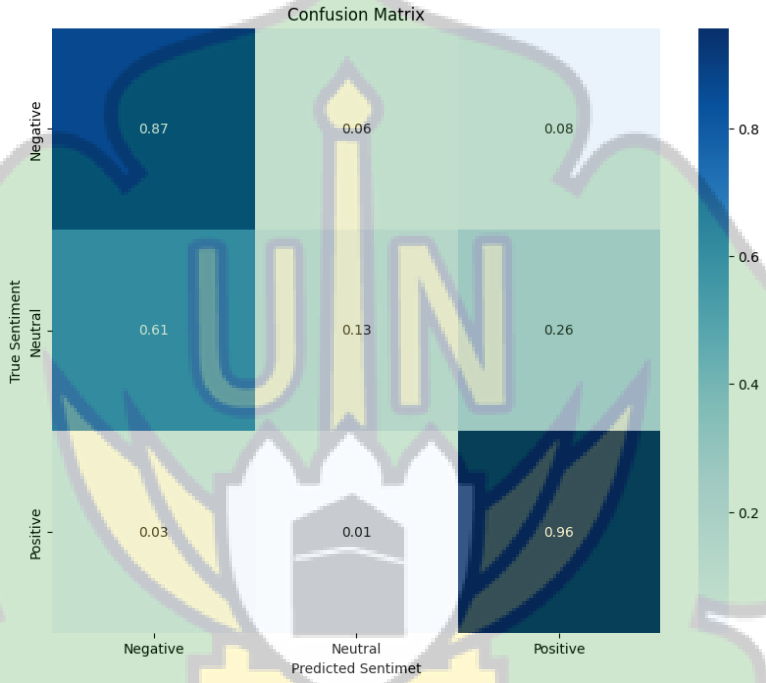


Gambar 4.23 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 10



Gambar 4.24 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.24** sentimen positif yang dapat diprediksi dengan benar sebesar 0.95, sentimen netral 0.13 dan sentimen negatif 0.88. Model mengalami kesusahan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.73 dan positif 0.15. **Gambar 4.25** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 32 dan *epoch* 10.



Gambar 4.25 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 10

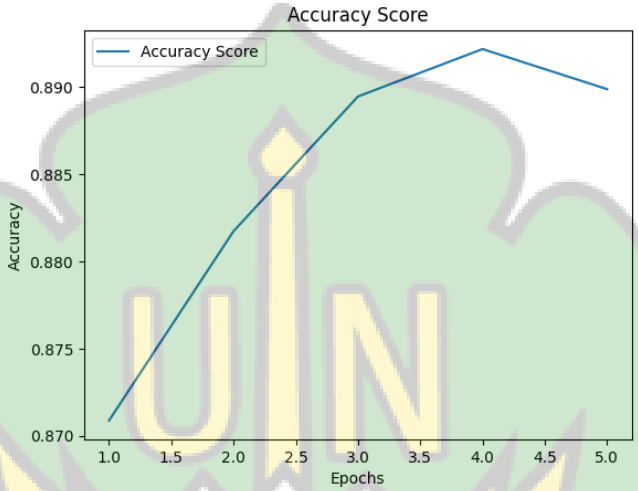
Pada **Gambar 4.25** sentimen positif yang dapat diprediksi dengan benar sebesar 0.96, sentimen netral 0.13 dan sentimen negatif 0.87. Model mengalami kesusahan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.61 dan positif 0.26.

Dari diagram *confusion matrix* menunjukkan bahwa hasil prediksi model memakai dataset pelabelan sesuai *rating* sangat tinggi dalam melakukan analisis sentimen pada label positif dan negatif. Tetapi pada label netral model kesulitan dalam memprediksi label sesuai dengan sentimen sebenarnya.

4.8.2 Grafik dan Diagram Model dari Pelabelan Menggunakan TextBlob

4.8.2.1 Grafik Model dari Pelabelan Menggunakan TextBlob

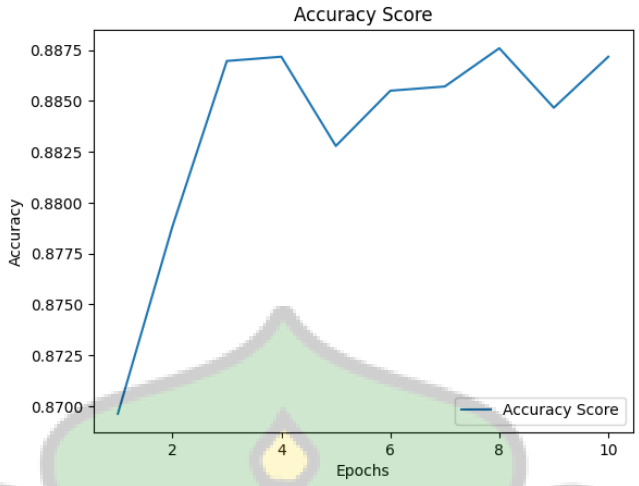
Pada tahapan ini dilakukan plot menjadi grafik dari hasil model dengan pelabelan menggunakan TextBlob. **Gambar 4.26** menunjukkan nilai *accuracy* dari model dengan *batch size* 16 dan *epoch* 5.



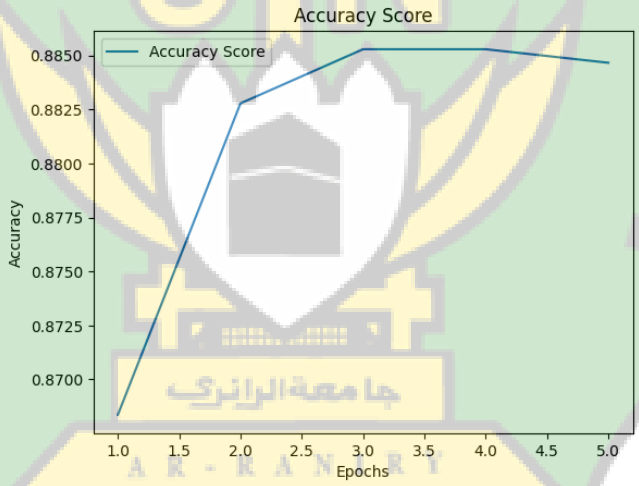
Gambar 4.26 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.26** *accuracy* awal 0.87 naik 0.02 dari *epoch* 1 ke *epoch* 4, lalu mengalami penurunan saat menuju *epoch* 5 sebesar 0.02 yang berakhir dengan nilai 0.888. Grafik yang awalnya naik lalu tiba-tiba turun disebabkan oleh model yang mengalami *overfitting*. **Gambar 4.27** menunjukkan nilai *accuracy* model dengan *batch size* 16 dan *epoch* 10.

Pada **Gambar 4.27** nilai *accuracy* awal 0.87 naik 0.0175 pada *epoch* 4, lalu mengalami penurunan 0.0050 pada *epoch* 5. Kemudian *accuracy* naik kembali 0.0050 pada *epoch* 8 lalu turun 0.0025 tetapi naik kembali dengan nilai yang sama pada *epoch* 10. Nilai *accuracy* berakhir dengan nilai 0.8875 pada *epoch* 10. Grafik yang naik turun disebabkan oleh model yang mengalami *overfitting*. **Gambar 4.28** menunjukkan grafik *accuracy* model dengan *batch size* 32 dan *epoch* 5.

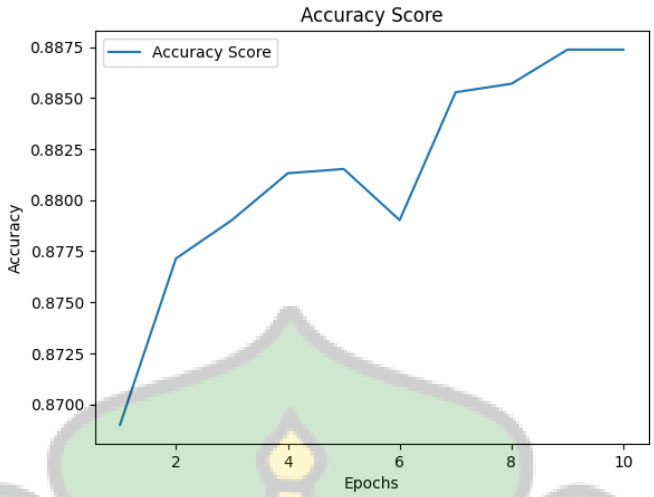


Gambar 4.27 Grafik *Accuracy* Model dengan *Batch Size* 16 dan *Epoch* 10



Gambar 4.28 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.28** nilai *accuracy* awal 0.8700 naik 0.0150 pada *epoch* 3 dan nilai ini tetap bertahan pada *epoch* 4. Tetapi pada *epoch* 5 mengalami sedikit penurunan sebesar 0.002 dengan akhir nilai *accuracy* sebesar 0.8830. Grafik yang awalnya naik perlahan turun disebabkan oleh model yang mengalami *overfitting*. **Gambar 4.29** menunjukkan grafik *accuracy* model dengan *batch size* 32 dan *epoch* 10.

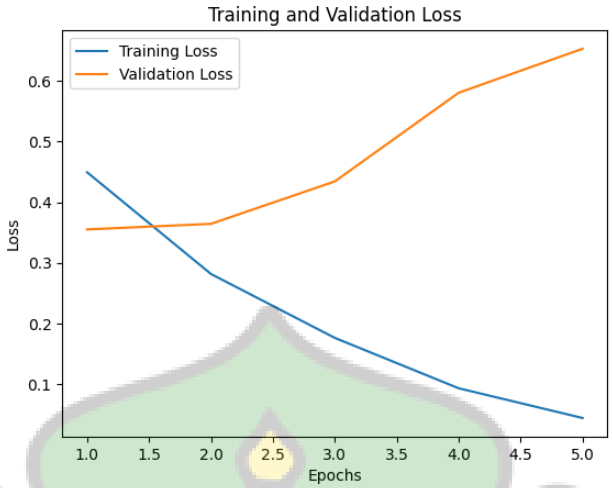


Gambar 4.29 Grafik *Accuracy* Model dengan *Batch Size* 32 dan *Epoch* 10

Pada **Gambar 4.29** nilai *accuracy* awal 0.8700 naik 0.012 pada *epoch* 5 tetapi turun 0.003 pada *epoch* 6. Kemudian nilai *accuracy* naik secara signifikan pada *epoch* 9 sebesar 0.0085 dan terus bertahan sampai *epoch* 10. Hasil akhir dari nilai *accuracy* pada *epoch* 10 adalah 0.8875 Grafik yang awalnya naik lalu mengalami sedikit penurunan kemudian naik secara signifikan yang berakhir dengan situasi stabil juga dipengaruhi oleh model yang mengalami *overfitting*.

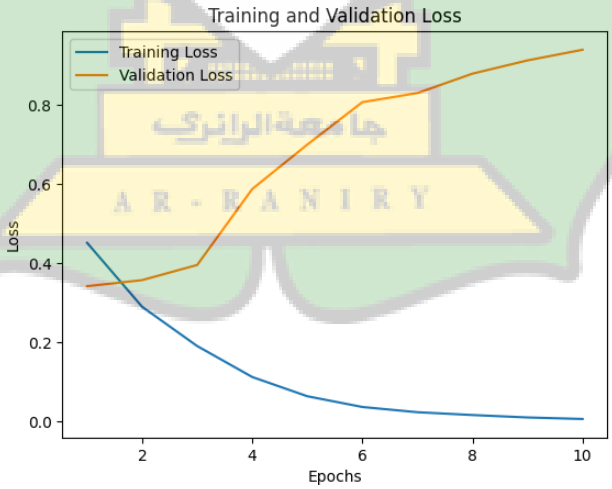
Dari hasil analisis grafik *accuracy* model dengan memakai dataset pelabelan menggunakan TextBlob, didapatkan model dengan *accuracy* terbaik pada model dengan *batchsize* 16 dan *epoch* 5 dengan nilai akhir 0.888. Semakin besar *batchsize* dan *epoch* yang digunakan, *accuracy* model semakin menurun. Ini disebabkan oleh model yang mengalami *overfitting*.

Setelah mendapatkan hasil grafik dari proses plot nilai *accuracy* pada model dari pelabelan menggunakan TextBlob, lalu dilakukan plot pada *training loss* dan *validation loss* dalam bentuk grafik pada setiap model pelabelan menggunakan TextBlob. **Gambar 4.30** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 16 dan *epoch* 5.



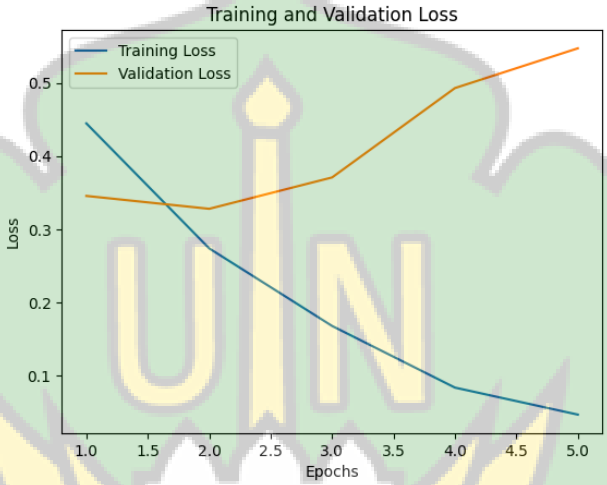
Gambar 4.30 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 5

Pada **Gambar 4.30** *training loss* menurun dari angka 0.46 menjadi <0.05 pada *epoch* 5. Tetapi *validation loss* meningkat dari angka 0.35 menjadi 0.60 pada *epoch* 5, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.31** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 16 dan *epoch* 10.



Gambar 4.31 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 16 dan *Epoch* 10

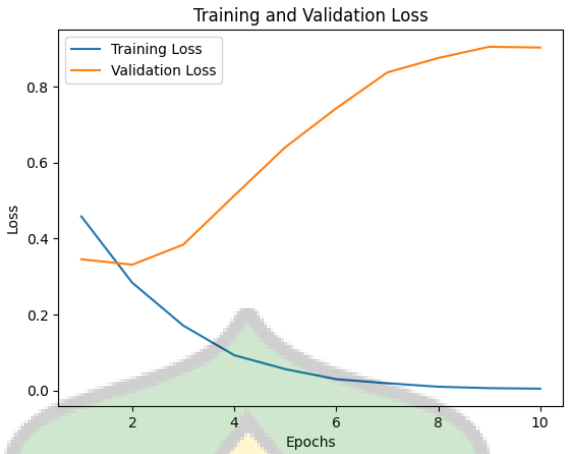
Pada **Gambar 4.31** *training loss* menurun dari angka 0.46 menjadi 0.05 pada *epoch* 6 dan bertahan dengan nilai yang sama sampai *epoch* 10. Tetapi *validation loss* meningkat dari angka 0.37 menjadi 0.80 pada *epoch* 10, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.32** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 32 dan *epoch* 5.



Gambar 4.32 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.32** *training loss* menurun dari angka 0.45 menjadi <0.05 pada *epoch* 5. Tetapi *validation loss* meningkat dari angka 0.35 menjadi 0.50 pada *epoch* 5, menunjukkan bahwa terjadi *overfitting* pada model. **Gambar 4.33** menunjukkan grafik *training* dan *validation loss* pada model dengan menggunakan *batch size* 32 dan *epoch* 10 pada dataset pelabelan menggunakan TextBlob.

Pada **Gambar 4.33** *training loss* menurun dari angka 0.45 menjadi <0.05 pada *epoch* 6 dan bertahan dengan nilai yang sama sampai *epoch* 10. Tetapi *validation loss* meningkat dari angka 0.35 menjadi 0.80 pada *epoch* 9 dan mengalami sedikit penurunan pada *epoch* 10, menunjukkan bahwa terjadi *overfitting* pada model.

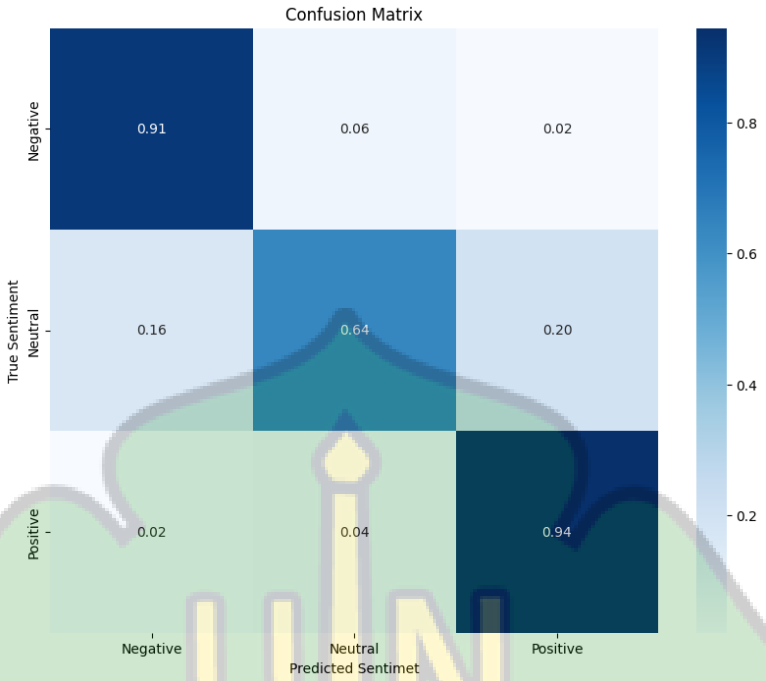


Gambar 4.33 Grafik *Training* dan *Validation Loss* dengan *Batch Size* 32 dan *Epoch* 10

Dari hasil *training* dan *validation loss* yang telah diplotkan ke dalam bentuk grafik menunjukkan bahwa, *training loss* pada model dengan memakai dataset pelabelan menggunakan *TextBlob* mengalami penurunan yang drastis. Hal ini memperlihatkan model semakin baik dalam menyesuaikan diri dengan data pelatihan dan melakukan prediksi yang lebih akurat. Namun, *validation loss* pada semua model dengan memakai dataset pelabelan menggunakan TextBlob mengalami kenaikan secara signifikan yang menunjukkan terjadinya *overfitting*. *Overfitting* terjadi ketika model terlalu kompleks dan telah "menghafal" data *training* hingga tingkat di mana tidak dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

4.8.2.2 Diagram *Confusion Matrix* dari Pelabelan Menggunakan TextBlob

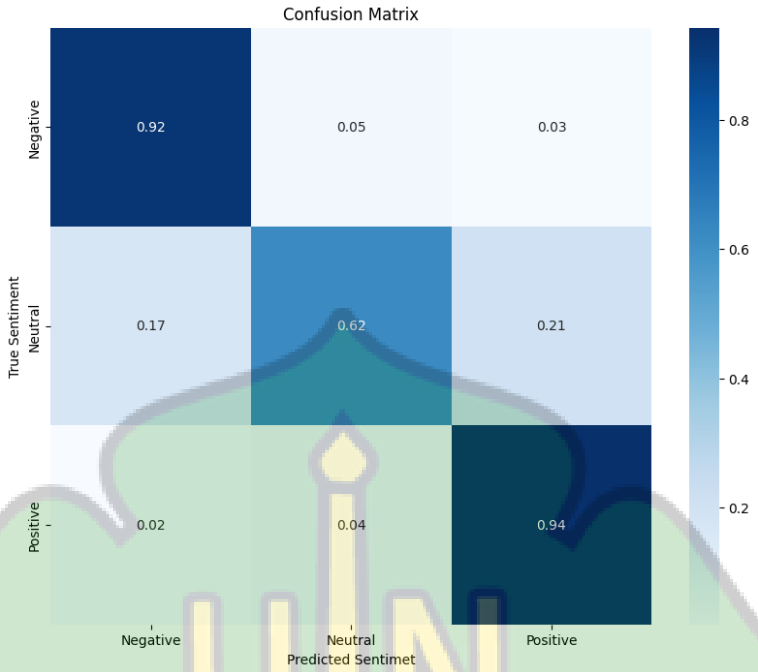
Setelah memplot *training loss* dan *validation loss* dalam bentuk grafik, langkah selanjutnya melakukan plot dari hasil model dalam melakukan analisis sentimen pada label positif, netral dan negatif dalam bentuk diagram *confusion matrix* pada model pelabelan menggunakan TextBlob. **Gambar 4.34** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 16 dan *epoch* 5.



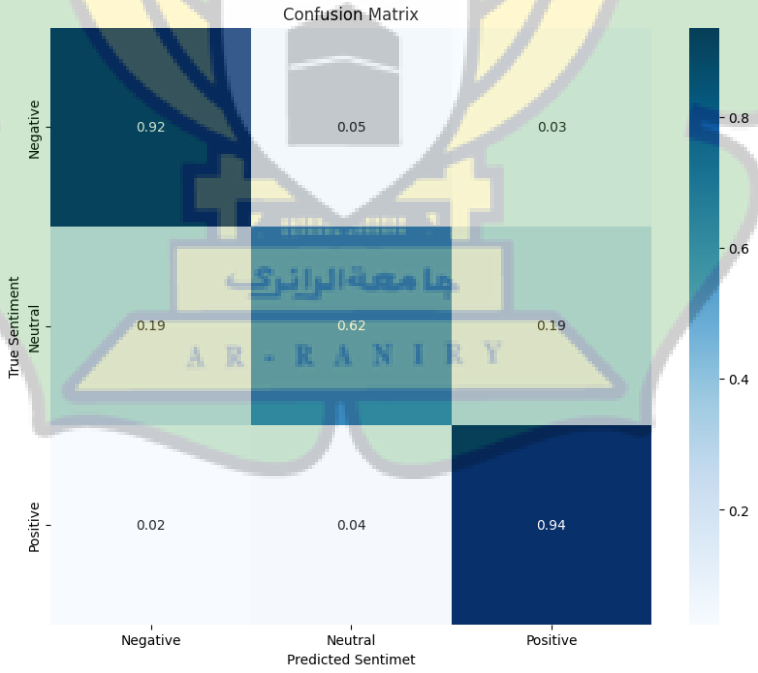
Gambar 4.34 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch 5*

Pada **Gambar 4.34** sentimen positif yang dapat diprediksi dengan benar sebesar 0.94, sentimen netral 0.64 dan sentimen negatif 0.91. Model kesulitan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.16 dan positif 0.20. **Gambar 4.35** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 16 dan *epoch* 10.

Pada **Gambar 4.35** sentimen positif yang dapat diprediksi dengan benar sebesar 0.94, sentimen netral 0.62 dan sentimen negatif 0.92. Model kesulitan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.17 dan positif 0.21. **Gambar 4.36** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 32 dan *epoch* 5.

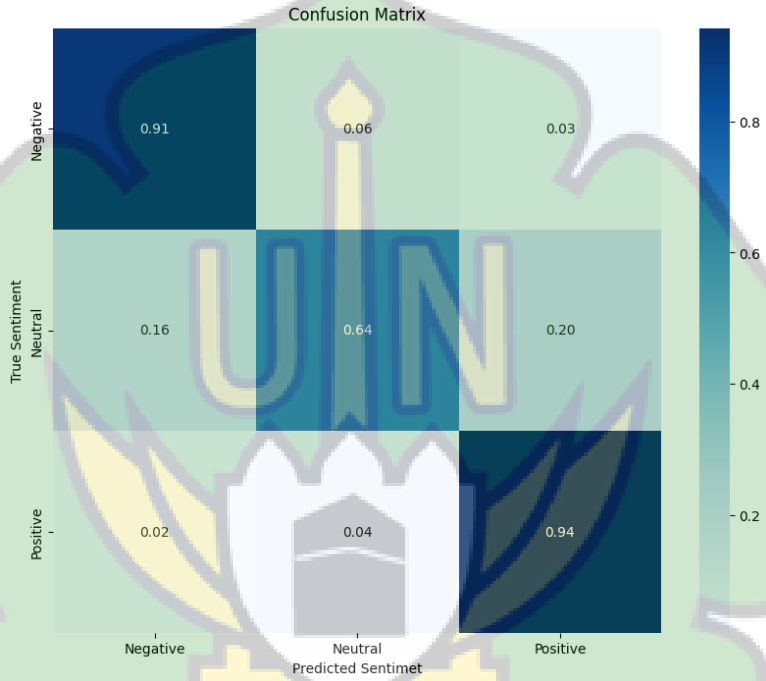


Gambar 4.35 Diagram *Confusion Matrix* dengan *Batch Size* 16 dan *Epoch* 10



Gambar 4.36 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 5

Pada **Gambar 4.36** sentimen positif yang dapat diprediksi dengan benar sebesar 0.94, sentimen netral 0.62 dan sentimen negatif 0.92. Model kesulitan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.19 dan positif 0.19. **Gambar 4.37** menunjukkan diagram *confusion matrix* pada model dengan *batch size* 32 dan *epoch* 10.



Gambar 4.37 Diagram *Confusion Matrix* dengan *Batch Size* 32 dan *Epoch* 10

Pada **Gambar 4.37** sentimen positif yang dapat diprediksi dengan benar sebesar 0.94, sentimen netral 0.64 dan sentimen negatif 0.91. Model kesulitan dalam melakukan prediksi pada sentimen netral, model mengklasifikasi label sentimen netral menjadi label negatif sebesar 0.16 dan positif 0.20.

Dari diagram *confusion matrix* menunjukkan bahwa hasil prediksi model memakai dataset pelabelan menggunakan TextBlob sangat tinggi dalam melakukan analisis sentimen pada label positif dan negatif. Tetapi pada label netral model sedikit kesulitan dalam memprediksi label sesuai dengan sentimen sebenarnya.

4.9 Perbandingan Hasil Pelabelan Sesuai *Rating* dan TextBlob

Dari semua hasil evaluasi performa model dengan pengujian menggunakan *hyperparameter* yang bervariasi pada 2 dataset yang menggunakan pelabelan berbeda. Maka didapatkan nilai performa model tertinggi pada model dengan menggunakan *hyperparameters batch size* 16 dan *epoch* 5 pada dataset pelabelan sesuai *rating*. Hal ini dikarenakan oleh pemilihan *batch size* 16 yang merujuk pada jumlah sampel data yang diproses oleh model dalam satu iterasi lebih sedikit daripada *batch size* 32 sehingga data lebih akurat. Namun pada kasus pemilihan *epoch*, IndoBERT-Large phase-1 lebih efektif pada pemilihan *epoch* yang lebih kecil. Pemilihan *epoch* yang lebih besar menyebabkan *overfiitting* pada model. *Overfitting* terjadi saat model mempelajari data terlalu detail dan dataset yang digunakan kurang bervariatif. *Vocabulary* pada model IndoBERT-Large phase 1 sudah sangat komplek sehingga penggunaan dataset yang kurang bervariatif menjadi faktor utama terjadi *overfitting* pada model. **Tabel 4.9** menunjukkan perbandingan nilai evaluasi dari semua model IndoBERT-Large yang telah dibuat.

Tabel 4.9 Perbandingan Nilai Evaluasi Performa Model

Jenis Pelabelan	Model		Performa Model			
	Batch Size	*Epoch*	*Accuracy*	*Precision*	*Recall*	*F* 1-*Score*
Label *Rating*	16	5	**0.903**	**0.900**	**0.903**	**0.901**
	16	10	0.902	0.896	0.902	0.898
	32	5	0.898	0.894	0.898	0.895
	32	10	0.898	0.888	0.898	0.892
Label TextBlob	16	5	0.889	0.886	0.889	0.887
	16	10	0.887	0.883	0.887	0.884
	32	5	0.884	0.881	0.884	0.882
	32	10	0.887	0.884	0.887	0.885

Pada **Tabel 4.9** menunjukkan model IndoBERT-Large phase 1 yang menggunakan *batch size* 16 dan *epoch* 5 mendapatkan nilai evaluasi performa model paling tinggi. Didapatkan hasil akhir ketika sudah dibulatkan dengan nilai *accuracy* 0.903, *precision* 0.900, *recall* 0.903 dan *f*1-*score* 0.901.

Dari semua nilai *accuracy* yang telah diplotkan ke dalam bentuk grafik, nilai *accuracy* dari semua model yang dilatih memakai dataset pelabelan menggunakan TextBlob mempunyai riwayat grafik yang lebih bagus daripada model yang dilatih memakai dataset pelabelan sesuai *rating*. Ini menunjukkan hasil yang berbanding terbalik dengan hasil nilai dari performa model yang menunjukkan nilai lebih bagus menggunakan dataset pelabelan sesuai *rating* daripada menggunakan TextBlob. Kasus ini disebabkan oleh nilai evaluasi performa model pada matriks *accuracy, precision, recall* dan *f*1-*score* hanya mengambil nilai kelas positif dan negatif saja, tidak melibatkan kelas netral saat menghitung keseluruhan hasil dari evaluasi performa model.

Pada setiap grafik *accuracy score* dari model yang menggunakan dataset pelabelan sesuai *rating* mengalami penurunan pada *epoch* 2 atau setelah melewati *epoch* 2 dan terus turun sampai pada *epoch* yang ditentukan. Tetapi pada setiap grafik *accuracy score* dari model yang dilatih dengan dataset pelabelan menggunakan TextBlob mengalami peningkatan dari *epoch* 1 sampai dengan *epoch* 4 selanjutnya nilai *accuracy* mengalami penurunan. Hasil ini menunjukkan bahwa model yang memakai dataset pelabelan sesuai *rating* bagus jika menggunakan *epoch* 1 dan 2. Sedangkan untuk model yang memakai dataset pelabelan menggunakan TextBlob bagus jika menggunakan *epoch* 4 dan 5 tetapi pada model dengan *batch size* 32 dan *epoch* 10 mengalami peningkatan yang signifikan setelah mengalami penurunan pada *epoch* 6. Model yang memakai dataset pelabelan sesuai *rating* lebih mudah terjadi *overfitting* daripada model yang memakai dataset palabelan menggunakan TextBlob.

Dari hasil *training* dan *validation loss* yang telah diplotkan ke dalam bentuk grafik menunjukkan bahwa, *training loss* pada semua model dengan menggunakan 2 dataset pelabelan yang berbeda mengalami penurunan yang drastis. Hal ini memperlihatkan model semakin baik dalam menyesuaikan diri dengan data pelatihan dan melakukan prediksi yang lebih akurat. Namun, *validation loss* pada semua model dengan menggunakan 2 dataset pelabelan yang berbeda mengalami kenaikan secara signifikan yang menandakan terjadinya *overfitting*. *Overfitting*

terjadi ketika model terlalu kompleks dan telah "menghafal" data *training* hingga tingkat di mana tidak dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Training loss menurun mendekati angka 0 saat berada *epoch* 5 atau *epoch* 6 dan angka tersebut terus stabil sampai pada *epoch* 10. Sebaliknya, *validation loss* meningkat selaras dengan semakin tinggi *epoch* yang digunakan. *Validation loss* pada model dengan *batch size* 15 dan 32 dengan *epoch* 10 yang memakai dataset pelabelan menggunakan TextBlob menyetuh angka diatas 0.80, memperlihatkan angka yang cukup tinggi untuk kasus *overfitting.*

Dari diagram *confusion matrix* menunjukkan bahwa hasil prediksi model sangat tinggi dalam melakukan analisis sentimen pada label positif dan negatif. Tetapi pada label netral model kesulitan dalam memprediksi label sesuai dengan sentimen sebenarnya. Model yang memakai dataset pelabelan sesuai *rating* lebih kesulitan meprediksi sentimen netral daripada model yang memakai dataset pelabelan menggunakan TextBlob. Hal ini disebabkan pada dataset pelabelan sesuai *rating* mengandung sentimen yang berbanding terbalik dan rancu dengan *rating* yang diberikan oleh pengguna yang berkomentar. Misal pemberian *rating* 3 saat berkomentar tetapi mengandung kata yang positif atau negatif sehingga ini menjadi faktor utama model kesulitan memprediksi label netral pada dataset pelabelan sesuai *rating*. **Tabel 4.10** menunjukkan perbandingan hasil analisis sentimen yang ditampilkan oleh *confusion matrix*.

Tabel 4.10 Perbandingan Hasil Analisis Sentimen

Jenis Pelabelan	Model		Sentimen yang Berhasil Diprediksi dengan Benar		
	Batch Size	*Epoch*	Positif	Netral	Negatif
Label *Rating*	16	5	**0.96**	0.20	0.88
	16	10	**0.96**	0.18	0.88
	32	5	0.95	0.13	0.88
	32	10	**0.96**	0.13	0.87
Label TextBlob	16	5	0.94	**0.64**	0.91
	16	10	0.94	0.62	**0.92**
	32	5	0.94	0.62	**0.92**
	32	10	0.94	**0.64**	0.91

Pada **Tabel 4.10** menunjukkan nilai prediksi yang benar tertinggi pada label positif didapatkan oleh 3 model yaitu model dengan memakai *batch size* 16 dan *epoch* 5 pada pelabelan sesuai *rating*, *batch size* 32 dan *epoch* 5 pada pelabelan sesuai *rating*, dan *batch size* 32 dan *epoch* 10 pada pelabelan sesuai *rating* dengan akurasi prediksi yang didapatkan 0.96. Pada pelabelan menggunakan TextBlob memiliki nilai akurasi prediksi 0.94 terhadap label positif pada semua model. Nilai akurasi prediksi pelabelan sesuai *rating* lebih tinggi 0.01 sampai dengan 0.02 dari pelabelan menggunakan TextBlob. Hal ini menunjukkan pelabelan sesuai *rating* lebih baik terhadap analisis sentimen pada label positif dibandingkan pelabelan menggunakan TextBlob.

Akurasi prediksi yang benar tertinggi pada label netral didapatkan oleh 2 model, yaitu model dengan memakai *batch size* 16 dan *epoch* 5 pada pelabelan menggunakan TextBlob, dan *batch size* 32 dan *epoch* 10 pada pelabelan menggunakan TextBlob, dengan akurasi prediksi yang didapatkan 0.64. Pelabelan menggunakan TextBlob memiliki akurasi prediksi 0.62 dan 0.64 pada semua model terhadap sentimen netral. Sedangkan pelabelan sesuai *rating* memiliki nilai akurasi prediksi yang sangat rendah pada label netral dengan nilai yang didapatkan antara 0.13 – 0.20. Hasil ini mempresentasikan pelabelan menggunakan TextBlob lebih baik dalam menganalisis sentimen pada label netral dibandingkan pelabelan sesuai *rating.*

Pada label negatif didapatkan akurasi prediksi tertinggi pada 2 model yaitu model dengan memakai *batch size* 16 dan *epoch* 10 pada pelabelan menggunakan TextBlob, dan *batch size* 32 dan *epoch* 5 pada pelabelan menggunakan TextBlob, dengan akurasi prediksi yang didapatkan 0.92. Pelabelan menggunakan TextBlob memiliki akurasi prediksi 0.91 dan 0.92 terhadap label negatif pada semua model. Sedangkan pelabelan sesuai *rating* memiliki akurasi prediksi 0.87 dan 0.88. Nilai ini menunjukkan pelabelan menggunakan TextBlob lebih baik melakukan analisis pada label negatif dibandingkan pelabelan sesuai *rating*. Dari semua label yang telah dianalisis, maka didapatkan hasil pelabelan menggunakan TextBlob memiliki model dengan prediksi yang lebih akurat dibandingkan pelabelan sesuai *rating.*

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari Penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan antara lain:

1. Performa model IndoBERT-Large phase 1 dengan pembagian data *training* 80% dan data *validation* 20% didapatkan hasil terbaik pada model dengan *batch size* 16 dan *epoch* 5 yang menggunakan dataset pelabelan sesuai *rating* dengan *accuracy* 0.903, *precision* 0.900, *recall* 0.903 dan *f*1-*score* 0.901.
2. Hasil evaluasi menggunakan *confusion matrix* menunjukkan model dengan *batchsize* 16, *epoch* 5 dan *batch size* 32, *epoch* 10 pada dataset pelabelan sesuai TextBlob mendapatkan hasil terbaik dalam melakukan analisis sentimen. Nilai yang didapatkan sama besar yaitu 0.94 untuk label positif, 0.64 label netral dan 0.91 label negatif.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya

1. Dalam penelitian ini kata'bsi' adalah kata kunci yang banyak terdapat pada dataset tetapi kata 'bsi' tidak terdapat dalam *vocabulary*. Penulis menyarankan agar penelitian selanjutnya mencoba untuk menambahkan kata-kata yang sesuai dengan topik penelitian ke dalam *vocabulary*, dengan tujuan untuk mengetahui dampak terhadap performa model.
2. Menggunakan jumlah data yang seimbang antara sentimen positif, netral dan negatif serta menggunakan metode pelabelan dataset lainnya untuk mengetahui pengaruh pada performa model.
3. Melakukan penelitian dengan memanfaatkan model IndoBERT lainnya, seperti IndoBERT-Base, IndoBERT-liteBase dan IndoBERT-liteLarge untuk memajukan penelitian dalam pengolahan bahasa alami (NLP) berbahasa Indonesia.

DAFTAR PUSTAKA

Akhmad, E. P. A. (2023). Analisis Sentimen Ulasan Aplikasi DLU Ferry Pada Google Play Store Menggunakan Bidirectional Encoder Representations from Transformers. *Jurnal Aplikasi Pelayaran Dan Kepelabuhan, Volume 13, Nomer 2*, *13*(1), 10–27. https://doi.org/10.30649/japk.v13i2.94

Arifiyanti, A. A., Shantika, N. R., & Syafira, A. O. (2022). *ANALISIS SENTIMEN ULASAN PENGGUNA BSI MOBILE PADA*. 283–288.

Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1–6. https://doi.org/10.51983/ajcst-2019.8.s2.2037

Chen, Q., Zhuo, Z., & Wang, W. (2019). *BERT for Joint Intent Classification and Slot Filling*. http://arxiv.org/abs/1902.10909

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

Elngar, A. A., Arafa, M., Fathy, A., & Moustafa, B. (2021). Image Classification Based On CNN: A Survey. *Journal of Cybersecurity and Information Management (JCIM)*, *6*(June), 1–162. papers3://publication/uuid/186091F2-049F-46DC-ACC1-5010D2F513E9

Evasaria M. Sipayung, Herastia Maharani, & Ivan Zefanya. (2016). Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier. *Jurnal Sistem Informasi (JSI)*, *8*(1), 2355–4614. http://ejournal.unsri.ac.id/index.php/jsi/index

Firmansyah, D., & Ahsan, M. (2023). Monitoring Kualitas Pada Aplikasi MyPertamina Berdasarkan Rating Pengguna di Google Play Menggunakan Diagram Kendali p. *Jurnal Sains Dan Seni ITS*, *12*(2). https://doi.org/10.12962/j23373520.v12i2.115315

Fransiscus, & Girsang, A. S. (2022). Sentiment Analysis of COVID-19 Public Activity Restriction (PPKM) Impact using BERT Method. *International Journal of Engineering Trends and Technology*, *70*(12), 281–288. https://doi.org/10.14445/22315381/IJETT-V70I12P226

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. 1–17. http://arxiv.org/abs/2008.05756

Guntara, R. G. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *5*(1), 55–60. https://doi.org/10.47233/jteksis.v5i1.750

Herwinsyah, & Witanti, A. (2022). Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Sistem Informasi Dan Informatika (Simika)*, *5*(1), 59–67. https://doi.org/10.47080/simika.v5i1.1411

Hidayat, I. R., & Maharani, W. (2022). General Depression Detection Analysis Using IndoBERT Method. *International Journal on Information and Communication Technology (IJoICT)*, *8*(1), 41–51. https://doi.org/10.21108/ijoict.v8i1.634

Jayanti, F., & Arista, N. T. (2019). Persepsi Mahasiswa Terhadap Pelayanan Perpustakaan Universitas Trunojoyo Madura. *Competence : Journal of Management Studies*, *12*(2), 205–223. https://doi.org/10.21107/kompetensi.v12i2.4958

Kavabilla, F. E., Widiharih, T., & Warsito, B. (2023). Analisis Sentimen Pada Ulasan Aplikasi Investasi Online Ajaib Pada Google Play Menggunakan Metode Support Vector Machine Dan Maximum Entropy. *Jurnal Gaussian*, *11*(4), 542–553. https://doi.org/10.14710/j.gauss.11.4.542-553

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

Mahfudiyah, N., & Alamsyah, A. (2022). Analisis Persepsi Konsumen Terhadap Kualitas Layanan Gojek Menggunakan Sentiment Analysis Dan Topic Modeling Berdasarkan Deep Learning IndoBERT. *E-Proceeding of Management*, *9*(4), 1812–1817.

Nongthombam, K., & Sharma, D. (2021). Data Analysis Using Python. *International Journal of Engineering Research and Technology (IJERT)*, *10*(07), 463–468.

Nugroho, K. S., Sukmadewa, A. Y., Wuswilahaken Dw, H., Bachtiar, F. A., & Yudistira, N. (2021). BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews. *ACM International Conference Proceeding Series*, 258–264. https://doi.org/10.1145/3479645.3479679

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

Rantemangiling, Y., Mamesah, E. L., & Setiabudhi, D. O. (2022). Analisis Yuridis Mengenai Merger Bank Syariah Mandiri, Bri Syariah, Dan Bni Syariah Menjadi Bank Syariah Indonesia (BSI). *Lex Crimen*, *11*(5), 1–15.

Sardanto, R., & Ratnanto, S. (2016). Pengaruh Persepsi Konsumen Terhadap Keputusan Pada Angkringan Kota Kediri. *Jurnal Benefit*, *3*(1), 31–45.

Sari, D. M., Fasa, M. I., & Suharto, S. (2021). Manfaat Dan Risiko Penggunaan Layanan Perbankan Melalui Aplikasi Mobile Banking. *Al-Infaq: Jurnal Ekonomi Islam*, *12*(2), 170. https://doi.org/10.32507/ajei.v12i2.892

Sudhir, P., & Suresh, V. D. (2021). Comparative study of various approaches, applications and classifiers for sentiment analysis. *Global Transitions Proceedings*, *2*(2), 205–211. https://doi.org/10.1016/j.gltp.2021.08.004

Supriyadi, P. F., & Sibaroni, Y. (2023). *Xiaomi Smartphone Sentiment Analysis on Twitter Social Media Using IndoBERT*. *10*(1), 19–30. https://doi.org/10.30865/jurikom.v10i1.5540

Utami, M. A. A. T., Silvianti, P., & Masjkur, M. (2023). Algoritme Support Vector Machine untuk Analisis Sentimen Berbasis Aspek Ulasan Game Online Mobile Legends: Bang-Bang. *Xplore: Journal of Statistics*, *12*(1), 63–77. https://doi.org/10.29244/xplore.v12i1.1064

Vaswani, A., Shazer, N., Pamar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Power Systems*, *Nips*, 47–82. https://doi.org/10.1007/978-3-319-29409-4_3

Wang, K., Wang, H., Liu, M., Xing, X., & Han, T. (2018). Survey on person re-identification based on deep learning. *CAAI Transactions on Intelligence Technology*, *3*(4), 219–227. https://doi.org/10.1049/trit.2018.1001

Wati, R., & Ernawati, S. (2021). Analisis Sentimen Persepsi Publik Mengenai PPKM Pada Twitter Berbasis SVM Menggunakan Python. *Jurnal Teknik Informatika UNIKA Santo Thomas*, *06*, 240–247. https://doi.org/10.54367/jtiust.v6i2.1465

Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., & Purwarianti, A. (2020).

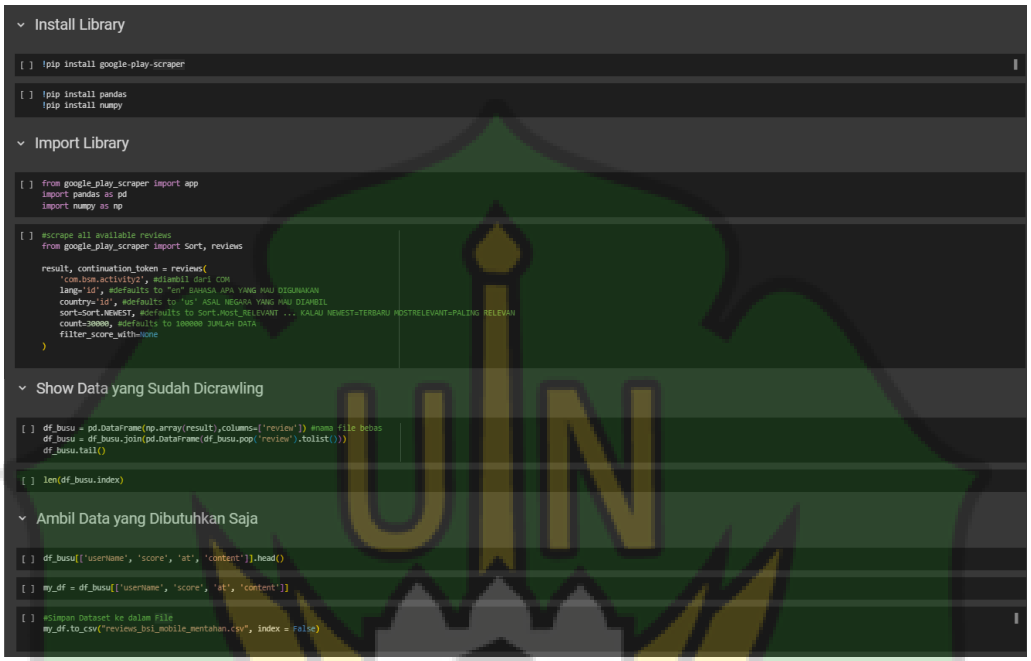
IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding. http://arxiv.org/abs/2009.05387

Zhou, Q., Lim, F. J., Yu, H., Xu, G., Ren, X., Liu, D., Wang, X., Mai, X., & Xu, H. (2021). A study on factors affecting service quality and loyalty intention in mobile banking. *Journal of Retailing and Consumer Services*, *60*, 0–22. https://doi.org/10.1016/j.jretconser.2020.102424

Zhu, L., Zhu, Z., Zhang, C., Xu, Y., & Kong, X. (2023). Multimodal sentiment analysis based on fusion methods: A survey. *Information Fusion*, *95*(February), 306–325. https://doi.org/10.1016/j.inffus.2023.02.028

LAMPIRAN

Lampiran 1. *Source Code* Untuk Pengumpulan Data



Install Library

```
[ ] !pip install google-play-scraper

[ ] !pip install pandas
    !pip install numpy
```

Import Library

```
[ ] from google_play_scraper import app
    import pandas as pd
    import numpy as np

[ ] #scrape all available reviews
    from google_play_scraper import Sort, reviews

    result, continuation_token = reviews(
        'com.bsm.activity2', #diambil dari COM
        lang='id', #defaults to "en" BAHASA APA YANG MAU DIGUNAKAN
        country='id', #defaults to 'us' ASAL NEGARA YANG MAU DIAMBIL
        sort=Sort.NEWEST, #defaults to Sort.MOST_RELEVANT ... KALAU NEWEST=TERBARU MOSTRELEVANT=PALING RELEVAN
        count=300000, #defaults to 100000 JUMLAH DATA
        filter_score_with=None
    )
```

Show Data yang Sudah Dicrawling

```
[ ] df_busu = pd.DataFrame(np.array(result),columns=['review']) #nama file bebas
    df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
    df_busu.tail()

[ ] len(df_busu.index)
```

Ambil Data yang Dibutuhkan Saja

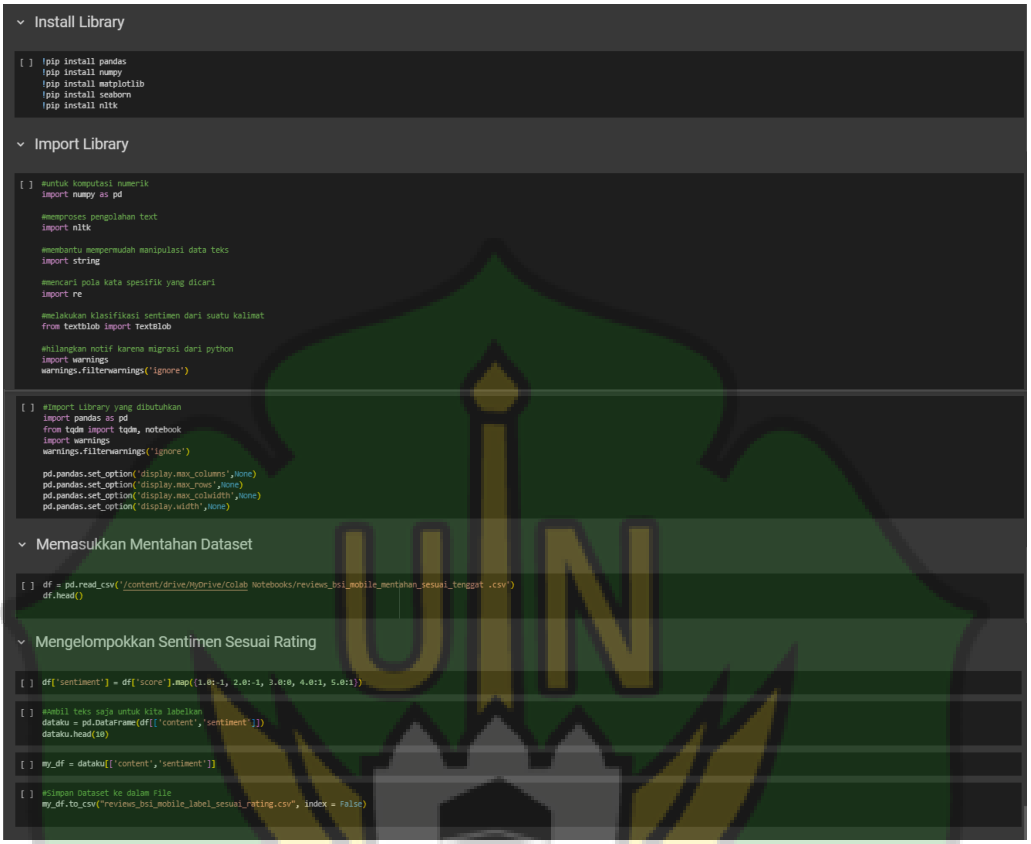
```
[ ] df_busu[['userName', 'score', 'at', 'content']].head()

[ ] my_df = df_busu[['userName', 'score', 'at', 'content']]

[ ] #Simpan Dataset ke dalam File
    my_df.to_csv("reviews_bsi_mobile_mentahan.csv", index = False)
```

جامعة الرانري
AR-RANIRY

Lampiran 2. *Source Code* Pelabelan Data Sesuai *Rating*



Install Library

```
[ ] !pip install pandas
    !pip install numpy
    !pip install matplotlib
    !pip install seaborn
    !pip install nltk
```

Import Library

```
[ ] #untuk komputasi numerik
    import numpy as pd

    #memproses pengolahan text
    import nltk

    #membantu mempermudah manipulasi data teks
    import string

    #mencari pola kata spesifik yang dicari
    import re

    #melakukan klasifikasi sentimen dari suatu kalimat
    from textblob import TextBlob

    #hilangkan notif karena migrasi dari python
    import warnings
    warnings.filterwarnings('ignore')
```

```
[ ] #Import Library yang dibutuhkan
    import pandas as pd
    from tqdm import tqdm, notebook
    import warnings
    warnings.filterwarnings('ignore')

    pd.pandas.set_option('display.max_columns',None)
    pd.pandas.set_option('display.max_rows',None)
    pd.pandas.set_option('display.max_colwidth',None)
    pd.pandas.set_option('display.width',None)
```

Memasukkan Mentahan Dataset

```
[ ] df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/reviews_bsi_mobile_mentahan_sesuai_tenggat .csv')
    df.head()
```

Mengelompokkan Sentimen Sesuai Rating

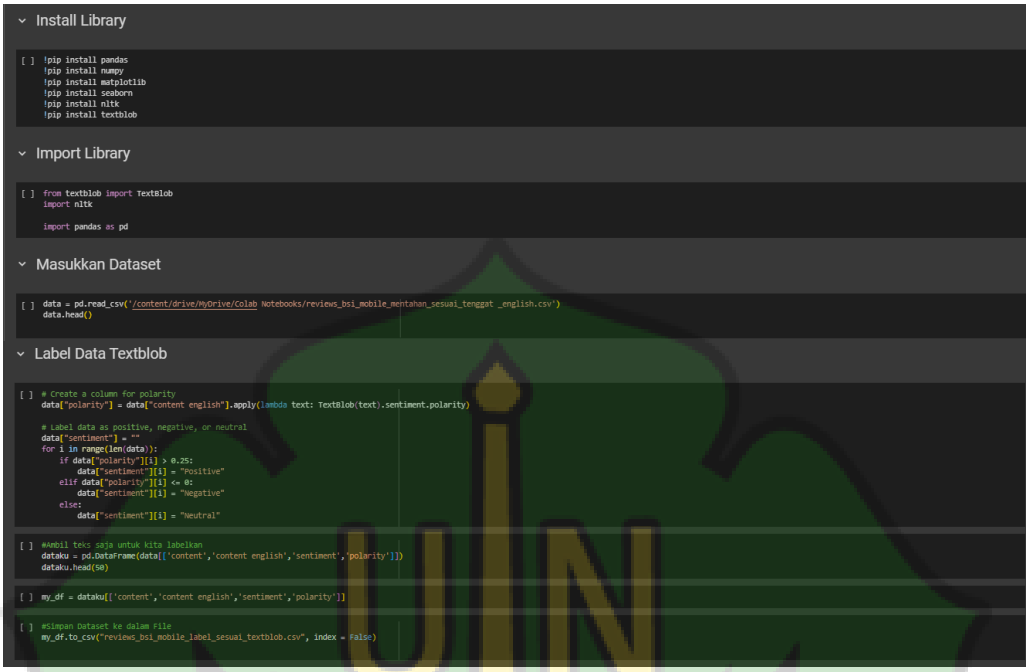
```
[ ] df['sentiment'] = df['score'].map({1.0:-1, 2.0:-1, 3.0:0, 4.0:1, 5.0:1})
```

```
[ ] #Ambil teks saja untuk kita labelkan
    dataku = pd.DataFrame(df[['content','sentiment']])
    dataku.head(10)
```

```
[ ] my_df = dataku[['content','sentiment']]
```

```
[ ] #Simpan Dataset ke dalam File
    my_df.to_csv("reviews_bsi_mobile_label_sesuai_rating.csv", index = False)
```

Lampiran 3. *Source Code* Pelabelan Menggunakan TextBlob



Install Library

```
[ ] !pip install pandas
    !pip install numpy
    !pip install matplotlib
    !pip install seaborn
    !pip install nltk
    !pip install textblob
```

Import Library

```
[ ] from textblob import TextBlob
    import nltk

    import pandas as pd
```

Masukkan Dataset

```
[ ] data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/reviews_bsi_mobile_mentahan_sesuai_tenggat _english.csv')
    data.head()
```

Label Data Textblob

```
[ ] # Create a column for polarity
    data["polarity"] = data["content english"].apply(lambda text: TextBlob(text).sentiment.polarity)

    # Label data as positive, negative, or neutral
    data["sentiment"] = ""
    for i in range(len(data)):
        if data["polarity"][i] > 0.25:
            data["sentiment"][i] = "Positive"
        elif data["polarity"][i] <= 0:
            data["sentiment"][i] = "Negative"
        else:
            data["sentiment"][i] = "Neutral"
```

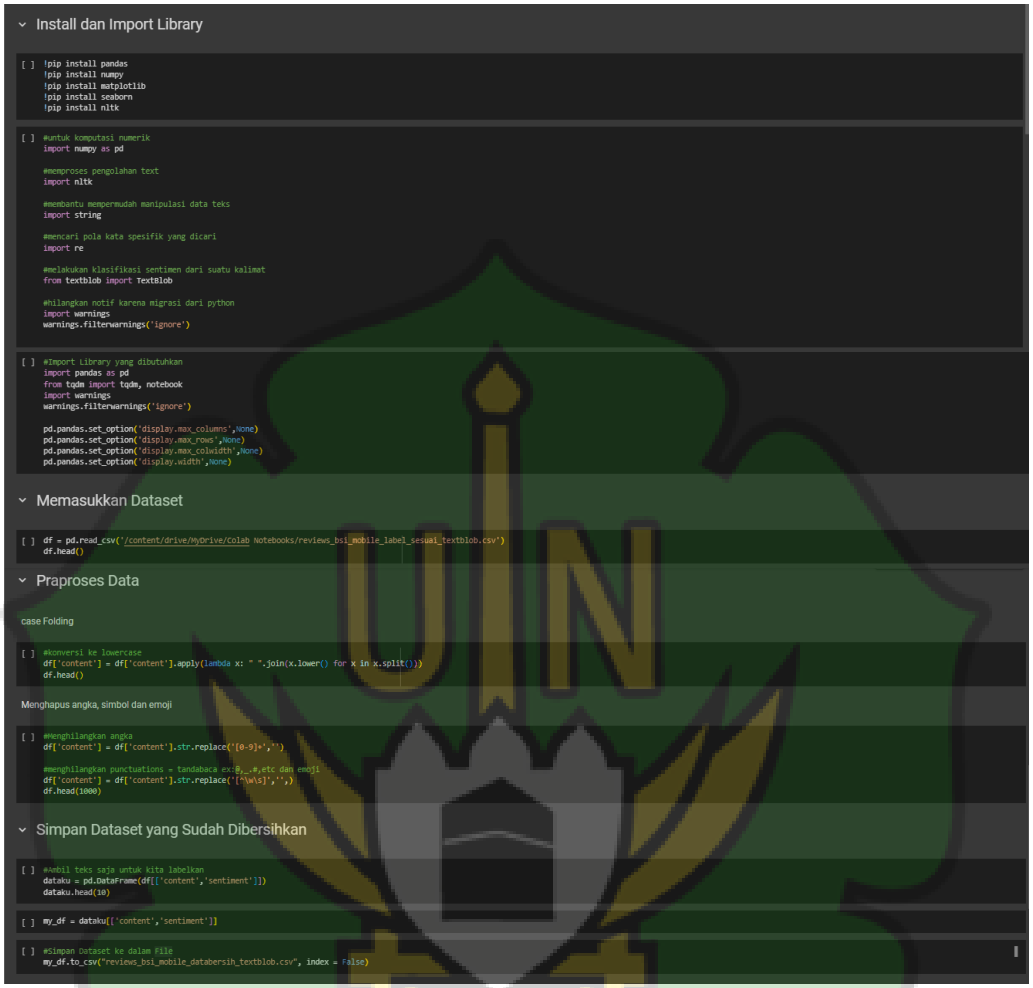
```
[ ] #Ambil teks saja untuk kita labelkan
    dataku = pd.DataFrame(data[['content','content english','sentiment','polarity']])
    dataku.head(50)
```

```
[ ] my_df = dataku[['content','content english','sentiment','polarity']]
```

```
[ ] #Simpan Dataset ke dalam File
    my_df.to_csv("reviews_bsi_mobile_label_sesuai_textblob.csv", index = False)
```

UIN
جامعة الرانري
AR-RANIRY

Lampiran 4. *Source Code* Untuk *Pre-Processing* Data



Install dan Import Library

```
[ ] !pip install pandas
    !pip install numpy
    !pip install matplotlib
    !pip install seaborn
    !pip install nltk
```

```
[ ] #untuk komputasi numerik
    import numpy as pd

    #memproses pengolahan text
    import nltk

    #membantu mempermudah manipulasi data teks
    import string

    #mencari pola kata spesifik yang dicari
    import re

    #melakukan klasifikasi sentimen dari suatu kalimat
    from textblob import TextBlob

    #hilangkan notif karena migrasi dari python
    import warnings
    warnings.filterwarnings('ignore')
```

```
[ ] #Import Library yang dibutuhkan
    import pandas as pd
    from tqdm import tqdm, notebook
    import warnings
    warnings.filterwarnings('ignore')

    pd.pandas.set_option('display.max_columns',None)
    pd.pandas.set_option('display.max_rows',None)
    pd.pandas.set_option('display.max_colwidth',None)
    pd.pandas.set_option('display.width',None)
```

Memasukkan Dataset

```
[ ] df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/reviews_bsi_mobile_label_sesuai_textblob.csv')
    df.head()
```

Praproses Data

case Folding

```
[ ] #konversi ke lowercase
    df['content'] = df['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))
    df.head()
```

Menghapus angka, simbol dan emoji

```
[ ] #menghilangkan angka
    df['content'] = df['content'].str.replace('[0-9]+','')

    #menghilangkan punctuations = tandabaca ex:@,_,#,etc dan emoji
    df['content'] = df['content'].str.replace('[^\w\s]','',)
    df.head(1000)
```

Simpan Dataset yang Sudah Dibersihkan

```
[ ] #Ambil teks saja untuk kita labelkan
    dataku = pd.DataFrame(df[['content','sentiment']])
    dataku.head(10)
```

```
[ ] my_df = dataku[['content','sentiment']]
```

```
[ ] #simpan Dataset ke dalam File
    my_df.to_csv("reviews_bsi_mobile_databersih_textblob.csv", index = False)
```

جامعة الرانري
AR-RANIRY

Lampiran 5. *Source Code* Untuk *Split Data* dan *Fine-Tuning* IndoBERT

Sentiment Analysis Using IndoBERT Fine-Tuning model

Install dan Import Library yang dibutuhkan

```
[ ] !pip install panda
    !pip install numpy
    !pip install matplotlib
    !pip install seaborn
    !pip install nltk
    !pip install Transformers
```

```
[ ] pip install git+https://github.com/andreinechaev/nvcc4jupyter.git
```

```
[ ] load_ext nvcc_plugin
```

```
[ ] %%cuda --name testGoogleColab.cu

    #include <stdio.h>

    #define gpuErrchk(ans) { gpuAssert((ans), __FILE__, __LINE__); }
    inline void gpuAssert(cudaError_t code, const char* file, int line, bool abort = true)
    {
        if (code != cudaSuccess)
        {
            fprintf(stderr, "GPUassert: %s %s %d\n", cudaGetErrorString(code), file, line);
            if (abort) exit(code);
        }
    }

    __global__ void add(int a, int b, int *c) {
        *c = a + b; }

    int main() {

        // --- Host declarations and initializations
        int a, b, c;
        a = 2;
        b = 6;

        // --- Device allocations
        int *d_c; gpuErrchk(cudaMalloc(&d_c, sizeof(int)));

        // --- Kernel execution
        add<<<1,1>>>(a, b, d_c);
        gpuErrchk(cudaPeekAtLastError());
        gpuErrchk(cudaDeviceSynchronize());
        // --- Moving the results from device to host
        gpuErrchk(cudaMemcpy(&c, d_c, sizeof(int), cudaMemcpyDeviceToHost));

        // --- Results printout
        printf("%d + %d is %d\n", a, b, c);
        return 0; }
```

```
[ ] !wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-10-1_10.1.243-1_amd64.deb
    !dpkg -i cuda-10-1_10.1.243-1_amd64.deb
    !apt-get update
    !apt-get install cuda
```

```
[ ] !nvcc --version
```

```
[ ] import torch
    import pandas as pd
    from tqdm import tqdm, notebook
    import warnings
    warnings.filterwarnings('ignore')

    pd.pandas.set_option('display.max_columns',None)
    pd.pandas.set_option('display.max_rows',None)
    pd.pandas.set_option('display.max_colwidth',None)
    pd.pandas.set_option('display.width',None)
```

Masukkan Dataset

```
[ ] df = pd.read_csv('/content/reviews_bsi_mobile_textblob_clear.csv')
    df.set_index('id', inplace=True)
    df.head()
```

```
[ ] df.describe()
```

```
[ ] df.dtypes
```

Buat variable label agar indexnya bisa dipanggil

```
[ ] possible_labels = df.sentiment.unique()
```

```
[ ]
    # Create label dictionary
    label_dict = {}
    for index, possible_label in enumerate(possible_labels):
      if possible_label == 'Positive':
        label_dict[possible_label] = 2
      elif possible_label == 'Neutral':
        label_dict[possible_label] = 1
      else:
        label_dict[possible_label] = 0
```

```
[ ] df['label'] = df.sentiment.replace(label_dict)
```

```
[ ] # Print updated DataFrame
    df.head(300)
```

Training/Validation Split Data

```
[ ] from sklearn.model_selection import train_test_split

    x_train,x_val,y_train,y_val=train_test_split(df.index.values,
                                                 df.label.values,
                                                 test_size=0.20,
                                                 random_state=20,
                                                 stratify=df.label.values)
```

```
[ ] df['data_type'] = ['not_set']*df.shape[0]
```

```
[ ] df.loc[x_train, 'data_type']= 'train'
    df.loc[x_val,'data_type'] = 'val'
```

```
[ ] df.groupby (['sentiment','label','data_type']).count()
```

```
[ ] #melihat dataset yang udah terbagi-bagi
    df.head(300)
```

Loading Tokenizer and Encoding our Data

```
[ ] !git clone https://github.com/IndoNLP/indonlu.git
```

```
[ ] from torch import nn
    from transformers import BertForSequenceClassification, BertConfig, BertTokenizer
    from torch.utils.data import Dataset, TensorDataset, DataLoader
    from torch.utils.data.dataset import random_split

    from indonlu.utils.forward_fn import forward_sequence_classification
```

```
[ ] tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-large-p1',
                                              do_lower_case=True)
```

```
[ ] encoded_data_train = tokenizer.batch_encode_plus(
        df[df.data_type=='train'].content.values,
        add_special_tokens=True,
        return_attention_mask=True,
        padding='longest',
        max_length=256,
        truncation=True,
        return_tensors='pt'
    )
    encoded_data_val = tokenizer.batch_encode_plus(
        df[df.data_type=='val'].content.values,
        add_special_tokens=True,
        return_attention_mask=True,
        padding='longest',
        max_length=256,
        truncation=True,
        return_tensors='pt'
    )

    input_ids_train = encoded_data_train['input_ids']
    attention_mask_train = encoded_data_train['attention_mask']
    labels_train = torch.tensor(df[df.data_type=='train'].label.values)

    input_ids_val = encoded_data_val['input_ids']
    attention_mask_val = encoded_data_val['attention_mask']
    labels_val = torch.tensor(df[df.data_type=='val'].label.values)
```

```
[ ] import pandas as pd

    # Mengkonversi hasil tokenisasi pelatihan menjadi DataFrame
    df_encoded_train = pd.DataFrame({
        'Text': df[df.data_type == 'train'].content.values,
        'Input IDs': list(input_ids_train),
        'Attention Mask': list(attention_mask_train),
        'Labels': labels_train
    })

    # Mengkonversi hasil tokenisasi validasi menjadi DataFrame
    df_encoded_val = pd.DataFrame({
        'Text': df[df.data_type == 'val'].content.values,
        'Input IDs': list(input_ids_val),
        'Attention Mask': list(attention_mask_val),
        'Labels': labels_val
    })
```

```
[ ] #Menampilkan proses tokenisasi train
    df_encoded_train.head(300)
```

```
[ ] #Menampilkan proses tokenisasi val
    df_encoded_val.head(300)
```

```
[ ] #Tensor Dataset berguna untuk mempackage data
    dataset_train = TensorDataset(input_ids_train, attention_mask_train, labels_train)
    dataset_val = TensorDataset(input_ids_val, attention_mask_val, labels_val)
```

```
[ ] len(dataset_train)
```

```
[ ] len(dataset_val)
```

Setting up IndoBERT Pretrained Model

```
[ ] from transformers import BertForSequenceClassification
```

```
[ ] model = BertForSequenceClassification.from_pretrained("indobenchmark/indobert-large-p1", num_labels=len(label_dict),
                                                          output_attentions=False,
                                                          output_hidden_states=False)
```

```
[ ] params = list(model.named_parameters())

    print('The BERT model has {:} different named parameters.\n'.format(len(params)))

    print('==== Embedding Layer ====\n')

    for p in params[0:5]:
        print("{:<55} {:>12}".format(p[0], str(tuple(p[1].size()))))

    print('\n==== First Transformer ====\n')

    for p in params[5:21]:
        print("{:<55} {:>12}".format(p[0], str(tuple(p[1].size()))))

    print('\n==== Output Layer ====\n')

    for name, p in params[-4:]:
        print("{:<55} {:>12}".format(name, str(tuple(p.size()))))
```

Creating Data Loaders

```
[ ] from torch.utils.data import DataLoader, RandomSampler, SequentialSampler
```

```
[ ] batch_size = 32

    dataloader_train = DataLoader(dataset_train,
                                  sampler=RandomSampler(dataset_train),
                                  batch_size=batch_size)

    dataloader_validation = DataLoader(dataset_val,
                                       sampler=RandomSampler(dataset_val),
                                       batch_size=batch_size)
```

Setting Up Optimiser and Scheduler

```
[ ] from transformers import AdamW, get_linear_schedule_with_warmup
```

```
[ ] optimizer = AdamW(model.parameters(),
                      lr=2e-5,
                      eps=1e-8)
```

```
[ ] epochs = 10
    scheduler = get_linear_schedule_with_warmup(optimizer,
                                                num_warmup_steps=0,
                                                num_training_steps=len(dataloader_train)*epochs)
```

Defining our Perfomance Metrics

```
[ ] import numpy as np
    from sklearn.metrics import f1_score, accuracy_score, precision_score, recall_score
```

```
[ ] def f1_score_func(preds, labels):
        preds_flat = np.argmax(preds, axis=1).flatten()
        labels_flat = labels.flatten()
        return f1_score(labels_flat, preds_flat, average='weighted')

    def precision_score_func(preds, labels):
        preds_flat = np.argmax(preds, axis=1).flatten()
        labels_flat = labels.flatten()
        return precision_score(labels_flat, preds_flat, average='macro')

    def recall_score_func(preds, labels):
        preds_flat = np.argmax(preds, axis=1).flatten()
        labels_flat = labels.flatten()
        return recall_score(labels_flat, preds_flat, average='macro')

    def accuracy_score_func(preds, labels):
        preds_flat = np.argmax(preds, axis=1).flatten()
        labels_flat = labels.flatten()
        return accuracy_score(labels_flat, preds_flat)
```

```
[ ] def accuracy_per_class(preds, labels):
      label_dict_inverse = {v: k for k, v in label_dict.items()}

      preds_flat = np.argmax(preds, axis=1).flatten()
      labels_flat = labels.flatten()

      for label in np.unique(labels_flat):
        y_preds = preds_flat[labels_flat==label]
        y_true = labels_flat[labels_flat==label]
        print(f'Class: {label_dict_inverse[label]}')
        print(f'Accuracy: {len(y_preds[y_preds==label])}/{len(y_true)}\n')
```

Creating our Training Loop

```
[ ] device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model.to(device)

    print(device)
```

```
[ ] def evaluate(dataloader_val):
        model.eval()
        loss_val_total = 0
        predictions, true_vals = [], []

        for batch in dataloader_val:
            batch = tuple(b.to(device) for b in batch)

            inputs = {
                'input_ids': batch[0],
                'attention_mask': batch[1],
                'labels': batch[2]
            }

            with torch.no_grad():
                outputs = model(**inputs)
                loss = outputs[0]
                logits = outputs[1]

                loss_val_total += loss.item()
                logits = logits.detach().cpu().numpy()
                label_ids = inputs['labels'].cpu().numpy()

                predictions.append(logits)
                true_vals.append(label_ids)

        loss_val_avg = loss_val_total / len(dataloader_val)
        predictions = np.concatenate(predictions, axis=0)
        true_vals = np.concatenate(true_vals, axis=0)

        return loss_val_avg, predictions, true_vals
```

```
[ ] training_stats = []

    for epoch_i in tqdm(range(epochs), desc='Epochs', leave=True, disable=False):
        model.train()
        loss_train_total = 0
        progress_bar = tqdm(dataloader_train, desc=f'Epoch {epoch_i+1}/{epochs}', leave=True, disable=False)

        for batch in progress_bar:
            model.zero_grad()
            batch = tuple(b.to(device) for b in batch)
            inputs = {
                'input_ids': batch[0],
                'attention_mask': batch[1],
                'labels': batch[2]
            }

            outputs = model(**inputs)
            loss = outputs[0]
            loss_train_total += loss.item()
            loss.backward()

            torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)

            optimizer.step()
            scheduler.step()

            progress_bar.set_postfix({'training_loss': '{0:.2f}'.format(loss.item() / len(batch))})

        torch.save(model.state_dict(), f'finetuned_BERT_epoch_{epoch_i}.model')

        tqdm.write(f'\nEpoch {epoch_i+1}')

        loss_train_avg = loss_train_total / len(dataloader_train)
        tqdm.write(f'Training loss: {loss_train_avg}')

        val_loss, predictions, true_vals = evaluate(dataloader_validation)
        val_f1 = f1_score_func(predictions, true_vals)
        val_precision = precision_score_func(predictions, true_vals)
        val_recall = recall_score_func(predictions, true_vals)
        val_accuracy = accuracy_score_func(predictions, true_vals)

        tqdm.write(f'Validation loss: {val_loss}')
        tqdm.write(f'F1 Score (Weighted): {val_f1}')
        tqdm.write(f'Precision score: {val_precision}')
        tqdm.write(f'Recall score: {val_recall}')
        tqdm.write(f'Accuracy score: {val_accuracy}')

        training_stats.append({
            'epoch': epoch_i + 1,
            'Training Loss': loss_train_avg,
            'Valid. Loss': val_loss,
            'F1 Score': val_f1,
            'Precision score': val_precision,
            'Recall score': val_recall,
            'Accuracy score': val_accuracy
        })
```

Showing Graph

```
[ ] import matplotlib.pyplot as plt

    epochs = [stat['epoch'] for stat in training_stats]
    training_losses = [stat['Training Loss'] for stat in training_stats]
    valid_losses = [stat['Valid. Loss'] for stat in training_stats]
    f1_scores = [stat['F1 Score'] for stat in training_stats]
    precision_scores = [stat['Precision score'] for stat in training_stats]
    recall_scores = [stat['Recall score'] for stat in training_stats]
    accuracy_scores = [stat['Accuracy score'] for stat in training_stats]

    # Plot training loss and validation loss
    plt.plot(epochs, training_losses, label='Training Loss')
    plt.plot(epochs, valid_losses, label='Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.title('Training and Validation Loss')
    plt.legend()
    plt.show()
```

```
[ ] # Plot accuracy score
    plt.plot(epochs, accuracy_scores, label='Accuracy Score')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.title('Accuracy Score')
    plt.legend()
    plt.show()
```

Show Model Perform

```
[ ] from sklearn.metrics import classification_report

    def evaluate_model(predictions, true_vals):
        pred_labels = np.argmax(predictions, axis=1)
        true_labels = true_vals

        report = classification_report(true_labels, pred_labels)
        print(report)

        precision = precision_score(true_labels, pred_labels, average='weighted')
        recall = recall_score(true_labels, pred_labels, average='weighted')
        f1 = f1_score(true_labels, pred_labels, average='weighted')
        accuracy = accuracy_score(true_labels, pred_labels)

        print(f"Precision: {precision}")
        print(f"Recall: {recall}")
        print(f"F1-Score: {f1}")
        print(f"Accuracy: {accuracy}")

    # Evaluating the model
    val_loss, predictions, true_vals = evaluate(dataloader_validation)
    evaluate_model(predictions, true_vals)
```

Show Confusion Matrix

```
[ ] from sklearn.metrics import confusion_matrix
    import seaborn as sns

    def plot_confusion_matrix(predictions, true_vals, labels):
        pred_labels = np.argmax(predictions, axis=1)
        true_labels = true_vals

        cm = confusion_matrix(true_labels, pred_labels)
        cm_norm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

        plt.figure(figsize=(10, 8))
        sns.heatmap(cm_norm, annot=True, cmap='Blues', fmt='.2f', xticklabels=labels, yticklabels=labels)
        plt.xlabel('Predicted Sentimet')
        plt.ylabel('True Sentiment')
        plt.title('Confusion Matrix')
        plt.show()

    # Evaluating the model and plotting the confusion matrix
    val_loss, predictions, true_vals = evaluate(dataloader_validation)
    labels = ['Negative', 'Neutral', 'Positive']  # ganti dengan label yang sesuai dengan dataset Anda
    plot_confusion_matrix(predictions, true_vals, labels)
```