Kelompok 6

- Azarine Aisyah Ramadhani (2206051550)
- Haifa Marwa Saniyyah (2206048783)
- Nadira Eka Rahmaharva (2206051525)
- Yiesha Reyhani Ghozali (2206828115)

```
# Import Library

!pip install -qq google-play-scraper
!pip install Sastrawi --q
!pip install openpyxl --q
!pip install transformers --q
!pip install tensorflow --q

import json
import pandas as pd
import gdown

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google_play_scraper import Sort, reviews, app
from datetime import datetime
from datetime import datetime

import nltk
nltk.download('stopwords')
nltk.download('punkt_tab')
nltk.download('averaged_perceptron_tagger')

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from transformers import BertTokenizer, TFBertForSequenceClassification, BertForSequenceClass
import tensorflow as tf

# Viz
from wordcloud import WordCloud
from plotly import graph_objs as go
import plotly.express as px
import plotly.figure_factory as ff
from collections import Counter

# Inisialisasi
tokenizer = word_tokenize
lemmatizer = WordNetLemmatizer()
```

```
import string
import re
```

```
                                         50.2/50.2 kB 4.5 MB/s eta 0:0
                                       209.7/209.7 kB 5.3 MB/s eta 0:0
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

Scraping Data

```
# Ambil data
result, continuation_token = reviews(
    'id.co.btn.mobilebanking.android',
    lang='id',
    sort=Sort.NEWEST,
    count=5000
)

# Tentukan batas tanggal dengan jam 13:00
batas_tanggal = datetime(2025, 4, 28, 13, 0)

# Filter hasil berdasarkan tanggal dan jam
filtered_reviews = [review for review in result if review['at'] <= batas_tanggal]

# Ambil 2500 data teratas dari filtered_reviews
results = filtered_reviews[:2500]
```

```
# Hasil Scraping
df = pd.DataFrame(results)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   reviewId              2500 non-null   object
 1   userName              2500 non-null   object
 2   userImage             2500 non-null   object
 3   content               2500 non-null   object
 4   score                 2500 non-null   int64
 5   thumbsUpCount         2500 non-null   int64
 6   reviewCreatedVersion  1916 non-null   object
 7   at                    2500 non-null   datetime64[ns]
 8   replyContent          2500 non-null   object
 9   repliedAt             2500 non-null   datetime64[ns]
 10  appVersion            1916 non-null   object
dtypes: datetime64[ns](2), int64(2), object(7)
memory usage: 215.0+ KB
```

```
df.tail()
```

	reviewId	userName	userImage	content	score
2495	d736de9a-542a-4828-8915-2622a7aff53b	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	MANTAPPP	5
2496	89388815-a9c8-4bd7-be36-1d56efd58e63	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	Akhirnya Mobile banking BTN bisa pake fitur si...	5
2497	03d5b3f1-5302-4ed6-b166-94b18c9b2c1d	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	Apk kagak jelass, pendftaran buat rek baru aja...	1

```
df['score'].value_counts()
```

score	count
5	1294
1	817
2	148
3	141
4	100

dtype: int64

```python
pd.set_option("display.max_colwidth", None)


# Cuplikan 50 ulasan pertama
df['content'].head(50)
```

	content
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.
1	terimakasih atas pelayan yg sangat memuaskan
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas
5	udah registrasi ulang..trus di suruh login 180 menit lagi.di coba lgi login suruh registrasi lagi.dah coba live chat jawaban ngg nyambung.di email jg.jawaban sama jg di suruh registrasi lagi..khn udah beberapa x registrasi..payah
6	Mantap
7	sangat lengkap dan mudah
8	apk nya bagus, tapi baru kali ini saya mengalami kendala tidak bisa masuk/login karena ada warning ada kendala di perangkat ini terdeteksi tidak mengunduh dengan store resmi, padahal saya install nya pake PlayStore, ini maksudnya gimana sih bikin cemas aja lama lama pake BTN..
9	dari semua bank ini yang paling ribet daftarnya susah acc ada aja kendala dari mulai foto tidak sesuai beneficiary lah kendala pas scan ktp lah ribet banget tinggal acc doang susahnya
10	APLIKASI GOBLOKK TOLOLL
11	gak bisa di pakai harus hapus aplikasi yg tidak terdaftar di play store. payah.
12	#sudah ada kemajuan
13	apk apa ni berani nya nipu org aja biar bnyk yg download wkwk ngakak lu buat promosi tp yg nipu org smpt ada yg butuh uang x gimana smpt org butuh x trus klen tipu kek gitu gimana dimana otak kalian dasar penipu
14	Sejak kemarin saya update, tiba-tiba aplikasi bale BTN tidak bisa digunakan/dibuka. Catatan "perangkat ini terdeteksi memiliki aplikasi yang diunduh bukan dari store resmi" mohon bantuannya kenapa tiba-tiba tidak bisa digunakan di HP saya.
15	apa" pake pulsa sangatt jauh beda dg bank yang lain ,yang lebih modern😅serasa jaman 90 an pake pulsa
16	aplikasinya tiba² dikatakan diunduh tidak resmi ga bisa dibuka gila

17	fitur lengkap bisa topup banyak pilihan dan sangat mudah
18	sangat sempurna sekali
19	tf duitnya ga sampe tapi saldo berkurang, apaaan!. edit;duitku yg di telan btn udah balik setelah 2 hari, ratingnya kunaikin dikit☝️, mohon sistemnya dibuat lebih baik lagi
20	nga jelas bgt ni pls 15rb abis cuma mau buka m banking nga bisa2, nga jelas lama1
21	jadi bingung sama apknya sekarang saya ketika mau login tiba² ga bisa udah masukin ID pengguna sama pasword tetap aja salah. tolong untuk di tanggapi sama bagian adminnya biar jangan kayak gitu entar para nasabah gak nyaman untuk menggunakannya.
22	Setelah update versi 2.1.0 malahan sy gk bisa login.... suruh hapus launcher,emg sy pake launcher pihak kedua....apa hubunganya keteranganya perangkat ini terdeteksi memiliki aplikasi yg di unduh bukan dari store resmi kocak🤭🤭😆😆😁😁
	Ceritanya mau tarik tunai 500.000 di salah-satu ATM BTN di Cempaka Putih, udah coba tiga
	banking benar saldo saya berkurang, padahal uang tidak keluar sama sekali di 2 ATM yg berbeda.
24	ok
25	Mantap
26	aplikasi gak jelas, susah amat mau daftar m-banking nya, masak gak semua hp langsung bisa buat daftar, dari mana aturan nya harus riset hp dulu baru bisa daftar m-banking
27	mau aktifin kartu ATM & mbanking aja susah bgt... EROR DARI JAM 2 SIANG, SAMPAI SEKARANG, SAMPAI ABIS PULSA, CUMA BUAT KODE OTP 👎
28	bagus
29	keren
30	aplikasi sangat bagus gampang di gunakan
31	Aplikasi yang sangat bagus: memudahkan pengguna untuk transaksi, fiturnya mudah dipahami dan digunakan, cocok untuk mahasiswa/pelajar. Sangat membantu untuk mengatur keuangan dan transaksi tanpa kartu. Terima kasih.
32	Aplikasi penipuan. Saya saya nabung kenapa saldo tidak berubah. Ditunggu sudah berhari hari saldo ga nambah nambah. HATI HATI INI APLIKASI TIPU TIPU
33	kecewa isi saldo spay malah nyangkut lalu saldo terpotong dan sampe sekarang uangnya belum di refund. maaf bintang 1
34	Beli Data internet, transaksi gagal terus 👎

Pre Processing

Pre Processing (1)

```
# Info mengenai data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   reviewId              2500 non-null   object        
 1   userName              2500 non-null   object        
 2   userImage             2500 non-null   object        
 3   content               2500 non-null   object        
 4   score                 2500 non-null   int64         
 5   thumbsUpCount         2500 non-null   int64         
 6   reviewCreatedVersion  1916 non-null   object        
 7   at                    2500 non-null   datetime64[ns]
 8   replyContent          2500 non-null   object        
 9   repliedAt             2500 non-null   datetime64[ns]
 10  appVersion            1916 non-null   object        
dtypes: datetime64[ns](2), int64(2), object(7)
memory usage: 215.0+ KB
```

```
# Kamus Alay (Indonesian Colloquial)
url = "https://raw.githubusercontent.com/nasalsabila/kamus-alay/master/colloquial-indonesian
kamus_alay_df = pd.read_csv(url)
print(kamus_alay_df.head())

# Convert df into dictionary
kamus_alay = dict(zip(kamus_alay_df['slang'], kamus_alay_df['formal']))
print(list(kamus_alay.items())[:10])
```

```
      slang    formal  In-dictionary  \
0      woww       wow              1
1     aminn      amin              1
2       met   selamat              1
3    netaas   menetas              1
4   keberpa  keberapa              0


0
1  Selamat ulang tahun kakak tulus semoga panjang umur kakak,sehat selalu j
2
3
4

   category1 category2 category3
0   elongasi         0         0
1   elongasi         0         0
2  abreviasi         0         0
3   afiksasi  elongasi         0
4  abreviasi         0         0
[('woww', 'wow'), ('aminn', 'amin'), ('met', 'selamat'), ('netaas', 'meneta
```

```
# Fungsi untuk Pre Processing
def lowercase(review_text):
    return review_text.lower()

def clean_text(review_text):
    # Case folding
    review_text = lowercase(review_text)

    # Remove emoji
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        "]+", flags=re.UNICODE)
    cleaned_text = emoji_pattern.sub(r'', review_text)

    # Remove hashtags
    cleaned_text = re.sub(r'#\w+', '', cleaned_text)

    # Remove numbers
    cleaned_text = re.sub(r'\d+', ' ', cleaned_text)

    # Remove punctuation
    cleaned_text = cleaned_text.translate(
        str.maketrans(string.punctuation, ' ' * len(string.punctuation))
    )

    # Remove superscript
    superscript_pattern = re.compile("["
        u"\U00002070"
        u"\U000000B9"
        u"\U000000B2-\U000000B3"
        u"\U00002074-\U00002079"
        u"\U0000207A-\U0000207E"
        u"\U0000200D"
        "]+", flags=re.UNICODE)
    cleaned_text = superscript_pattern.sub(r'', cleaned_text)

    # Normalized slang words
    words = cleaned_text.split()
    normalized_words = []
    for word in words:
        lower_word = word.lower()
        if lower_word in kamus_alay:
            normalized_words.append(kamus_alay[lower_word])
        else:
            normalized_words.append(word)
    cleaned_text = ' '.join(normalized_words)

    # Remove character repetition
    cleaned_text = re.sub(r'(.)\1+', r'\1', cleaned_text)

    # Remove word repetition
    cleaned_text = re.sub(r'\b(\w+)(?:\W\1\b)+', r'\1', cleaned_text, flags=re.IGNORECASE)

    # Remove extra whitespaces
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()

    # Stopwords Removal (without tokenization)
    stop_words = stopwords.words('indonesian') + stopwords.words('english') + ["yg",
                "gak", "ngisi", "udah", "d", "sih", "nya", "srg", "utk", "byk", "gk", "ga",
                "gua", "gweh", "lu", "lw"]

    # Split, filter stopwords, and rejoin as a string
    words = cleaned_text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    cleaned_text = ' '.join(filtered_words)

    return cleaned_text
```

```
# Apply Preprocessing (1)
df['preprocessing'] = df['content'].apply(clean_text)

# Hasil Preprocessing
df[['content', 'preprocessing']]
```

	content	preprocessing
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.	notifikasi aplikasi ilegal coba bale btn dihapus instal ulang username lupa klik lupa id siapkan kartu atm isi data pulsa minimal pastikan mpin coba login jam pastikan jaringan lancar pengalaman indak berhasil langsung banknya
1	terimakasih atas pelayan yg sangat memuaskan	terimakasih pelayan memuaskan
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	antre bikin rekening daftar isi formulir daftar eror aplikasinya buang
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	nih registrasi susah kartu dibilang beda bikin ribet
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas	bayar service ac apk bybtn dah janji tukang ac konfirmasi duit dibatalin penipuan pesan
...	...	...
2495	BTN mobil bebas biaya transfer. Login pakai sidik jari/face id yang meningkatkan keamanan, rekomendasi bgt deh!	btn mobil bebas biaya transfer login pakai sidik jari face id meningkatkan keamanan rekomendasi banget deh
	aplikasi BTN mobile banking yang mudah dan aman	aplikasi btn mobile banking

```
df2 = df[['content', 'preprocessing','score']]
```

```
# Drop rows dengan missing value pada kolom 'content'
df2.dropna(subset=['preprocessing'], inplace=True)

# Drop baris yang duplikat
df2.drop_duplicates(subset=['preprocessing'], keep='first', inplace=True)
```

```
df2
```

	content	preprocessing	score
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.	notifikasi aplikasi ilegal coba bale btn dihapus instal ulang username lupa klik lupa id siapkan kartu atm isi data pulsa minimal pastikan mpin coba login jam pastikan jaringan lancar pengalaman indak berhasil langsung banknya	1
1	terimakasih atas pelayan yg sangat memuaskan	terimakasih pelayan memuaskan	5
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	antre bikin rekening daftar isi formulir daftar eror aplikasinya buang	1
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	nih registrasi susah kartu dibilang beda bikin ribet	1
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas	bayar service ac apk bybtn dah janji tukang ac konfirmasi duit dibatalin penipuan pesan	1
...	...	...	...
2495	BTN mobil bebas biaya transfer. Login pakai sidik jari/face id yang meningkatkan keamanan, rekomendasi bgt deh!	btn mobil bebas biaya transfer login pakai sidik jari face id meningkatkan keamanan	5

Preprocessing (2)

```
df_eda = df2.copy()
def preprocess_eda(text):
    # Convert text to lowercase for case-insensitive matching
    text = text.lower()

    # Remove specific words - expanded list with word boundaries
    text = re.sub(r'\b(aplikasi|btn|mobile|aplikasinya|bale|apk|banget|kali|ya|bank)\b', '',

    # Remove extra whitespaces
    text = re.sub(r'\s+', ' ', text).strip()

    # Stopwords Removal (without tokenization)
    stop_words = stopwords.words('indonesian') + stopwords.words('english') + ["yg",
                "gak", "ngisi", "udah", "d", "sih", "nya", "srg", "utk", "byk", "gk", "ga",
                "gua", "gweh", "lu", "lw"]

    # Split, filter stopwords, and rejoin as a string
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    text = ' '.join(filtered_words)

    return text

df_eda['eda'] = df_eda['content'].apply(preprocess_eda)
df_eda.head()
```

Preprocessing (3)

```
def clean_text(review_text):
    # Remove emoji
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        "]+", flags=re.UNICODE)
    cleaned_text = emoji_pattern.sub(r'', review_text)

    # Remove hashtags
    cleaned_text = re.sub(r'#\w+', '', cleaned_text)

    # Remove numbers
    cleaned_text = re.sub(r'\d+', ' ', cleaned_text)

    # Remove punctuation
    cleaned_text = cleaned_text.translate(
        str.maketrans(string.punctuation, ' ' * len(string.punctuation))
    )

    # Remove superscript
    superscript_pattern = re.compile("["
        u"\U00002070"
        u"\U000000B9"
        u"\U000000B2-\U000000B3"
        u"\U00002074-\U00002079"
        u"\U0000207A-\U0000207E"
        u"\U0000200D"
        "]+", flags=re.UNICODE)
    cleaned_text = superscript_pattern.sub(r'', cleaned_text)

    # Normalized slang words
    words = cleaned_text.split()
    normalized_words = []
    for word in words:
        lower_word = word.lower()  # Just for checking against dictionary
        if lower_word in kamus_alay:
            normalized_words.append(kamus_alay[lower_word])
        else:
            normalized_words.append(word)  # Keep original case
    cleaned_text = ' '.join(normalized_words)

    # Remove character repetition
    cleaned_text = re.sub(r'(.)\1+', r'\1', cleaned_text)

    # Remove word repetition
    cleaned_text = re.sub(r'\b(\w+)(?:\W\1\b)+', r'\1', cleaned_text, flags=re.IGNORECASE)

    # Remove extra whitespaces
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()

    return cleaned_text
```

```
# Apply Preprocessing
df3 = df.copy()
df3['preprocessing'] = df3['content'].apply(clean_text)
df3[['content', 'preprocessing']]
```

	content	preprocessing
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.	Bagi yang tiba ada notifikasi aplikasi ilegal coba bale by BTN dihapus kemudian instal ulang Bagi yang username lupa bisa klik lupa ID Siapkan kartu ATM untuk isi data dan pulsa minimal ke Pastikan ingat MPIN Coba login lagi setelah jam Pastikan jaringan lancar Ini dari pengalaman saja kalau indak berhasil bisa langsung datang ke banknya
1	terimakasih atas pelayan yg sangat memuaskan	terimakasih atas pelayan yang sangat memuaskan
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	sudah antre bikin rekening sudah daftar isi formulir macam engak bisa daftar karena eror aplikasinya buang waktu saja
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	bagaimana Sih Nih Mau Registrasi saja Susah benar No Kartu saja benar kok Masa Dibilang Beda Bikin Ribet saja
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas	sudah bayar mau service ac lewat apk byBTN dah buat janji tapi tukang ac nya engak ada konfirmasi sama sekali Duit engak balik engak bisa dibatalin juga Penipuan jangan pesan lewat sini engak jelas
...	...	...
	BTN mobil bebas biaya transfer. Login pakai	BTN mobil bebas biaya transfer Login

```
df3 = df3[['content', 'preprocessing','score']]
```

```
# Menghapus baris dengan nilai yang hilang di kolom 'preprocessing'
df3.dropna(subset=['preprocessing'], inplace=True)

# Menghapus baris yang hanya berisi spasi kosong
df3 = df3[df3['preprocessing'].str.strip() != '']

# Menghapus baris duplikat, menyimpan kemunculan pertama
df3.drop_duplicates(subset=['preprocessing'], keep='first', inplace=True)
```

```
df3
```

	content	preprocessing	score
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.	Bagi yang tiba tiba ada notifikasi aplikasi ilegal coba bale by BTN dihapus kemudian instal ulang Bagi yang username lupa bisa klik lupa ID Siapkan kartu ATM untuk isi data dan pulsa minimal ke Pastikan ingat MPIN Coba login lagi setelah jam Pastikan jaringan lancar Ini dari pengalaman saja kalau indak berhasil bisa langsung datang ke banknya	1
1	terimakasih atas pelayan yg sangat memuaskan	terimakasih atas pelayan yang sangat memuaskan	5
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	sudah antre bikin rekening sudah daftar isi formulir macam engak bisa daftar karena eror aplikasinya buang waktu saja	1
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	bagaimana Sih Nih Mau Registrasi saja Susah benar No Kartu saja benar kok Masa Dibilang Beda Bikin Ribet saja	1
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas	sudah bayar mau service ac lewat apk byBTN dah buat janji tapi tukang ac nya engak ada konfirmasi sama sekali Duit engak balik engak bisa dibatalin juga Penipuan jangan pesan lewat sini engak jelas	1
...	...	...	...
	BTN mobil bebas biaya transfer. Login	BTN mobil bebas biaya transfer Login	

Labelling

```
# Labelling berdasarkan skor
df3['sentiment'] = df3['score'].apply(lambda x: 'positif' if x >= 4 else 'negatif')
df3
```

	content	preprocessing	score	sentiment
0	Bagi yang tiba tiba ada notifikasi aplikasi illegal coba bale by BTN dihapus kemudian install ulang. Bagi yang username lupa, bisa klik lupa ID. Siapkan kartu ATM untuk isi data dan pulsa minimal 1k. Pastikan ingat MPIN. Coba login lagi setelah 4 jam. Pastikan jaringan lancar Ini dari pengalaman aja, kalau nda berhasil bisa langsung datang ke banknya.	Bagi yang tiba ada notifikasi aplikasi ilegal coba bale by BTN dihapus kemudian instal ulang Bagi yang username lupa bisa klik lupa ID Siapkan kartu ATM untuk isi data dan pulsa minimal ke Pastikan ingat MPIN Coba login lagi setelah jam Pastikan jaringan lancar Ini dari pengalaman saja kalau indak berhasil bisa langsung datang ke banknya	1	negatif
1	terimakasih atas pelayan yg sangat memuaskan	terimakasih atas pelayan yang sangat memuaskan	5	positif
2	udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	sudah antre bikin rekening sudah daftar isi formulir macam engak bisa daftar karena eror aplikasinya buang waktu saja	1	negatif
3	gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	bagaimana Sih Nih Mau Registrasi saja Susah benar No Kartu saja benar kok Masa Dibilang Beda Bikin Ribet saja	1	negatif
4	udah bayar mau service ac lewat apk byBTN dh buat janji tp tukang ac nya gda konfirmasi sma sekali Duit ga balik ga bisa dibatalin juga, Penipuan jangan pesen" lewat sini ga jelas	sudah bayar mau service ac lewat apk byBTN dah buat janji tapi tukang ac nya engak ada konfirmasi sama sekali Duit engak balik engak bisa dibatalin juga Penipuan jangan pesan lewat sini engak jelas	1	negatif

```
# Export ke Excel
df3.to_excel('reviews_sentiment.xlsx', index=False)
```

Validasi manual

```
# Load Data (Preprocessing 1)
!gdown 196t8WbXbydGnOp38mdd4kY0IxroREHzU
data1 = pd.read_csv('/content/status ulasan - Sheet1.csv')
```

```
Downloading...
From: https://drive.google.com/uc?id=196t8WbXbydGnOp38mdd4kY0IxroREHzU
To: /content/status ulasan - Sheet1.csv
100% 190k/190k [00:00<00:00, 4.88MB/s]
```

```
data1 = data1[['content', 'sentiment']]
data1.head()
```

	content	sentiment
0	notifikasi aplikasi ilegal coba bale btn dihap...	negatif
1	antre bikin rekening daftar isi formulir dafta...	negatif
2	nih registrasi susah kartu dibilang beda bikin...	negatif
3	bayar service ac apk bybtn dah janji tukang ac...	negatif
4	registrasi ulang suruh login menit coba login ...	negatif

```
# Encode labels
data1['label'] = data1['sentiment'].replace({'negatif': 0, 'positif': 1})
data1
```

```
<ipython-input-26-5462cf646ac7>:2: FutureWarning: Downcasting behavior in `
  data['label'] = data['sentiment'].replace({'negatif': 0, 'positif': 1})
```

	content	sentiment	label
0	notifikasi aplikasi ilegal coba bale btn dihap...	negatif	0
1	antre bikin rekening daftar isi formulir dafta...	negatif	0
2	nih registrasi susah kartu dibilang beda bikin...	negatif	0
3	bayar service ac apk bybtn dah janji tukang ac...	negatif	0
4	registrasi ulang suruh login menit coba login ...	negatif	0
...	...	...	...
2275	aplikasinya membantu banget bayar tagihan tran...	positif	1
2276	kalo buka rekening btn apknya praktis banget l...	positif	1
2277	mbanking btn nih enak pakeknya enaknya buka re...	positif	1
2278	aplikasi btn mobile mudah biaya transfer bebas...	positif	1
2279	login aplikasi pakai sidik jari face id aplika...	positif	1

2280 rows × 3 columns

```
# Load Data (Preprocessing 3)
!gdown 16ANsWg8DwxnftxrEMc3LkpNwSUuj2vCH
data = pd.read_excel('/content/status ulasan.xlsx')
```

```
Downloading...
From: https://drive.google.com/uc?id=16ANsWg8DwxnftxrEMc3LkpNwSUuj2vCH
To: /content/status ulasan.xlsx
100% 227k/227k [00:00<00:00, 82.6MB/s]
```

```python
data = data[['preprocessing', 'sentiment']]
data.head()
```

	preprocessing	sentiment
0	Bagi yang tiba ada notifikasi aplikasi ilegal ...	negatif
1	sudah antre bikin rekening sudah daftar isi fo...	negatif
2	bagaimana Sih Nih Mau Registrasi saja Susah be...	negatif
3	sudah bayar mau service ac lewat apk byBTN dah...	negatif
4	sudah registrasi ulang terus di suruh login me...	negatif

```python
# Encode label
data.rename(columns={'preprocessing': 'content'}, inplace=True)
data['label'] = data['sentiment'].replace({'negatif': 0, 'positif': 1})
data
```

```
<ipython-input-4-45b2cd37cf66>:3: FutureWarning: Downcasting behavior in `r
  data['label'] = data['sentiment'].replace({'negatif': 0, 'positif': 1})
```

	content	sentiment	label
0	Bagi yang tiba ada notifikasi aplikasi ilegal ...	negatif	0
1	sudah antre bikin rekening sudah daftar isi fo...	negatif	0
2	bagaimana Sih Nih Mau Registrasi saja Susah be...	negatif	0
3	sudah bayar mau service ac lewat apk byBTN dah...	negatif	0
4	sudah registrasi ulang terus di suruh login me...	negatif	0
...	...	...	...
2342	Aplikasi BTN mobile banking sangat membantu pe...	positif	1
2343	BTN mobil bebas biaya transfer Login pakai sid...	positif	1
2344	aplikasi BTN mobile banking yang mudah dan ama...	positif	1
2345	Aku suka banget sama BTN mobile karena bebas b...	positif	1
2346	Bisa buka rekening dari hp jadi lebih mudah to...	positif	1

2347 rows × 3 columns

EDA

[ ] ↳ 10 cells hidden

Modelling

Model 1

```
# Split data
X = data1['content']
y = data1['label']

# Split dataset
X_train, X_val, y_train, y_val = train_test_split(
    X, y, test_size=0.3, random_state=25, stratify=y
)

X_val, X_test, y_val, y_test = train_test_split(
    X_val, y_val, test_size=0.5, random_state=25, stratify=y_val
)

print(f"Training set size: {X_train.shape[0]}")
print(f"Validation set size: {X_val.shape[0]}")
print(f"Testing set size: {X_test.shape[0]}")
```

```
Training set size: 1596
Validation set size: 342
Testing set size: 342
```

```
from transformers import AutoTokenizer

# Inisialisasi tokenizer
tokenizer = AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p1")

# Tokenisasi data
def tokenize_function(texts):
    return tokenizer(
        texts.tolist(),
        padding=True,
        truncation=True,
        max_length=100
    )

train_encodings = tokenize_function(X_train)
val_encodings = tokenize_function(X_val)
test_encodings = tokenize_function(X_test)
```

```
from torch.utils.data import Dataset # Import Dataset from torch.utils.data

class ReviewsDataset(Dataset):
    def __init__(self, encodings, label):
        self.encodings = encodings
        self.label = label

    def __len__(self):
        return len(self.label)

    def __getitem__(self, idx):
        # Check if idx is a list (for batching) or an integer (single item)
        if isinstance(idx, list):
            # If idx is a list, create a batch of items
            item = {key: torch.tensor([val[i] for i in idx]) for key, val in self.encodings.
            item['label'] = torch.tensor([self.label[i] for i in idx])
        else:
            # If idx is an integer, get a single item
            item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
            item['label'] = torch.tensor(self.label[idx])
        return item

train_dataset = ReviewsDataset(train_encodings, y_train.tolist())
val_dataset = ReviewsDataset(val_encodings, y_val.tolist())
test_dataset = ReviewsDataset(test_encodings, y_test.tolist())


from transformers import AutoModelForSequenceClassification, TrainingArguments, Trainer, get
from transformers import AutoTokenizer
from transformers import BertConfig
from torch.optim import AdamW # Import AdamW from torch.optim
import torch
import numpy as np
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

config = BertConfig.from_pretrained("indobenchmark/indobert-base-p1", num_labels=2, seed = 2

# Inisialisasi model
model = AutoModelForSequenceClassification.from_pretrained("indobenchmark/indobert-base-p1",

# Buat optimizer AdamW
optimizer = AdamW(model.parameters(), lr=3e-5, weight_decay=0.01)

# Scheduler learning rate
lr_scheduler = get_scheduler(
    name="linear",
    optimizer=optimizer,
    num_warmup_steps=500,
    num_training_steps=3 * len(train_dataset) // 32
)

# Define compute_metrics function
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binar
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

# Training arguments
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=32,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    load_best_model_at_end=True,
    metric_for_best_model='accuracy',
    learning_rate=3e-5,
    eval_strategy='epoch',
    save_strategy='epoch'
)

# Inisialisasi trainer dengan optimizer dan scheduler custom
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics, # Pass the defined function
    optimizers=(optimizer, lr_scheduler)
)
```

```
Some weights of BertForSequenceClassification were not initialized from the
You should probably TRAIN this model on a down-stream task to be able to us
```

```
# Train model
trainer.train()
```

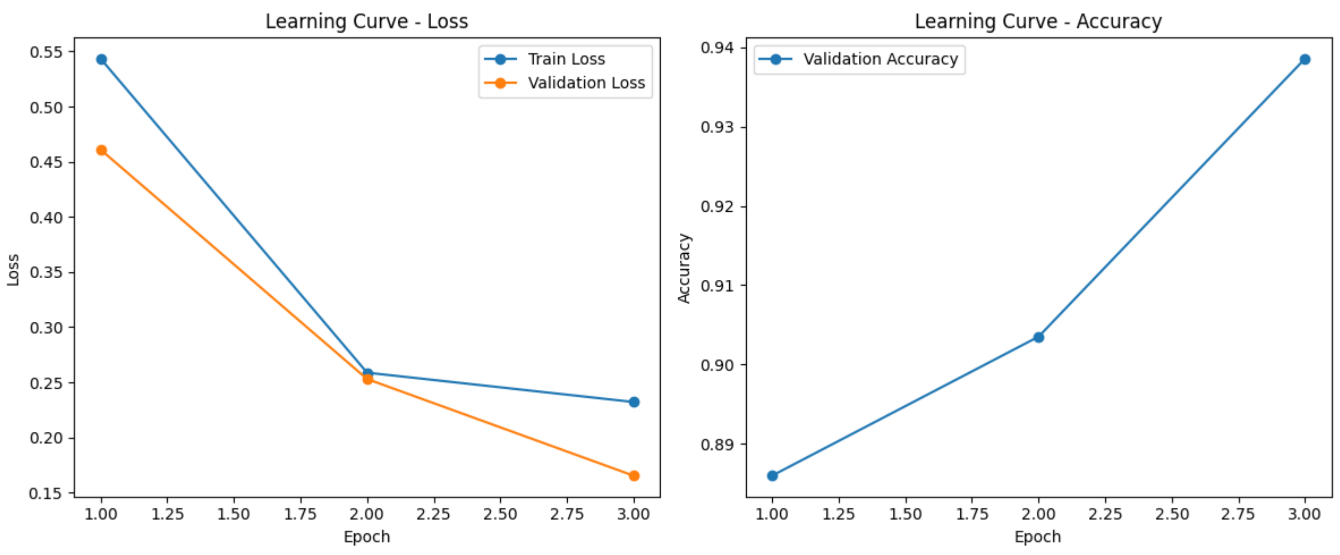
[150/150 01:50, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.543500	0.461016	0.885965	0.877743	0.958904	0.809249
2	0.258800	0.253052	0.903509	0.911528	0.850000	0.982659
3	0.232200	0.165409	0.938596	0.939481	0.936782	0.942197

```
TrainOutput(global_step=150, training_loss=0.3816517361005147, metrics=
```

```
import matplotlib.pyplot as plt
```

```python
log_history = trainer.state.log_history

# List untuk menyimpan data per epoch
train_epochs = []
train_loss = []

eval_epochs = []
eval_loss = []
eval_accuracy = []

# Ekstrak data dari log_history
for log in log_history:
    # Training loss per epoch
    if 'loss' in log and 'epoch' in log:
        train_epochs.append(log['epoch'])
        train_loss.append(log['loss'])
    # Evaluation metrics per epoch
    if 'eval_loss' in log and 'epoch' in log:
        eval_epochs.append(log['epoch'])
        eval_loss.append(log['eval_loss'])
    if 'eval_accuracy' in log and 'epoch' in log:
        eval_accuracy.append(log['eval_accuracy'])

# Plotting
plt.figure(figsize=(12, 5))

# Plot Loss (Training & Validation)
plt.subplot(1, 2, 1)
plt.plot(train_epochs, train_loss, label='Training Loss', marker='o')
plt.plot(eval_epochs, eval_loss, label='Validation Loss', marker='o')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training & Validation Loss per Epoch')
plt.legend()

# Plot Validation Accuracy
plt.subplot(1, 2, 2)
plt.plot(eval_epochs, eval_accuracy, label='Validation Accuracy', color='green', marker='o')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Validation Accuracy per Epoch')
plt.legend()

plt.tight_layout()
plt.show()
```



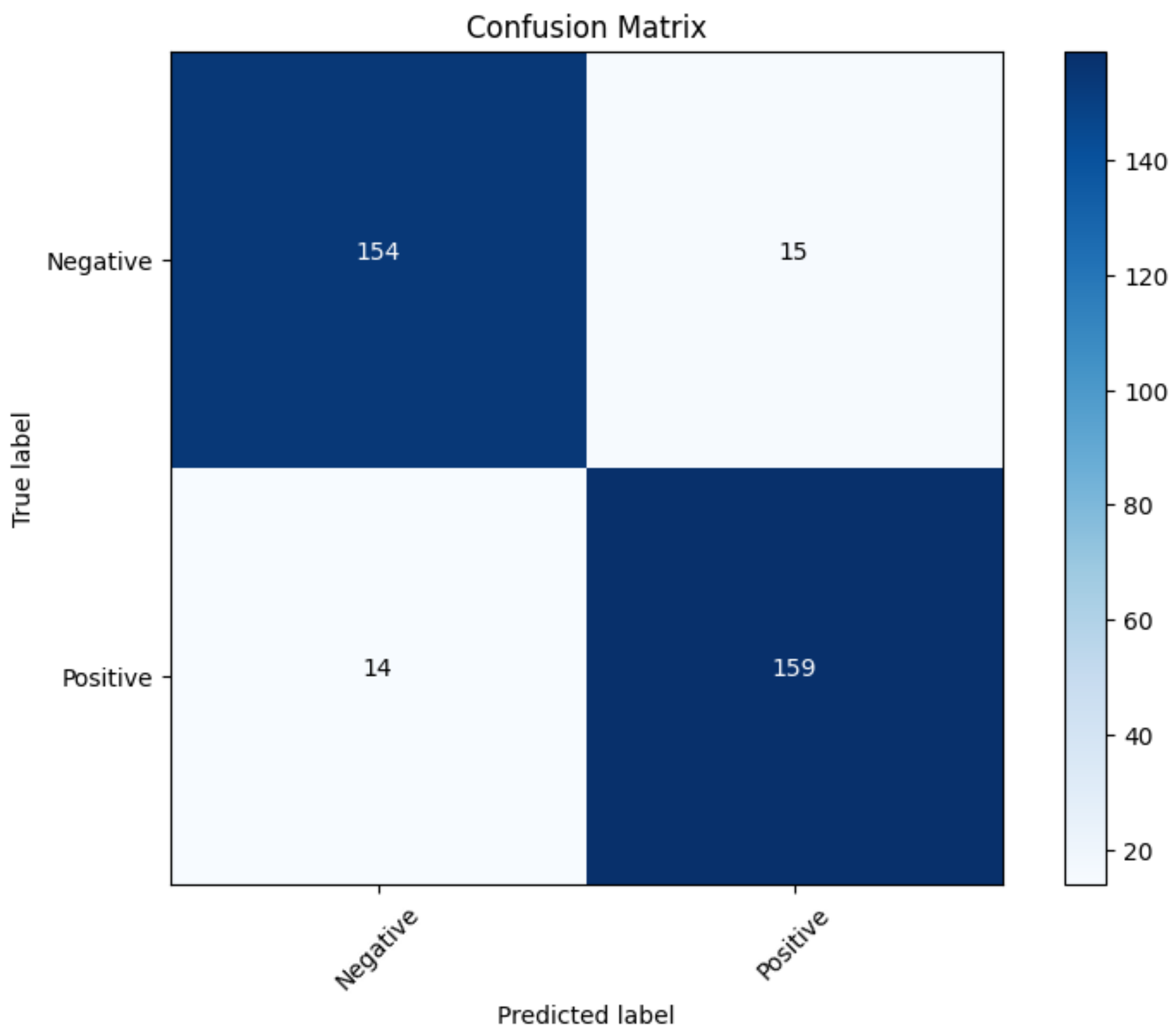
```
# Prediksi pada data testing
predictions = trainer.predict(test_dataset)
y_pred = np.argmax(predictions.predictions, axis=1)

# Nilai sebenarnya (ground truth)
y_true = predictions.label_ids

# Hitung test accuracy
test_accuracy = accuracy_score(y_true, y_pred)

# Tampilkan test accuracy
print(f"Test Accuracy: {test_accuracy}")
```

```
Test Accuracy: 0.9152046783625731
```

```
cr = classification_report(y_true, y_pred)
cm = confusion_matrix(y_true, y_pred)

print("=== Classification Report ===")
print(cr)
print("=== Confusion Matrix (===")
print(cm)
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.92      0.91      0.91       169
           1       0.91      0.92      0.92       173

    accuracy                           0.92       342
   macro avg       0.92      0.92      0.92       342
weighted avg       0.92      0.92      0.92       342

=== Confusion Matrix (===
[[154  15]
 [ 14 159]]
```

```
# Plot the confusion matrix
plt.figure(figsize=(8, 6))
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()

classes = ['Negative', 'Positive']
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)

thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```


Confusion Matrix
True label
Negative
Positive
154
15
14
159
Predicted label
140
120
100
80
60
40
20

Model 2

```
data.head()
```

	content	sentiment	label
0	Bagi yang tiba ada notifikasi aplikasi ilegal coba bale by BTN dihapus kemudian instal ulang Bagi yang username lupa bisa klik lupa ID Siapkan kartu ATM untuk isi data dan pulsa minimal ke Pastikan ingat MPIN Coba login lagi setelah jam Pastikan jaringan lancar Ini dari pengalaman saja kalau indak berhasil bisa langsung datang ke banknya	negatif	0
1	sudah antre bikin rekening sudah daftar isi formulir macam engak bisa daftar karena eror aplikasinya buang waktu saja	negatif	0
2	bagaimana Sih Nih Mau Registrasi saja Susah benar No Kartu saja	negatif	0

```
import os
import random
import numpy as np
import tensorflow as tf

def set_seed(seed=25):
    # Pengaturan untuk Python core
    os.environ['PYTHONHASHSEED'] = str(seed)
    random.seed(seed)

    # Pengaturan untuk NumPy
    np.random.seed(seed)

    # Pengaturan untuk TensorFlow
    tf.random.set_seed(seed)

    # Pengaturan tambahan untuk TensorFlow
    os.environ['TF_DETERMINISTIC_OPS'] = '1'
    os.environ['TF_CUDNN_DETERMINISTIC'] = '1'


    # Jika menggunakan GPU
    try:
        tf.config.experimental.set_memory_growth(
            tf.config.list_physical_devices('GPU')[0], True)
    except:
        pass

# Aplikasikan seed
set_seed(25)

# Jika menggunakan Keras, tambahkan ini:
from tensorflow import keras
keras.utils.set_random_seed(25)
```

```
# Split data
X = data['content']
y = data['label']

# Split dataset
X_train, X_val, y_train, y_val = train_test_split(
    X, y, test_size=0.3, random_state=25, stratify=y
)

X_val, X_test, y_val, y_test = train_test_split(
    X_val, y_val, test_size=0.5, random_state=25, stratify=y_val
)

print(f"Training set size: {X_train.shape[0]}")
print(f"Validation set size: {X_val.shape[0]}")
print(f"Testing set size: {X_test.shape[0]}")
```

```
Training set size: 1642
Validation set size: 352
Testing set size: 353
```

```
from transformers import AutoTokenizer

# Inisialisasi tokenizer
tokenizer = AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p1")

# Tokenisasi data
def tokenize_function(texts):
    return tokenizer(
        texts.tolist(),
        padding=True,
        truncation=True,
        max_length=100
    )

train_encodings = tokenize_function(X_train)
val_encodings = tokenize_function(X_val)
test_encodings = tokenize_function(X_test)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
  warnings.warn(
tokenizer_config.json: 100%                    2.00/2.00 [00:00<00:00, 27.8B/s]
config.json: 100%                              1.53k/1.53k [00:00<00:00, 19.4kB/s]
vocab.txt: 100%                                229k/229k [00:00<00:00, 566kB/s]
special_tokens_map.json: 100%                  112/112 [00:00<00:00, 1.70kB/s]
```

```
from torch.utils.data import Dataset # Import Dataset from torch.utils.data
import torch

class ReviewsDataset(Dataset):
    def __init__(self, encodings, label):
        self.encodings = encodings
        self.label = label

    def __len__(self):
        return len(self.label)

    def __getitem__(self, idx):
        # Check if idx is a list (for batching) or an integer (single item)
        if isinstance(idx, list):
            # If idx is a list, create a batch of items
            item = {key: torch.tensor([val[i] for i in idx]) for key, val in self.encodings.
            item['label'] = torch.tensor([self.label[i] for i in idx])
        else:
            # If idx is an integer, get a single item
            item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
            item['label'] = torch.tensor(self.label[idx])
        return item

train_dataset = ReviewsDataset(train_encodings, y_train.tolist())
val_dataset = ReviewsDataset(val_encodings, y_val.tolist())
test_dataset = ReviewsDataset(test_encodings, y_test.tolist())


from transformers import AutoModelForSequenceClassification, TrainingArguments, Trainer, get
from torch.optim import AdamW
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

# Konfigurasi model
config = BertConfig.from_pretrained(
    "indobenchmark/indobert-base-p1",
    num_labels=2,
    seed=25
)

model = AutoModelForSequenceClassification.from_pretrained(
    "indobenchmark/indobert-base-p1",
    config=config
)

# Optimizer
optimizer = AdamW(model.parameters(), lr=2e-5, weight_decay=0.01)

# Parameter training
train_batch_size = 32
num_train_epochs = 5
train_dataset_size = len(train_dataset)
total_training_steps = (train_dataset_size // train_batch_size) * num_train_epochs

# Scheduler LR
lr_scheduler = get_scheduler(
    name="linear",
    optimizer=optimizer,
    num_warmup_steps=500,
    num_training_steps=total_training_steps
)

# Fungsi evaluasi
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binar
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

# TrainingArguments dengan evaluasi per step dan save per step
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=train_batch_size,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    learning_rate=2e-5,

    # Penting: evaluasi dan save per step, agar load_best_model_at_end bisa jalan
    eval_strategy="steps",
    save_strategy="steps",
    eval_steps=10,
    save_steps=10,

    load_best_model_at_end=True,
    metric_for_best_model='accuracy',
    greater_is_better=True,

    # Optional: untuk menghindari terlalu banyak checkpoint
    save_total_limit=3
)

# Inisialisasi Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics,
    optimizers=(optimizer, lr_scheduler)
)
```

```
pytorch_model.bin: 100%                                   498M/498M [00:04<00:00, 223MB/s]
Some weights of BertForSequenceClassification were not initialized from the
You should probably TRAIN this model on a down-stream task to be able to us
model.safetensors: 100%                                   498M/498M [00:04<00:00, 93.0MB/s]
```

```
model
```

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(50000, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768,
bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768,
bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768,
bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=2, bias=True)
)
```

```
# Train model
trainer.train()
```

```
wandb: WARNING The `run_name` is currently set to the same value as `Trai
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: 
wandb: You can find your API key in your browser here: https://wandb.ai/au
wandb: Paste an API key from your profile and hit enter: ··········
wandb: WARNING If you're specifying your api key in code, ensure this code
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: haifamarwas (haifamarwas-universitas-indone
```

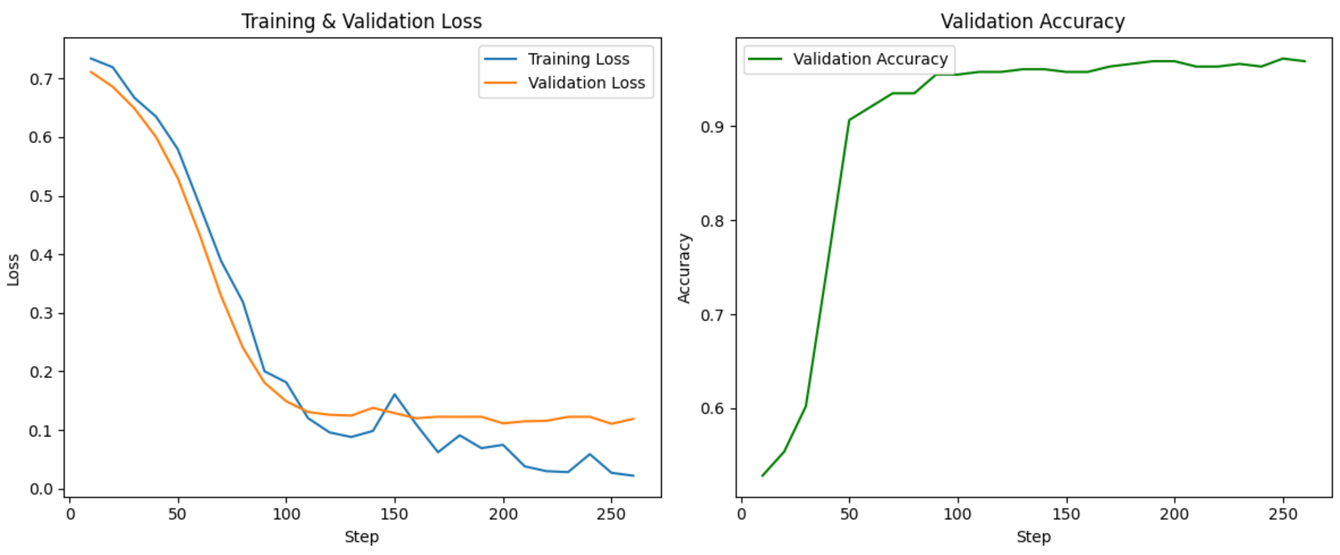
Tracking run with wandb version 0.19.10
Run data is saved locally in `/content/wandb/run-20250508_121107-iix7z0ii`
Syncing run **./results** to Weights & Biases (docs)
View project at https://wandb.ai/haifamarwas-universitas-indonesia/huggingface
View run at https://wandb.ai/haifamarwas-universitas-indonesia/huggingface/runs/iix7z0ii

[260/260 19:21, Epoch 5/5]

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
10	0.733900	0.710896	0.528409	0.178218	0.642857	0.103448
20	0.719100	0.685770	0.553977	0.255924	0.729730	0.155172
30	0.667000	0.648731	0.602273	0.385965	0.814815	0.252874
40	0.634700	0.599731	0.752841	0.690391	0.906542	0.557471
50	0.579300	0.529765	0.906250	0.903790	0.917160	0.890805
60	0.484400	0.433929	0.920455	0.919075	0.924419	0.913793
70	0.388300	0.328447	0.934659	0.933333	0.941520	0.925287
80	0.318800	0.240510	0.934659	0.932153	0.957576	0.908046
90	0.200200	0.180623	0.954545	0.953488	0.964706	0.942529
100	0.181400	0.149156	0.954545	0.953216	0.970238	0.936782
110	0.120500	0.130779	0.957386	0.956522	0.964912	0.948276
120	0.095900	0.126018	0.957386	0.956772	0.959538	0.954023
130	0.088100	0.124568	0.960227	0.959770	0.959770	0.959770
140	0.098500	0.137887	0.960227	0.959064	0.976190	0.942529
150	0.161000	0.128839	0.957386	0.956268	0.970414	0.942529
160	0.109400	0.120075	0.957386	0.956772	0.959538	0.954023
170	0.062100	0.122735	0.963068	0.962751	0.960000	0.965517
180	0.091000	0.122420	0.965909	0.965517	0.965517	0.965517
190	0.069100	0.122685	0.968750	0.968300	0.971098	0.965517
200	0.074700	0.111365	0.968750	0.968300	0.971098	0.965517
210	0.038000	0.114926	0.963068	0.962099	0.976331	0.948276
220	0.029800	0.115683	0.963068	0.962319	0.970760	0.954023

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
230	0.028200	0.122523	0.965909	0.965116	0.976471	0.954023

```python
import matplotlib.pyplot as plt

log_history = trainer.state.log_history

# List untuk menyimpan data
train_steps = []
train_loss = []

eval_steps = []
eval_loss = []
eval_accuracy = []

# Ekstrak data dari log_history
for log in log_history:
    # Training loss biasanya muncul saat step training
    if 'loss' in log and 'step' in log:
        train_steps.append(log['step'])
        train_loss.append(log['loss'])
    # Evaluation metrics muncul saat evaluasi
    if 'eval_loss' in log and 'step' in log:
        eval_steps.append(log['step'])
        eval_loss.append(log['eval_loss'])
    if 'eval_accuracy' in log and 'step' in log:
        eval_accuracy.append(log['eval_accuracy'])

# Plotting
plt.figure(figsize=(12, 5))

# Plot Loss (Training & Validation)
plt.subplot(1, 2, 1)
plt.plot(train_steps, train_loss, label='Training Loss')
plt.plot(eval_steps, eval_loss, label='Validation Loss')
plt.xlabel('Step')
plt.ylabel('Loss')
plt.title('Training & Validation Loss')
plt.legend()

# Plot Validation Accuracy
plt.subplot(1, 2, 2)
plt.plot(eval_steps, eval_accuracy, label='Validation Accuracy', color='green')
plt.xlabel('Step')
plt.ylabel('Accuracy')
plt.title('Validation Accuracy')
plt.legend()

plt.tight_layout()
plt.show()
```



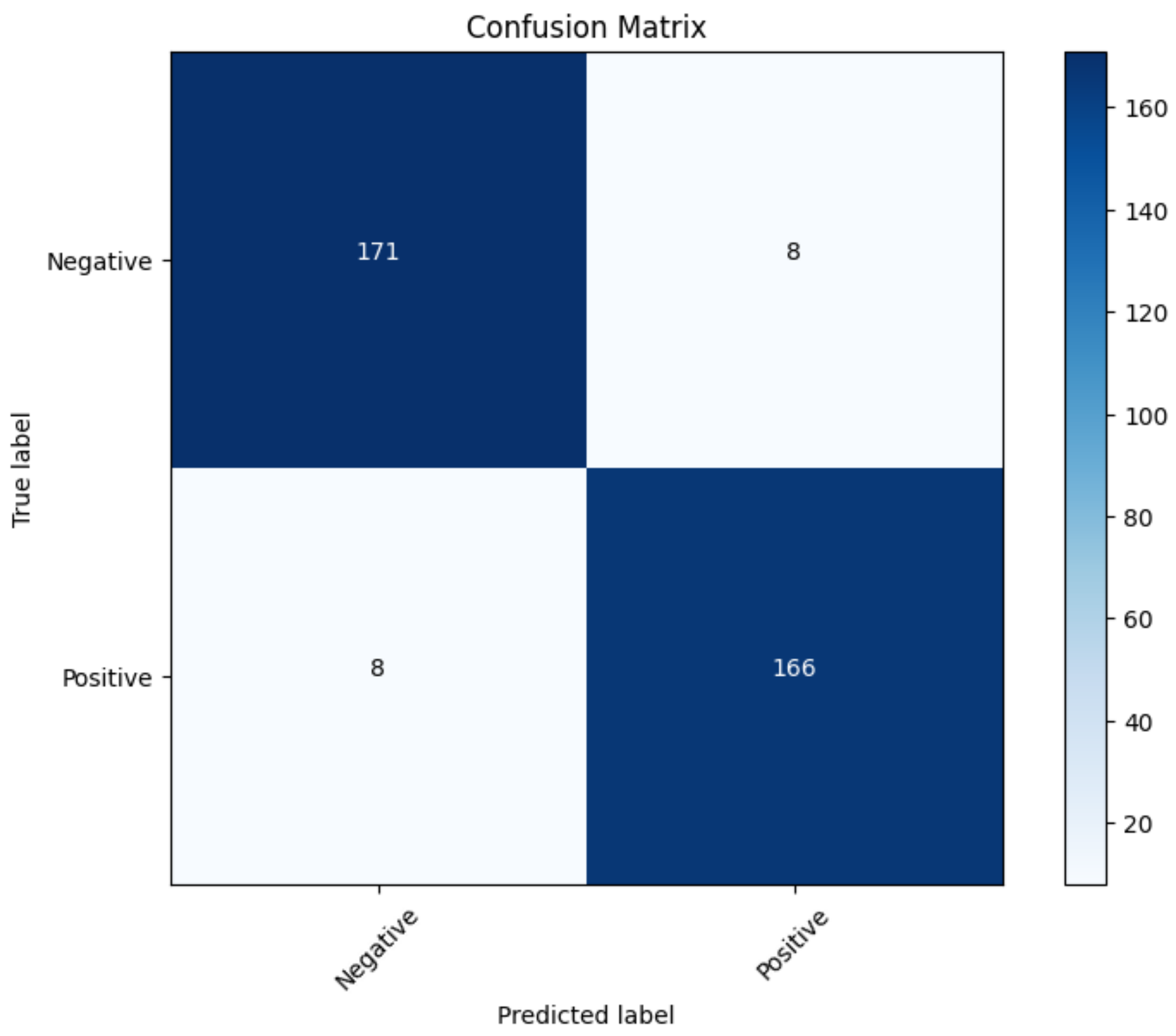
```
# Import Library Tambahan

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report


# Prediksi pada Data Testing
predictions = trainer.predict(test_dataset)
y_pred = np.argmax(predictions.predictions, axis=1)
y_true = predictions.label_ids

# Menghitung dan Menampilkan Test Accuracy
test_accuracy = accuracy_score(y_true, y_pred)
print(f"Test Accuracy: {test_accuracy}")
```

```
Test Accuracy: 0.9546742209631728
```

```
# Classification Report

cr = classification_report(y_true, y_pred)
cm = confusion_matrix(y_true, y_pred)

print("=== Classification Report ===")
print(cr)
print("=== Confusion Matrix (===")
print(cm)
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       179
           1       0.95      0.95      0.95       174

    accuracy                           0.95       353
   macro avg       0.95      0.95      0.95       353
weighted avg       0.95      0.95      0.95       353

=== Confusion Matrix (===
[[171   8]
 [  8 166]]
```

```
# Plot Confusion Matrix
plt.figure(figsize=(8, 6))
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()

classes = ['Negative', 'Positive']
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)

thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```



Model 3

```
from transformers import AutoModelForSequenceClassification, TrainingArguments, Trainer, get
from torch.optim import AdamW
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

# Konfigurasi model
config = BertConfig.from_pretrained(
    "indobenchmark/indobert-base-p1",
    num_labels=2,
    seed=25
)

model = AutoModelForSequenceClassification.from_pretrained(
    "indobenchmark/indobert-base-p1",
    config=config
)
```

```
# Optimizer
optimizer = AdamW(model.parameters(), lr=2e-5, weight_decay=0.01)

# Parameter training
train_batch_size = 32
num_train_epochs = 7
train_dataset_size = len(train_dataset)
total_training_steps = (train_dataset_size // train_batch_size) * num_train_epochs

# Scheduler LR
lr_scheduler = get_scheduler(
    name="linear",
    optimizer=optimizer,
    num_warmup_steps=500,
    num_training_steps=total_training_steps
)

# Fungsi evaluasi
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binar
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

# TrainingArguments dengan evaluasi per step dan save per step
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=train_batch_size,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    learning_rate=2e-5,

    # Penting: evaluasi dan save per step, agar load_best_model_at_end bisa jalan
    eval_strategy="steps",
    save_strategy="steps",
    eval_steps=10,
    save_steps=10,

    load_best_model_at_end=True,
    metric_for_best_model='accuracy',
    greater_is_better=True,

    # Optional: untuk menghindari terlalu banyak checkpoint
    save_total_limit=3
)
```

```
# Inisialisasi Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics,
    optimizers=(optimizer, lr_scheduler)
)
```

```
Some weights of BertForSequenceClassification were not initialized from the
You should probably TRAIN this model on a down-stream task to be able to us
```

```
# Train model
trainer.train()
```

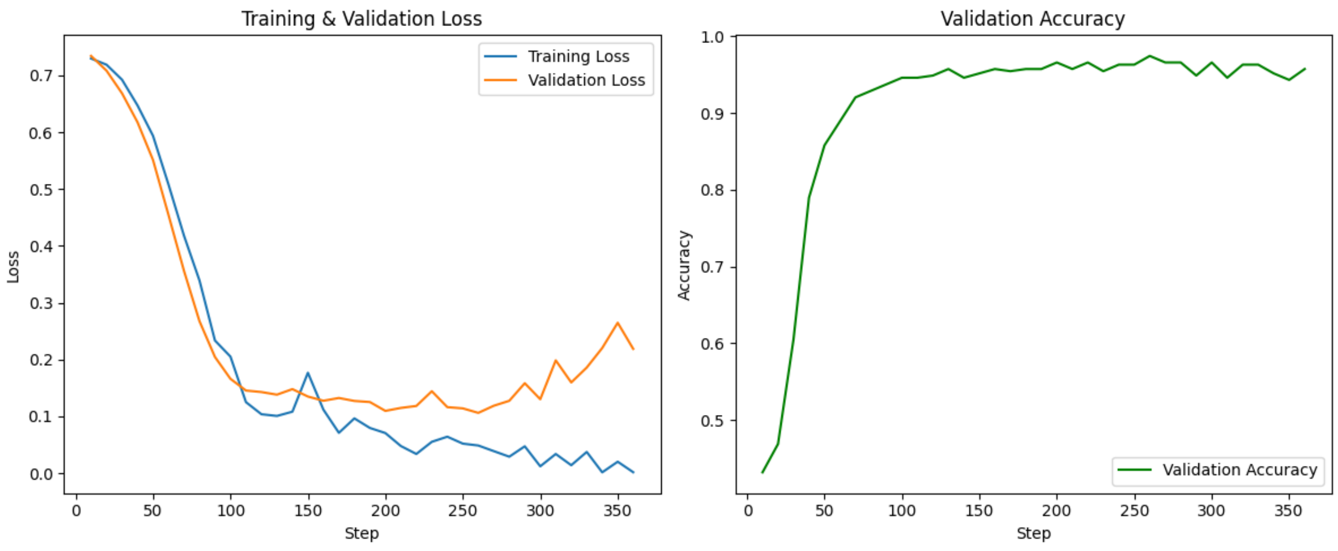
[364/364 27:44, Epoch 7/7]

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
10	0.729600	0.734138	0.431818	0.586777	0.458065	0.816092
20	0.718700	0.708212	0.468750	0.604651	0.478261	0.821839
30	0.692300	0.668344	0.605114	0.666667	0.572016	0.798851
40	0.647100	0.617460	0.789773	0.774390	0.824675	0.729885
50	0.593600	0.551079	0.857955	0.851190	0.882716	0.821839
60	0.507900	0.454359	0.889205	0.884273	0.914110	0.856322
70	0.417900	0.355961	0.920455	0.920455	0.910112	0.931034
80	0.339100	0.267440	0.928977	0.927954	0.930636	0.925287
90	0.233400	0.204406	0.937500	0.936416	0.941860	0.931034
100	0.205100	0.166032	0.946023	0.944928	0.953216	0.936782
110	0.125000	0.145421	0.946023	0.945245	0.947977	0.942529
120	0.103700	0.142766	0.948864	0.948571	0.943182	0.954023
130	0.100700	0.138157	0.957386	0.957265	0.949153	0.965517
140	0.108200	0.147898	0.946023	0.943953	0.969697	0.919540
150	0.176700	0.134857	0.951705	0.950725	0.959064	0.942529
160	0.111800	0.127436	0.957386	0.956772	0.959538	0.954023
170	0.071000	0.132294	0.954545	0.954286	0.948864	0.959770
180	0.096400	0.127097	0.957386	0.957265	0.949153	0.965517
190	0.079600	0.125196	0.957386	0.957507	0.944134	0.971264
200	0.070600	0.109607	0.965909	0.965714	0.960227	0.971264

200	0.070600	0.109607	0.965909	0.965714	0.960227	0.971264
210	0.047800	0.114816	0.957386	0.957265	0.949153	0.965517
220	0.033800	0.118288	0.965909	0.964912	0.982143	0.948276
230	0.055200	0.144229	0.954545	0.954023	0.954023	0.954023
240	0.064200	0.116181	0.963068	0.962319	0.970760	0.954023
250	0.052000	0.114047	0.963068	0.962963	0.954802	0.971264
260	0.048700	0.106112	0.974432	0.973913	0.982456	0.965517
270	0.038800	0.118658	0.965909	0.964912	0.982143	0.948276
280	0.029000	0.127342	0.965909	0.964912	0.982143	0.948276
290	0.047300	0.158320	0.948864	0.948864	0.938202	0.959770
300	0.012100	0.130109	0.965909	0.965116	0.976471	0.954023

```
import matplotlib.pyplot as plt

log_history = trainer.state.log_history

# List untuk menyimpan data
train_steps = []
train_loss = []

eval_steps = []
eval_loss = []
eval_accuracy = []

# Ekstrak data dari log_history
for log in log_history:
    # Training loss biasanya muncul saat step training
    if 'loss' in log and 'step' in log:
        train_steps.append(log['step'])
        train_loss.append(log['loss'])
    # Evaluation metrics muncul saat evaluasi
    if 'eval_loss' in log and 'step' in log:
        eval_steps.append(log['step'])
        eval_loss.append(log['eval_loss'])
    if 'eval_accuracy' in log and 'step' in log:
        eval_accuracy.append(log['eval_accuracy'])

# Plotting
plt.figure(figsize=(12, 5))

# Plot Loss (Training & Validation)
plt.subplot(1, 2, 1)
plt.plot(train_steps, train_loss, label='Training Loss')
plt.plot(eval_steps, eval_loss, label='Validation Loss')
plt.xlabel('Step')
plt.ylabel('Loss')
plt.title('Training & Validation Loss')
plt.legend()
```

```
# Plot Validation Accuracy
plt.subplot(1, 2, 2)
plt.plot(eval_steps, eval_accuracy, label='Validation Accuracy', color='green')
plt.xlabel('Step')
plt.ylabel('Accuracy')
plt.title('Validation Accuracy')
plt.legend()

plt.tight_layout()
plt.show()
```



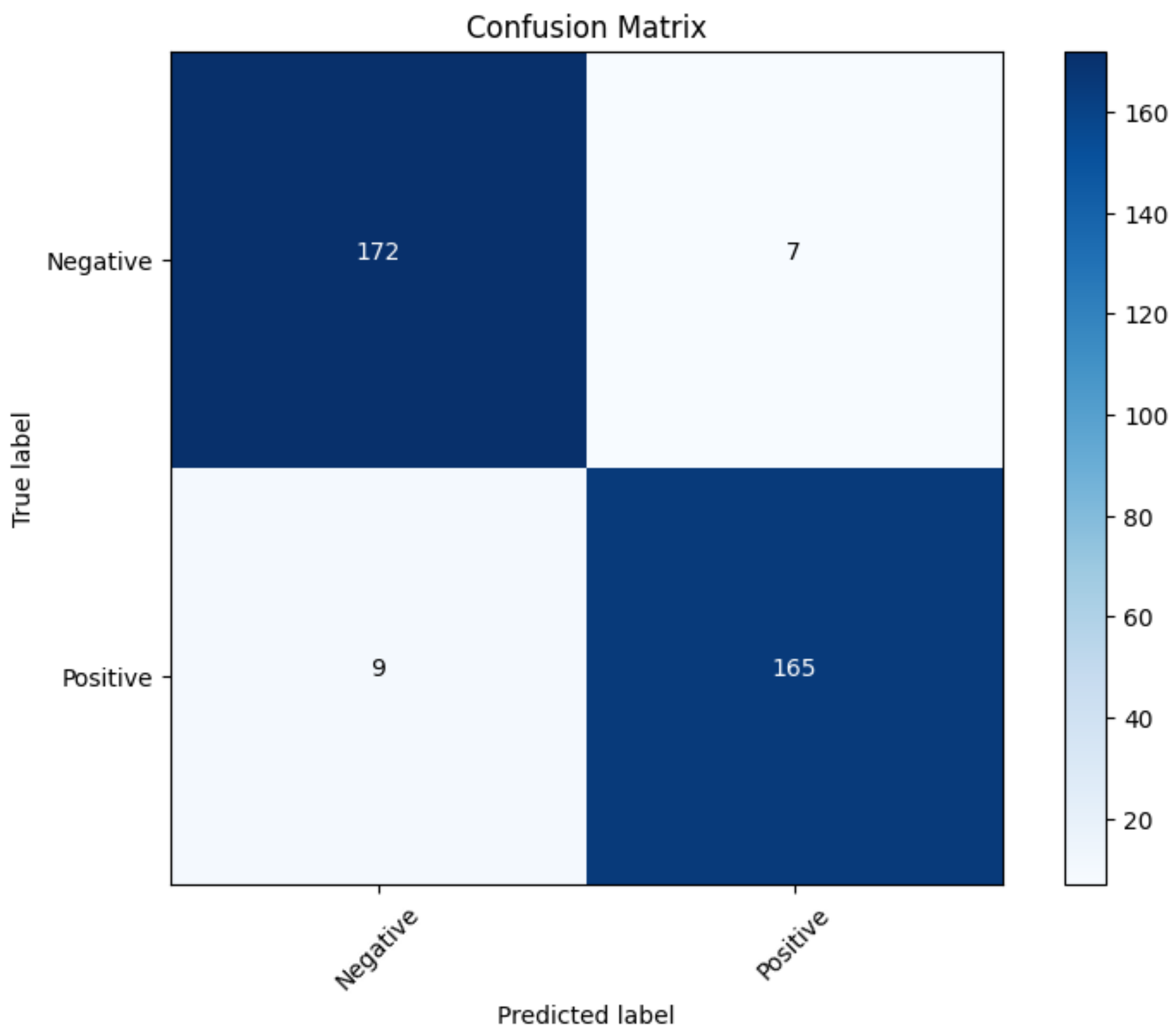
```
# Prediksi pada Data Testing
predictions = trainer.predict(test_dataset)
y_pred = np.argmax(predictions.predictions, axis=1)
y_true = predictions.label_ids

# Menghitung dan Menampilkan Test Accuracy
test_accuracy = accuracy_score(y_true, y_pred)
print(f"Test Accuracy: {test_accuracy}")
```

```
Test Accuracy: 0.9546742209631728
```

```
# Classification Report

cr = classification_report(y_true, y_pred)
cm = confusion_matrix(y_true, y_pred)

print("=== Classification Report ===")
print(cr)
print("=== Confusion Matrix (===")
print(cm)
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.95      0.96      0.96       179
           1       0.96      0.95      0.95       174

    accuracy                           0.95       353
   macro avg       0.95      0.95      0.95       353
weighted avg       0.95      0.95      0.95       353

=== Confusion Matrix (===
[[172   7]
 [  9 165]]
```

```
# Plot Confusion Matrix
plt.figure(figsize=(8, 6))
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()

classes = ['Negative', 'Positive']
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)

thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```



Confusion Matrix
True label
Negative
Positive
172
7
9
165
Predicted label
Negative
Positive
160
140
120
100
80
60
40
20

Model 4

```
import os
import random
import numpy as np
import tensorflow as tf

def set_seed(seed=25):
    # Pengaturan untuk Python core
    os.environ['PYTHONHASHSEED'] = str(seed)
    random.seed(seed)

    # Pengaturan untuk NumPy
    np.random.seed(seed)

    # Pengaturan untuk TensorFlow
    tf.random.set_seed(seed)

    # Pengaturan tambahan untuk TensorFlow
    os.environ['TF_DETERMINISTIC_OPS'] = '1'
    os.environ['TF_CUDNN_DETERMINISTIC'] = '1'


    # Jika menggunakan GPU
    try:
        tf.config.experimental.set_memory_growth(
            tf.config.list_physical_devices('GPU')[0], True)
    except:
        pass

# Aplikasikan seed
set_seed(25)

# Jika menggunakan Keras, tambahkan ini:
from tensorflow import keras
keras.utils.set_random_seed(25)
```

```
# Split data
X_eda = df_eda['eda']
y_eda = df_eda['label']

# Split dataset
X_eda_train, X_eda_val, y_eda_train, y_eda_val = train_test_split(
    X_eda, y_eda, test_size=0.3, random_state=25, stratify=y_eda
)

X_eda_val, X_eda_test, y_eda_val, y_eda_test = train_test_split(
    X_eda_val, y_eda_val, test_size=0.5, random_state=25, stratify=y_eda_val
)

print(f"Training set size: {X_eda_train.shape[0]}")
print(f"Validation set size: {X_eda_val.shape[0]}")
print(f"Testing set size: {X_eda_test.shape[0]}")
```

```
Training set size: 1642
Validation set size: 352
Testing set size: 353
```

```
from transformers import AutoTokenizer

# Inisialisasi tokenizer
tokenizer = AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p1")

# Tokenisasi data
def tokenize_function(texts):
    return tokenizer(
        texts.tolist(),
        padding=True,
        truncation=True,
        max_length=100
    )

train_eda_encodings = tokenize_function(X_eda_train)
val_eda_encodings = tokenize_function(X_eda_val)
test_eda_encodings = tokenize_function(X_eda_test)
```

```
from torch.utils.data import Dataset # Import Dataset from torch.utils.data
import torch

class ReviewsDataset(Dataset):
    def __init__(self, encodings, label):
        self.encodings = encodings
        self.label = label

    def __len__(self):
        return len(self.label)

    def __getitem__(self, idx):
        # Check if idx is a list (for batching) or an integer (single item)
        if isinstance(idx, list):
            # If idx is a list, create a batch of items
            item = {key: torch.tensor([val[i] for i in idx]) for key, val in self.encodings.
            item['label'] = torch.tensor([self.label[i] for i in idx])
        else:
            # If idx is an integer, get a single item
            item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
            item['label'] = torch.tensor(self.label[idx])
        return item

train_eda_dataset = ReviewsDataset(train_eda_encodings, y_eda_train.tolist())
val_eda_dataset = ReviewsDataset(val_eda_encodings, y_eda_val.tolist())
test_eda_dataset = ReviewsDataset(test_eda_encodings, y_eda_test.tolist())


from transformers import AutoModelForSequenceClassification, TrainingArguments, Trainer, get
from torch.optim import AdamW
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

# Konfigurasi model
config = BertConfig.from_pretrained(
    "indobenchmark/indobert-base-p1",
    num_labels=2,
    seed=25
)

model = AutoModelForSequenceClassification.from_pretrained(
    "indobenchmark/indobert-base-p1",
    config=config
)

# Optimizer
optimizer = AdamW(model.parameters(), lr=2e-5, weight_decay=0.01)

# Parameter training
train_batch_size = 32
num_train_epochs = 5
train_dataset_size = len(train_dataset)
total_training_steps = (train_dataset_size // train_batch_size) * num_train_epochs

# Scheduler LR
lr_scheduler = get_scheduler(
    name="linear",
    optimizer=optimizer,
    num_warmup_steps=500,
    num_training_steps=total_training_steps
)

# Fungsi evaluasi
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binar
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

# TrainingArguments dengan evaluasi per step dan save per step
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=train_batch_size,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    learning_rate=2e-5,

    # Penting: evaluasi dan save per step, agar load_best_model_at_end bisa jalan
    eval_strategy="steps",
    save_strategy="steps",
    eval_steps=10,
    save_steps=10,

    load_best_model_at_end=True,
    metric_for_best_model='accuracy',
    greater_is_better=True,

    # Optional: untuk menghindari terlalu banyak checkpoint
    save_total_limit=3
)

# Inisialisasi Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_eda_dataset,
    eval_dataset=val_eda_dataset,
    compute_metrics=compute_metrics,
    optimizers=(optimizer, lr_scheduler)
)
```

```
Some weights of BertForSequenceClassification were not initialized from the
You should probably TRAIN this model on a down-stream task to be able to us
```

```
# Train model
trainer.train()
```

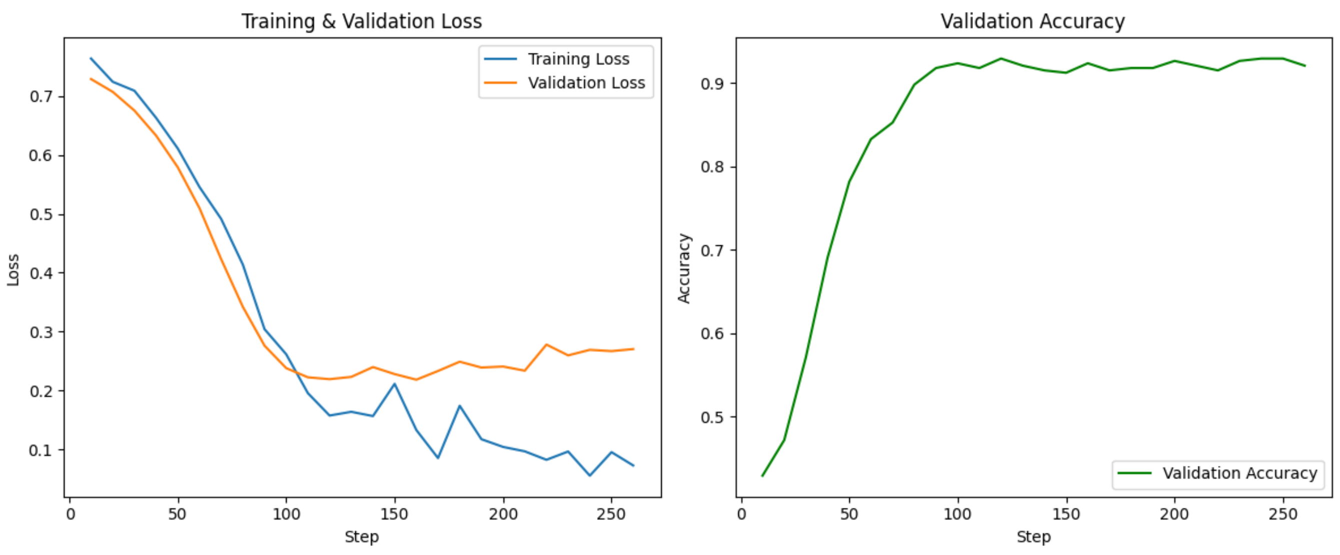
[260/260 06:46, Epoch 5/5]

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
10	0.763900	0.729023	0.428977	0.199203	0.324675	0.143678
20	0.724400	0.707301	0.471591	0.335714	0.443396	0.270115
30	0.709200	0.675487	0.571023	0.494983	0.592000	0.425287
40	0.663100	0.632954	0.690341	0.635452	0.760000	0.545977
50	0.611000	0.579035	0.781250	0.771513	0.797546	0.747126
60	0.545600	0.509669	0.832386	0.830946	0.828571	0.833333
70	0.491400	0.422927	0.852273	0.857923	0.817708	0.902299
80	0.413700	0.341881	0.897727	0.894118	0.915663	0.873563
90	0.304000	0.275969	0.917614	0.914956	0.934132	0.896552
100	0.261300	0.237762	0.923295	0.919881	0.950920	0.890805
110	0.195200	0.222174	0.917614	0.915452	0.928994	0.902299
120	0.157200	0.219019	0.928977	0.927114	0.940828	0.913793
130	0.163600	0.222882	0.920455	0.919075	0.924419	0.913793
140	0.156300	0.239576	0.914773	0.909639	0.955696	0.867816
150	0.211100	0.227656	0.911932	0.907463	0.944099	0.873563
160	0.132600	0.218126	0.923295	0.920354	0.945455	0.896552
170	0.084900	0.232953	0.914773	0.914773	0.904494	0.925287
180	0.173800	0.248714	0.917614	0.913947	0.944785	0.885057
190	0.117000	0.238794	0.917614	0.917847	0.905028	0.931034
200	0.103900	0.240505	0.926136	0.924855	0.930233	0.919540
210	0.096400	0.233420	0.920455	0.917647	0.939759	0.896552
220	0.082000	0.277807	0.914773	0.909091	0.961538	0.862069
230	0.096100	0.259450	0.926136	0.924855	0.930233	0.919540
240	0.055000	0.268874	0.928977	0.927114	0.940828	0.913793
250	0.095000	0.266702	0.928977	0.927114	0.940828	0.913793
260	0.072400	0.270170	0.920455	0.917647	0.939759	0.896552

```
TrainOutput(global_step=260, training_loss=0.2877014634700922, metrics=
{'train_runtime': 407.0406, 'train_samples_per_second': 20.17
```

```
import matplotlib.pyplot as plt
```

```
log_history = trainer.state.log_history

# List untuk menyimpan data
train_steps = []
train_loss = []

eval_steps = []
eval_loss = []
eval_accuracy = []

# Ekstrak data dari log_history
for log in log_history:
    # Training loss biasanya muncul saat step training
    if 'loss' in log and 'step' in log:
        train_steps.append(log['step'])
        train_loss.append(log['loss'])
    # Evaluation metrics muncul saat evaluasi
    if 'eval_loss' in log and 'step' in log:
        eval_steps.append(log['step'])
        eval_loss.append(log['eval_loss'])
    if 'eval_accuracy' in log and 'step' in log:
        eval_accuracy.append(log['eval_accuracy'])

# Plotting
plt.figure(figsize=(12, 5))

# Plot Loss (Training & Validation)
plt.subplot(1, 2, 1)
plt.plot(train_steps, train_loss, label='Training Loss')
plt.plot(eval_steps, eval_loss, label='Validation Loss')
plt.xlabel('Step')
plt.ylabel('Loss')
plt.title('Training & Validation Loss')
plt.legend()

# Plot Validation Accuracy
plt.subplot(1, 2, 2)
plt.plot(eval_steps, eval_accuracy, label='Validation Accuracy', color='green')
plt.xlabel('Step')
plt.ylabel('Accuracy')
plt.title('Validation Accuracy')
plt.legend()

plt.tight_layout()
plt.show()
```



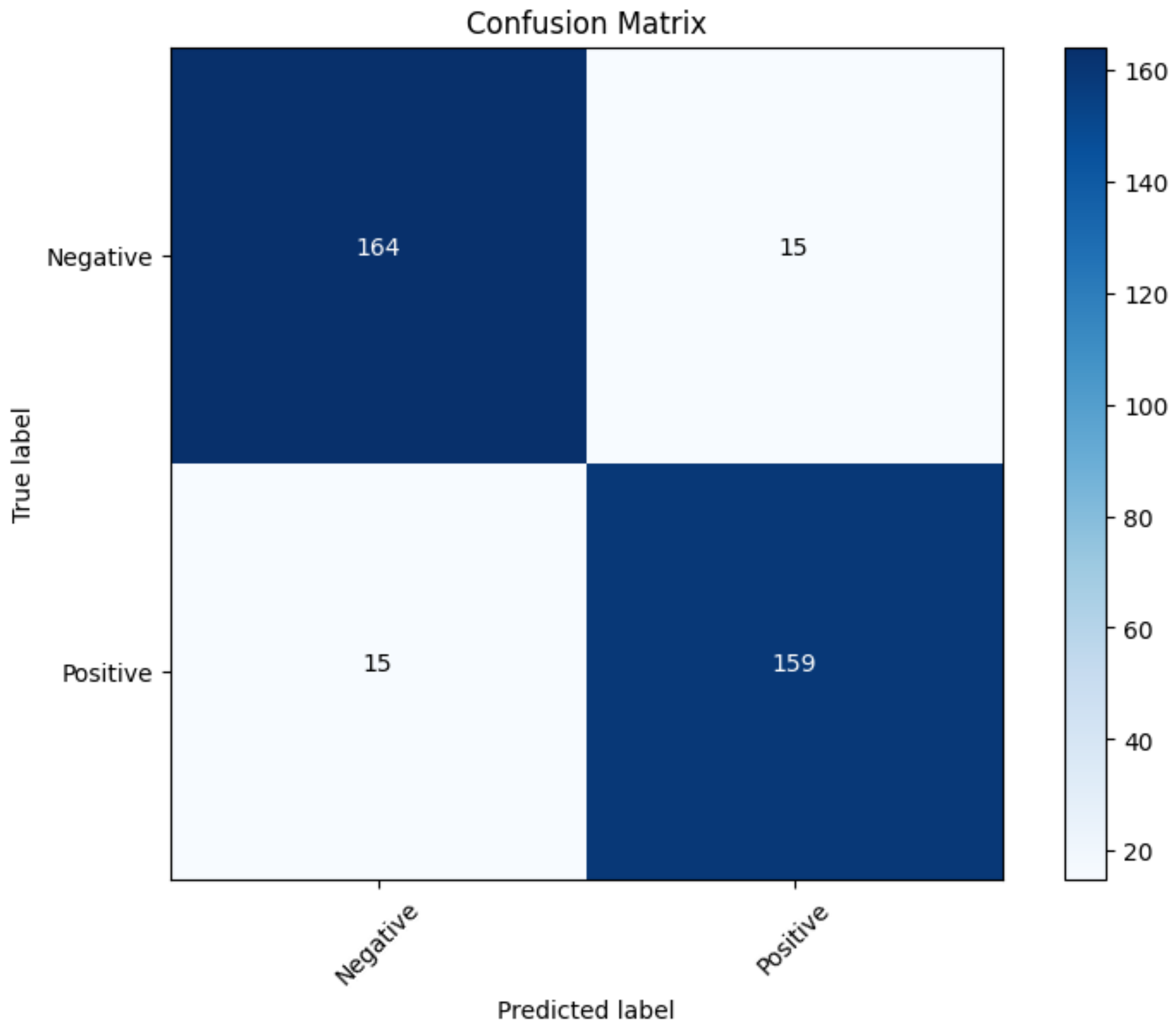
```
# Prediksi pada Data Testing
predictions = trainer.predict(test_eda_dataset)
y_pred = np.argmax(predictions.predictions, axis=1)
y_true = predictions.label_ids

# Menghitung dan Menampilkan Test Accuracy
test_accuracy = accuracy_score(y_true, y_pred)
print(f"Test Accuracy: {test_accuracy}")
```

```
Test Accuracy: 0.9150141643059491
```

```
# Classification Report

cr = classification_report(y_true, y_pred)
cm = confusion_matrix(y_true, y_pred)

print("=== Classification Report ===")
print(cr)
print("=== Confusion Matrix (===")
print(cm)
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.92      0.92      0.92       179
           1       0.91      0.91      0.91       174

    accuracy                           0.92       353
   macro avg       0.91      0.91      0.91       353
weighted avg       0.92      0.92      0.92       353

=== Confusion Matrix (===
[[164  15]
 [ 15 159]]
```

```
# Plot Confusion Matrix
plt.figure(figsize=(8, 6))
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()

classes = ['Negative', 'Positive']
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)

thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```


Confusion Matrix
True label
Negative
Positive
164
15
15
159
Predicted label
Negative
Positive
160
140
120
100
80
60
40
20