Analisis Sentimen pada Ulasan Pengguna Aplikasi balé by BTN di Play Store

Kelompok 6

Anggota :

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

Daftar Isi

- Pendahuluan
- Preprocessing
- EDA
- Modelling
- Evaluasi

Latar Belakang

- Digitalisasi mendorong Bank Tabungan Negara (BTN) menghadirkan Balé by BTN untuk memberikan layanan yang cepat, aman, dan efisien kepada nasabah.

- Persepsi dan kepuasan pengguna berperan penting dalam kesuksesan aplikasi. Ulasan di Google Play Store mencerminkan opini dan pengalaman pengguna yang perlu dianalisis untuk memahami sentimen yang berkembang.

- Penelitian ini menerapkan analisis sentimen dengan model IndoBERT untuk mengidentifikasi sentimen dalam ulasan, menemukan kata-kata yang dominan, dan mengevaluasi performa model dalam mengklasifikasi sentimen pengguna terhadap aplikasi Balé by BTN.

Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, berikut rumusan masalah yang kami tentukan:

Problem 01

Bagaimana sentimen pengguna aplikasi Balé by BTN yang terlihat melalui ulasan di Google Play Store?

Problem 02

Apa saja kata-kata yang sering muncul untuk menggambarkan ulasan dengan sentimen positif dan negatif?

Problem 03

Bagaimana performa model IndoBERT dalam menganalisis sentimen dalam ulasan pengguna?

Tujuan

Berikut tujuan dari proyek yang kami lakukan berdasarkan rumusan yang sudah kami tentukan:

Tujuan 01

Mengetahui sentimen pengguna aplikasi Balé by BTN yang tercermin melalui ulasan di Google Play Store

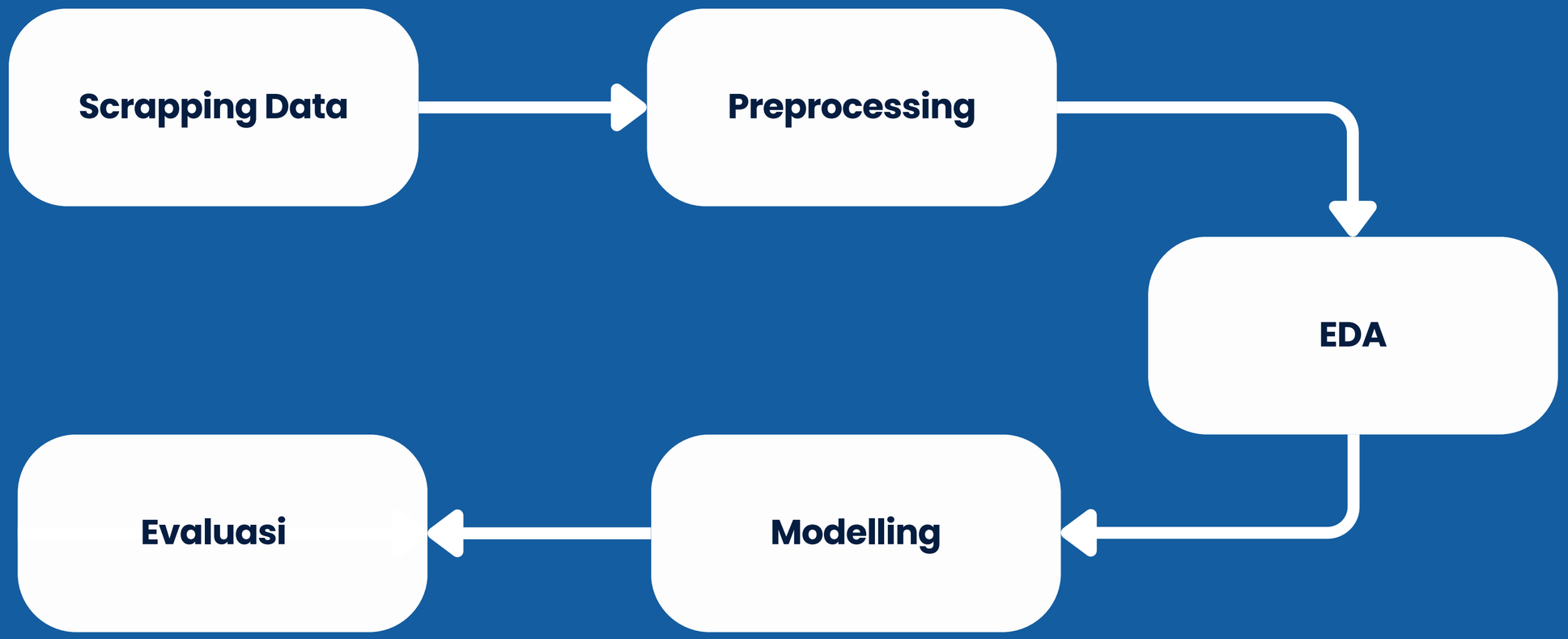
Tujuan 02

Mengidentifikasi kata-kata yang sering muncul dalam ulasan untuk merepresentasikan sentimen positif dan negatif

Tujuan 03

Menganalisis performa model IndoBERT dalam mengklasifikasi dan memprediksi sentimen dalam ulasan pengguna

Alur Pengerjaan



Data

Pengumpulan data (*scraping)* dilakukan dengan menggunakan *package* "google-play-scraper" di python. Pada proses scraping, diambil 2500 ulasan dengan bahasa Indonesia paling baru, berdasarkan Play Store, pada rentang waktu Oktober 2024–April 2025.

Hasil *Scraping*:

Data terdiri dari 2500 ulasan.

- **reviewId** : id unik untuk setiap ulasan.
- **userName** : nama pengguna yang memberikan ulasan.
- **userImage** : URL gambar profil pengguna.
- **content** : teks ulasan dari pengguna.
- **score** : rating yang diberikan pengguna (1-5).
- **thumbsUpCount** : jumlah *like* dari pengguna lain.
- **reviewCreatedVersion** : versi aplikasi ketika ulasan dibuat.
- **at** : tanggal dan waktu ulasan dibuat.
- **replyContent** : teks balasan dari *developer.*
- **repliedAt** : waktu balasan dari *developer* dibuat.

Data

Cuplikan 5 data pertama :

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt
0	d7aa5f09-43de-4640-a98a-6fe557e494b8	Siti Nurhalimah	https://play-lh.googleusercontent.co...	Aplikasi sangat membantu	5	0	2.0.1	2025-04-10 04:05:36	Terima kasih Ibu Siti Nurhalimah tel...	2025-04-10 04:54:38
1	ca512324-c225-44fc-ba02-50568f44bcda	Budi Mulyono	https://play-lh.googleusercontent.co...	biar otak selalu aktif dimasa tua	5	0	2.0.1	2025-04-10 01:13:58	Terima kasih Ibu Budi Mulyono telah ...	2025-04-10 04:57:55
2	0002179b-b611-4f75-8441-e27282b3701c	Syaibani Abdurrahman	https://play-lh.googleusercontent.co...	tolong buatkan fitur qris tampil bia...	3	0	None	2025-04-09 23:09:09	Terima kasih Bapak Syaibani Abdurrah...	2025-04-10 04:58:23
3	7437196d-b1be-41b2-8876-df4cc1b92689	Erik Wijaya	https://play-lh.googleusercontent.co...	sering banget gangguan mau login aja...	1	0	2.0.1	2025-04-09 20:34:06	Bapak Erik Wijaya kami sampaikan moh...	2025-04-10 04:58:42
4	63e4f975-7998-4de1-a5dc-10e12f12d41a	wahyu nugraha	https://play-lh.googleusercontent.co...	aplikasi buruk, gangguan gak jelas.	1	0	2.0.1	2025-04-09 19:38:37	Bapak Wahyu Nugraha kami sampaikan m...	2025-04-10 04:58:54

Data

Cuplikan ulasan :

“mohon maaf bapk/ibuk, ini knp ya saya mau login susah sekali ya masuk ke aplikasi bale by btn. dan saya juga baru kirim dana ke tabungan TDK ada notifikasinya yg masuk ya. tolong dong BPK/ibuk penyelasannya.”

“uang tf sering ilang ga masuk, aplikasi bobrok 👎“

“menyala abangku🔥🔥, aplikasi banking modern dengan tampilan simple jadi lebih mudah digunakan😍, tapi juga dapat digunakan untuk buka rekening kpr, qris dll, jadi nyaman dipake terus😘”

“fiture sangat lengkap, dapat digunakan dengan baik dalam transaksi memenuhi kebutuhan sehari hari“

Labelling

Labelling awal dilakukan berdasarkan kolom ‘score’, dengan :

- 1 – 3 = negatif
- 4 – 5 = positif

Kemudian, dilakukan validasi manual :

content	score	sentiment
udh antre bikin rekening, udh daftar isi formulir macem2, ga bisa daftar Krn eror aplikasinya. buang2 waktu aja	1	negatif
gimana Sih Nih Mau Registrasi aja Susah Bener,No Kartu aja Bener Ko Masa Dibilang Beda... Bikin Ribet aja	1	negatif
mohon maaf kepada developer baru saja saya.beli token listrik melalui aplikasi ini. tapi token nya ga masuk saldo saya hilang 50000 ini bagaimana tolong diperbaikin	2	negatif
saya verifikasi akun online gagal mulu padahal blom pernah daftar min	2	negatif
Jadi gampang banget buka rekening dari hp, nggak perlu repot repot ke atm ambil uang.	3	positif
Fitur tarik tanpa kartu hilang	3	negatif
Lelet dari tdi ngisi gopay tapi truss aja blom masuk masuk sampe sekarang udah 2 kali ngisi tetep aja ngak masuk masuk gimana si... Mohon di perbaikin aja	4	negatif
Aplikasi yg bagus jadi lebih mudah bertransaksi	4	positif
Aplikasi yang gampang digunakan untuk mentransfer uang cek saldo gampang pokoknya recommended banget deh temen temen	5	positif
Ngrestart terus aplikasi nya. Baru login langsung logout lg. Mau transaksi jd susah.	5	negatif

Pre-Processing

Beberapa tahapan pre-processing yang akan dilakukan pada data :

Case Folding	Mengubah teks menjadi huruf kecil untuk menyeragamkan.
Hapus Karakter Non-huruf	Menghapus karakter selain huruf, seperti angka, tanda baca, atau simbol lainnya yang kurang relevan untuk analisis teks.
Normalisasi teks	Menggunakan daftar kata baku/kamus untuk mengubah teks menjadi bentuk yang sesuai.
Hapus *whitespace*	Menghapus spasi berlebih.
Hapus Stopwords	Menghapus kata-kata umum yang sering muncul dan tidak memberikan informasi yang bermakna
Hapus missing value & duplikat	Memastikan data tidak memiliki missing value dan duplikat.

- Normalisasi teks menggunakan **Kamus Alay - Colloquial Indonesian Lexicon**
- Stopwords (Indonesia & Inggris) menggunakan library NLTK (Natural Language Toolkit)

Pre-Processing

Dilakukan 3 jenis pre-processing yang berbeda pada data.

01

Hapus *whitespace*	**Case Folding**	**Hapus Karakter Non-huruf**
Normalisasi teks	**Hapus Stopwords**	**Hapus missing value & duplikat**

02

Hapus *whitespace*	**Case Folding**	**Hapus Karakter Non-huruf**
Normalisasi teks	**Hapus Stopwords**	**Hapus missing value & duplikat**

Dengan tambahan stopwords :

aplikasi, aplikasinya, apk, bank, btn, bale, banget, kali, mobile, ya

03

Hapus *whitespace*	**Hapus Karakter Non-huruf**	**Normalisasi teks**	**Hapus missing value & duplikat**

Pre-Processing

Sebelum

"Apa pake pulsa sangatt jauh beda dg bank yang lain, yang lebih modern😅serasa jaman 90 an pake pulsa"

Setelah

01 Setelah pre-processing: **2280** ulasan

"pakai pulsa beda bank modern serasa jaman pakai pulsa"

02 Setelah pre-processing: **2347** ulasan

"pakai pulsa beda modern serasa jaman pakai pulsa"

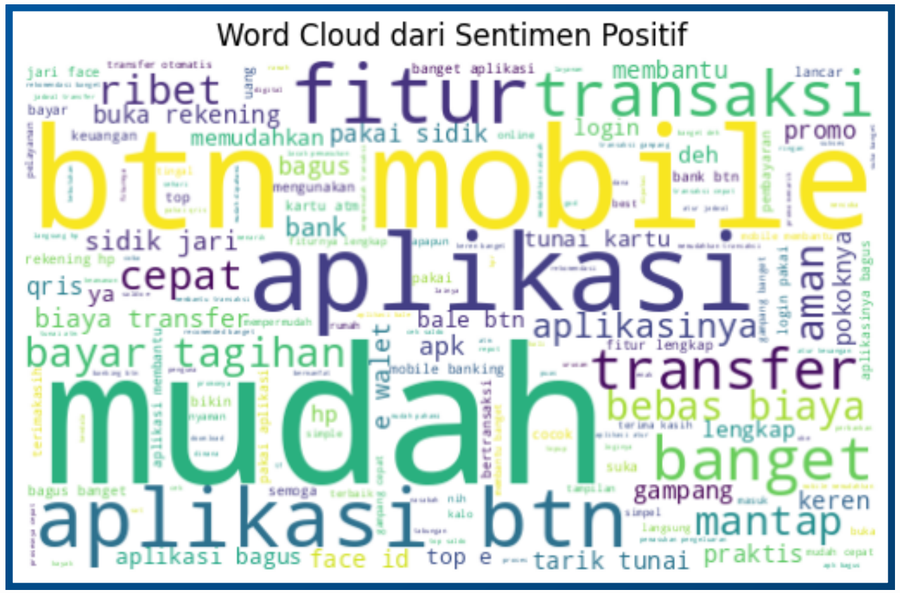
03 Setelah pre-processing: **2347** ulasan

"Apa pakai pulsa sangat jauh beda dengan bank yang lain yang lebih modern serasa jaman an pakai pulsa"

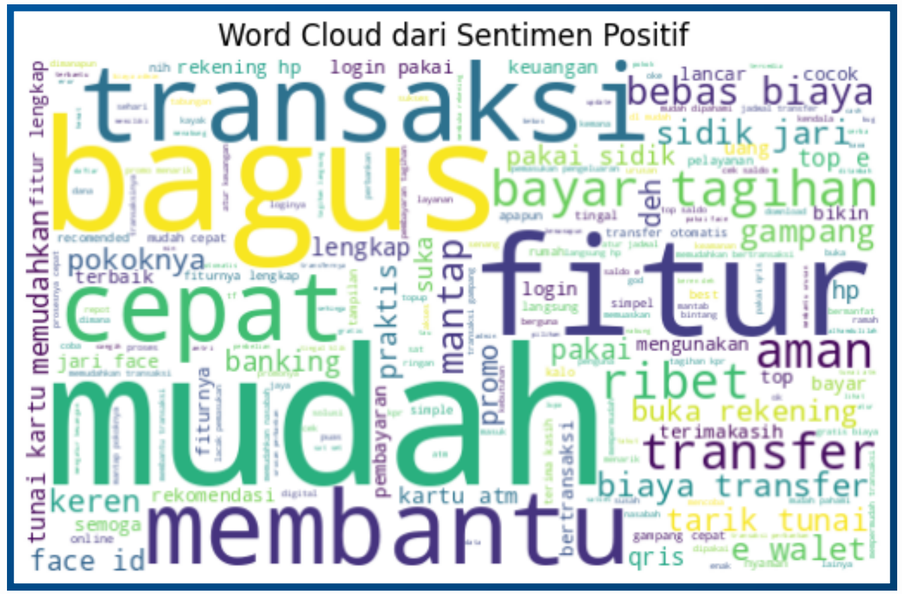
- Ketiga data hasil pre-processing akan digunakan dalam modelling.
- Data hasil pre-processing (02) digunakan untuk proses EDA karena telah melalui pembersihan stop words yang lebih sesuai dibandingkan 01 & 03.

EDA

Before



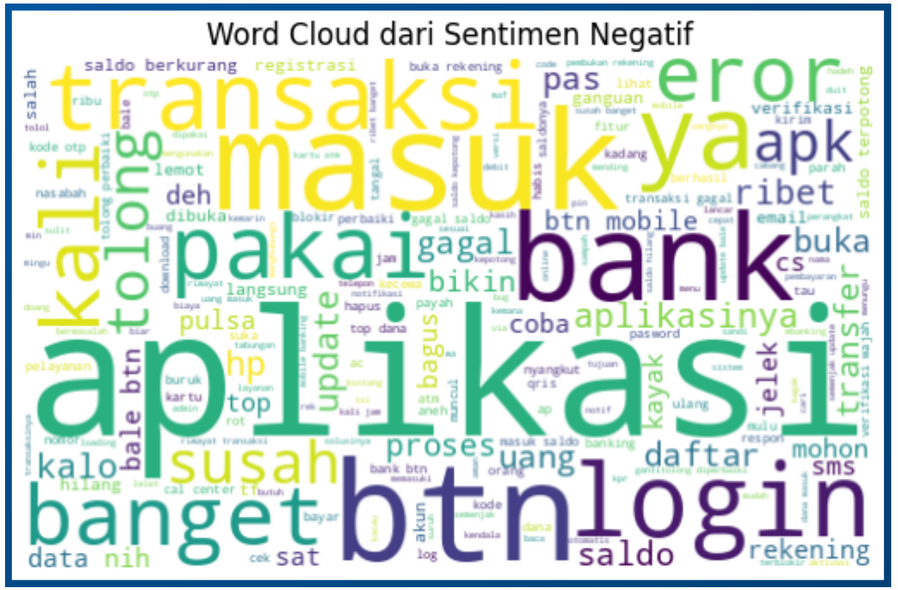
After



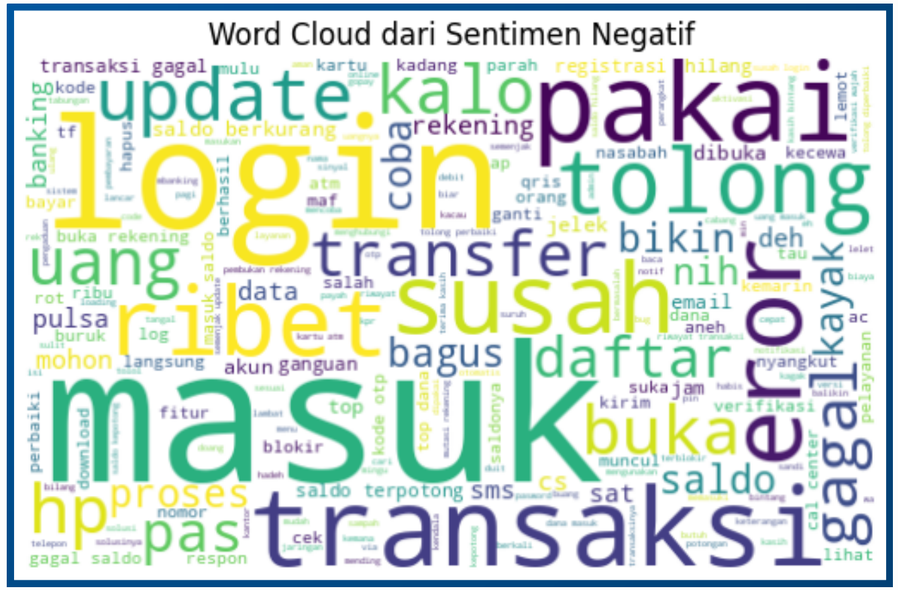
Pada **gambar "Before", word cloud masih didominasi kata-kata umum seperti “btn”, “mobile”, dan “aplikasi” yang tidak merepresentasikan sentimen positif secara jelas**, meskipun sudah melalui tahapan pre-processing. Hal ini disebabkan oleh belum dihapusnya kata-kata yang spesifik terkait nama atau elemen teknis aplikasi. Setelah dilakukan perbaikan pre-processing seperti dilihat pada **gambar "After", hasil word cloud menjadi lebih relevan dengan dominasi kata bermakna seperti “bagus”, “cepat”, dan “membantu” yang benar-benar mencerminkan sentimen positif** dari pengguna.

EDA

Before



After

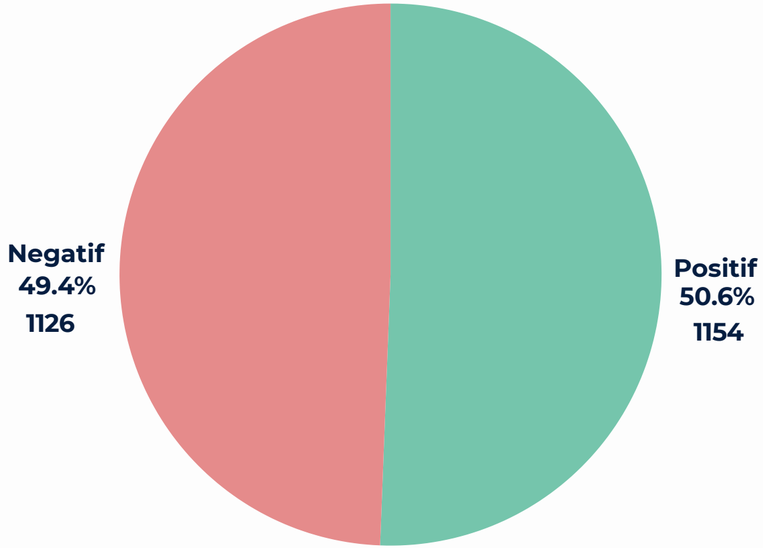


Pada **gambar "Before", word cloud masih dipenuhi kata umum seperti “aplikasi”, “btn”, dan “bank” yang tidak menggambarkan sentimen negatif** secara spesifik. Setelah dilakukan penyempurnaan pre-processing, word cloud pada **gambar "After" menampilkan kata-kata seperti “login”, “gagal”, dan “susah” yang lebih relevan dengan keluhan pengguna**, sehingga memberikan insight yang lebih bermakna dalam analisis sentimen negatif.

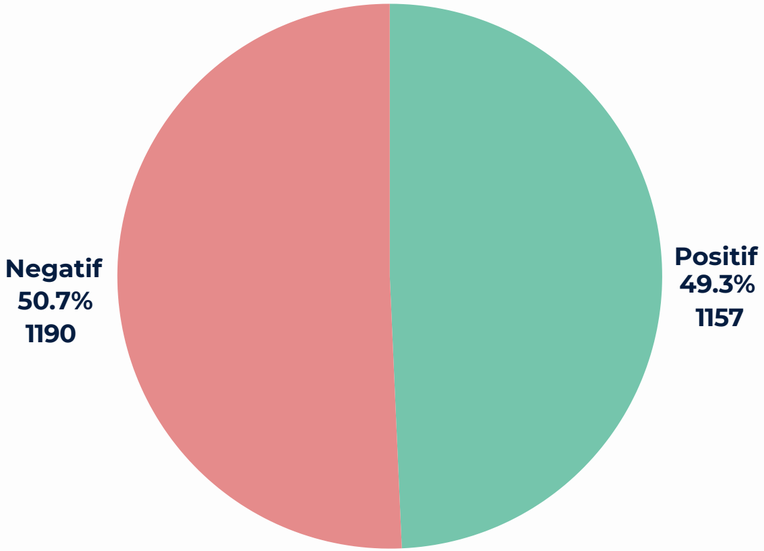
EDA

Distribusi Sentimen

Before



After



Pada pie chart distribusi sentimen di atas, terlihat adanya perbedaan jumlah data antara Before dan After, dari total 2280 (1.154 positif dan 1.126 negatif) data menjadi 2347 (1.157 positif dan 1.190 negatif). Hal ini disebabkan oleh perbedaan strategi pre-processing. **Pada "Before" (tahapan pre-processing yang kami coba lakukan pertama kali), dilakukan penghapusan stopwords pada data yang akan digunakan untuk modelling**, sehingga banyak data menjadi kosong dan dianggap sebagai missing value. Akibatnya, jumlah data yang bisa dianalisis berkurang. Sebaliknya, **pada "After", tidak dilakukan penghapusan stopwords pada data untuk modelling**, sehingga lebih banyak data yang dipertahankan. **Distribusi sentimen pun mengalami sedikit pergeseran, dari dominan positif (50.6%) menjadi sedikit lebih banyak negatif (50.7%)** setelah perubahan preprocessing. Meski terjadi sedikit pergeseran proporsi distribusi sentimen, tetapi **secara keseluruhan distribusi sentimen pada kedua data tetap cukup seimbang.**

Metode

Untuk mencapai tujuan penelitian, analisis sentimen terhadap ulasan pengguna akan dilakukan dengan menggunakan **BERT** (B*idirectional Encoder Representations from Transformers*). BERT merupakan model bahasa *pre-trained* yang telah mencapai performa *state-of-the-art* dalam berbagai permasalahan *Natural Language Processing* (NLP), dengan kemampuan merepresentasikan konteks secara dua arah (*bidirectional*).

Dalam implementasinya, digunakan **IndoBERT**, yaitu adaptasi dari BERT yang telah dilatih secara khusus pada data berbahasa Indonesia. Model yang digunakan adalah **indobenchmark/indobert-base-p1**, salah satu jenis base dengan performa yang solid dalam klasifikasi teks.

Model ini kemudian disesuaikan (*fine-tuned*) menggunakan dataset ulasan untuk menghasilkan prediksi sentimen yang relevan.

Modelling

Split Data

Dataset dibagi menjadi tiga bagian utama:

- Training (70%) : melatih model
- Validation (15%) : mengukur performa saat training
- Testing (15%) : evaluasi akhir model

Tokenization

Teks diproses menggunakan **tokenizer IndoBERT** agar dapat dibaca oleh model. Tokenizer mengubah teks menjadi angka dengan format yang sesuai.

Dataset Preparation

Hasil tokenisasi diorganisasi menjadi format yang dapat digunakan untuk *training* model. Label sentimen (positif & negatif) disiapkan untuk digunakan dalam klasifikasi.

Training

Parameter *training* diatur melalui **TrainingArguments**, seperti jumlah epoch, batch size, dan learning rate. Evaluasi dilakukan setiap epoch dan model terbaik otomatis disimpan.

Modelling

Beberapa model yang akan diterapkan untuk klasifikasi sentimen :

Model 1

Modelling dilakukan menggunakan dataset hasil pre-processing jenis pertama

Model 2

Modelling dilakukan menggunakan dataset hasil pre-processing jenis ketiga dengan epoch 5

Model 3

Modelling dilakukan menggunakan dataset hasil pre-processing jenis ketiga dengan epoch 7

Model 4

Modelling dilakukan menggunakan dataset hasil pre-processing jenis kedua

Keempat model ini nantinya akan dibandingkan untuk mendapatkan model dengan **performa terbaik.**

Model 1

Tokenization

```python
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
    )
```

Teks Asli	Tokenization	Input IDs
aplikasi btn mobile mantap fitur melacak pemasukan pengeluaran transaksipun aman terpercaya	['aplikasi', 'btn', 'mobile', 'mantap', 'fitur', 'melacak', 'pemasukan', 'pengeluaran', 'transaksi', '##pun', 'aman', 'terpercaya']	[2, 1339, 13097, 5872, 7424, 2631, 14011, 14086, 6089, 3108, 453, 1703, 3076, 3]

Training

```python
# 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'
)
```

Hyperparameter	Nilai
Batch Size	32
Learning Rate	3e-5
Epoch	3
Warmup Steps	500
Weight Decay	0.01

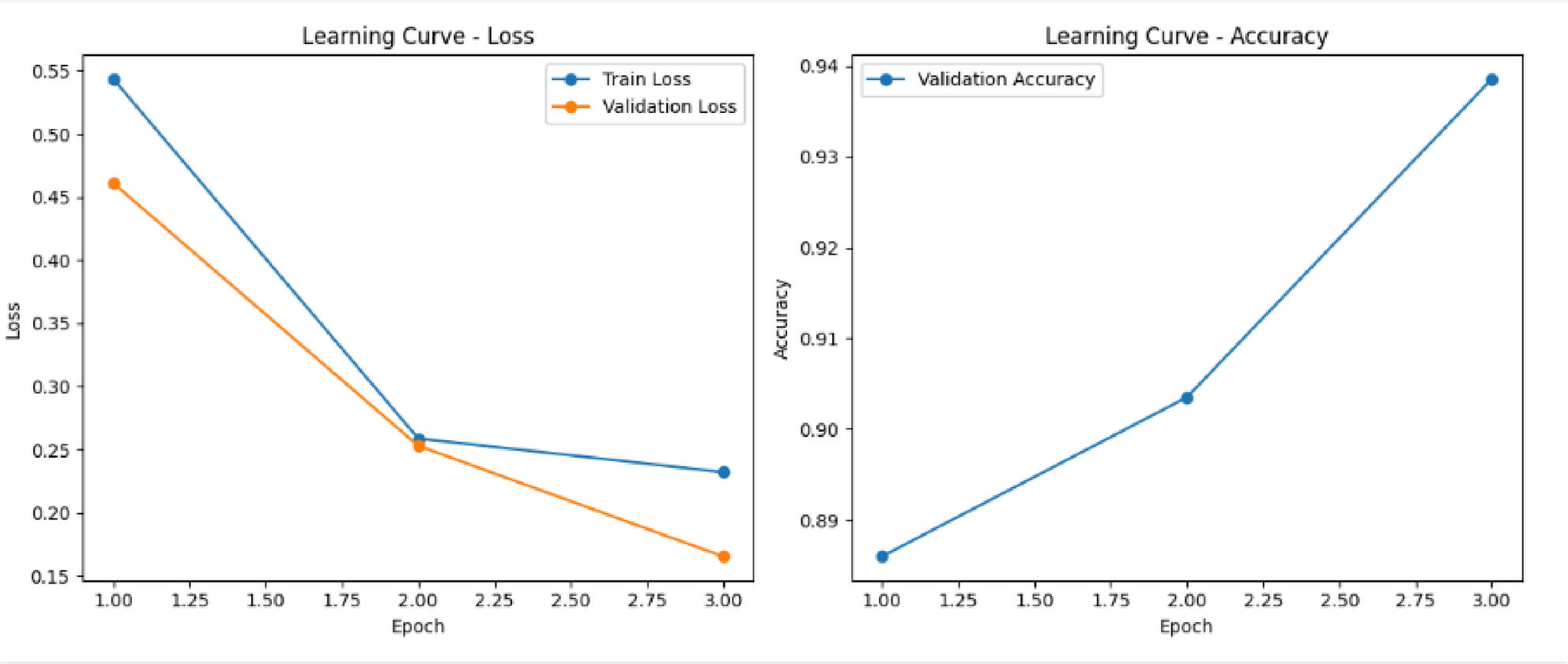
- **Warmup Steps** : Jumlah langkah awal di mana learning rate akan perlahan-lahan meningkat dari 0 ke nilai learning_rate. Digunakan untuk membantu stabilisasi training di proses awal.
- **Weight Decay** : Teknik regularisasi untuk mencegah overfitting dengan menambahkan penalti terhadap besar bobot model.

Training

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	Nilai
global_step	150
training_loss	0.3816
train_runtime	110.3716
train_samples_per_second	43.381
train_steps_per_second	1.359

Model dilatih selama **150 langkah (global step), yang berarti telah dilakukan 150 kali proses pembaruan bobot berdasarkan batch data training**. Selama proses ini, diperoleh **nilai loss sebesar 0.3816, yang menunjukkan seberapa besar kesalahan prediksi model terhadap data latih**—semakin rendah nilainya, semakin baik model dalam mempelajari pola data. **Proses training berlangsung selama sekitar 110 detik**, dengan kecepatan pemrosesan mencapai **43 sampel per detik**, dan **1.36 langkah pembaruan per detik**. Hasil ini menunjukkan bahwa pelatihan berjalan secara efisien, dan model menunjukkan performa yang cukup menjanjikan.



Pada grafik di atas, terlihat bahwa **nilai loss, baik pada data pelatihan maupun validasi, mengalami penurunan seiring bertambahnya epoch**. Hal ini menunjukkan bahwa **model mampu belajar dengan stabil tanpa indikasi overfitting**. Sementara itu, grafik **akurasi validasi menunjukkan tren peningkatan dari sekitar 88.6% menjadi hampir 93.8%**, yang menunjukkan bahwa **performa model terhadap data yang tidak dilatih semakin baik seiring bertambahnya epoch**. Secara keseluruhan, hasil ini mencerminkan proses pelatihan yang efektif dan progresif.

Model 2

Tokenization

```python
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
    )
```

Teks Asli	Tokenization	Input IDs
Aplikasi BTN Mobile mantap apalagi ada fitur yang bisa melacak pemasukan dan pengeluaran jadi transaksipun jadi aman dan terpercaya	['aplikasi', 'btn', 'mobile', 'mantap', 'apalagi', 'ada', 'fitur', 'yang', 'bisa', 'melacak', 'pemasukan', 'dan', 'pengeluaran', 'jadi', 'transaksi', '##pun', 'jadi', 'aman', 'dan', 'terpercaya']	[2, 1339, 13097, 5872, 7424, 2087, 176, 2631, 34, 166, 14011, 14086, 41, 6089, 472, 3108, 453, 472, 1703, 41, 3076, 3]

Training

```
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,
    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,
```

Hyperparameter	Nilai
Batch Size	32
Learning Rate	2e-5
Epoch	5
Warmup Steps	500
Weight Decay	0.01
Eval Steps	10

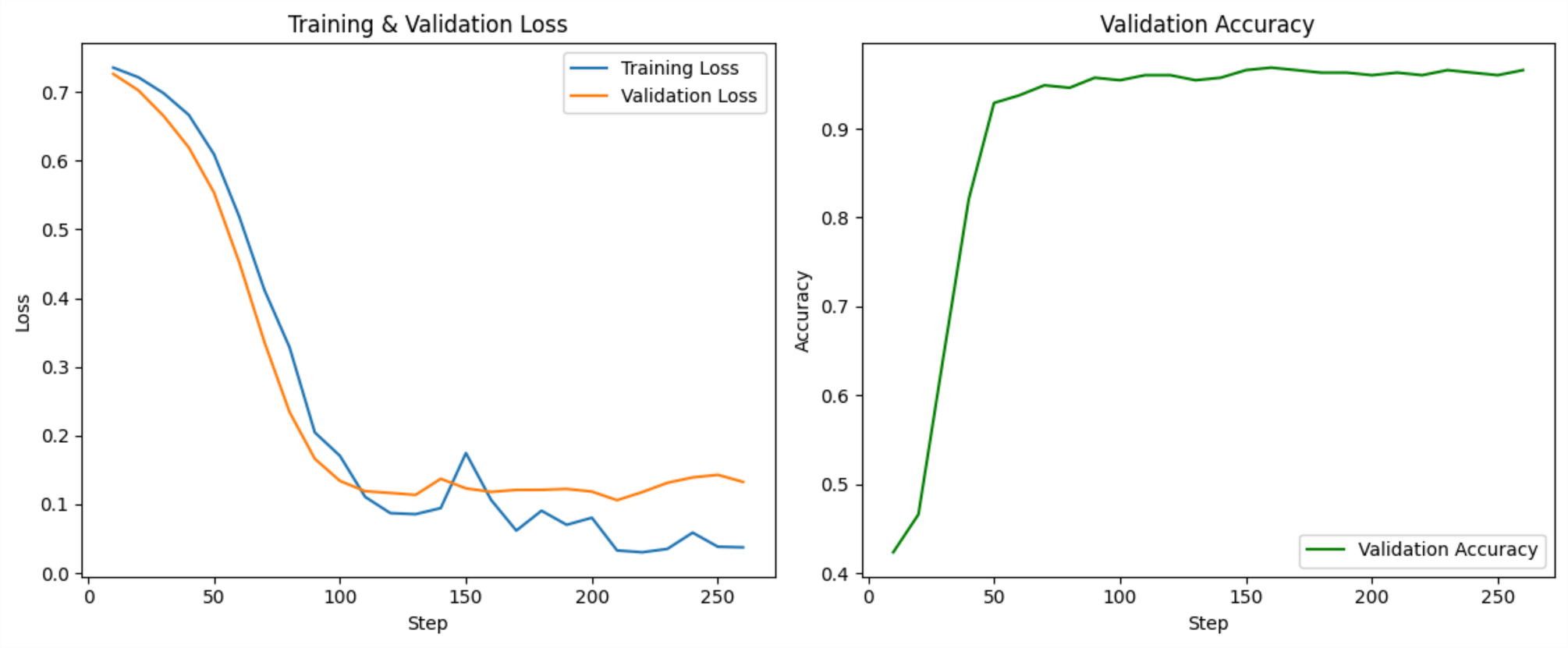
- **Eval Steps** : model akan dievaluasi setiap 10 langkah training untuk memantau performa dan mencegah overfitting secara real-time.

Training

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	0.082100	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
230	0.028200	0.122523	0.965909	0.965116	0.976471	0.954023
240	0.058900	0.122611	0.963068	0.962099	0.976331	0.948276
250	0.027000	0.110703	0.971591	0.971264	0.971264	0.971264
260	0.022100	0.119007	0.968750	0.968116	0.976608	0.959770

TrainOutput	Nilai
global_step	260
training_loss	0.234
train_runtime	1191.419
train_samples_per_second	6.891
train_steps_per_second	0.218

Hasil training menunjukkan **total 260 langkah (global step)** dengan **rata-rata loss sebesar 0.234**. Proses training **berlangsung selama sekitar 20 menit (1191 detik)**, dengan kecepatan sekitar **6.89 sampel per detik** dan **0.218 langkah pembaruan per detik.** Angka-angka ini mengindikasikan proses pelatihan berjalan lancar dengan performa yang stabil.



Pada grafik di atas, terlihat bahwa **nilai loss** pada data pelatihan dan validasi **mengalami penurunan signifikan** seiring bertambahnya langkah epoch, **kemudian stabil pada nilai yang rendah** tanpa adanya lonjakan pada validation loss. Hal ini menunjukkan bahwa **model belajar secara stabil dan tidak mengalami overfitting**. Sementara itu, **grafik akurasi validasi menunjukkan peningkatan pesat hingga 96.8%**, dan **tetap stabil hingga akhir pelatihan.** Secara keseluruhan, hasil ini mencerminkan proses pelatihan yang efektif dan model yang mampu menggeneralisasi dengan baik pada data baru.

Model 3

Tokenization

```python
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
    )
```

Teks Asli	Tokenization	Input IDs
Aplikasi BTN Mobile mantap apalagi ada fitur yang bisa melacak pemasukan dan pengeluaran jadi transaksipun jadi aman dan terpercaya	['aplikasi', 'btn', 'mobile', 'mantap', 'apalagi', 'ada', 'fitur', 'yang', 'bisa', 'melacak', 'pemasukan', 'dan', 'pengeluaran', 'jadi', 'transaksi', '##pun', 'jadi', 'aman', 'dan', 'terpercaya']	[2, 1339, 13097, 5872, 7424, 2087, 176, 2631, 34, 166, 14011, 14086, 41, 6089, 472, 3108, 453, 472, 1703, 41, 3076, 3]

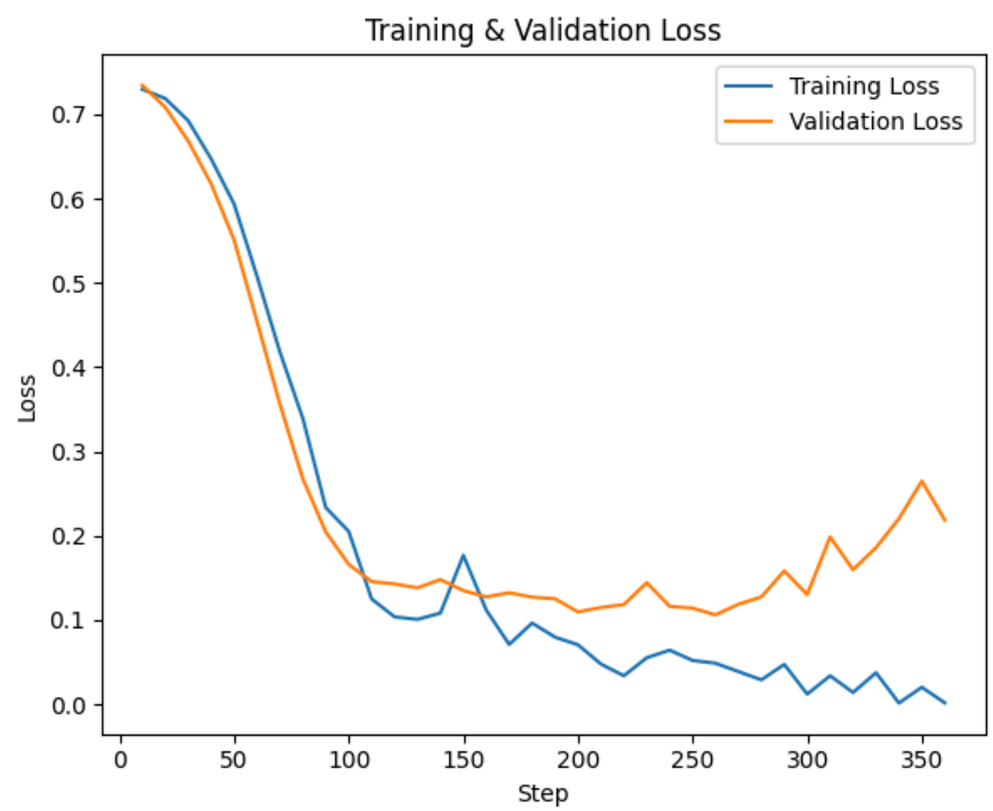
Training

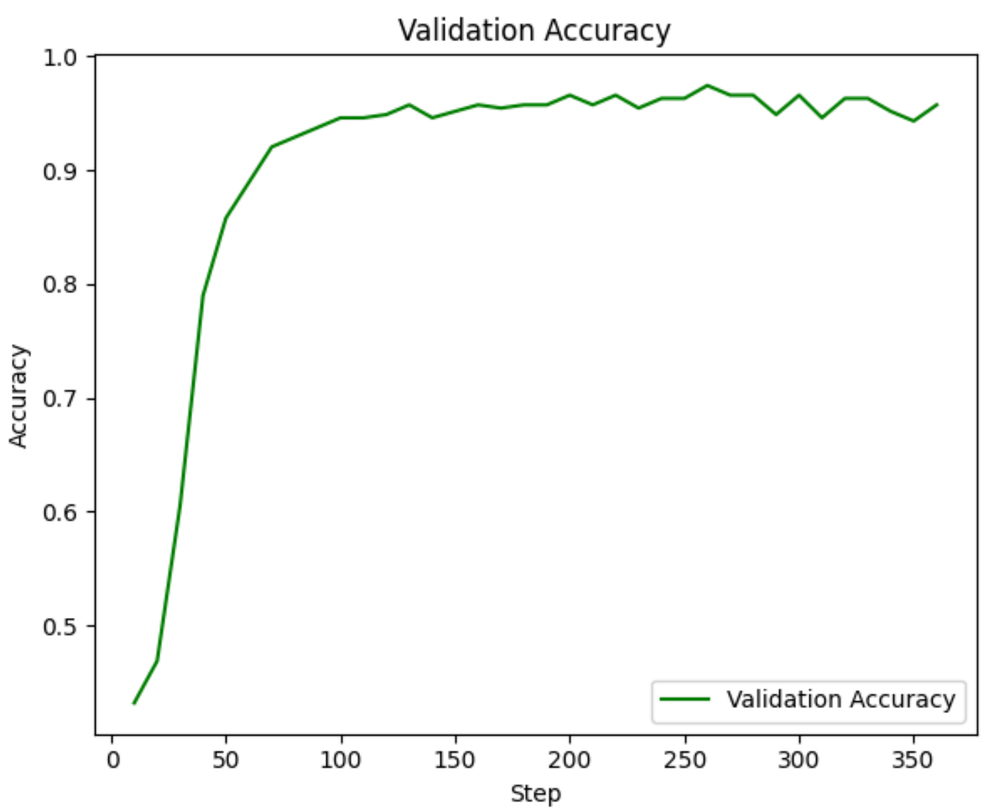
```
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=train_batch_size,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    learning_rate=2e-5,
    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,
```

Hyperparameter	**Nilai**
Batch Size	32
Learning Rate	2e-5
Epoch	7
Warmup Steps	500
Weight Decay	0.01
Eval Steps	10

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
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
310	0.033800	0.198483	0.946023	0.947075	0.918919	0.977011
320	0.014000	0.159626	0.963068	0.962099	0.976331	0.948276
330	0.037400	0.185848	0.963068	0.961877	0.982036	0.942529
340	0.001500	0.220233	0.951705	0.951289	0.948571	0.954023
350	0.020200	0.264704	0.943182	0.943503	0.927778	0.959770
360	0.001700	0.218716	0.957386	0.955752	0.981818	0.931034

TrainOutput	**Nilai**
global_step	360
training_loss	0.183
train_runtime	1665.354
train_samples_per_second	6.902
train_steps_per_second	0.219





Pada grafik di atas, terlihat bahwa **nilai loss** pada data pelatihan dan validasi **mengalami penurunan yang cukup tajam pada awal pelatihan**. Namun, **validation loss mulai menunjukkan sedikit kenaikan dan fluktuasi**. Kenaikan ini bisa menjadi **indikasi awal terjadinya overfitting**, di mana model mulai terlalu menyesuaikan diri dengan data pelatihan dan performanya pada data validasi sedikit menurun. Sementara itu, **grafik akurasi validasi menunjukkan peningkatan pesat hingga 95.7%** dan **tetap stabil hingga akhir pelatihan**. Secara keseluruhan, hasil ini mencerminkan model yang efektif dan mampu menggeneralisasi dengan baik, namun perlu perhatian lebih untuk menghindari overfitting pada tahap pelatihan selanjutnya.

Model 4

Tokenization

```python
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
    )
```

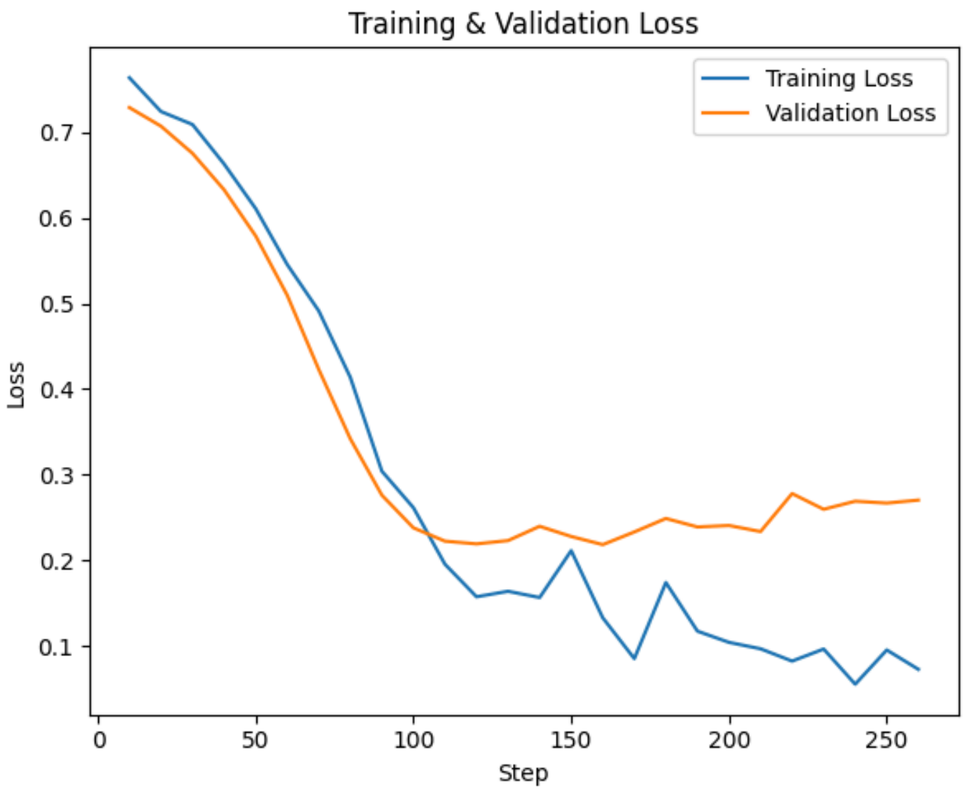
Teks Asli	Tokenization	Input IDs
mantap fitur melacak pemasukan pengeluaran transaksipun aman terpercaya	['mantap', 'fitur', 'melacak', 'pemasukan', 'pengeluaran', 'transaksi', '##pun', 'aman', 'terpercaya']	[2, 7424, 2631, 14011, 14086, 6089, 3108, 453, 1703, 3076, 3]

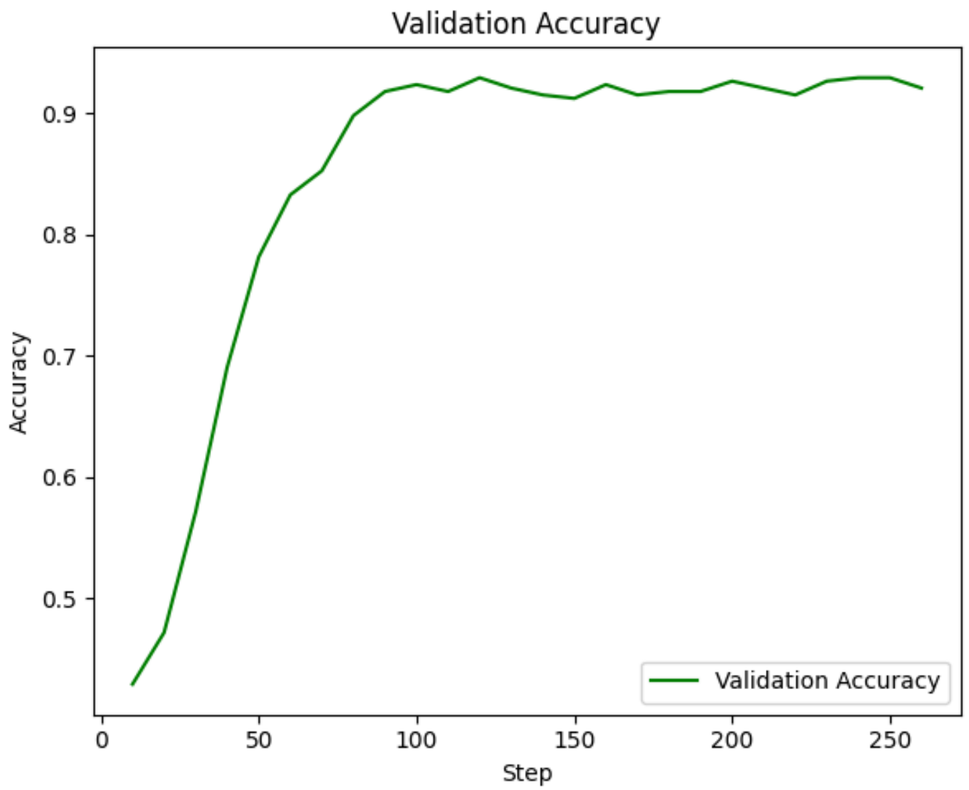
Training

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	Nilai
global_step	260
training_loss	0.2877
train_runtime	407.0406
train_samples_per_second	20.17
train_steps_per_second	0.639

Hasil training menunjukkan **total 260 langkah (global step)** dengan **rata-rata loss sebesar 0.2877**. Proses training **berlangsung selama sekitar 6 menit (407 detik)**, dengan kecepatan sekitar **20.17 sampel per detik** dan **0.639 langkah pembaruan per detik.** Angka-angka ini mengindikasikan proses pelatihan berjalan lancar dengan performa yang stabil.





Pada grafik di atas, terlihat bahwa **nilai loss** pada data pelatihan dan validasi **mengalami penurunan menurun secara signifikan terutama selama 100 langkah pertama.** Kemudian, t**raining loss terus turun** sementara **validation loss stabil dengan sedikit fluktuasi** tanpa lonjakan tajam, menandakan **model belajar konsisten tanpa overfitting**. Sementara itu, **grafik akurasi validasi menunjukkan peningkatan hingga 93%** dan **tetap stabil hingga akhir pelatihan meskipun terdapat sedikit variasi, namun tetap pada tingkat yang tinggi**. Secara keseluruhan, hasil ini menggambarkan proses pelatihan yang berjalan efektif dan model yang mampu melakukan generalisasi dengan baik pada data validasi.

Evaluasi

Precision

Mengukur jumlah sentimen yang berhasil diprediksi dengan benar dari seluruh prediksi yang dibuat untuk sentimen tersebut.

$$P = \frac{tp}{tp + fp}$$

Recall

Mengukur jumlah sentimen yang berhasil diprediksi dengan benar dari semua sentimen sebenarnya.

$$R = \frac{tp}{tp + fn}$$

F1-Score

Weighted average dari Precision dan Recall.

$$F_1 = F = 2 * \frac{P * R}{P + R}$$

Accuracy

Mengukur seberapa banyak prediksi model yang benar dibandingkan dengan seluruh prediksi yang dilakukan, tanpa membedakan kelas.

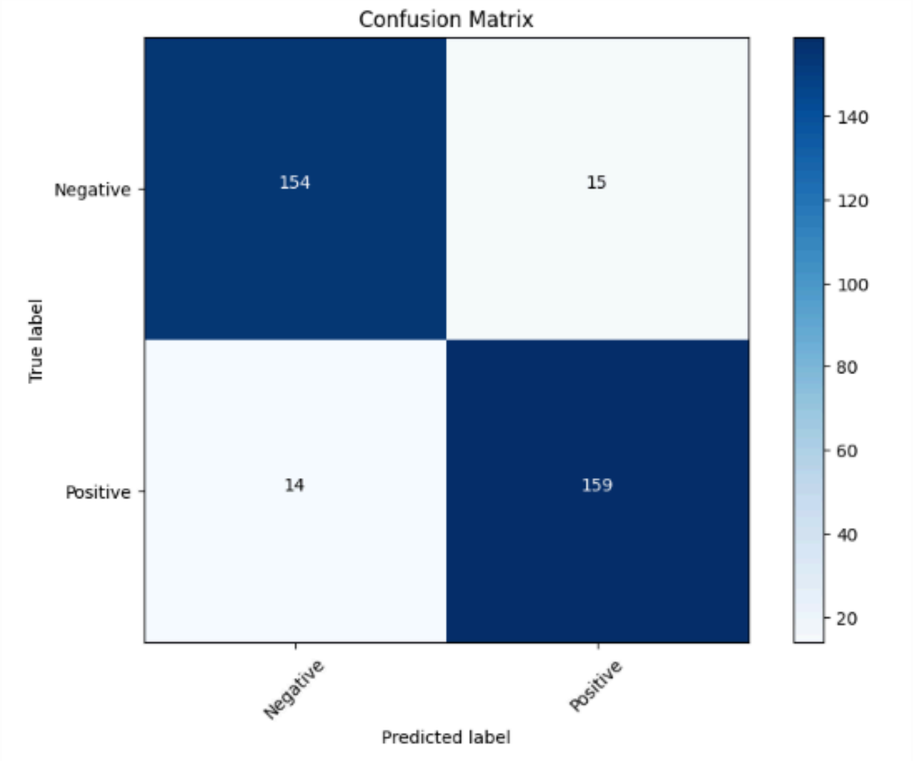
$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

Model 1

```
=== 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
```



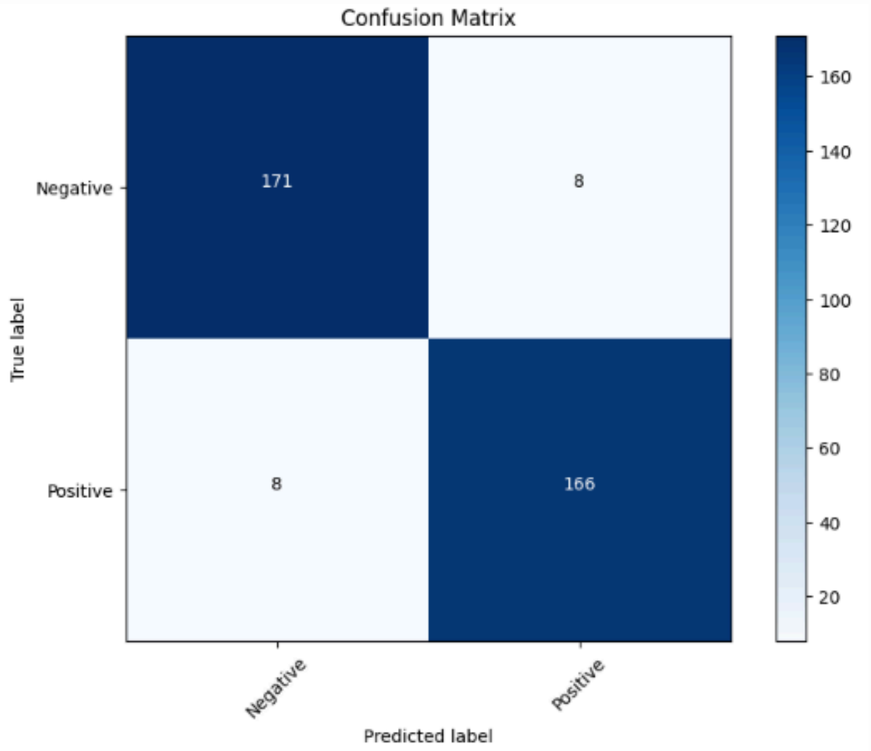
- **Precision** untuk kelas negatif adalah 0.92, hal ini menunjukkan bahwa **dari semua prediksi negatif, 92% benar-benar negatif**. Sementara, **untuk prediksi positif, 91% benar**. Model sedikit lebih percaya diri dalam memprediksi negatif.
- **Recall** untuk kelas negatif adalah 0.91, hal ini menunjukkan bahwa **dari semua data negatif yang seharusnya terdeteksi, 91% berhasil ditemukan**. Sementara **untuk positif, 92% berhasil ditemukan**. Model sedikit lebih sensitif terhadap data positif.
- **F1-score** menunjukkan nilai 0.91 untuk negatif dan 0.92 untuk positif, yang merupakan harmonisasi antara precision dan recall. Artinya, **performa model dalam menangani kedua kategori sangat seimbang**.
- **Akurasi** total model adalah 0.92, yang menunjukkan bahwa **92% data diklasifikasikan dengan benar** sehingga dapat dikatakan **model memiliki performa yang baik**.
- Untuk **kelas negatif**, model berhasil mengklasifikasikan **154** ulasan dengan benar (**True Negative**), sedangkan **15** ulasan keliru diprediksi sebagai positif (**False Positive**).
- Untuk **kelas positif**, model mengenali **159** ulasan dengan benar (**True Positive**) dan hanya salah pada **14** ulasan yang terprediksi sebagai negatif (**False Negative**).

Model 2

```
=== 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
```



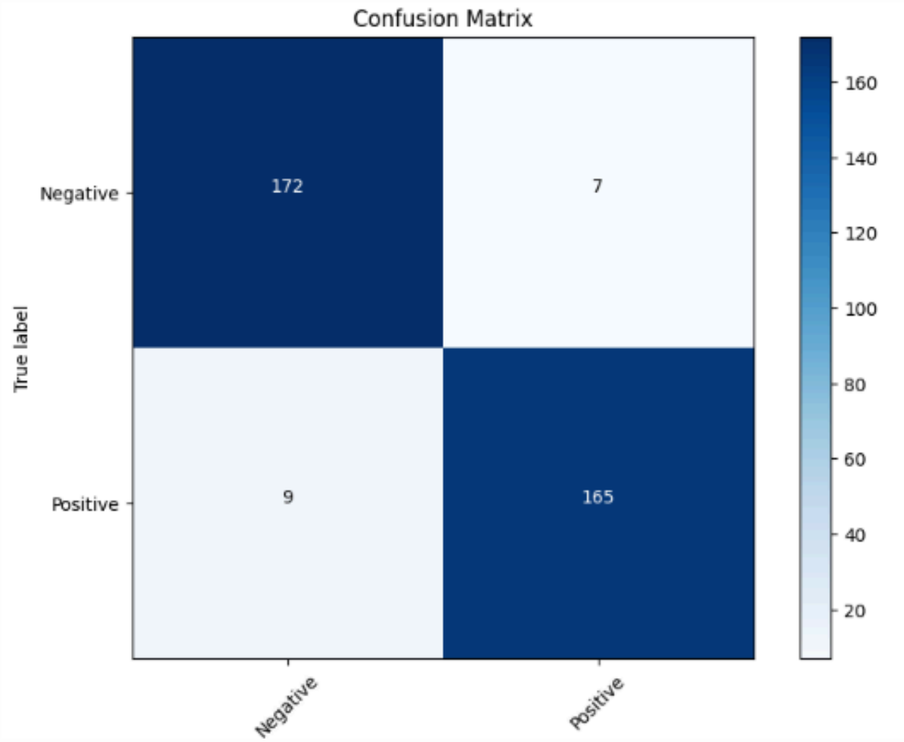
- **Precision** untuk kelas negatif adalah 0.96, hal ini menunjukkan bahwa **dari semua prediksi negatif, 96% benar-benar negatif**. Sementara, **untuk prediksi positif, 92% benar**. Model 2 juga sedikit lebih percaya diri dalam memprediksi negatif.
- **Recall** untuk kelas negatif adalah 0.96, hal ini menunjukkan bahwa **dari semua data negatif yang seharusnya terdeteksi, 96% berhasil ditemukan**. Sementara **untuk positif, 95% berhasil ditemukan**. Model sedikit lebih sensitif terhadap data negatif.
- **F1-score** menunjukkan nilai 0.96 untuk negatif dan 0.95 untuk positif, yang merupakan harmonisasi antara precision dan recall. Artinya, **performa model dalam menangani kedua kategori sangat seimbang**.
- **Akurasi** total model adalah 0.95, yang menunjukkan bahwa **95% data diklasifikasikan dengan benar** sehingga dapat dikatakan **model memiliki performa yang sangat baik**.
- Untuk **kelas negatif**, model berhasil mengklasifikasikan **171** ulasan dengan benar (**True Negative**), sedangkan **8** ulasan keliru diprediksi sebagai positif (**False Positive**).
- Untuk **kelas positif**, model mengenali **166** ulasan dengan benar (**True Positive**) dan hanya salah pada **8** ulasan yang terprediksi sebagai negatif (**False Negative**).

Model 3

```
=== 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
```



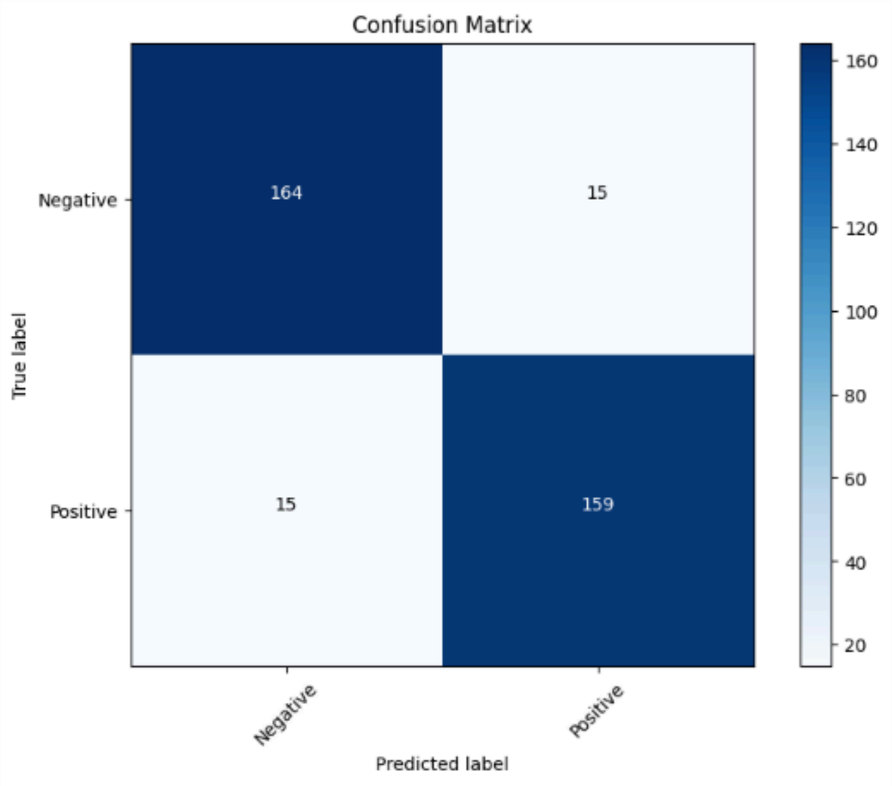
- **Precision** untuk kelas negatif adalah 0.95, hal ini menunjukkan bahwa **dari semua prediksi negatif, 95% benar-benar negatif**. Sementara, **untuk prediksi positif, 96% benar**. Dengan penambahan 2 epoch, model sedikit lebih percaya diri dalam memprediksi positif.
- **Recall** untuk kelas negatif adalah 0.96, hal ini menunjukkan bahwa **dari semua data negatif yang seharusnya terdeteksi, 96% berhasil ditemukan**. Sementara **untuk positif, 95% berhasil ditemukan**. Penambahan epoch tidak berpengaruh dan model tetap sedikit lebih sensitif terhadap data negatif.
- **F1-score** menunjukkan nilai 0.96 untuk negatif dan 0.95 untuk positif, yang merupakan harmonisasi antara precision dan recall. Artinya, **performa model dalam menangani kedua kategori sangat seimbang**.
- **Akurasi** total model adalah 0.95, yang menunjukkan bahwa **95% data diklasifikasikan dengan benar** sehingga dapat dikatakan **model memiliki performa yang baik**.
- Untuk **kelas negatif**, model berhasil mengklasifikasikan **172** ulasan dengan benar (**True Negative**), sedangkan **7** ulasan keliru diprediksi sebagai positif (**False Positive**).
- Untuk **kelas positif**, model mengenali **165** ulasan dengan benar (**True Positive**) dan hanya salah pada **9** ulasan yang terprediksi sebagai negatif (**False Negative**).

Model 4

```
=== 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
```



- **Precision** untuk kelas negatif adalah 0.92, hal ini menunjukkan bahwa **dari semua prediksi negatif, 92% benar-benar negatif**. Sementara, **untuk prediksi positif, 91% benar**. Model sedikit lebih percaya diri dalam memprediksi negatif.
- **Recall** untuk kelas negatif adalah 0.92, hal ini menunjukkan bahwa **dari semua data negatif yang seharusnya terdeteksi, 92% berhasil ditemukan**. Sementara **untuk positif, 91% berhasil ditemukan**. Dengan data yang melalui penyempurnaan pre-processing, model sedikit lebih sensitif terhadap data negatif.
- **F1-score** menunjukkan nilai 0.92 untuk negatif dan 0.91 untuk positif, yang merupakan harmonisasi antara precision dan recall. Artinya, **performa model dalam menangani kedua kategori sangat seimbang**.
- **Akurasi** total model adalah 0.92, yang menunjukkan bahwa **92% data diklasifikasikan dengan benar** sehingga dapat dikatakan **model memiliki performa yang baik**.
- Untuk **kelas negatif**, model berhasil mengklasifikasikan **164** ulasan dengan benar (**True Negative**), sedangkan **15** ulasan keliru diprediksi sebagai positif (**False Positive**).
- Untuk **kelas positif**, model mengenali **159** ulasan dengan benar (**True Positive**) dan hanya salah pada **15** ulasan yang terprediksi sebagai negatif (**False Negative**).

Kesimpulan

01

Sentimen pengguna terhadap aplikasi Balé by BTN menunjukkan persebaran yang cukup seimbang, dengan sedikit dominasi ulasan negatif (50,7%) dibandingkan ulasan positif (49,3%). Hal ini mencerminkan bahwa pengalaman pengguna bersifat beragam, dengan adanya sejumlah keluhan namun juga apresiasi terhadap aplikasi.

02

Kata-kata yang sering muncul dalam ulasan memperkuat gambaran sentimen tersebut. **Ulasan negatif didominasi oleh kata seperti "masuk", "gagal", "login", dan "susah"**, yang menggambarkan keluhan pengguna terhadap masalah teknis saat penggunaan. Sementara itu, **ulasan positif banyak memuat kata seperti "mudah", "bagus", "cepat", dan "membantu"**, yang menunjukkan kepuasan terhadap kemudahan dan manfaat aplikasi

03

IndoBERT menunjukkan performa unggul dalam analisis sentimen ulasan aplikasi, dengan keempat model mencapai **akurasi di atas 91% serta nilai precision, recall, dan f1-score yang tinggi**. Hal ini menegaskan bahwa IndoBERT mampu memahami konteks bahasa Indonesia secara efektif dalam tugas analisis sentimen. Di antara model yang diuji, **Model 2 menjadi yang terbaik dengan akurasi tertinggi sebesar 95% dan f1-score 0.95–0.96**. Learning curve-nya pun stabil tanpa indikasi overfitting, menandakan proses pelatihan yang optimal.

REFERENSI

Dalianis, H. (2018). Evaluation Metrics and evaluation. In Springer eBooks (pp. 45–53).

Wijaya, D. R. G. (2023). *Penerapan Analisis Sentimen dan Pendeteksian Topik pada Ulasan Pengguna Aplikasi MyPertamina di Play Store*. University of Indonesia.

Link Google Colaboratory:
https://colab.research.google.com/drive/1191akBfZSccSXdX61W7OGk433GwZGxi?usp=sharing

THANK YOU!