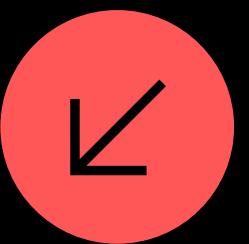


Sentiment Analysis on Political Posts in X/Twitter

Weekly Assignment 2

GENESIS

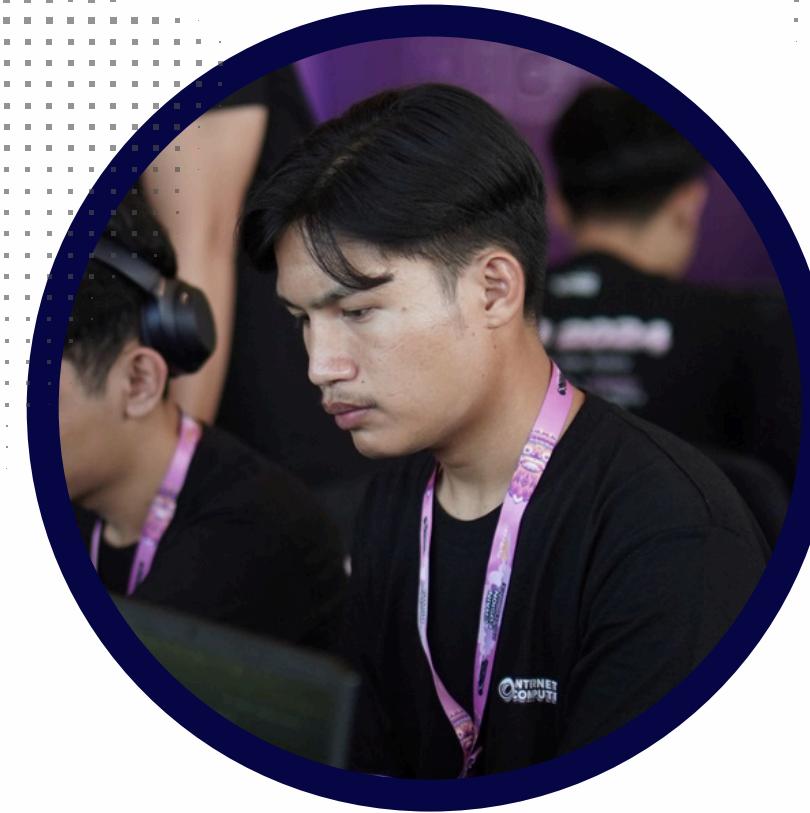




Muhammad Zuama



Juan Hendy Irmanto



Rizky Ahsan Syarief



Samuel David Sutanto



Jihan Salma
Ramadhanti Widodo



Mohammad Arief
Rajendra

LATAR BELAKANG

Pemilu Presiden 2024 memicu diskusi intens di platform X (Twitter), di mana masyarakat aktif menyuarakan opini politik melalui tweet dan komentar. Analisis sentimen digunakan untuk memahami persepsi publik terhadap calon presiden dan isu politik.

Penelitian oleh Rizki dkk. (2024) menunjukkan bahwa algoritma Naive Bayes mampu mengklasifikasikan sentimen secara efektif, mengungkap pola dan tren opini publik selama masa Pilpres. Temuan ini menunjukkan potensi media sosial sebagai cerminan dinamika politik masyarakat.



Tujuan



- Memberikan wawasan kuantitatif mengenai proporsi sentimen yang berbeda dalam diskusi politik di aplikasi X.
- Menemukan model klasifikasi untuk menganalisis sentimen dengan Performa terbaik.

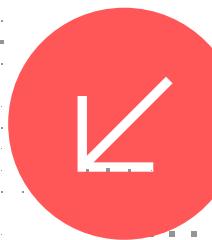
Susunan Project

- Mengumpulkan 1.815 data terkait topik politik yang relevan dari aplikasi X yang telah disediakan oleh tim Indonesia AI
- Mencari dan membandingkan model klasifikasi sentimen dengan akurasi terbaik dalam mengidentifikasi sentimen positif, negatif, dan netral.
- Menghasilkan visualisasi data yang komprehensif dan mudah dipahami untuk menyajikan hasil analisis sentimen, seperti grafik distribusi sentimen dan confusion matrix
- Menyusun laporan analisis sentimen yang merinci temuan utama, implikasi, dan rekomendasi berdasarkan data yang diperoleh.



Business & Data Understanding

Business & Data Understanding



Data yang digunakan adalah tweet.csv yang disediakan oleh Indonesia AI, yang menyatukan berbagai post di X/Twitter mengenai politik Indonesia.

Jumlah Observasi: **1.815 observasi**

Jumlah Fitur: **2 variabel, yang termasuk sentimen dan post/tweet**

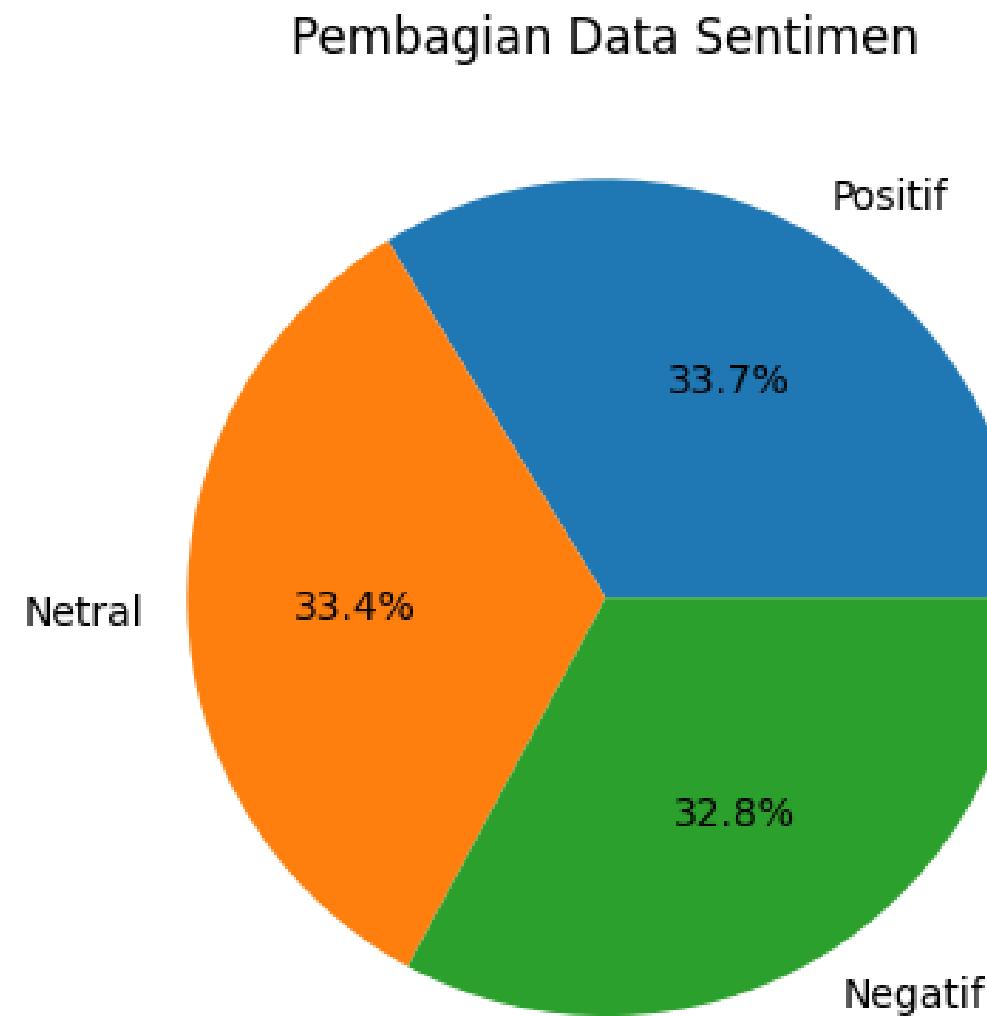
Business & Data Understanding

Fitur-fitur yang dapat ditemukan dalam dataset meliputi:

- *Tweet*: konten dari post pada X/Twitter mengenai politik Indonesia
- *Sentimen*: klasifikasi sentimen setiap post yang terkandung dalam variabel *Tweet*, dibagi menjadi kategori positif, netral, dan negatif

Karena data text memiliki perbedaan dalam konten dan format, preprocessing harus dilakukan untuk menyelaraskan data yang dimasukan kedalam algoritma machine learning.

Business & Data Understanding



Distribusi data variabel *Sentimen* cukup seimbang, karena setiap kategori setidaknya merepresentasikan 33% dari observasi dalam dataset tersebut.

Data cleaning & Analysis

Data Cleaning & Analysis

```
data = data.drop(columns=['Unnamed: 0'])
data

  sentimen      tweet
0    negatif  Kata @prabowo Indonesia tidak dihargai bangsa ...
1     netral  Batuan Langka, Tasbih Jokowi Hadiah dari Habib...
2     netral  Di era Jokowi, ekonomi Indonesia semakin baik....
3    positif  Bagi Sumatera Selatan, Asian Games berdampak p...
4    negatif  Negara kita ngutang buat bngun infrastruktur y...
...        ...
1810    netral  Negarawan sejati sll bangga dan mengedepankan ...
1811    netral  1. HRS ceramah di Damai Indonesiaku 2. Perekon...
1812    netral  Mari bangun bangsa dgn mendukung perekonomian ...
1813    netral  Bantu majukan perekonomian bangsa bersama Pak ...
1814    netral  Pak @jokowi mengubah cara pandang ekonomi. Kin...
1815 rows x 2 columns

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1815 entries, 0 to 1814
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   sentimen    1815 non-null   object 
 1   tweet       1815 non-null   object 
dtypes: object(2)
memory usage: 28.5+ KB
```

Dalam Data Cleaning dan Analisis Dilakukan beberapa langkah

- **Cek Nilai Kosong**
- **Cek Nilai Duplikat**
- **Cek Keseimbangan Data**
- **Hapus Kolom yang Tidak Perlu**

Data Cleaning & Analysis

```
import re
import pandas as pd
import numpy as np
import random
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

def preprocessing(text):
    # case folding
    text = text.lower()

    # remove punctuation and non-alphabetic characters
    text = re.sub(r'[^w\s]', '', text)

    # Menghapus link menggunakan regex
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)

    # Menghapus hashtag menggunakan regex
    text = re.sub(r'#\S+', '', text)

    # remove numbers
    text = re.sub(r'\d+', '', text)

    # stopword removal
    factory = StopWordRemoverFactory()
    stopwords = factory.get_stop_words()
    words = text.split()
    text = " ".join([word for word in words if word not in stopwords])

    return text

data["tweet"] = data["tweet"].apply(preprocessing)
data

sentimen
0      negatif      kata prabowo indone
```



```
# Contoh data
# data = pd.read_csv('tweet.csv') # kalau sudah ada data csv

# Inisialisasi stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Integrasi tqdm dengan pandas apply
tqdm.pandas(desc="Stemming progress")

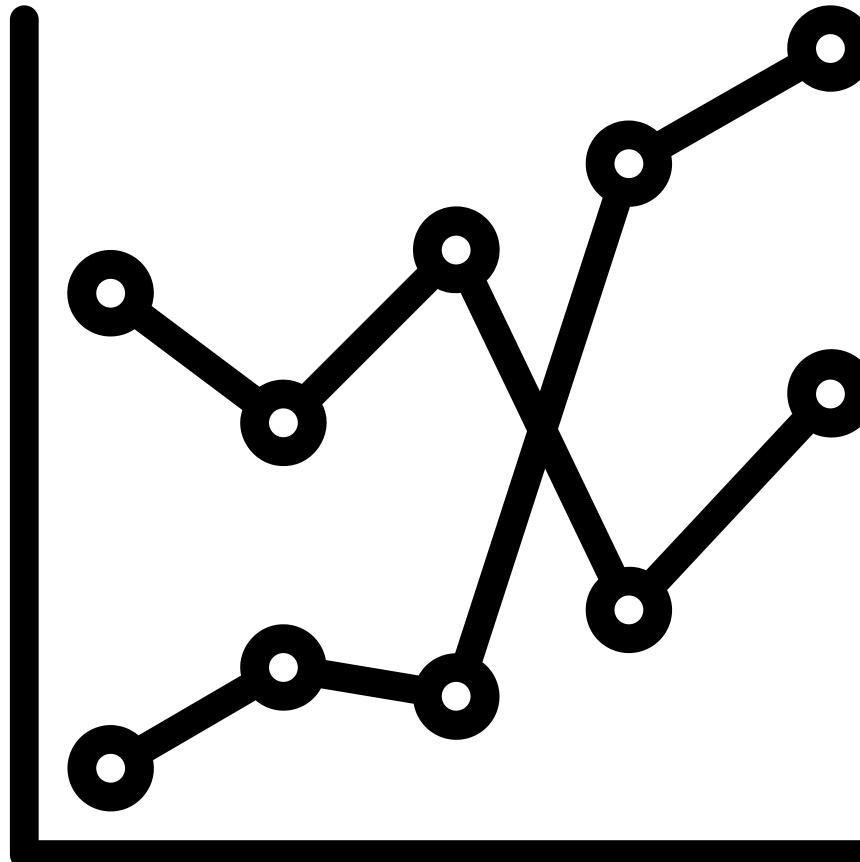
# Terapkan stemming dengan progress bar, pastikan input string
data["tweet"] = data["tweet"].astype(str).progress_apply(lambda x: stemmer.stem(x))

print(data)
```

- Menggunakan NLTK dan Sastrawi untuk menghapus stopword dan melakukan stemming bahasa Indonesia.
- Hasil akhirnya disimpan dalam kolom baru bernama `text_shallow` yang digunakan sebagai input untuk algoritma shallow learning.
- Melakukan vektorisasi TF-IDF untuk mengubah `text_shallow` menjadi fitur numerik (X), dan dipasangkan dengan label sentimen (y) untuk pelatihan model.

Data Modeling

Data Modeling



Shallow Learning Methods:

- Logistic Regression
- Multinomial Naive Bayes (MultinomialNB)
- Support Vector Machine (SVM)
- Random Forest

Deep Learning Methods:

- GRU
- LSTM

Data Modeling

```
from tensorflow.keras.models import Sequential    Import "tensorflow.keras.models" could not be resolved
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout    Import "tensorflow.keras.layers" could not be resolved
from tensorflow.keras.optimizers import Adam    Import "tensorflow.keras.optimizers" could not be resolved

model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=64, input_length=max_length),
    LSTM(128, return_sequences=True),
    LSTM(64),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax') # Output layer for multi-class classification
])

# Define learning rate
learning_rate = 0.001

# Compile the model with the specified learning rate
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate=learning_rate), metrics=['accuracy'])

# Display the model summary
model.summary()
✓ 0.1s

WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs slowly on M1/M2 Macs, please use the legacy Keras optimizer instead.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on M1/M2 Macs. Falling back to the legacy Keras optimizer.
Model: "sequential_5"

Layer (type)          Output Shape         Param #
=====
embedding_5 (Embedding)    (None, 30, 64)      384000
lstm_10 (LSTM)           (None, 30, 128)     98816
lstm_11 (LSTM)           (None, 64)          49408
dense_10 (Dense)          (None, 128)          8320
dropout_5 (Dropout)        (None, 128)          0
dense_11 (Dense)          (None, 3)            387
=====
Total params: 540931 (2.06 MB)
Trainable params: 540931 (2.06 MB)
Non-trainable params: 0 (0.00 Byte)
```

```
from tensorflow.keras.models import Sequential    Import "tensorflow.keras.models" could not be resolved
from tensorflow.keras.layers import Embedding, GRU, Dense, Dropout    Import "tensorflow.keras.layers" could not be resolved
# Define the model architecture using the hyperparameters
model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=128, input_length=max_length),    IndentationError: unexpected indent
    GRU(256, return_sequences=True),
    GRU(128),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax') # Output layer for multi-class classification
])

# Compile the model
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(learning_rate=0.001),
    metrics=['accuracy']
)

# Display the model summary
model.summary()
✓ 0.1s

WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs slowly on M1/M2 Macs, please use the legacy Keras optimizer instead.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on M1/M2 Macs. Falling back to the legacy Keras optimizer.
Model: "sequential_8"

Layer (type)          Output Shape         Param #
=====
embedding_8 (Embedding)    (None, 30, 128)     768000
gru_16 (GRU)             (None, 30, 256)     296448
gru_17 (GRU)             (None, 128)          148224
dense_16 (Dense)          (None, 256)          33024
dropout_8 (Dropout)        (None, 256)          0
dense_17 (Dense)          (None, 3)            771
=====
Total params: 1246467 (4.75 MB)
Trainable params: 1246467 (4.75 MB)
Non-trainable params: 0 (0.00 Byte)
```

Arsitektur Model LSTM

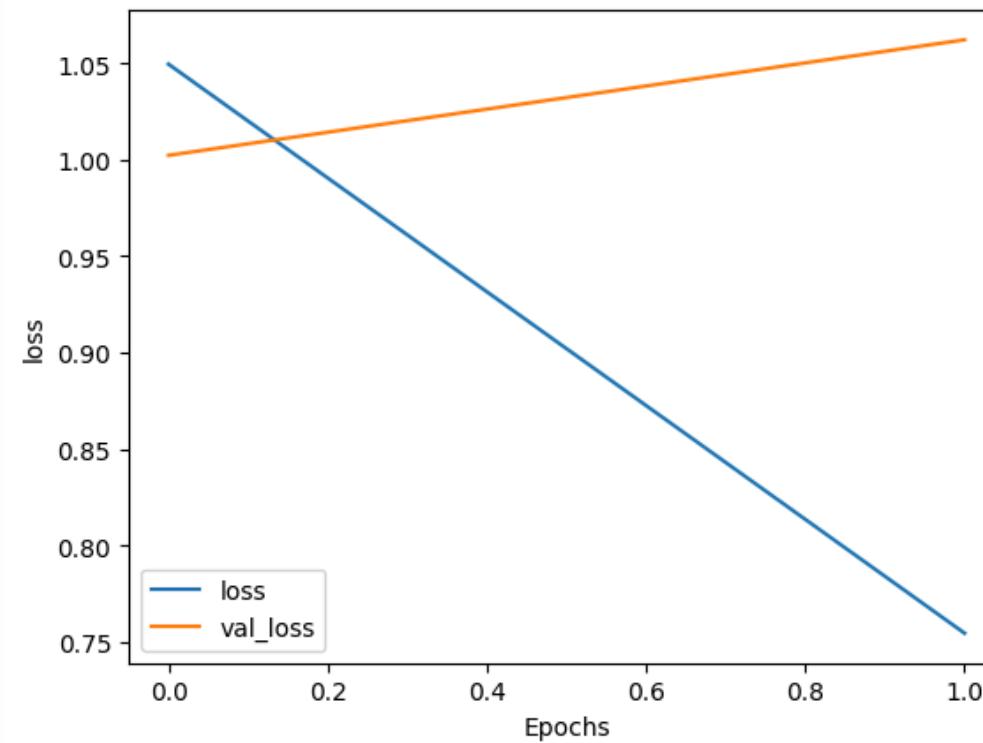
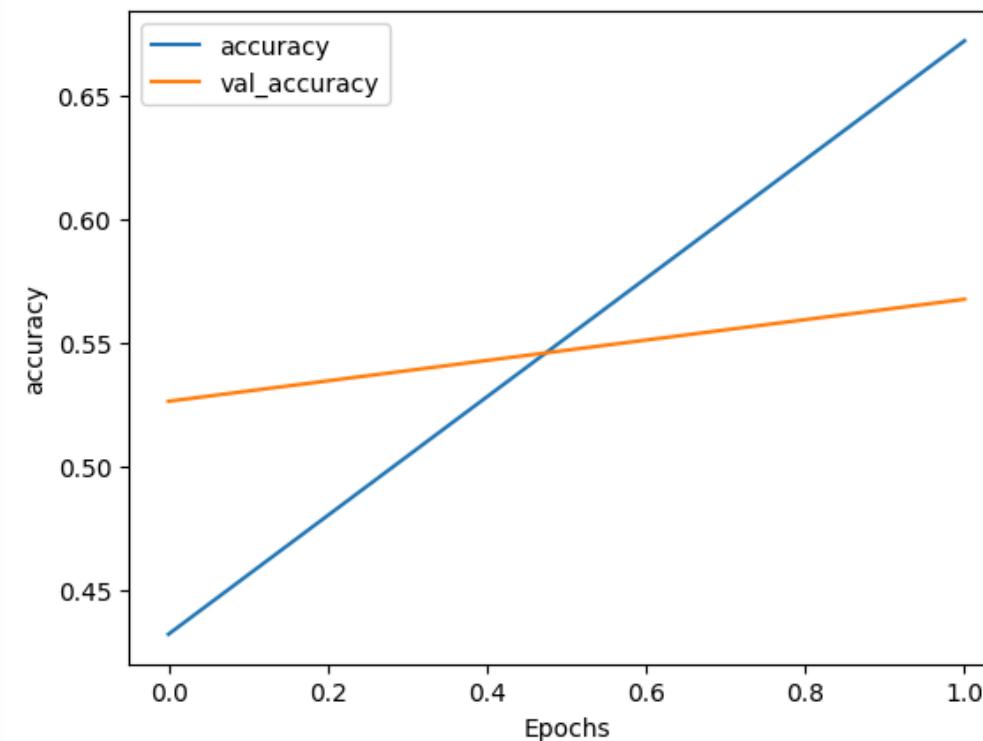
Arsitektur Model GRU

Model Training and Evaluation 2

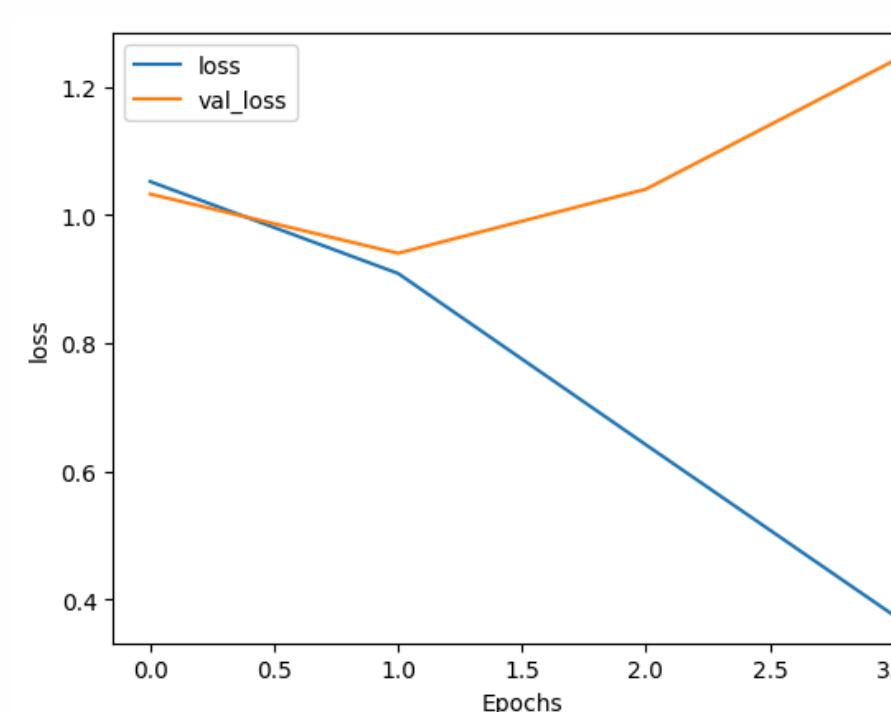
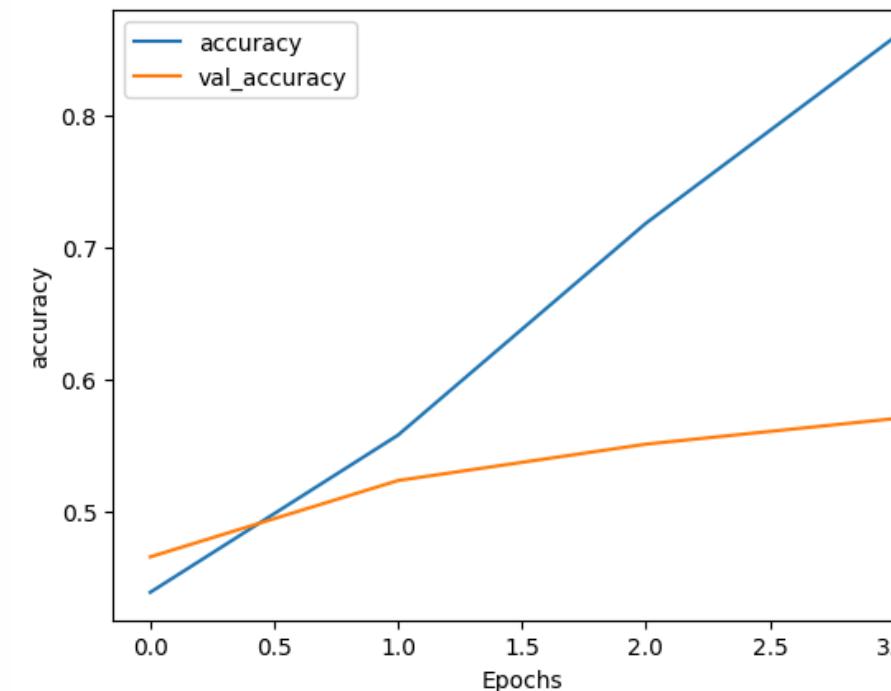
Perbandingan Hasil Train dan Evaluasi Model

	Model	Accuracy	Precision	Recall	F1 Score
0	Naive Bayes	0.6419	0.6509	0.6457	0.6417
1	SVM	0.6143	0.6178	0.6143	0.6117
2	LSTM	0.5702	0.5724	0.5702	0.571
3	GRU	0.5675	0.5813	0.5675	0.5469

Model Evaluation



LSTM

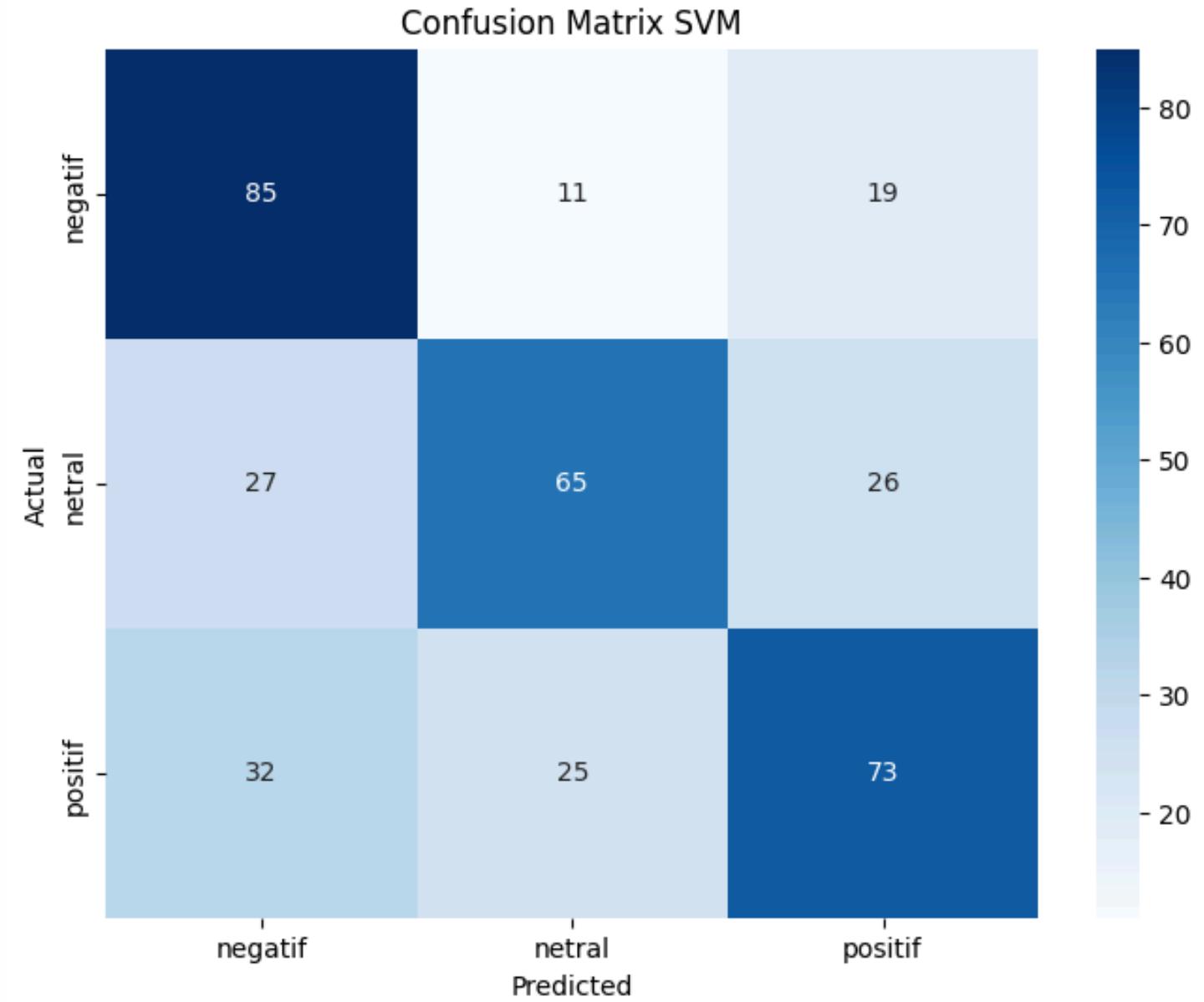


GRU

Namun, berdasarkan figur training dan loss dari model deep learning, terlihat bahwa masih terjadi **overfitting**.

Ini tetap terjadi saat menggunakan data *clean* maupun *raw* dari kolom tweet.

Confussion Matrik



Dalam confussion matrik di samping

Baris 1 (Actual: Negatif):

- ✓ 85 diprediksi Negatif dengan benar (True Negative)
- ✗ 11 salah diprediksi sebagai Netral
- ✗ 19 salah diprediksi sebagai Positif

Baris 2 (Actual: Netral):

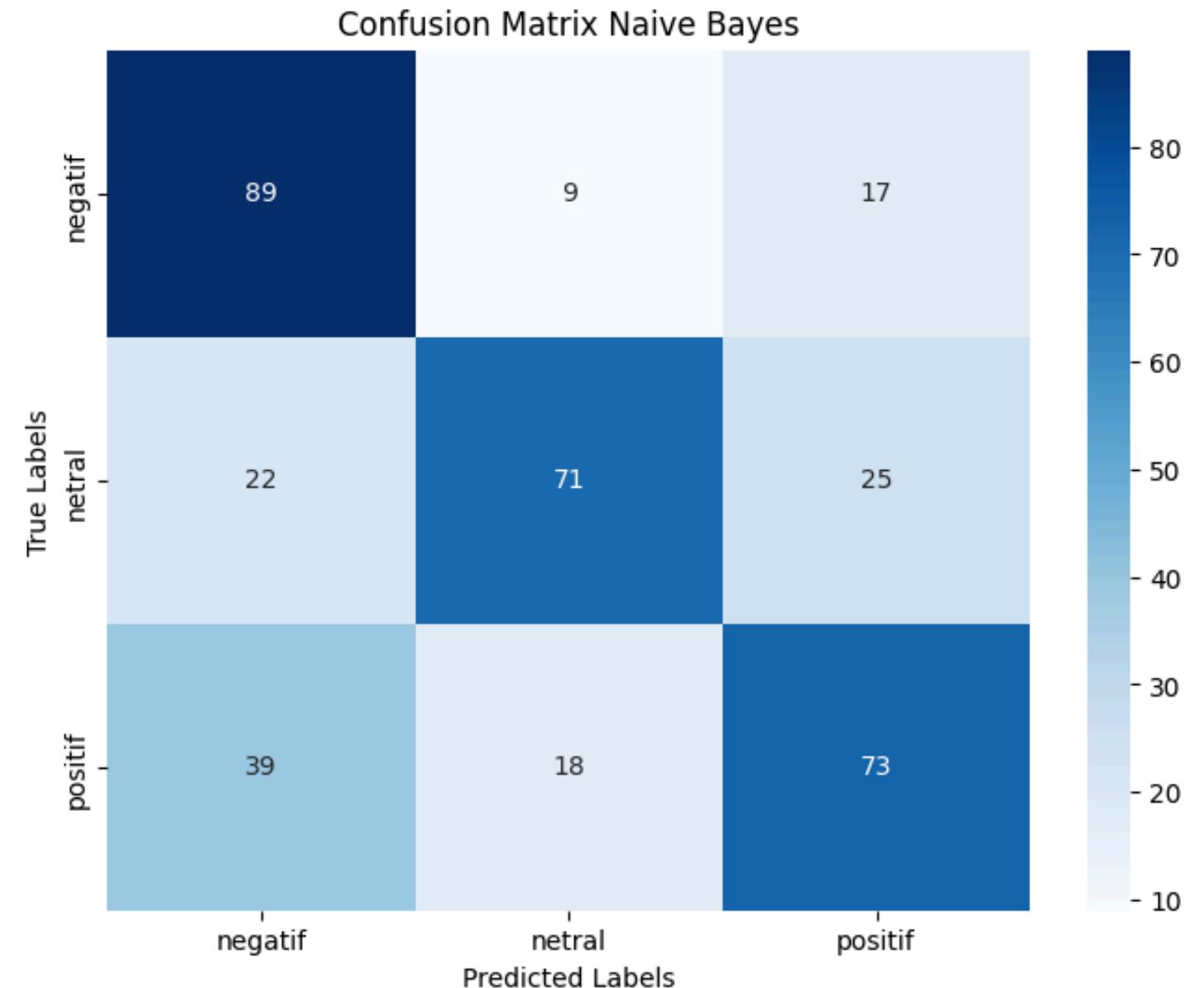
- ✗ 27 salah diprediksi sebagai Negatif
- ✓ 65 diprediksi Netral dengan benar (True Netral)
- ✗ 26 salah diprediksi sebagai Positif

Baris 3 (Actual: Positif):

- ✗ 32 salah diprediksi sebagai Negatif
- ✗ 25 salah diprediksi sebagai Netral
- ✓ 73 diprediksi Positif dengan benar (True Positive)

Model menunjukkan performa yang baik dalam mengklasifikasikan kelas Negatif dan Positif, namun masih kesulitan membedakan kelas Netral yang sering tertukar dengan dua kelas lainnya. Oleh karena itu, diperlukan peningkatan akurasi khusus pada kelas Netral agar hasil klasifikasi lebih seimbang.

Confussion Matrik



Dalam confussion matrik di samping

Baris 1 (Actual: Negatif):

- ✓ 89 diprediksi Negatif dengan benar (True Negative)
- ✗ 9 salah diprediksi sebagai Netral
- ✗ 17 salah diprediksi sebagai Positif

Baris 2 (Actual: Netral):

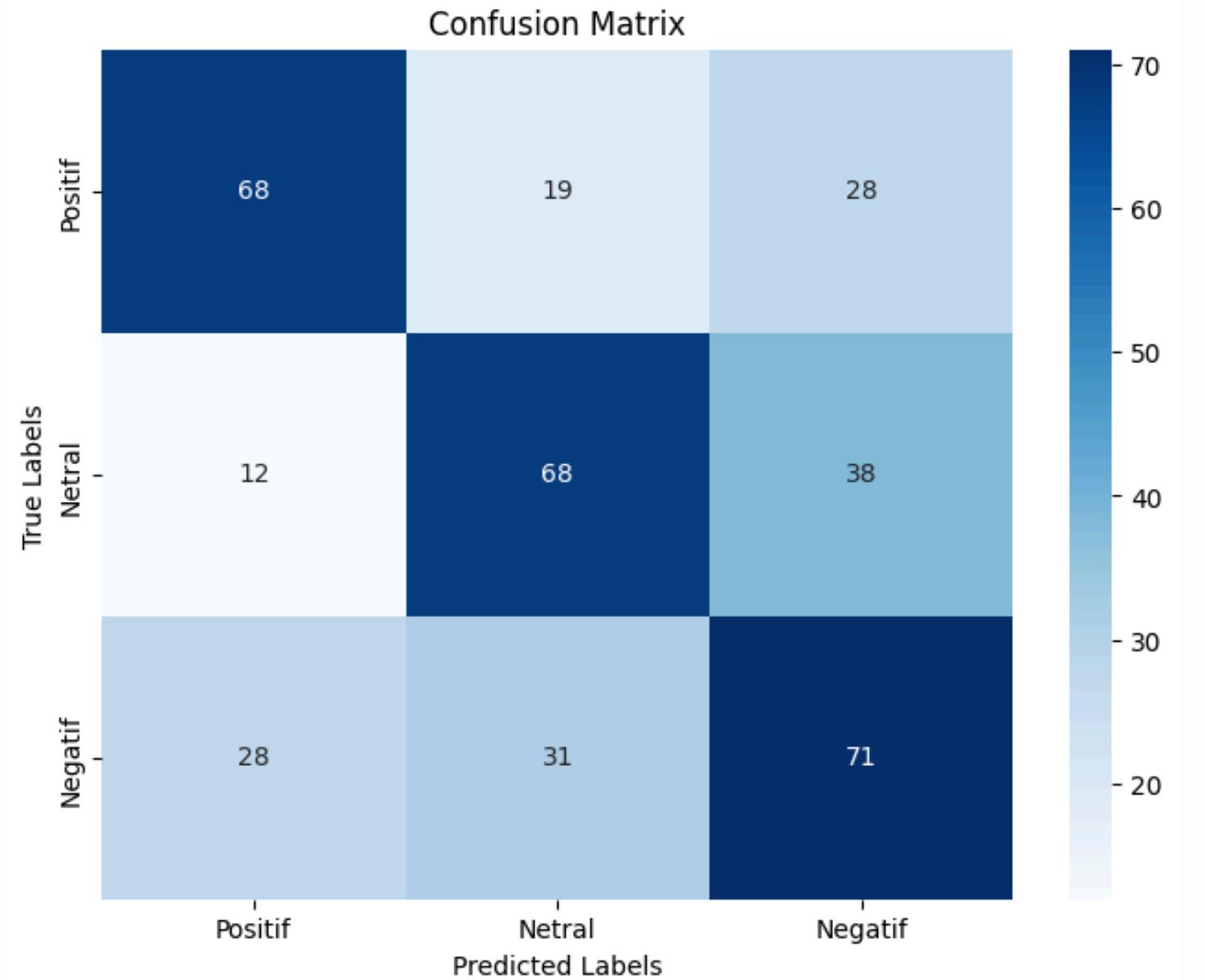
- ✗ 22 salah diprediksi sebagai Negatif
- ✓ 71 diprediksi Netral dengan benar (True Netral)
- ✗ 25 salah diprediksi sebagai Positif

Baris 3 (Actual: Positif):

- ✗ 39 salah diprediksi sebagai Negatif
- ✗ 18 salah diprediksi sebagai Netral
- ✓ 73 diprediksi Positif dengan benar (True Positive)

Model cukup akurat dalam mengenali kelas Positif dan Negatif, namun masih kesulitan membedakan kelas Netral yang sering salah diklasifikasikan. Perlu perbaikan pada identifikasi fitur Netral.

Confussion Matrik



Model ini cukup baik untuk kelas Negatif dan Netral, tapi kurang akurat untuk kelas Positif yang banyak salah klasifikasi.

Dalam confussion matrik di samping

Baris 1 (Actual: negatif):

- ✓ 71 diprediksi Negatif dengan benar (True Negative)
- ✗ 31 salah diprediksi sebagai Netral
- ✗ 28 salah diprediksi sebagai Positif

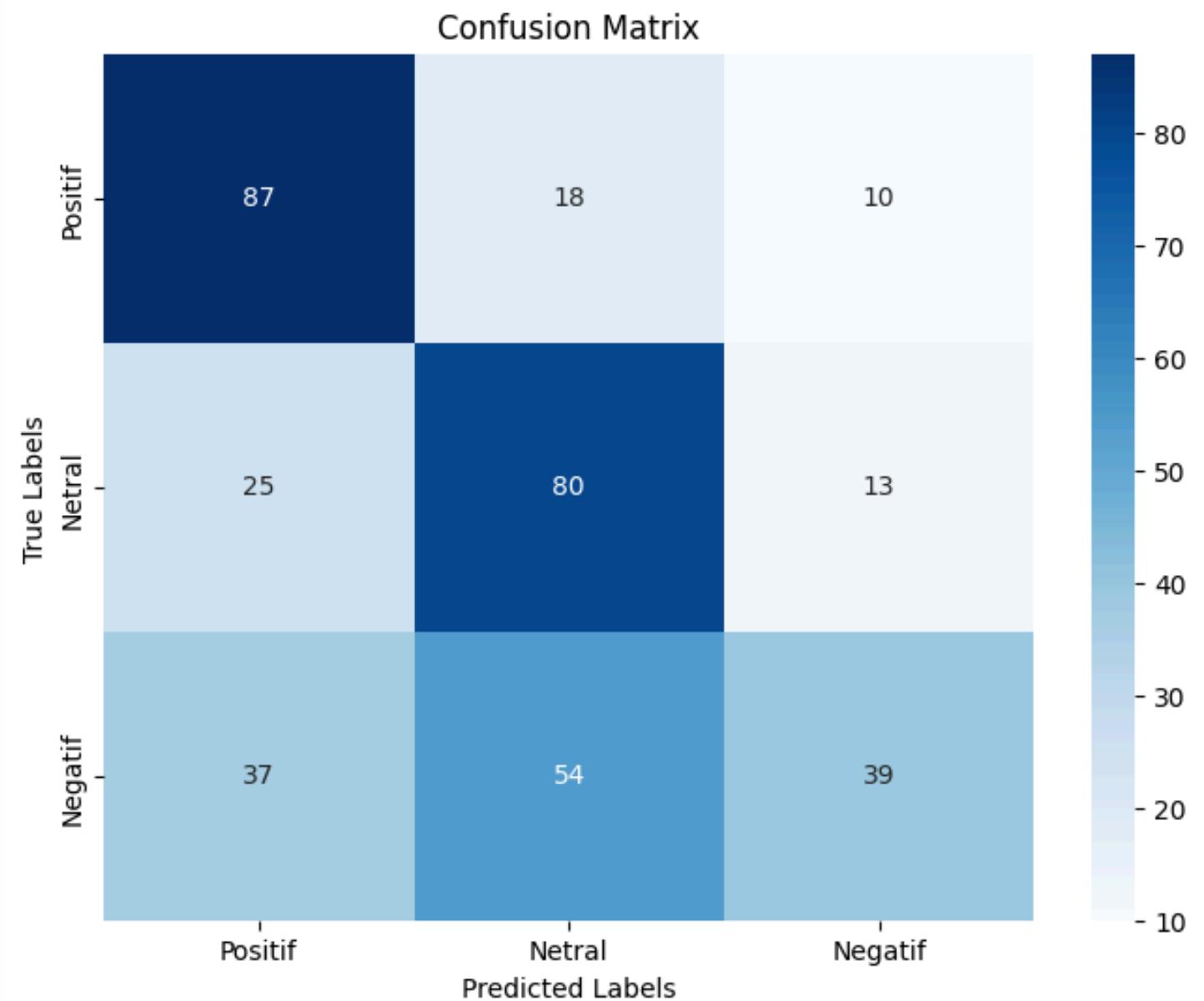
Baris 2 (Actual: Netral):

- ✗ 12 salah diprediksi sebagai Negatif
- ✓ 68 diprediksi Netral dengan benar (True Netral)
- ✗ 38 salah diprediksi sebagai Positif

Baris 3 (Actual: Positif):

- ✗ 68 salah diprediksi sebagai Negatif
- ✗ 19 salah diprediksi sebagai Netral
- ✓ 28 diprediksi Positif dengan benar (True Positive)

Confussion Matrik



GRU

Model ini menunjukkan performa terbaik pada kelas Positif dengan 87 prediksi benar, cukup baik di kelas Netral dengan 80 benar, namun kurang akurat pada kelas Negatif yang banyak salah klasifikasi ke Netral dan Positif.

Dalam confussion matrik di samping

Baris 1 (Actual: Negatif):

- ✓ 39 diprediksi Negatif dengan benar (True Negative)
- ✗ 54 salah diprediksi sebagai Netral
- ✗ 37 salah diprediksi sebagai Positif

Baris 2 (Actual: Netral):

- ✗ 25 salah diprediksi sebagai Negatif
- ✓ 80 diprediksi Netral dengan benar (True Netral)
- ✗ 13 salah diprediksi sebagai Positif

Baris 3 (Actual: Positif):

- ✗ 10 salah diprediksi sebagai Negatif
- ✗ 18 salah diprediksi sebagai Netral
- ✓ 87 diprediksi Positif dengan benar (True Positive)

Model Evaluation

```
Accuracy: 0.7521
Precision: 0.7562
Recall: 0.7562
F1 Score: 0.7521
Confusion Matrix:
[[91 21]
 [39 91]]

Classification Report:
precision    recall    f1-score   support
0            0.70      0.81      0.75      112
1            0.81      0.70      0.75      130

accuracy                           0.75      242
macro avg       0.76      0.76      0.75      242
weighted avg    0.76      0.75      0.75      242
```

Karena pada 2 model dengan performa tertinggi memiliki kekurangan dalam memprediksi class netral maka

Pelatihan ulang tanpa menyertakan data dengan label netral menghasilkan akurasi sebesar 75%, yang lebih tinggi dibandingkan pelatihan sebelumnya yang mencakup sentimen netral. Hal ini menunjukkan bahwa penghapusan kelas netral dapat meningkatkan kinerja model dengan mengurangi ambiguitas dalam klasifikasi.

Model Deployment



Streamlit

Pilih Halaman

Prediksi Sentimen

Masukkan teks untuk prediksi sentimen:

```
@DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut
```

Pilih jenis model:

Konvensional

Deep Learning

Pilih model konvensional untuk prediksi:

Naive Bayes

SVM

Prediksi

Prediksi Sentimen

Halaman ini digunakan untuk melakukan prediksi sentimen berdasarkan input manual.

Masukkan teks untuk prediksi sentimen:

```
@DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut
```

Pilih jenis model:

- Konvensional
- Deep Learning

Pilih model konvensional untuk prediksi:

- Naive Bayes
- SVM

Prediksi

Hasil Prediksi

Tweet: @DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut

Model yang digunakan: Naive Bayes

Prediksi Sentimen: negatif

Pilih Halaman

Prediksi Sentimen

Masukkan teks untuk prediksi sentimen:

```
@DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut
```

Pilih jenis model:

Konvensional

Deep Learning

Pilih model deep learning untuk prediksi:

LSTM

GRU

Prediksi

Prediksi Sentimen

Halaman ini digunakan untuk melakukan prediksi sentimen berdasarkan input manual.

Masukkan teks untuk prediksi sentimen:

```
@DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut
```

Pilih jenis model:

- Konvensional
- Deep Learning

Pilih model deep learning untuk prediksi:

- LSTM
- GRU

Prediksi

Hasil Prediksi

Tweet: @DivHumas_Polri Gakan di ladenin sama pak pulisi Karna pak pulisi di gaji sama @jokowi pake duit @jokowi bukaan sama RAKYAT!!! lapor nya nanti aja tunggu udah ganti presiden #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrutal #UASdiftnahKejiDanBrut

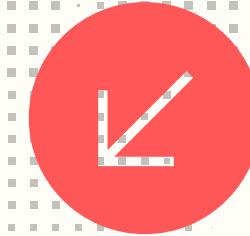
Model yang digunakan: LSTM

Prediksi Sentimen: Negatif

Depk

Deploy :

Hasil



- Model Naive Bayes menunjukkan performa terbaik dengan akurasi 64%, mengungguli SVM serta model deep learning seperti LSTM dan GRU. Hal ini menegaskan efektivitas pendekatan probabilistik dalam klasifikasi teks, terutama pada dataset dengan ukuran terbatas.
- Setelah menghapus kelas netral dan mengubah skema klasifikasi menjadi dua kelas, akurasi meningkat signifikan hingga 75%. Peningkatan ini menunjukkan bahwa penyederhanaan kelas dapat mengurangi ambiguitas dan meningkatkan ketepatan model.



Genesis

**Terimakasih
Atas Perhatiannya**