



**Hermawan
Rahmatina Ari A
Rona Wenas**

ANALISA SENTIMENT PADA TWEET

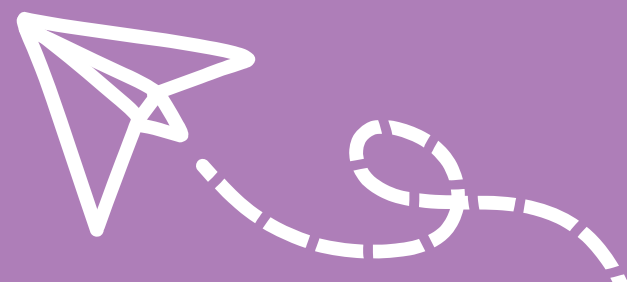
Challenge Platinum level
DSC Binar Wave 7

Latar belakang

Menurut KBBI, sentimen berarti pendapat yang didasarkan pada perasaan yang berlebihan terhadap sesuatu.

Analisis sentimen sendiri merupakan proses menganalisis teks untuk menentukan apakah emosi pada teks tersebut positif, negative, atau netral.

Analisis sentimen tidak hanya digunakan di media social seperti Twitter saja. Banyak perusahaan melakukan analisis sentiment untuk meningkatkan produk dan layanan, seperti analisis berdasarkan ulasan dari pelanggan



Tujuan

1. Mengidentifikasi sebaran data tweet positif, negatif, dan netral
2. Membuat model yang optimal untuk memprediksi sentimen dari sebuah teks
3. Membuat API yang dapat mengklasifikasikan sentimen yang dihasilkan

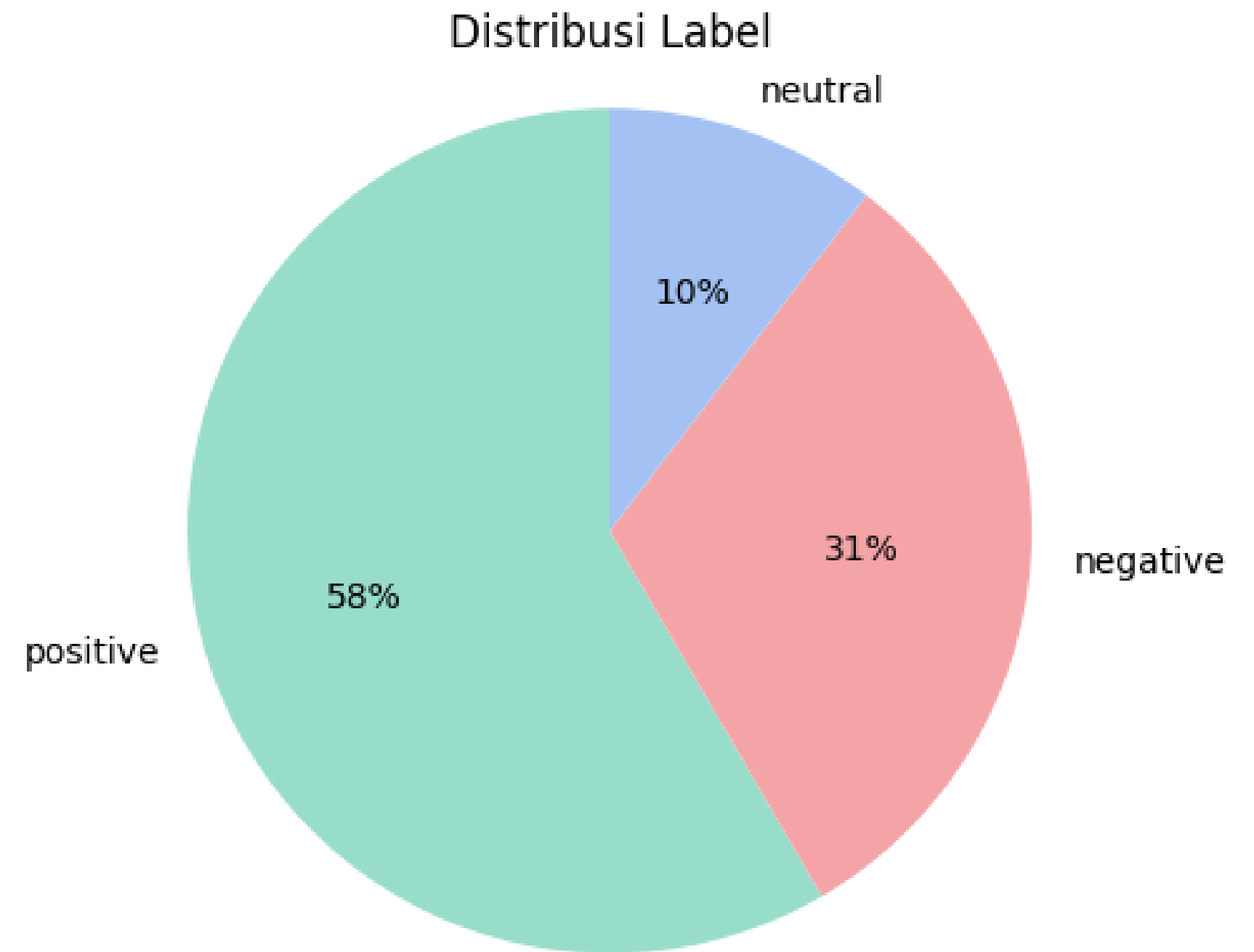


Exploratory Data Analysis

Sentiment Positif: 6416 Tweet

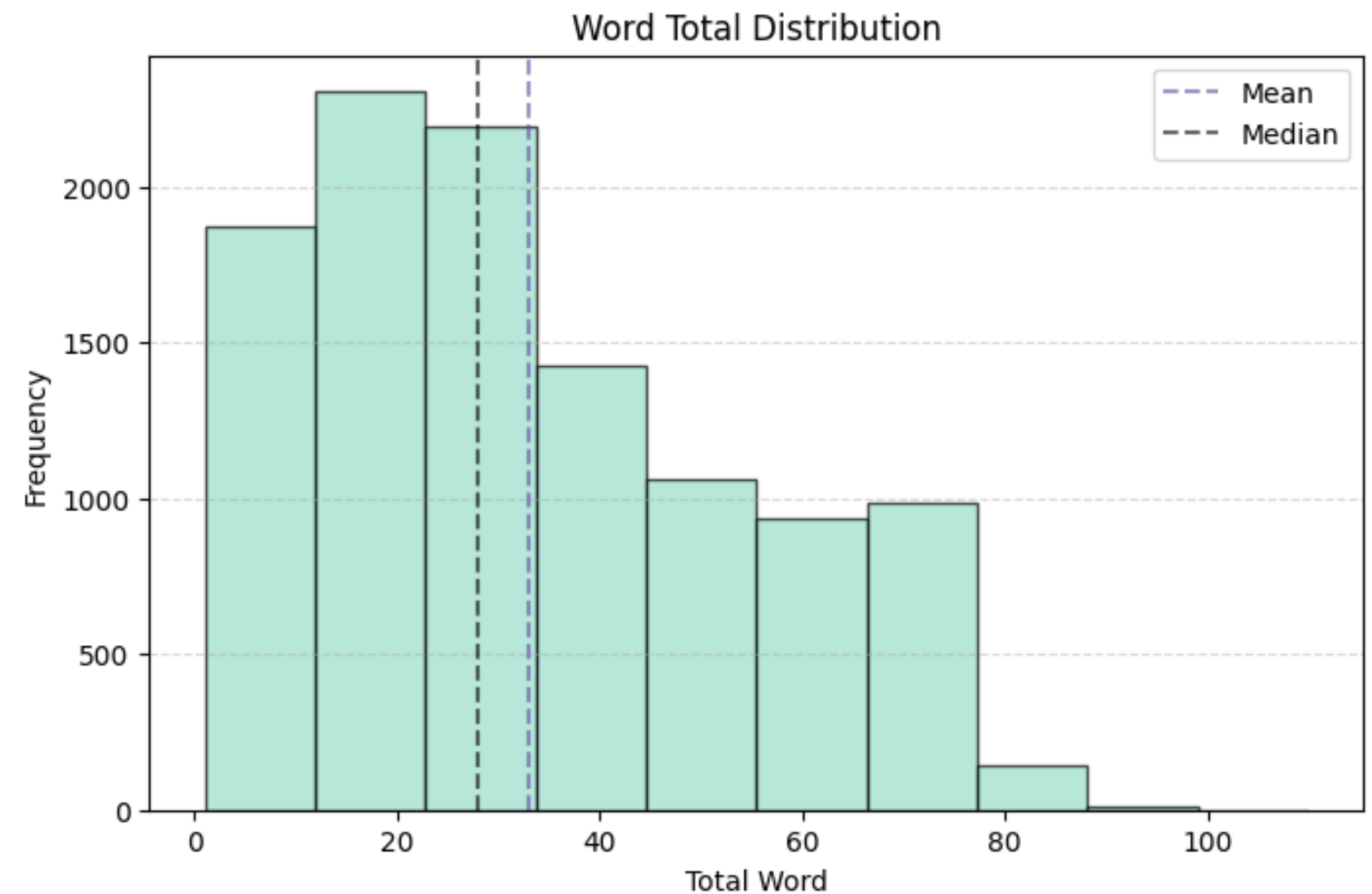
Sentiment Negatif: 3436 Tweet

Sentiment Netral: 1148 Tweet



Exploratory Data Analysis

- Modus: 20 – 30 kata
- Median: 28 kata
- Mean: 33 kata



Data Preparation

11.000

Baris data

2

Kolom.

Tweet Bahasa Indonesia

Label positif, negatif, netral

0

Missing Values

67

Data duplikat

```
#Cek Data  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 11000 entries, 0 to 10999  
Data columns (total 2 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   Kalimat    11000 non-null   object  
1   Label      11000 non-null   object  
dtypes: object(2)  
memory usage: 172.0+ KB
```

```
# Cek Missing Values
```

```
df.isna().sum()
```

```
Kalimat    0  
Label      0  
dtype: int64
```

```
# Cek duplikasi data
```

```
print('{} data duplikat'.format(df.duplicated().sum()))
```

```
67 data duplikat
```

Data Preparation

11.000

Baris data

- 6416 sentimen positif
- 3436 sentimen negatif
- 1148 sentimen netral

OVERSAMPLING

Jumlah data pada label netral terlalu sedikit/tidak berimbang (imbalance), sehingga melakukan oversampling pada data sentimen netral agar jumlah datanya sama dengan jumlah data sentimen negatif

```
# Label classification

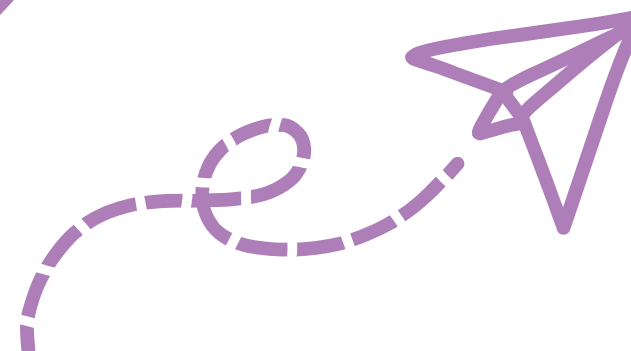
# Filter data by label
df_positive = data[data['Label'] == 'positive']
df_negative = data[data['Label'] == 'negative']
df_neutral = data[data['Label'] == 'neutral']

# Oversample neutral label to match negative label
df_neutral_over = df_neutral.sample(df_negative.shape[0], replace=True)

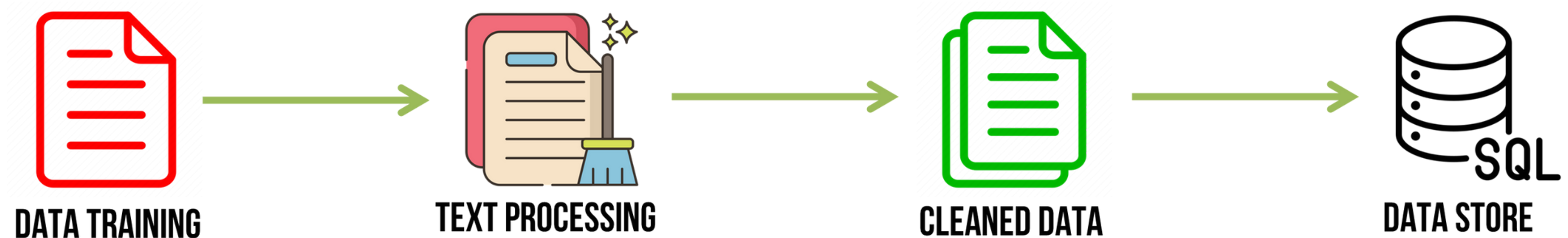
# Merge all data
data = pd.concat([df_positive, df_negative, df_neutral_over])

# Data Check
label_counts = data['Label'].value_counts()
print(label_counts)
```

```
positive    6383
negative    3412
neutral     3412
Name: Label, dtype: int64
```



Text Normalization



Sebelum: nkri bersyariah , fpi , revolusi akhlaq , suara keadilan , __laugh__ semua pada tersandung kasus , fpi , ulama nya tersandung obrolan mesum tidak beran

Sesudah: negara kesatuan republik indonesia bersyariah front pembela islam revolusi akhlaq suara keadilan laugh semua pada tersandung kasus front pembela islam

Machine Learning Preparation

KLASIFIKASI LABEL FEATURE

Memisahkan fitur (X) dan label (Y) pada dataset



FEATURE EXTRACTION

Mengekstraksi fitur dan mengubah data teks menjadi bilangan vector
(Tokenizer dan Pad Sequence)

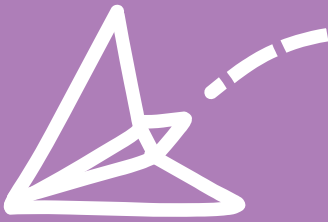


TRAIN – TEST DATA SPLIT

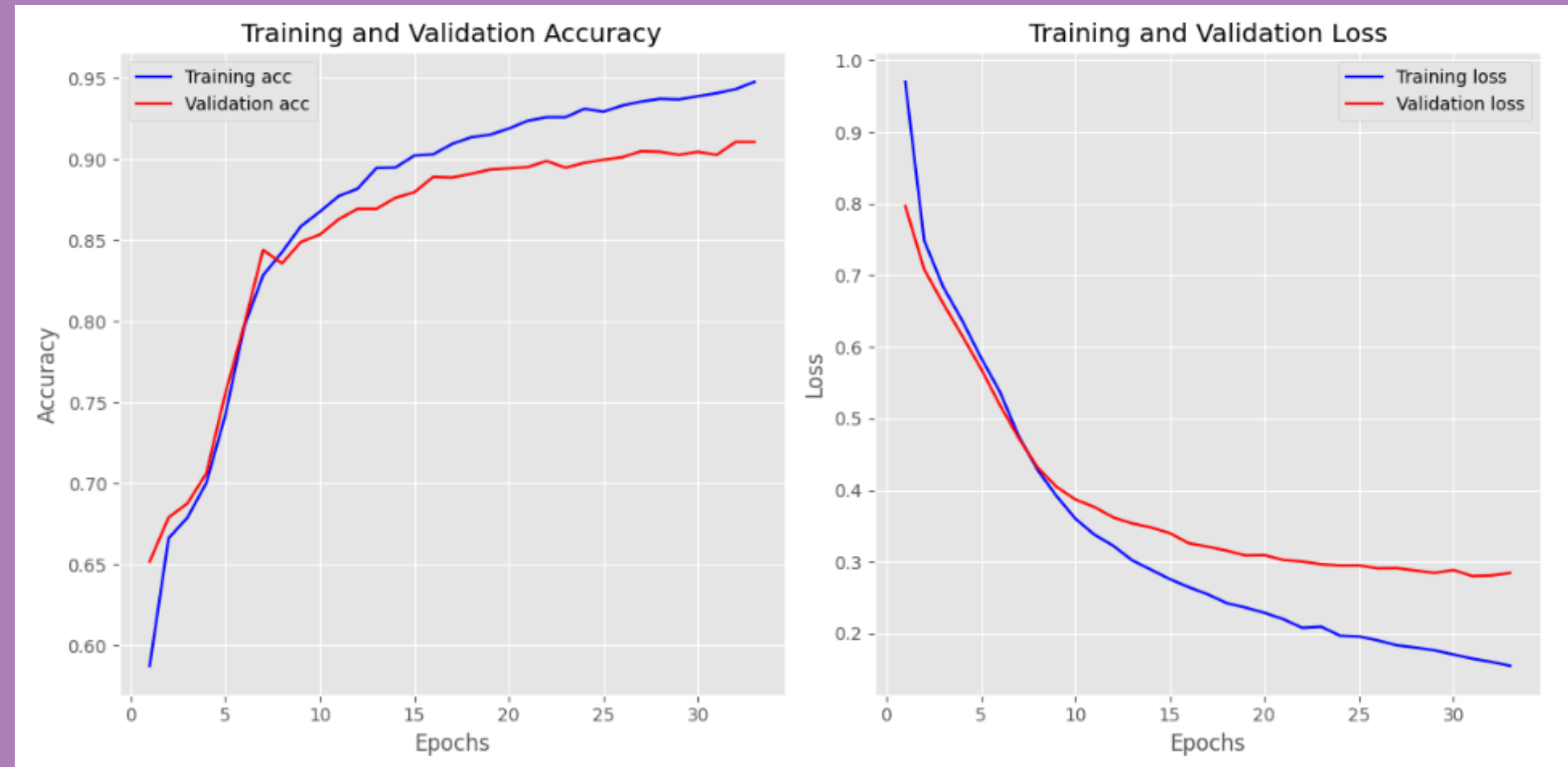
Membagi data menjadi data train dan data test dengan perbandingan 80:20

Hasil Analisis

MODEL TRAINING DAN EVALUATION (LSTM)

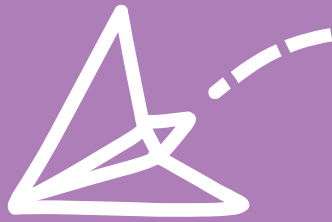


Parameter	
Activation	Softmax
Learning Rate	0.00005
Epoch	50
Batch size	32
Embedding dimension	200
Cross validation	5
Regularization	
Dropout	0.8
Early Stopping	
• Monitor	val_loss
• Mode	min
• Verbose	1
• Patience	2



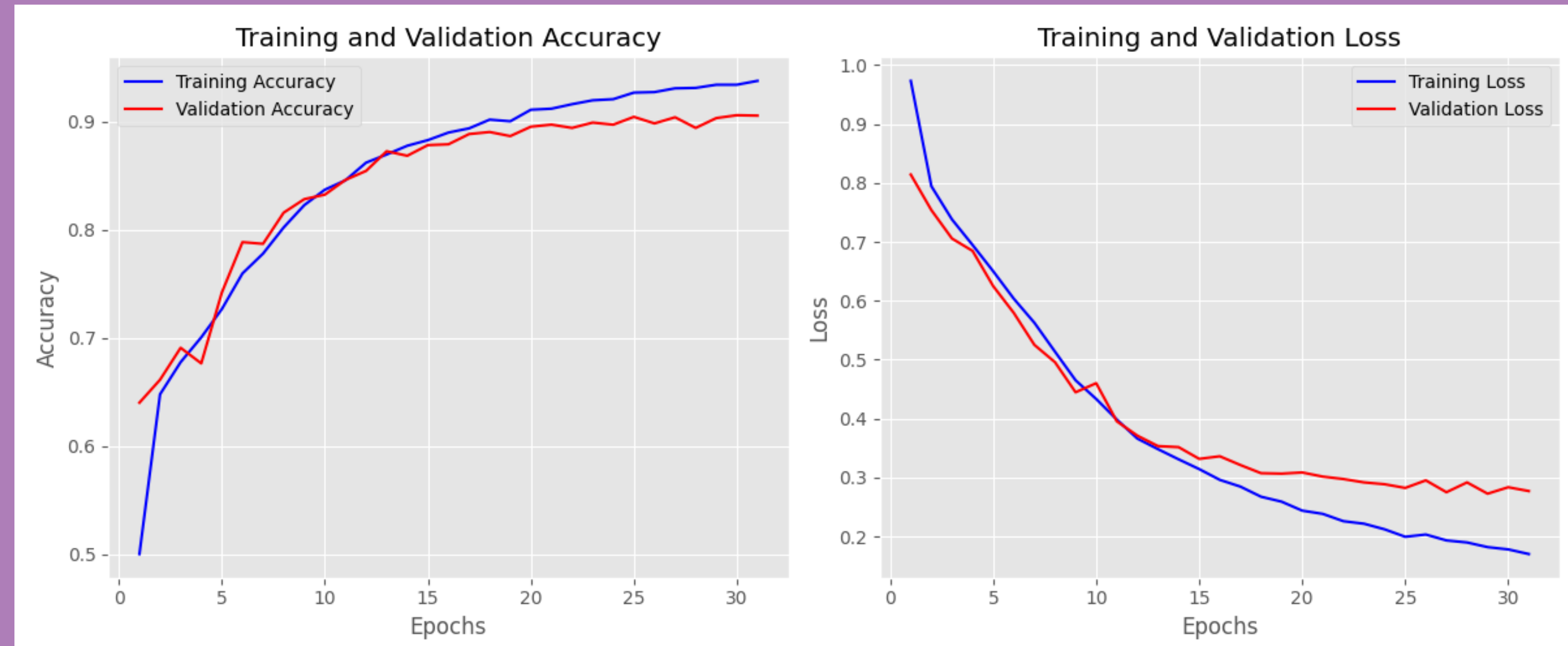
Dengan menggunakan model LSTM dihasilkan rata-rata akurasi sebesar 0.94 dan capaian loss pada iterasi ke-5 Sebesar 0.15 dan val_loss 0.28

Hasil Analisis



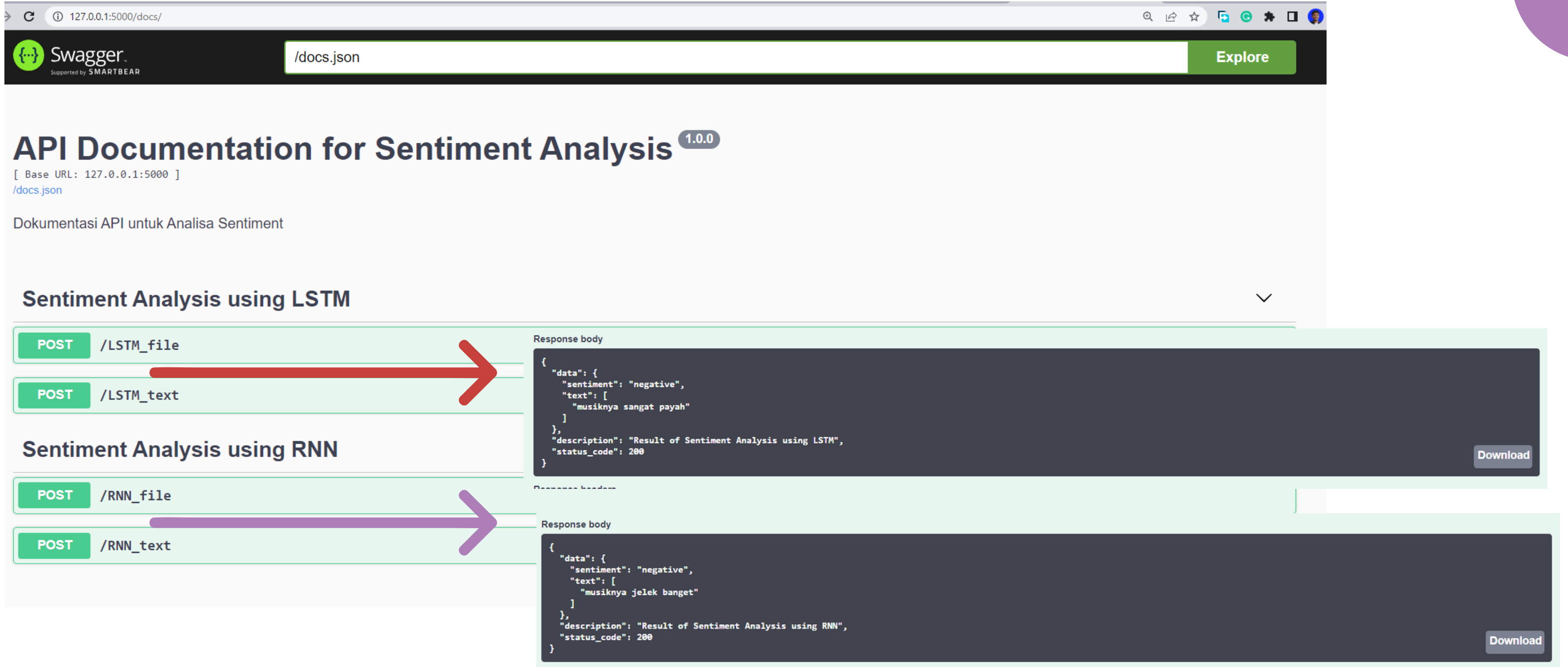
MODEL TRAINING DAN EVALUATION (RNN)

Parameter	
Activation	Softmax
Learning Rate	0.00005
Epoch	50
Batch size	32
Embedding dimension	200
Cross validation	5
Regularization	
Dropout	0.8
Early Stopping	
• Monitor	val_loss
• Mode	min
• Verbose	1
• Patience	2



Dengan menggunakan pendekatan model RNN dihasilkan rata-rata akurasi sebesar 0.9 dan capaian loss pada iterasi ke-5 Sebesar 0.17 dan val_loss 0.27

Prediksi dan Deployment



The image shows a Swagger API documentation page for a Sentiment Analysis service. The page is titled "API Documentation for Sentiment Analysis 1.0.0" and includes a base URL of "127.0.0.1:5000/docs/". The documentation is organized into two main sections: "Sentiment Analysis using LSTM" and "Sentiment Analysis using RNN". Each section lists two endpoints: a POST endpoint for file uploads and a POST endpoint for text input. A red arrow points from the LSTM endpoints to a sample response body, and a purple arrow points from the RNN endpoints to another sample response body. Both response bodies show a "negative" sentiment for the input text "musiknya sangat payah" and "musiknya jelek banget" respectively. The response also includes a description and a status code of 200. A "Download" button is present next to each response body.

Swagger
Supported by SMARTBEAR

/docs.json Explore

API Documentation for Sentiment Analysis 1.0.0

[Base URL: 127.0.0.1:5000]
</docs.json>

Dokumentasi API untuk Analisa Sentiment

Sentiment Analysis using LSTM

POST /LSTM_file

POST /LSTM_text

Response body

```
{
  "data": {
    "sentiment": "negative",
    "text": [
      "musiknya sangat payah"
    ]
  },
  "description": "Result of Sentiment Analysis using LSTM",
  "status_code": 200
}
```

Download

Sentiment Analysis using RNN

POST /RNN_file

POST /RNN_text

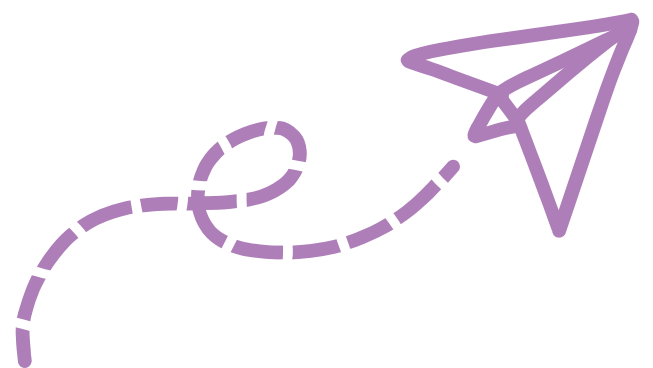
Response body

```
{
  "data": {
    "sentiment": "negative",
    "text": [
      "musiknya jelek banget"
    ]
  },
  "description": "Result of Sentiment Analysis using RNN",
  "status_code": 200
}
```

Download



Hasil


1. Dataset terdiri dari 58% data sentiment positif (6416 data), 31% data sentiment negative (3436 data), dan 10% data sentiment netral (1148 data)
 2. Model LSTM memiliki performa yang lebih baik dari model RNN, dibuktikan dengan hasil capaian akurasi yang lebih tinggi yakni 0.94 dan loss yang lebih rendah yaitu 0.16
- 



Kesimpulan

Secara kesimpulan, deep learning merupakan teknik yang sangat kuat dalam mengenali pola-pola rumit pada data dan menghasilkan hasil yang lebih akurat dibandingkan dengan pendekatan machine learning tradisional.

Kelebihan utama deep learning adalah **fleksibilitasnya** yang memungkinkan mengolah berbagai jenis data, termasuk gambar, suara, sinyal, dan sebagainya. Namun, deep learning juga memiliki beberapa kendala, yaitu membutuhkan data yang cukup besar, apabila jumlah dataset terbatas maka berpotensi menyebabkan overfitting.




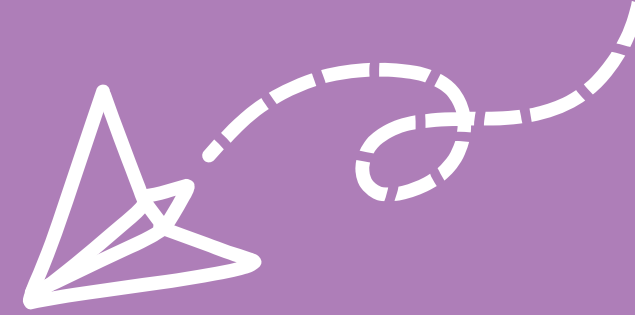
Selain itu, deep learning juga memerlukan **hardware** yang kuat dan komputasi yang intensif untuk melatih model dengan efisien.



Rekomendasi

Rekomendasi dari kami, apabila data yang diolah masih tergolong sedikit dan tidak rumit, sebaiknya menggunakan algoritma yang sederhana yakni menggunakan machine learning konvensional. Deep Learning akan powerfull apabila dataset yang diolah jumlahnya sangat besar dan kompleks/rumit, jika jumlah data terlalu kecil dan simpel maka beresiko terjadi overfitting.





Terima kasih

KONTAK :



Hermawan



Rahmatina Ari Apriliana



Ronna Wenas

DOKUMENTASI :



GitHub Repository.

