

# SENTIMENT ANALYSIS

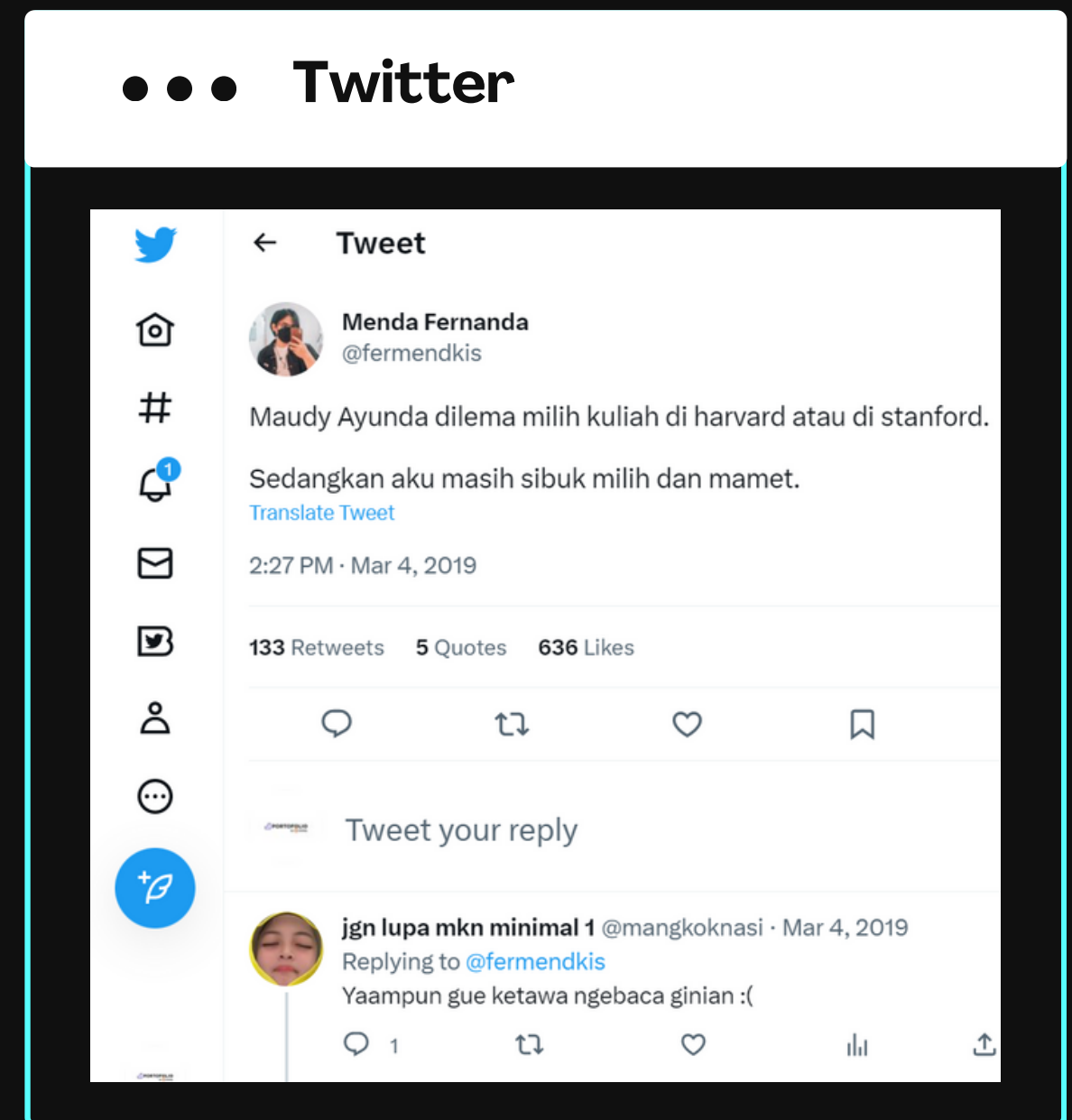
Bagas Hada - Robby Julian

# PENGERTIAN

- Sentiment analysis adalah proses mengidentifikasi perasaan dan pendapat dalam text.
- Sentiment analysis menggunakan NLP untuk mengidentifikasi sentiment dan memberikan interpretasi hasilnya

# LATAR BELAKANG

- Berkembangnya teknologi natural language processing (NLP) mempermudah analisis sentimen di media sosial.
- Analisis sentimen dapat memberikan wawasan tentang tanggapan pelanggan dan opini publik di Indonesia.



# METODE PENELITIAN

**1**

EDA (Exploratory Data Analysis)

**3**

Features Extraction & Tst Split

**2**

Data Cleansing

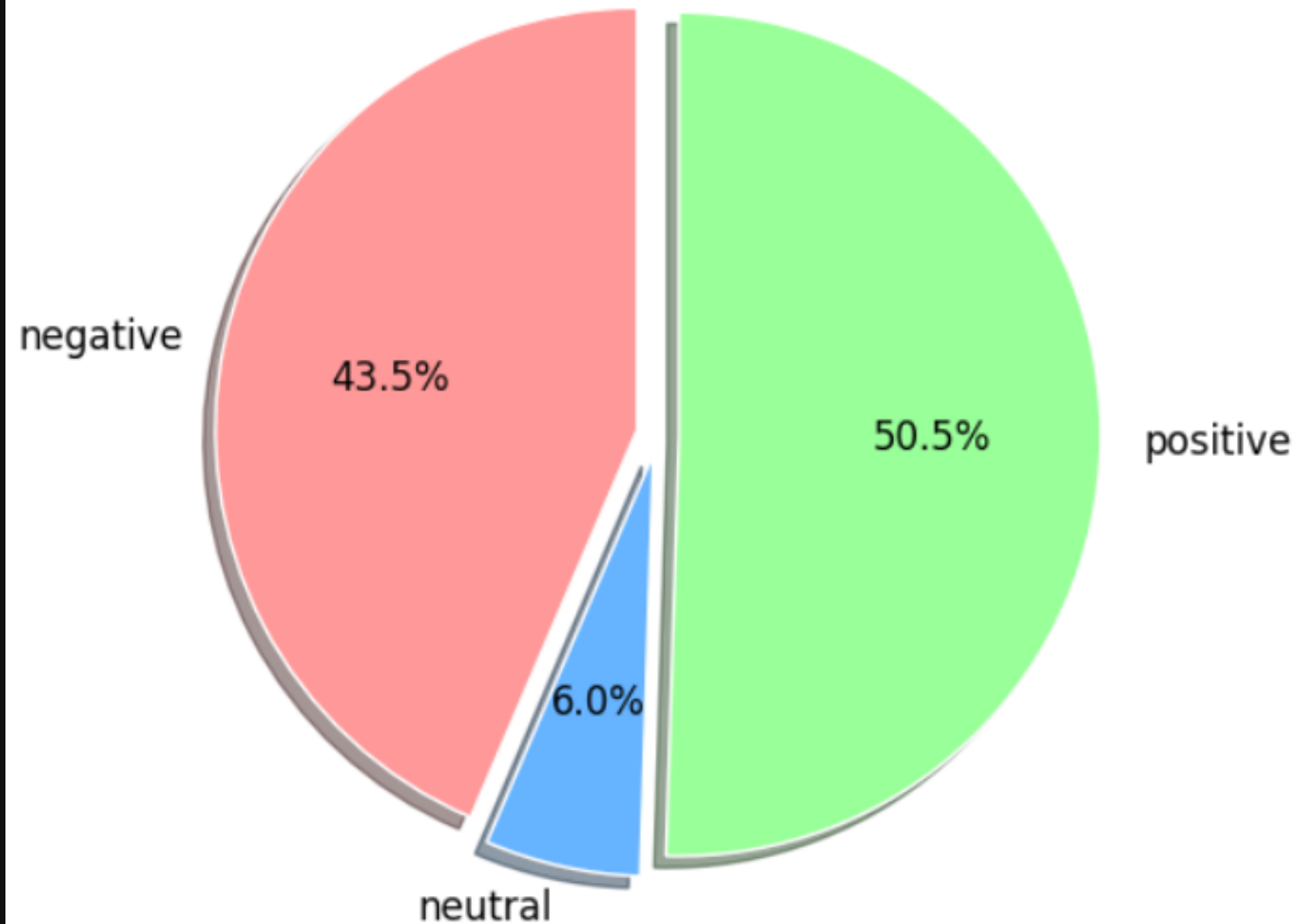
**4**

Modeling

# EDA (EXPLORATORY DATA ANALYSIS)

# EDA (EXPLORATORY DATA ANALYSIS)

Data Label



Note

Presentase pengguna lebih banyak menuliskan kata-kata dalam riview **positif** dibanding menuliskan kata-kata riview negatif atau netral

# EDA (EXPLORATORY DATA ANALYSIS)

## Positive



## Negative



## Neutral



**DATA CLEANSING**

# DATA CLEANSING

**1** Normalize

**2** Punctuation

**3** Stopword

**4** Lowercase

## ● ● ● Cleaning

```
import re
import pandas as pd
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
import re
from nltk.corpus import stopwords

stop_words = set(stopwords.words('indonesian'))

def cleansing(sent):
    string = sent.lower()
    string = re.sub(r'^a-zA-z0-9',' ', string)
    string = re.sub(r'^\w',' ', string)

    words = nltk.word_tokenize(string)
    words = [word for word in words if word not in stop_words]

    text = ' '.join(words)
    return text
```



# DATA CLEANSING

## ● ● ● Before

| index | text  | label    |
|-------|---|----------|
| 0     | Nikmati cicilan 0% hingga 12 bulan untuk pemesanan tiket pesawat air asia dengan kartu kredit bni!  | neutral  |
| 1     | Kue-kue yang disajikan bikin saya bernostalgia. Semuanya tipikal kue zaman dulu, baik dari penampilan maupun rasa. Kuenya enak dan harganya juga murah.   | positive |
| 2     | Ibu pernah bekerja di grab indonesia  | neutral  |
| 3     | Paling suka banget makan siang di sini ayam sama sambalnya enak banget harganya luar biasa hemat, rasa ayamnya meresap sampai ketulangnya, es lidah buayanya juga segar bikin adem perut setelah makan sambal yang pedas, pelayannya sigap dan ramah, yang aku suka di tempat kasir ada tulisan 10% disumbangkan untuk beramal, buat makan jadi lebih enak ke perut | positive |
| 4     | Pelayanan bus DAMRI sangat baik   | positive |

## ● ● ● After

| index | text   | label    |
|-------|--|----------|
| 0     | nikmati cicilan 0 12 pemesanan tiket pesawat air asia kartu kredit bni   | neutral  |
| 1     | kue kue disajikan bikin bernostalgia tipikal kue zaman penampilan kuenya enak harganya murah   | positive |
| 2     | grab indonesia   | neutral  |
| 3     | suka banget makan siang ayam sambalnya enak banget harganya hemat ayamnya meresap ketulangnya es lidah buayanya segar bikin adem perut makan sambal pedas pelayannya sigap ramah suka kasir tulisan 10 disumbangkan beramal makan enak perut | positive |
| 4     | pelayanan bus damri  | positive |

# FEATURES EXTRACTION & TEST SPLIT

# FEATURES EXTRACTION



## TF-IDF

Metode yang mengekspresikan bobot kata dalam dokumen teks dengan menggunakan rumus yang mengukur frekuensi kemunculan kata dalam dokumen dan di seluruh dokumen yang tersedia.



## Feature Extrac

```
# from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

count_vect = TfidfVectorizer()
# count_vect = CountVectorizer()
count_vect.fit(data_preprocessed)

X = count_vect.transform(data_preprocessed)
print("Feature Extraction Done !")
```

# TRAIN TEST SPLIT



## Test Split

Teknik yang membagi dataset menjadi data training dan data testing untuk membangun dan mengevaluasi model pembelajaran mesin.



## Test Split

```
from sklearn.model_selection import train_test_split

classes = df.label
classes

X_train, X_test, y_train, y_test = train_test_split(X, classes, test_size= 0.2)
```

**MODELING**

Studi yang dilakukan memanfaatkan 3 (tiga) jenis model *machine learning* yang menggunakan arsitektur yang berbeda, yaitu:

- Recurrent Neural Network (RNN)
- Long Short Term Memory Networks (LSTM)
- MultiLayer Perception (MLP)

## Recurrent Neural Network (RNN)

RNN (Recurrent Neural Network) adalah salah satu jenis arsitektur model dalam *deep learning* yang dirancang untuk mengatasi masalah pengolahan data berkelanjutan atau *sequential* data, seperti urutan kata dalam sebuah kalimat atau deret waktu.

RNN bekerja dengan cara memproses input data pada setiap waktu atau langkah iterasi dan menghasilkan keluaran dan juga mengirimkan sinyal kembali ke jaringan neural network. Setiap unit pada RNN menerima input dari unit sebelumnya dan juga mempertahankan sebuah state atau kondisi yang menyimpan informasi dari iterasi sebelumnya.

Implementasi RNN dalam studi ini dapat dilihat pada halaman-halaman selanjutnya ...

```
Start
|
Input data at time step t
|
v
RNN Unit
|
v
Output at time step t
|
v
Signal sent back to the network
|
v
Update state and go to the next time step
|
v
End
```



```
neg = df.loc[df['label']=='negative'].text.tolist()
pos = df.loc[df['label']=='positive'].text.tolist()
net = df.loc[df['label']=='neutral'].text.tolist()

neg_label = df.loc[df['label']=='negative'].label.tolist()
pos_label = df.loc[df['label']=='positive'].label.tolist()
net_label = df.loc[df['label']=='neutral'].label.tolist()

total_data = pos+neg+net
labels = pos_label+neg_label + net_label
```

```
[ ] import pickle
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from collections import defaultdict

max_features = 100000
tokenizer = Tokenizer(num_words=max_features, split= ' ', lower=True)
tokenizer.fit_on_texts(total_data)
with open('tokenizer.pickle','wb') as handle:
    pickle.dump(tokenizer, handle, protocol= pickle.HIGHEST_PROTOCOL)
    print("tokenizer.pickle has created !")

X = tokenizer.texts_to_sequences(total_data)

vocab_size = len(tokenizer.word_index)
maxlen = max(len(x) for x in X)

X = pad_sequences(X)
with open('x_pad_sequences.pickle','wb') as handle:
    pickle.dump(X, handle, protocol = pickle.HIGHEST_PROTOCOL)
    print('x_pad_sequences.pickle has created !')
```

Ekstraksi dilakukan terhadap label-label dan data-data terkait pada setiap label di data frame yang digunakan

Terhadap data-data yang telah diekstraksi dilakukan proses tokenizing (mengubah data teks menjadi numerikal) dan padding (penyeragaman panjang sekuens numerikal yang telah terbentuk), kemudian disimpan kedalam bentuk file dalam variabel "X" untuk penggunaan lebih lanjut menggunakan pickle.

```
[ ] Y = pd.get_dummies(labels)
    Y = Y.values

    with open('y_labels.pickle','wb') as handle:
        pickle.dump(Y, handle, protocol = pickle.HIGHEST_PROTOCOL)
        print('y_labels.pickle has created !')
```

```
from sklearn.model_selection import train_test_split

file = open('x_pad_sequences.pickle','rb')
X = pickle.load(file)
file.close()

file = open('y_labels.pickle','rb')
Y = pickle.load(file)
file.close()
```

```
#training
import numpy as np
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, SpatialDropout1D, SimpleRNN, Activation
from tensorflow.keras import optimizers
from tensorflow.keras.callbacks import EarlyStopping, TensorBoard
from tensorflow.keras.layers import Flatten
from tensorflow.keras import backend as K

embed_dim = 100
units = 64

model_rnn = Sequential()
model_rnn.add(Embedding(max_features, embed_dim, input_length=X.shape[1]))
model_rnn.add(SimpleRNN(units, dropout=0.2))
model_rnn.add(Dense(3, activation='softmax'))

sgd = optimizers.Adam(lr=0.001)
model_rnn.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
print(model_rnn.summary())

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=3)
history = model_rnn.fit(X_train, y_train, epochs=10, batch_size=10, validation_data=(X_test, y_test), verbose=1, callbacks=[es])
```

Proses konversi menjadi vektor/array dilakukan kepada setiap label menggunakan one-hot encoding dan kemudian disimpan dalam variabel "Y" dalam bentuk file menggunakan pickle

Terhadap data dalam variabel X dan Y dilakukan menjadi data training dan testing untuk keperluan pelatihan model RNN.

Proses training dilakukan terhadap model RNN dengan menggunakan data-data yang telah di-define.

```
model_rnn.save('model_rnn.h5')
print("Model has been created !")
```

Model RNN kemudian disimpan/di-export untuk penggunaan lebih lanjut



```
#evaluation
from sklearn import metrics

predictions = model_rnn.predict(X_test)
y_pred= predictions
matrix_test = metrics.classification_report(y_test.argmax(axis=1), y_pred.argmax(axis=1))
print("Testing Done !")
print(matrix_test)
```

Terhadap model RNN dilakukan evaluasi untuk mengukur akurasi dalam skor precision, recall, and F1-score bagi ketiga label ("Positive", "Negative", dan "Neutral") beserta rata-rata setiap skor.

# Hasil evaluasi skor akurasi model RNN dan uji coba prediksi label (1)

```
#evaluation
from sklearn import metrics

predictions = model_rnn.predict(X_test)
y_pred= predictions
matrix_test = metrics.classification_report(y_test.argmax(axis=1), y_pred.argmax(axis=1))
print("Testing Done !")
print(matrix_test)
```

169/169 [=====] - 2s 14ms/step

Testing Done !

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.77      | 0.76   | 0.77     | 2274    |
| 1            | 0.75      | 0.46   | 0.57     | 353     |
| 2            | 0.78      | 0.82   | 0.80     | 2759    |
| accuracy     |           |        | 0.77     | 5386    |
| macro avg    | 0.77      | 0.68   | 0.71     | 5386    |
| weighted avg | 0.77      | 0.77   | 0.77     | 5386    |

# Hasil evaluasi skor akurasi model RNN dan uji coba prediksi label (2)

GET

/sentiment-rnn Sentiment Analytics Rnn

Parameters

Cancel

| Name  | Description                                      |
|---|--|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="makanannya gak enak"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/sentiment-rnn?text=makanannya%20gak%20enak' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-rnn?text=makanannya%20gak%20enak
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "neutral"   },   "message": "Success" }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 68 content-type: application/json date: Mon, 10 Apr 2023 12:15:02 GMT server: uvicorn</pre></div></div> |

# Hasil evaluasi skor akurasi model RNN dan uji coba prediksi label (3)

GET

/sentiment-rnn Sentiment Analytics Rnn

Parameters

Cancel

| Name  | Description                                       |
|---|---|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="bajunya keren banget"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/sentiment-rnn?text=bajunya%20keren%20banget' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-rnn?text=bajunya%20keren%20banget
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "neutral"   },   "message": "Success" }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 68 content-type: application/json date: Mon, 10 Apr 2023 12:19:37 GMT server: uvicorn</pre></div></div> |

## Hasil evaluasi skor akurasi model RNN dan uji coba prediksi label (4), menggunakan input file

POST

/sentiment-rnn-file

Sentiment Analytics Rnn File

Cancel

Reset

Parameters

No parameters

Request body required

multipart/form-data

file \* required

string(\$binary)

Choose File

sample\_data\_input.csv

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/sentiment-rnn-file' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_data_input.csv;type=text/csv'
```

Request URL

http://127.0.0.1:8000/sentiment-rnn-file

Server response

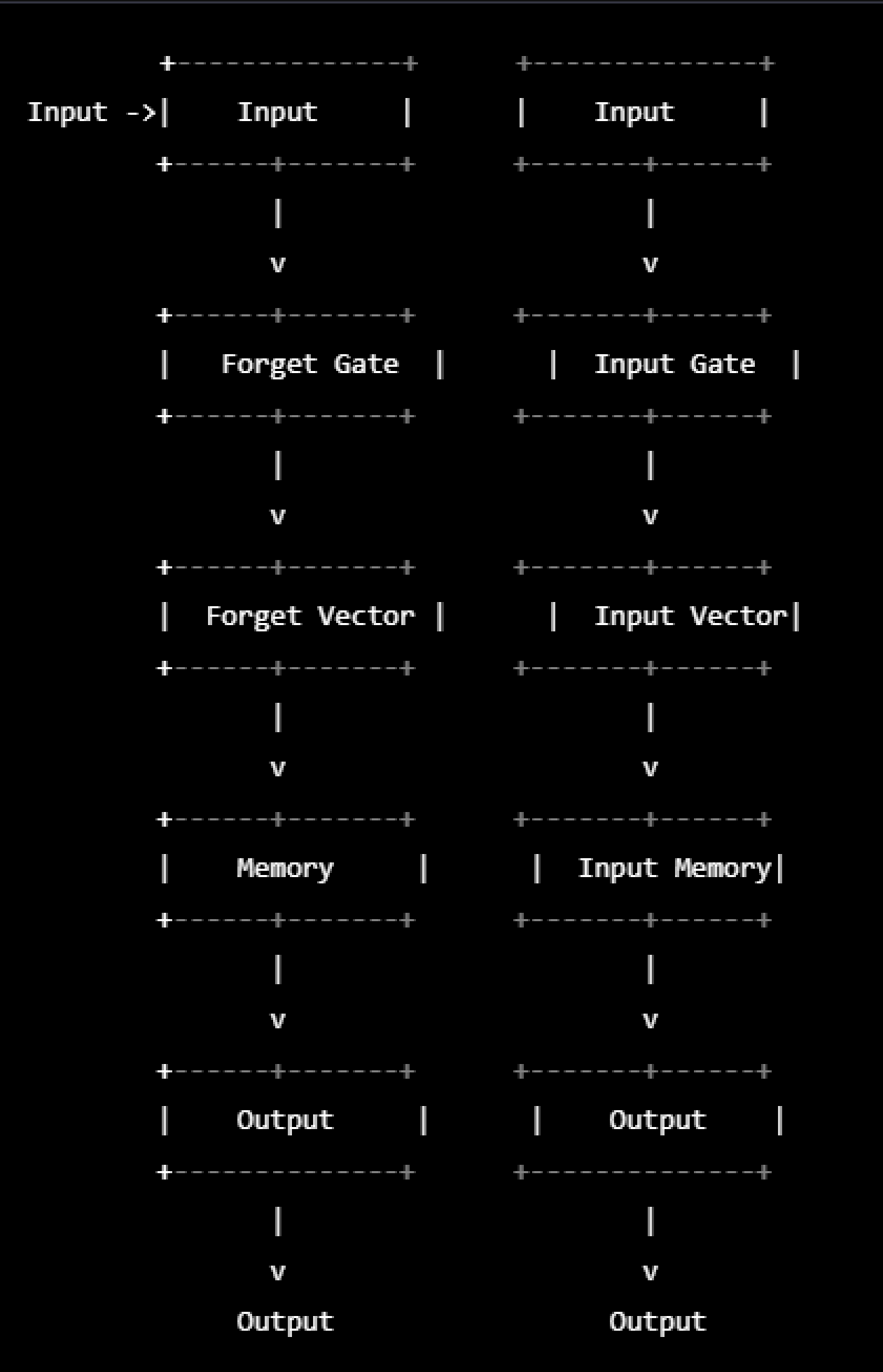
| Code | Details   |
|------|---|
| 200  | <div>Response body</div> <div><pre>{   "ok": true,   "code": 200,   "data": {     "original_text": "tidurnya kotor bantalnya bau badan gatal kayak hewannya tidur nyenyak",     "sentiment": "neutral"   },   "message": "Success" }</pre></div> <div>Download</div> <div>Response headers</div> <div><pre>content-length: 161 content-type: application/json date: Mon, 10 Apr 2023 15:24:37 GMT server: uvicorn</pre></div> |

## Long Short Term Memory Networks (LSTM)

LSTM adalah model yang dapat mengingat informasi dari peristiwa masa lalu dalam urutan dan menggunakannya untuk menginformasikan prediksi untuk peristiwa masa depan. Hal ini dilakukan dengan menggunakan "sel memori" yang dapat selektif menyimpan atau mengabaikan informasi, dan "pintu" (gates) yang mengontrol aliran informasi ke dan dari sel memori.

LSTM sendiri merupakan penerapan turunan dari RNN untuk menjawab masalah vanishing gradient yang dapat muncul dalam arsitektur RNN.

[Implementasi LSTM dalam studi ini dapat dilihat pada halaman-halaman selanjutnya ...](#)



```
[ ] #training
import numpy as np
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, SpatialDropout1D, SimpleRNN, Activation
from tensorflow.keras import optimizers
from tensorflow.keras.callbacks import EarlyStopping, TensorBoard
from tensorflow.keras.layers import Flatten
from tensorflow.keras import backend as K

embed_dim = 100
units = 64

model_lstm = Sequential()
model_lstm.add(Embedding(max_features, embed_dim, input_length=X.shape[1]))
model_lstm.add(LSTM(units, dropout=0.2))
model_lstm.add(Dense(3, activation='softmax'))

adam = optimizers.Adam(lr=0.001)
model_lstm.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
print(model_lstm.summary())

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=3)
history = model_lstm.fit(X_train, y_train, epochs=10, batch_size=10, validation_data=(X_test, y_test), verbose=1, callbacks=[es])
```

Proses training dilakukan terhadap model LSTM dengan menggunakan data-data yang telah di-define sebelumnya.

```
[ ] model_lstm.save('model_lstm.h5')
print("Model has been created !")
```

Model LSTM kemudian disimpan/di-export untuk penggunaan lebih lanjut



```
#evaluation
from sklearn import metrics

predictions = model_lstm.predict(X_test)
y_pred= predictions
matrix_test = metrics.classification_report(y_test.argmax(axis=1), y_pred.argmax(axis=1))
print("Testing Done !")
print(matrix_test)
```

Terhadap model LSTM dilakukan evaluasi untuk mengukur akurasi dalam skor precision, recall, and F1-score bagi ketiga label ("Positive", "Negative", dan "Neutral") beserta rata-rata setiap skor.



## Hasil evaluasi skor akurasi model LSTM dan uji coba prediksi labe (1)

```
#evaluation
from sklearn import metrics

predictions = model_lstm.predict(X_test)
y_pred= predictions
matrix_test = metrics.classification_report(y_test.argmax(axis=1), y_pred.argmax(axis=1))
print("Testing Done !")
print(matrix_test)
```

169/169 [=====] - 2s 13ms/step  
Testing Done !

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.80      | 0.85   | 0.82     | 2274    |
| 1            | 0.72      | 0.61   | 0.66     | 353     |
| 2            | 0.85      | 0.83   | 0.84     | 2759    |
| accuracy     |           |        | 0.82     | 5386    |
| macro avg    | 0.79      | 0.76   | 0.78     | 5386    |
| weighted avg | 0.82      | 0.82   | 0.82     | 5386    |

# Hasil evaluasi skor akurasi model LSTM dan uji coba prediksi label (2)

GET

/sentiment-lstm Sentiment Analytics Lstm

Parameters

Cancel

| Name  | Description                                      |
|---|--|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="makanannya gak enak"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/sentiment-lstm?text=makanannya%20gak%20enak' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-lstm?text=makanannya%20gak%20enak
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "neutral"   },   "message": "Success" }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 68 content-type: application/json date: Mon, 10 Apr 2023 12:21:49 GMT server: uvicorn</pre></div></div> |

# Hasil evaluasi skor akurasi model LSTM dan uji coba prediksi label (3)

GET

/sentiment-lstm Sentiment Analytics Lstm

Parameters

Cancel

| Name  | Description                                       |
|---|---|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="bajunya bagus banget"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/sentiment-lstm?text=bajunya%20bagus%20banget' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-lstm?text=bajunya%20bagus%20banget
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "neutral"   },   "message": "Success" }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 68 content-type: application/json date: Mon, 10 Apr 2023 12:23:25 GMT server: uvicorn</pre></div></div> |

Hasil evaluasi skor akurasi model LSTM dan uji coba prediksi label (4), menggunakan input file

POST

/sentiment-lstm-file

Sentiment Analytics Lstm File

Parameters

No parameters

Cancel

Reset

Request body

required

multipart/form-data

file \* required

string(\$binary)

Choose File

sample\_data\_input.csv

Execute

Clear

Responses

Curl

curl -X 'POST' \
'http://127.0.0.1:8000/sentiment-lstm-file' \
-H 'accept: application/json' \
-H 'Content-Type: multipart/form-data' \
-F 'file=@sample\_data\_input.csv;type=text/csv'

Request URL

http://127.0.0.1:8000/sentiment-lstm-file

Server response

Code

Details

200

Response body

{
 "ok": true,
 "code": 200,
 "data": {
 "original\_text": "tidurnya kotor bantalnya bau badan gatal kayak hewannya tidur nyenyak",
 "sentiment": "neutral"
 },
 "message": "Success"
}

Download

Response headers

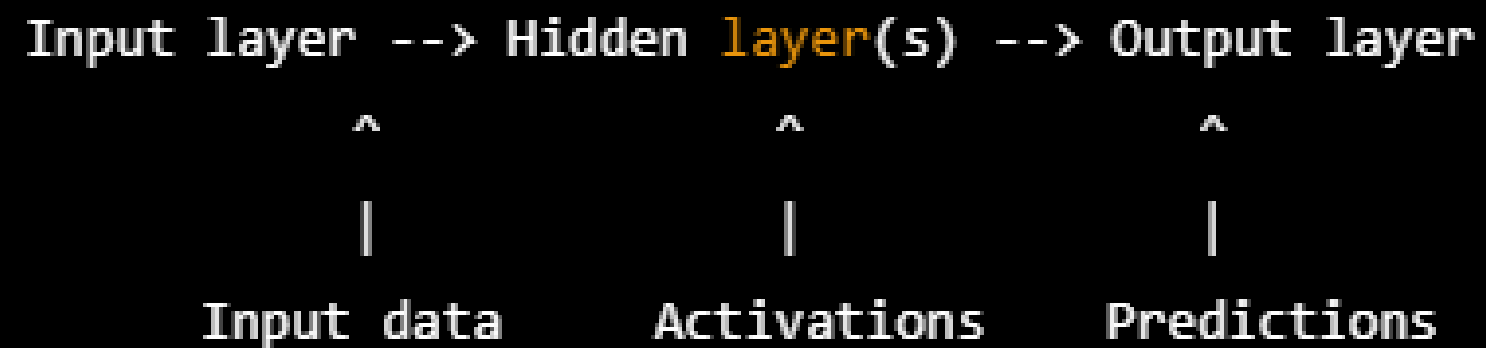
content-length: 161
content-type: application/json
date: Mon, 10 Apr 2023 15:23:29 GMT
server: uvicorn

## MultiLayer Perception (MLP)

MLP adalah model arsitektur yang umum digunakan untuk task klasifikasi dan regresi dalam machine learning.

MLP terdiri dari satu input layer, satu atau lebih hidden layer, dan satu output layer. Selama training, MLP diberikan input data, untuk kemudian ditransformasi menggunakan activation function pada setiap hidden layer dan diteruskan ke layer berikutnya.

Implementasi LSTM dalam studi ini dapat dilihat pada halaman-halaman selanjutnya ...



```
[ ] #training
from sklearn.neural_network import MLPClassifier

model_mlp = MLPClassifier()
model_mlp.fit(X_train, y_train)

print("Training Selesai !")
```

Proses training dilakukan terhadap model MLP dengan menggunakan data-data yang telah di-define sebelumnya.

```
[ ] pickle.dump(model_mlp, open("model_mlp.p", "wb"))
```

Model MLP kemudian disimpan/di-export untuk penggunaan lebih lanjut

```
[ ] #evaluations
from sklearn.metrics import classification_report

test = model_mlp.predict(X_test)

print("Testing Selesai !")

print(classification_report(y_test, test))
```

Terhadap model RNN dilakukan evaluasi untuk mengukur akurasi dalam skor precision, recall, and F1-score bagi ketiga label ("Positive", "Negative", dan "Neutral") beserta rata-rata setiap skor.

## Hasil evaluasi skor akurasi model MLP dan uji coba prediksi label (1)

```
[ ] #evaluations
    from sklearn.metrics import classification_report

    test = model_mlp.predict(X_test)

    print("Testing Selesai !")

    print(classification_report(y_test, test))
```

Testing Selesai !

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| negative     | 0.79      | 0.81   | 0.80     | 2356    |
| neutral      | 0.76      | 0.57   | 0.65     | 317     |
| positive     | 0.81      | 0.82   | 0.82     | 2713    |
| accuracy     |           |        | 0.80     | 5386    |
| macro avg    | 0.79      | 0.73   | 0.76     | 5386    |
| weighted avg | 0.80      | 0.80   | 0.80     | 5386    |

# Hasil evaluasi skor akurasi model MLP dan uji coba prediksi label (2)

GET

/sentiment-mlp

Sentiment Analytics Mlp

^

Parameters

Cancel

| Name  | Description                                      |
|---|--|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="makanannya gak enak"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/sentiment-mlp?text=makanannya%20gak%20enak' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-mlp?text=makanannya%20gak%20enak
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "negative"   },   "message": "Success" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 69 content-type: application/json date: Mon, 10 Apr 2023 12:24:50 GMT server: uvicorn</pre></div></div> |



# Hasil evaluasi skor akurasi model MLP dan uji coba prediksi label (3)

GET

/sentiment-mlp Sentiment Analytics Mlp

^

Parameters

Cancel

| Name  | Description                                       |
|---|---|
| <b>text</b> * required<br>string<br>(query) | <input type="text" value="bajunya bagus banget"/> |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/sentiment-mlp?text=bajunya%20bagus%20banget' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/sentiment-mlp?text=bajunya%20bagus%20banget
```

Server response

| Code | Details  |
|------|--|
| 200  | <div><div>Response body</div><div><pre>{   "ok": true,   "code": 200,   "data": {     "data": "positive"   },   "message": "Success" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 69 content-type: application/json date: Mon, 10 Apr 2023 12:25:47 GMT server: uvicorn</pre></div></div> |

# Hasil evaluasi skor akurasi model MLP dan uji coba prediksi label (4), menggunakan input file

POST

/sentiment-mlp-file

Sentiment Analytics Mlp File

^

Parameters

No parameters

Cancel

Reset

Request body required

multipart/form-data

file \* required

string(\$binary)

Choose File

sample\_data\_input.csv

Execute

Clear

Responses

Curl

curl -X 'POST' \  
'http://127.0.0.1:8000/sentiment-mlp-file' \  
-H 'accept: application/json' \  
-H 'Content-Type: multipart/form-data' \  
-F 'file=@sample\_data\_input.csv;type=text/csv'

Request URL

http://127.0.0.1:8000/sentiment-mlp-file

Server response

Code

Details

200

Response body

```
{  
  "ok": true,  
  "code": 200,  
  "data": {  
    "original_text": "tidurnya kotor bantalnya bau badan gatal kayak hewannya tidur nyenyak",  
    "sentiment": "negative"  
  },  
  "message": "Success"  
}
```

Download

Response headers

```
content-length: 162  
content-type: application/json  
date: Mon, 10 Apr 2023 15:20:31 GMT  
server: uvicorn
```

# KESIMPULAN

- 1 Metoda MLP, LSTM, dan RNN masing-masing memberikan hasil yang cukup baik dalam klasifikasi teks. Namun, pada beberapa kasus MLP memberikan hasil yang lebih baik dari pada LSTM atau RNN.
- 2 Pada kasus yang sama sebaiknya dilakukan balancing dahulu pada data, agar mendapatkan hasil klasifikasi yang lebih signifikan.
- 3 Precision, recall, f1-score pada setiap kelas cukup tinggi, menandakan bahwa metode yang digunakan mampu mengelompokkan teks ke dalam kelas yang tepat dengan tingkat keakuratan yang tinggi

**THANK  
YOU**