



Analisis Sentimen

Pramudya Rachadiansyach Putra
Yosef Bintang Pakpahan
Puji Maryane Putri Nabila
Ambika Arya Maheswara

PENDAHULUAN

Era teknologi yang berkembang pesat, melahirkan berbagai website dan aplikasi yang menjadi wadah bagi masyarakat untuk berdiskusi dan mengungkapkan pendapat. Informasi yang terkandung di dalamnya sangat berharga, menggambarkan sentimen khalayak terhadap berbagai isu. Sentimen pengguna tidak hanya memberikan umpan balik, tetapi juga membantu dalam pengambilan keputusan di masa depan.

Twitter, sebagai salah satu platform media sosial paling populer di Indonesia dengan lebih dari 18 juta pengguna pada tahun 2022 menurut We Are Social, menjadi pusat perhatian penelitian ini. Tweet, atau pesan singkat di Twitter, memiliki beragam variasi, termasuk teks. Melalui penelitian ini, kami bertujuan membentuk model Artificial Neural Network dan Long Short-Term Memory (LSTM) Neural Network. Model ini akan mengklasifikasikan teks tweet ke dalam tiga sentimen: negatif, netral, dan positif. Penelitian ini menjadi langkah signifikan dalam memahami dinamika opini publik di kancah media sosial yang dinamis di Indonesia.

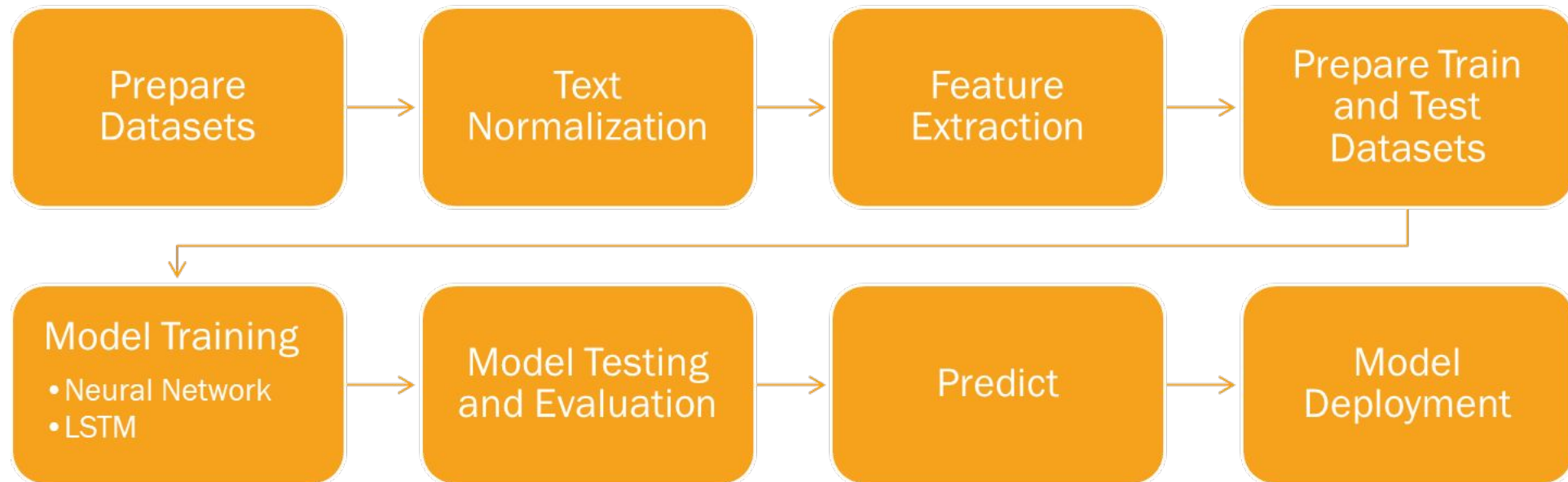


METODE

PENELITIAN

METODE PENELITIAN:

TAHAPAN PENELITIAN



METODE PENELITIAN

Prepare Datasets

Dataset untuk membuat model analisis sentimen

	text	label
0	warung ini dimiliki oleh pengusaha pabrik tahu...	positive
1	mohon ulama lurus dan k212 mmbri hujjah partai...	neutral
2	lokasi strategis di jalan sumatera bandung . t...	positive
3	betapa bahagia nya diri ini saat unboxing pake...	positive
4	duh . jadi mahasiswa jangan sombong dong . kas...	negative
...
10995	tidak kecewa	positive
10996	enak rasa masakan nya apalagi keping yang me...	positive
10997	hormati partai-partai yang telah berkoalisi	neutral
10998	pagi pagi di tol pasteur sudah macet parah , b...	negative
10999	meskipun sering belanja ke yogya di riau junct...	positive

11000 rows x 2 columns

Dataset untuk analisis sentimen

abusive.csv

citation.bib

data.csv

new_kamusalay.csv

README.md

	text
0	- disaat semua cowok berusaha melacak perhatia...
1	RT USER: USER siapa yang telat ngasih tau elu?...
2	41. Kadang aku berfikir, kenapa aku tetap perc...
3	USER USER AKU ITU AKU\In\InKU TAU MATAMU SIBIT T
4	USER USER Kaum cebong kapir udah ke

df_test.shape
(13169, 1)

	slang	normal
0	anakjakartaasik	anak jakarta asyik asyik
1	pakcikdahtua	pak cik sudah tua
2	pakcikmudalagi	pak cik muda lagi
3	t3tapjokowi	tetap jokowi
4	3x	tig

df_slang.shape
(15167, 2)

+ Stopwords by Sastrawi

METODE PENELITIAN

Text Normalization

Check Null Values

```
-----  
Training dataset Null Values :  
text      0  
label     0  
dtype: int64  
-----
```

```
-----  
Test dataset Null Values :  
text      0  
dtype: int64  
-----
```

Drop Duplicated Values

Before:

```
-----  
Training dataset duplicated Values : 67  
-----
```

```
-----  
Test dataset duplicated Values : 146  
-----
```

After:

```
-----  
Jumlah Duplicated Values pada Training Dataset setelah dibersihkan : 0  
-----
```

```
-----  
Jumlah Duplicated Values pada Test Dataset setelah dibersihkan : 0  
-----
```

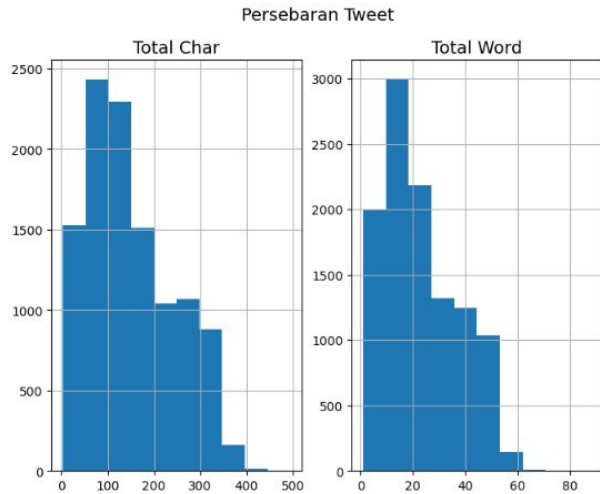
Cleansing Data

1. Regex
 - Lowercase
 - Remove emoji, url, non-alphanumeric
 - etc
2. Slang Words Normalization
3. Stopwords Removal

**Normal Text
using
Exploratory
Data Analysis
(EDA)**

Text Normalization Result

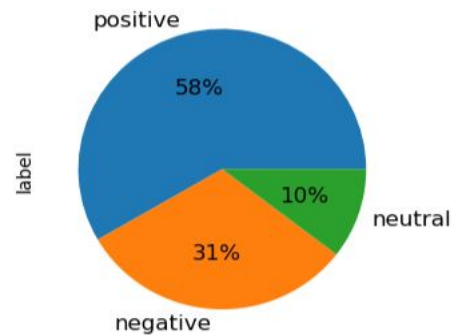
The Description of Dataset to create a sentiment analysis model



```
Mean: total char 151.565718
      total word  23.168206
```



Proporsi Label Data Train



The Description of Dataset to create a sentiment analysis



METODE PENELITIAN

Feature Extraction

```
# Vectorization --> dimana text "diubah" menjadi vector/numerical  
# Nama dari proses ini adalah Feature Extraction  
  
count_vect = CountVectorizer()  
count_vect.fit(data_preprocessed)  
  
X = count_vect.transform(data_preprocessed)
```


METODE PENELITIAN

Prepare Train & Test Datasets

```
from sklearn.model_selection import train_test_split

classes = df_train['sentiment']

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

Neural network
menggunakan sklearn

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

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

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=1)
```

LSTM menggunakan
pickle

METODE PENELITIAN

Model Training: Neural Network

Neural network yang digunakan pada penelitian ini adalah Artificial Neural Network dan Long Short Term Memory Neural Network

METODE PENELITIAN

Model Training: LSTM

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 88, 256)	25600000
spatial_dropout1d (Spatial Dropout1D)	(None, 88, 256)	0
lstm (LSTM)	(None, 32)	36992
dense (Dense)	(None, 3)	99

METODE PENELITIAN

Model Testing & Evaluation: Neural Network

Teknik pengukuran dalam model testing & evaluation

- Accuracy
- Confusion Matrix
- F-Measure

METODE PENELITIAN

Model Testing & Evaluation: LSTM

Testing selesai					
	precision	recall	f1-score	support	
0	0.83	0.82	0.82	668	
1	0.78	0.78	0.78	231	
2	0.92	0.93	0.92	1288	
accuracy			0.88	2187	
macro avg	0.84	0.84	0.84	2187	
weighted avg	0.88	0.88	0.88	2187	

Evaluasi menggunakan K-fold Cross Validation dibagi menjadi 5 training dan menghasilkan rata-rata accuracy = 0.883310470964792

METODE PENELITIAN

Predict: Neural Network

Model predict berfungsi untuk melakukan tes apakah model yang kita buat bisa “berjalan” dengan baik atau tidak.

METODE PENELITIAN

Predict: LSTM

```
input_text = """
cinta memang buta
"""

sentiment = ['negative', 'neutral', 'positive']

text = Cleansing(input_text)
text = [text]
predicted = tokenizer.texts_to_sequences(text)
guess = pad_sequences(predicted, maxlen=X.shape[1])

model = load_model('model.h5')
prediction = model.predict(guess)
polarity = np.argmax(prediction[0])

print("Text: ",text[0])
print("Sentiment: ",sentiment[polarity])

1/1 [=====] - 1s 515ms/step
Text: cinta memang buta
Sentiment: positive
```

METODE PENELITIAN

Model Deployment

```
# Swagger UI
app = Flask(__name__)
app.json_encoder = LazyJSONEncoder
swagger_template = dict(
    info = {
        'title' : LazyString(lambda: 'API Documentation for Deep Learning API'),
        'version' : LazyString(lambda: '1.0.0'),
        'description' : LazyString(lambda: 'Dokumentasi API untuk Deep Learning API')
    },
    host = LazyString(lambda: request.host)
)

swagger_config = {
    'headers' : [],
    'specs': [
        {
            'endpoint' : 'docs',
            'route' : '/docs.json'
        }
    ],
    'static_url_path' : '/flaskgger_static',
    'swagger_ui' : True,
    'specs_route' : '/docs/'
}

swagger = Swagger(app, template = swagger_template,
                  config = swagger_config)
```




HASIL DAN

PEMBAHASAN

HASIL DAN PEMBAHASAN

Neural Network Calculation

Initial Value

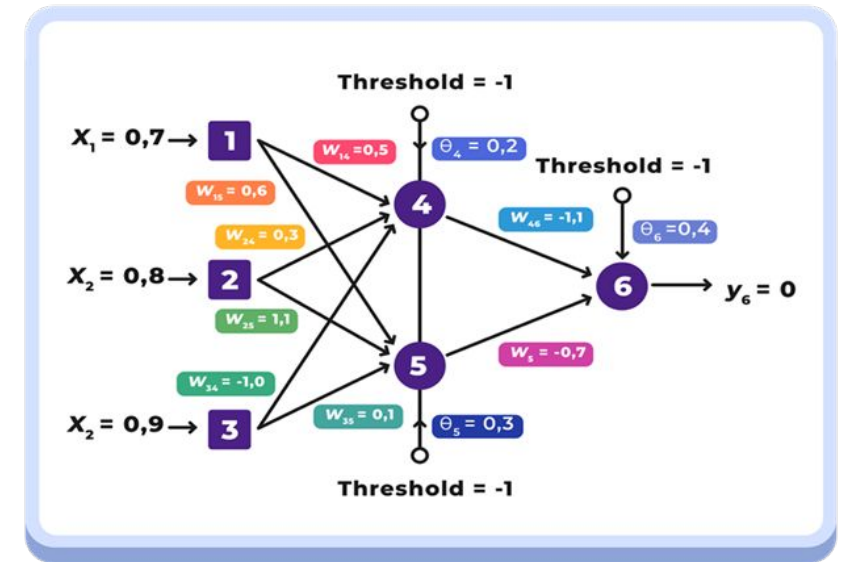
x_1	x_2	x_3	α	Threshold	$Y_{d,6}$
0.7	0.8	0.9	0.1	-1	0

Initial Random

W_{14}	W_{15}	W_{24}	W_{25}	W_{34}	W_{35}	W_{46}	W_{56}	θ_4	θ_5	θ_6
0.5	0.6	0.3	1.1	-1	0.1	-1.1	-0.7	0.2	0.3	0.4

Updated Weights

W_{14}	W_{15}	W_{24}	W_{25}	W_{34}	W_{35}	θ_4	θ_5	θ_6
0.5006	0.6003	0.3007	1.1003	-0.9992	0.1004	0.1991	0.2995	0.4034



Selisih antara Initial Value dengan Updated Weight minim = Data pelatihan konsisten

HASIL DAN PEMBAHASAN

Neural Network

Training

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

[ ] model = MLPClassifier()
    model.fit(X_train, y_train)
```

▼ MLPClassifier
MLPClassifier()

Testing

```
[ ] # dump model ke dalam file pickle
    # Menyimpan data yang sudah diuji coba dalam sebuah file

    pickle.dump(model, open("model.p", "wb"))

▶ # testing


    from sklearn.metrics import classification_report

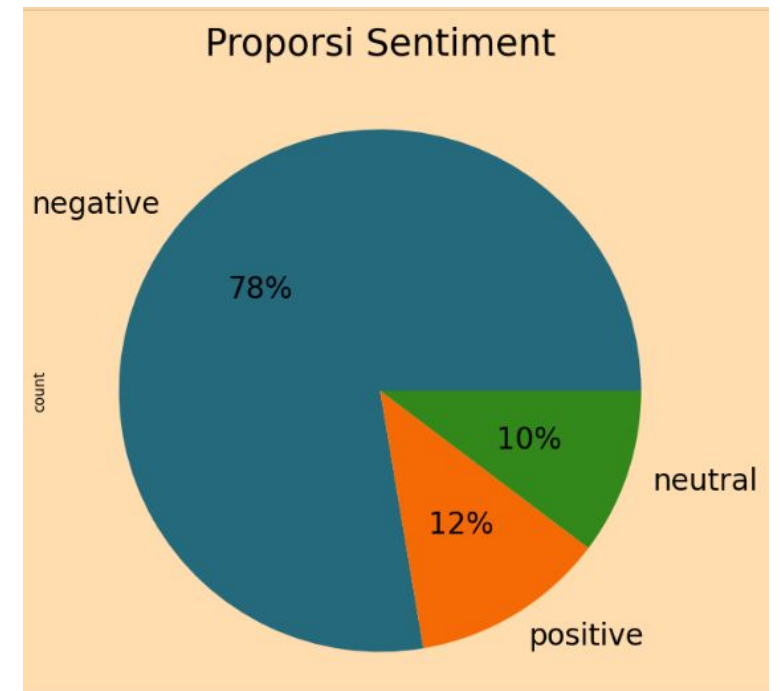
    test = model.predict(X_test)
```

HASIL DAN PEMBAHASAN

Neural Network

Hasil Testing

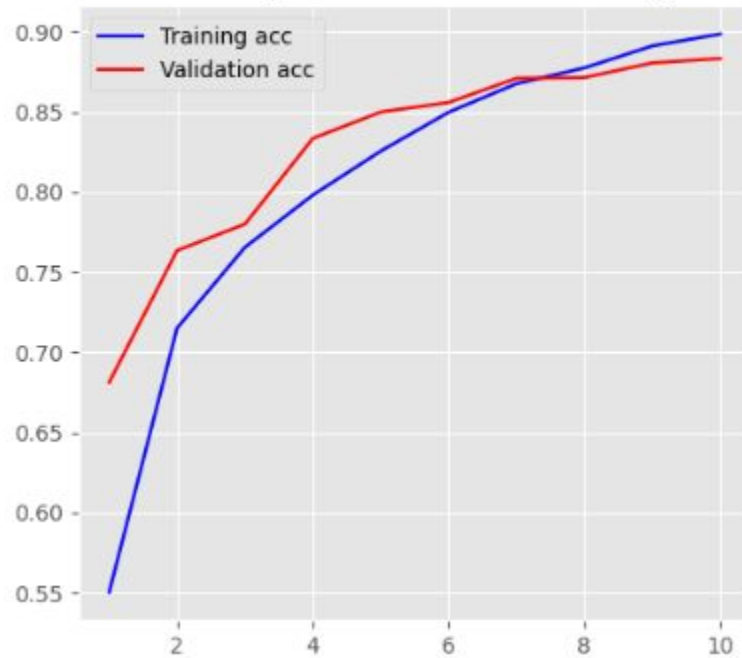
	<code>print(classification_report(y_test, test))</code>			
	precision	recall	f1-score	support
negative	0.77	0.81	0.79	686
neutral	0.80	0.63	0.71	242
positive	0.88	0.90	0.89	1272
accuracy			0.84	2200
macro avg	0.82	0.78	0.80	2200
weighted avg	0.84	0.84	0.84	2200



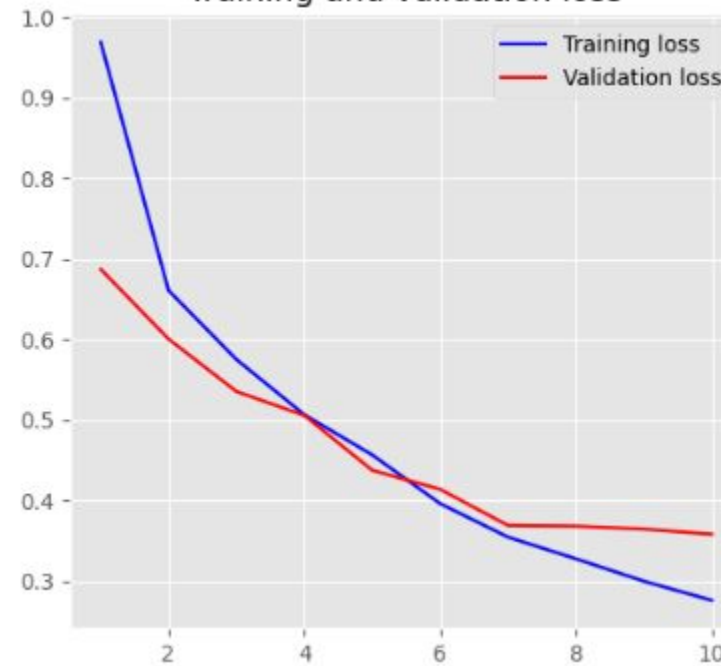
HASIL DAN PEMBAHASAN

LSTM

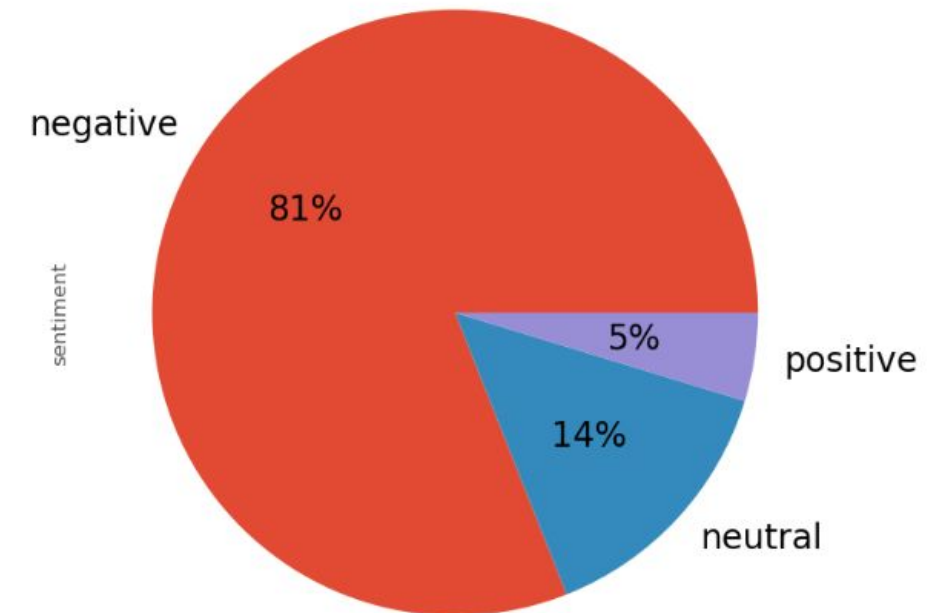
Training and validation accuracy



Training and validation loss



Proporsi Label



KESIMPULAN

Rata-rata metrics akurasi kedua model, yaitu ANN dan LSTM, masing-masing menunjukkan nilai 84,7% dan 88,3%. Nilai tersebut sudah dapat terbilang cukup bagus sehingga model yang telah dibuat dapat diaplikasikan pada dataset testing yang menghasilkan klasifikasi sentimen dengan proporsi pada pie chart yang telah dihasilkan.

Proporsi sentimen dan proporsi label menunjukkan mayoritas data validasi memiliki sentimen negatif. Hal tersebut wajar dikarenakan data validasi yang digunakan merupakan data tweet yang sudah dioptimasi untuk pendeteksian hate speech dan abusive tweet.