

Sentiment Model dan API untuk Analisis Sentimen

Anggota:

- Aditya Fadillah Naim Ahmad
- Jose Alfred Benaya
- Thomas Ken Ronaldi

By Group 3

Data Science Wave 11



Pendahuluan

Dalam era digital yang semakin berkembang, data yang berasal dari platform online seperti media sosial, forum, situs berita, dan ulasan produk telah menjadi sumber informasi yang berlimpah. Informasi ini tidak hanya mencerminkan perasaan dan opini individu, tetapi juga menciptakan jejak digital yang kaya akan wawasan tentang bagaimana orang-orang merespons berbagai aspek kehidupan mereka, termasuk produk, layanan, merek, isu sosial, dan banyak lagi.

Analisis sentimen adalah alat yang penting dalam memahami dan mengeksplorasi kerumitan dunia digital ini. Ini merupakan pendekatan analitis yang digunakan untuk mengidentifikasi, mengukur, dan memahami perasaan, opini, dan sikap yang terkandung dalam teks dan konten digital. Dengan menganalisis sentimen ini, kita dapat mengungkapkan wawasan berharga tentang bagaimana masyarakat merespons berbagai isu dan topik yang relevan.



Model analisis sentimen, yang merupakan bagian integral dari bidang pengolahan bahasa alami (Natural Language Processing, NLP), memungkinkan komputer untuk memproses dan memahami makna di balik kata-kata manusia. Ini memungkinkan berbagai aplikasi penting di berbagai sektor, termasuk bisnis, pemasaran, politik, penelitian pasar, dan banyak lagi. Model-model ini membantu organisasi dan individu untuk merespons perubahan opini publik, meningkatkan layanan pelanggan, memantau merek, dan mengambil keputusan berdasarkan analisis sentiment

Dalam penelitian ini, kami akan mencoba menganalisis dataset IndoNLP yang berasal dari Hugging face dan membuat model *sentiment analysis* dari data set tersebut. menginvestigasi tweet di Indonesia yang memanfaatkan Data Science dan teknologi seperti Python FAST API, Jupyter Notebook, IDE Pycharm. kami akan melakukan pembersihan data, analisis deskriptif, dan menampilkan data yang dianalisis secara deskriptif.

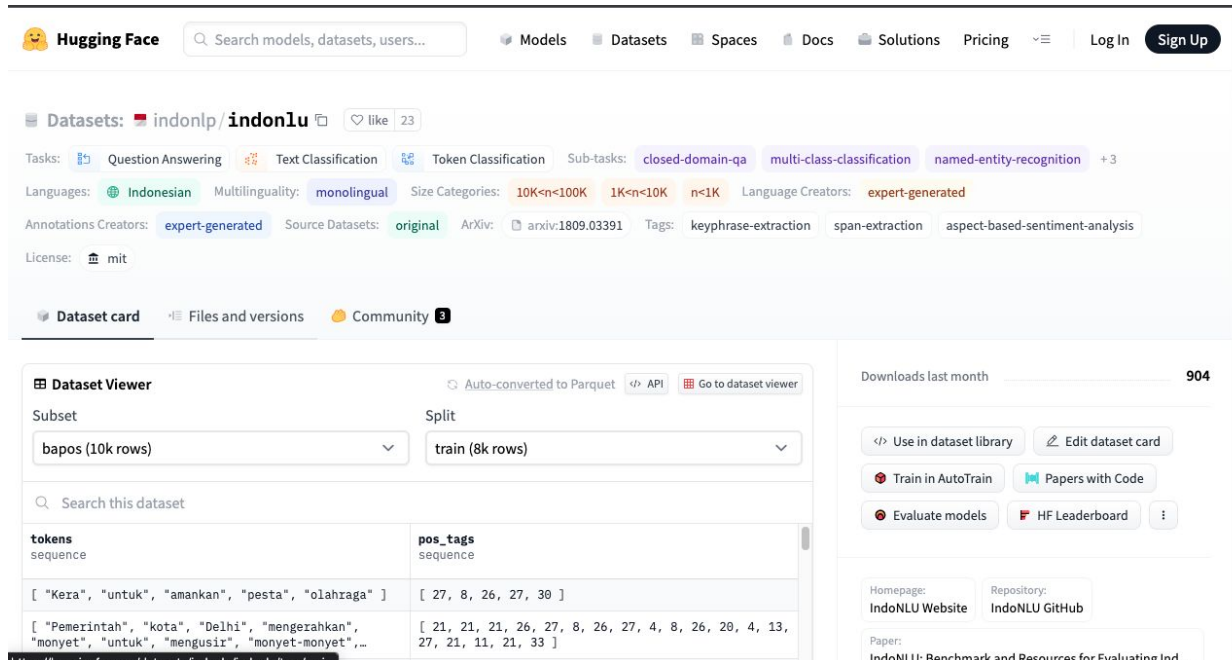
Studi ini didasarkan pada kumpulan dataset IndoNLP dari hugging face. Dataset tersebut kami lakukan analisis deskriptif dalam bentuk visual dan selanjutnya dilakukan Analisis Sentimen menggunakan metode Neural Network dan LSTM



Dataset

Data yang diambil merupakan data Indonlp/indonlu yang diakses pada lama Hugging face pada link:

<https://huggingface.co/datasets/indonlp/indonlu>



The screenshot shows the Hugging Face dataset page for IndonLP/IndoNLU. The page includes a search bar, navigation links (Models, Datasets, Spaces, Docs, Solutions, Pricing, Log In, Sign Up), and a dataset card for 'indonlp/indonlu'. The dataset card displays various filters such as Tasks (Question Answering, Text Classification, Token Classification), Languages (Indonesian), and Size Categories (10K<n<100K, 1K<n<10K, n<1K). It also shows the number of likes (23) and the license (MIT). The 'Dataset Viewer' section is visible, showing a subset of 'bapos (10k rows)' and a split of 'train (8k rows)'. The viewer displays a table with 'tokens sequence' and 'pos_tags sequence'.

tokens sequence	pos_tags sequence
["Kera", "untuk", "amankan", "pesta", "olahraga"]	[27, 8, 26, 27, 30]
["Pemerintah", "kota", "Delhi", "mengarahkan", "monyet", "untuk", "mengusir", "monyet-monyet", ...]	[21, 21, 21, 26, 27, 8, 26, 27, 4, 8, 26, 20, 4, 13, 27, 21, 11, 21, 33]

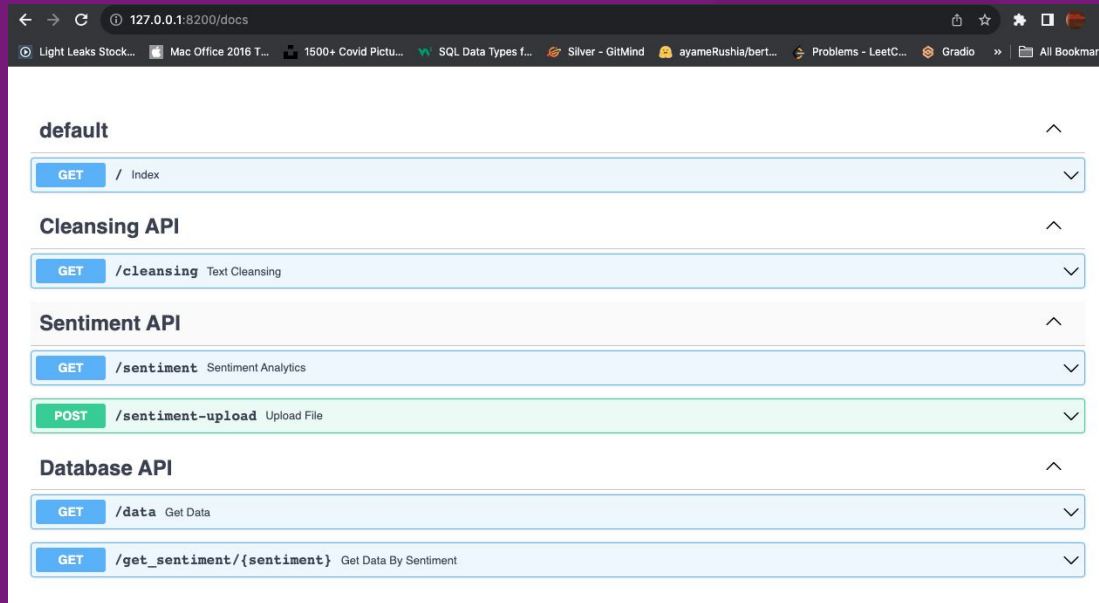


Dari dataset tersebut ditemukan sebanyak **12760** data dengan 2 fitur yaitu, **text** dan **label**

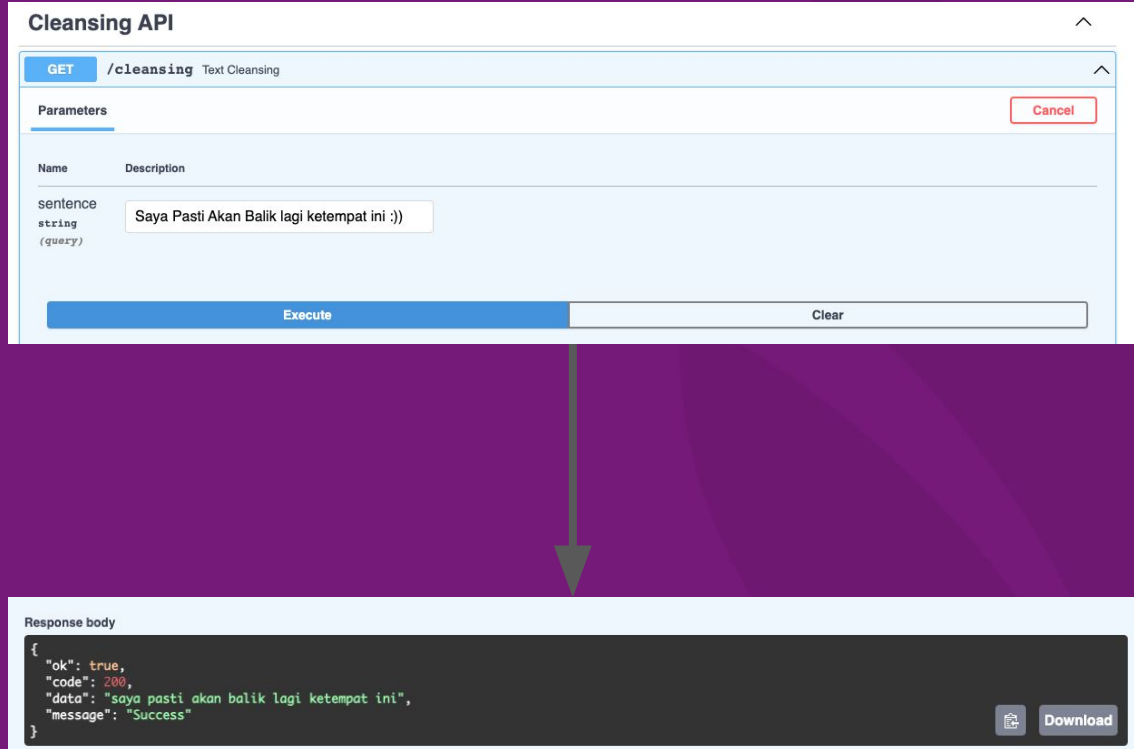
API yang dibangun ada 2, yaitu
Cleansing API, Sentiment API.

Cleansing API akan menerima
input berupa teks yang
selanjutnya dibersihkan menjadi
teks tanpa spesial karakter.

Sentiment API bertanggung
jawab untuk menerima file
upload yang akan dihitung skor
apakah nilai teks tersebut, positif
atau negatif



pada Cleansing API, Bentuk teks yang mengandung icon ataupun tanda baca akan di cleansing seperti pada gambar berikut



The screenshot displays the 'Cleansing API' interface. At the top, it shows the method 'GET' and the endpoint '/cleansing' with a description 'Text Cleansing'. Below this is a 'Parameters' section with a table containing one row: 'sentence' (string, query) with the value 'Saya Pasti Akan Balik lagi ketempat ini :)'. A large blue 'Execute' button is visible. Below the interface, a green arrow points to the 'Response body' section, which shows a JSON object: { "ok": true, "code": 200, "data": "saya pasti akan balik lagi ketempat ini", "message": "Success" }. A 'Download' button is located at the bottom right of the response body.

Cleansing API

GET /cleansing Text Cleansing

Parameters

Name	Description
sentence string (query)	Saya Pasti Akan Balik lagi ketempat ini :))

Execute Clear

Response body

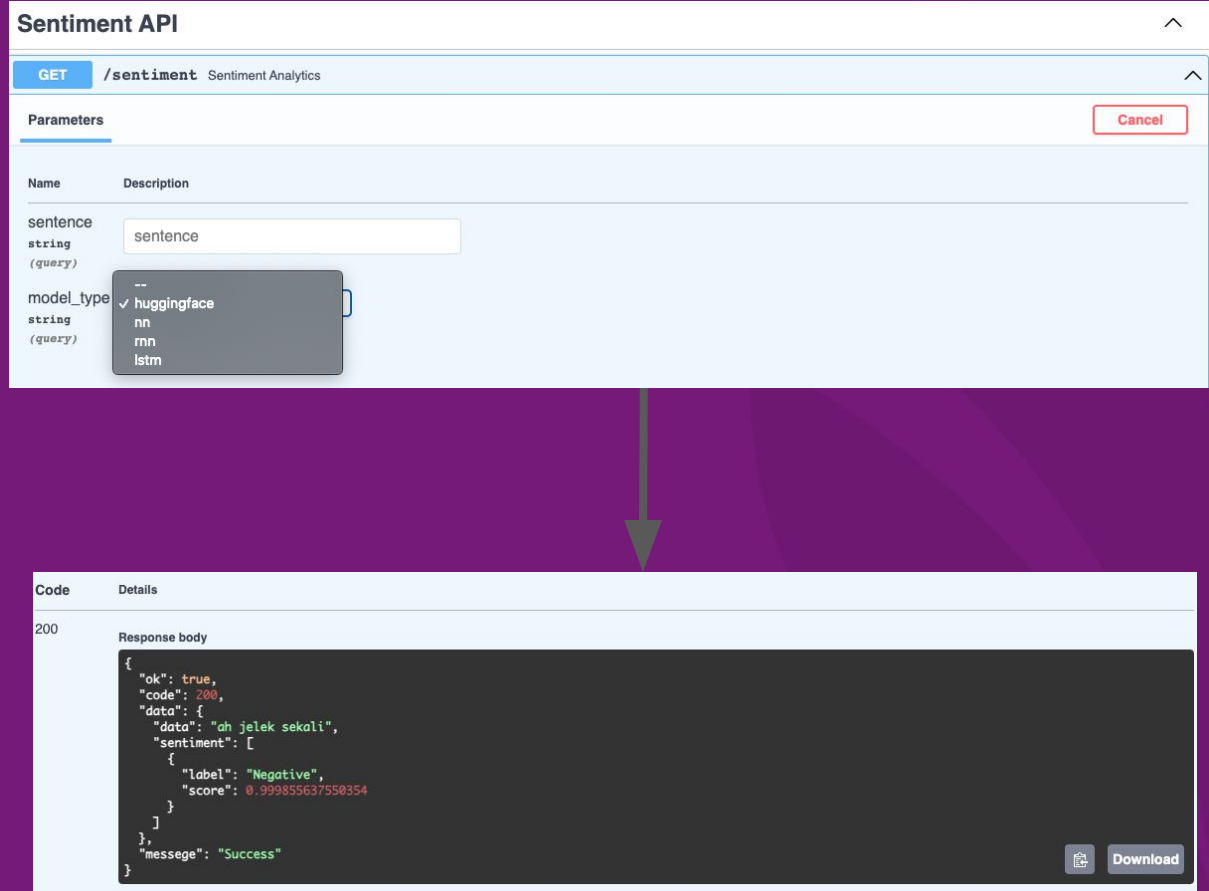
```
{
  "ok": true,
  "code": 200,
  "data": "saya pasti akan balik lagi ketempat ini",
  "message": "Success"
}
```

Download

1. Remove Label 'neutral'
2. Lowercase Letter
3. Remove Stopword
4. Lemmatize Word



Api Ini Memiliki 4 Model yang tersedia, Hugging face, NN, RNN dan LSTM



The screenshot displays the 'Sentiment API' interface. The top section shows the endpoint `/sentiment` with a 'GET' method and 'Sentiment Analytics' description. Below this, the 'Parameters' section is visible, containing two query parameters: `sentence` (string) and `model_type` (string). The `model_type` dropdown menu is open, showing four options: `huggingface` (selected), `nn`, `rnn`, and `lstm`. A green arrow points from the `model_type` dropdown to the 'Response body' section below. The 'Response body' section shows a JSON response with a status of 200, indicating a successful sentiment analysis. The response includes a 'data' object with a 'sentiment' array containing a 'label' of 'Negative' and a 'score' of 0.999855637550354. A 'Download' button is located at the bottom right of the response body.

Sentiment API

GET `/sentiment` Sentiment Analytics

Parameters

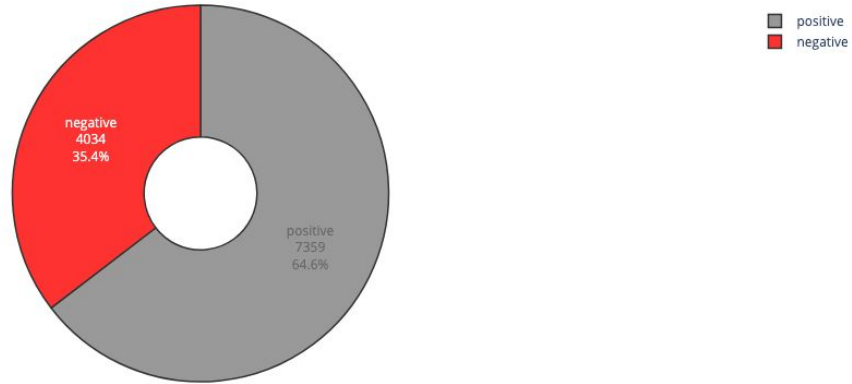
Name	Description
<code>sentence</code> string (query)	sentence
<code>model_type</code> string (query)	<ul style="list-style-type: none">huggingfacennrnnlstm

Response body

```
{
  "ok": true,
  "code": 200,
  "data": {
    "data": "ah jelek sekali",
    "sentiment": [
      {
        "label": "Negative",
        "score": 0.999855637550354
      }
    ]
  },
  "message": "Success"
}
```

Download

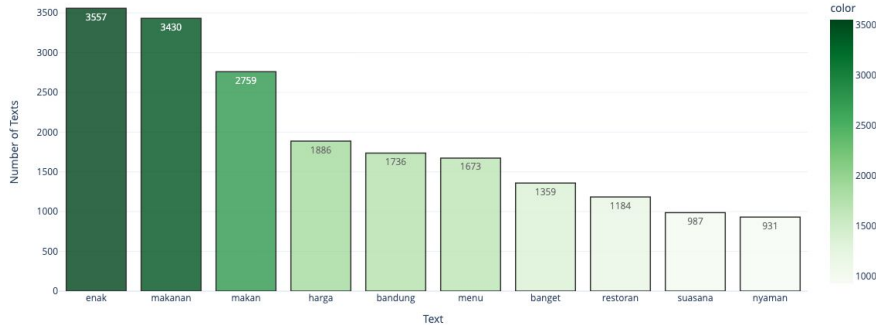
Sentiment Distribution



Dari total **12760** data didapatkan sebaran sebanyak **35,4%** untuk sentiment negatif, sementara untuk sentimen positif, yakni sebanyak **64,6%**

Hasil Analisa terhadap Dataset

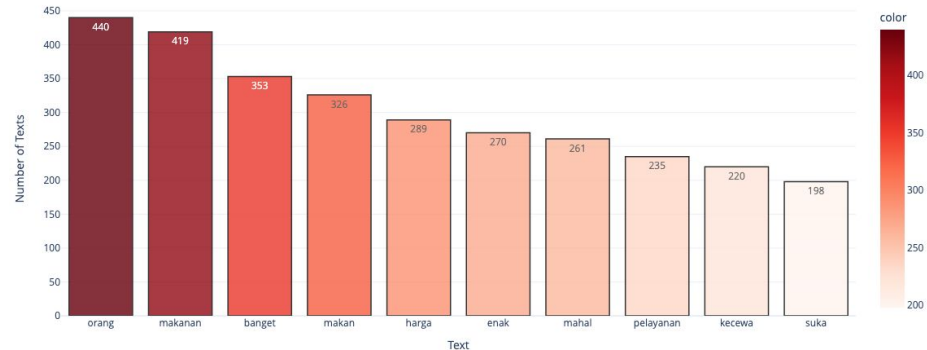
Top 10 Positive Sentiment Text Distribution



Teks dengan predikat sentimen positif terbanyak dipegang oleh kata **'enak'** sebanyak **3557 kata**, diikuti kata **'makanan'**, lalu **'makan'**, dst.

Sedangkan teks dengan nilai sentimen negatif terbanyak cenderung dikaitkan dengan kata **'orang'**, yaitu sebanyak **440 kata**, lalu diikuti dengan kata **'makanan'**, kemudian **'banget'**, dst.

Top 10 Negative Sentiment Text Distribution





Feature Extraction

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 from sklearn.feature_extraction.text import TfidfVectorizer
3
4 # count_vect = TfidfVectorizer()
5 count_vect = CountVectorizer()
6 count_vect.fit(data_preprocessed)
7
8 X = count_vect.transform(data_preprocessed)
9 print("Feature Extraction Done !")
```

```
Feature Extraction Done !
```

Pada dataset ini dilakukan pemrosesan teks dan pengolahan bahasa alami (Natural Language Processing, NLP) menggunakan CountVectorizer dan TfidfVectorizer yang diimport dari library Sklearn

Dataset X dan y akan dibagi menjadi subset pelatihan (80%) dan subset pengujian (20%),

Dalam pengujian Model Statistik dilakukan melalui fungsi train test split untuk mengukur kinerja model pada data yang belum pernah dilihat sebelumnya, sehingga dapat menghindari overfitting dan memberikan perkiraan yang lebih baik tentang seberapa baik model tersebut akan berfungsi dalam situasi dunia nyata

Model Neural Network MLP Classifier

Model *Deep Learning* dibangun menggunakan algoritma *simple Neural Network Multi-layer Perceptron* (MLP)..

Model dievaluasi dengan menggunakan beberapa parameter hasil pada tabel.

	Precision	Recall	F1 Score	Support
negative	0.83	0.85	0.84	792
positive	0.92	0.90	0.91	1487
Accuracy			0.89	2279
Macro avg	0.87	0.88	0.87	2279
Weighted avg	0.89	0.89	0.89	2279

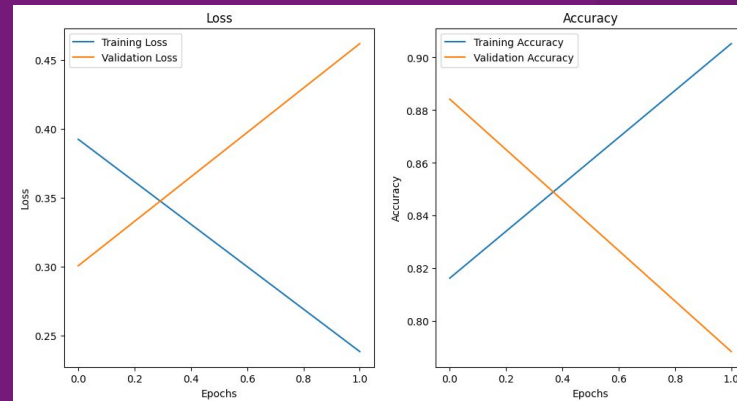
Hasil RNN

Model *Deep Learning* dibangun menggunakan algoritma *Recurrent Neural Network*.

Hyperparameter:

embedding_size = 100
units = 64
drop_out = 0.2
batch_size = 10
epochs=10
learning_rate = 0.01
verbose = 1
optimizer = Adam

Layer (type)	Output Shape	Param #
embedding_6 (Embedding)	(None, 96, 100)	10000000
simple_rnn_4 (SimpleRNN)	(None, 64)	10560
dense_5 (Dense)	(None, 2)	130



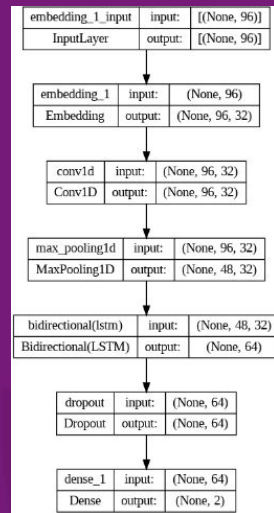
Hasil LSTM

Model *Deep Learning* dibangun menggunakan algoritma *Long-Short Term Memory*.

Hyperparameter yang digunakan:

`vocab_size = 5000`
`embedding_size = 32`
`epochs=20`
`learning_rate = 0.1`
`decay_rate = learning_rate / epochs`
`momentum = 0.8`
`batch size = 16`
`drop_out = 0.4`

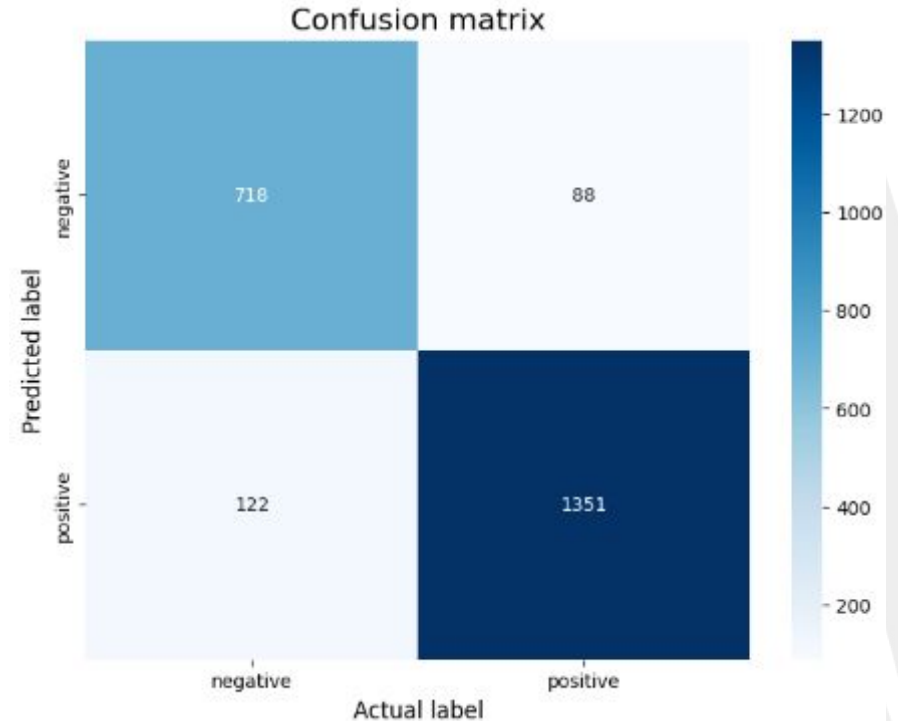
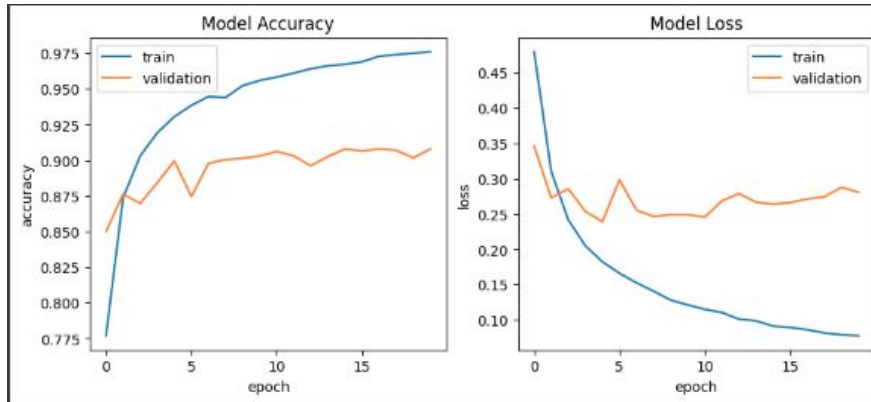
Hasil modeling dievaluasi dengan menggunakan *f1 Score*, menghasilkan akurasi, Prediksi dan recall sebesar 90%



```
1 # Evaluate model on the test set
2 from sklearn.metrics import f1_score
3
4 loss, accuracy, precision, recall = model.evaluate(X_test, y_test, verbose=0)
5 # Print metrics
6 print('')
7 print('Accuracy : {:.4f}'.format(accuracy))
8 print('Precision : {:.4f}'.format(precision))
9 print('Recall : {:.4f}'.format(recall))
```

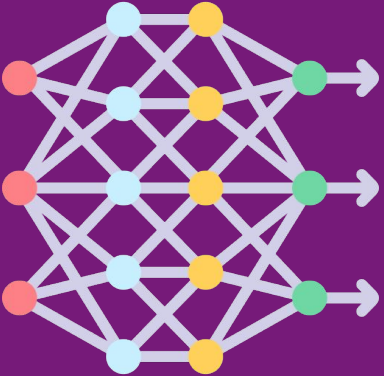
```
Accuracy : 0.9079
Precision : 0.9079
Recall : 0.9079
```

Hasil Modeling dengan menggunakan LSTM menghasilkan Akurasi dan Loss yang cukup baik. Matriks bisa dilihat pada gambar berikut



KESIMPULAN

1. Dari dataset IndoNlu didapatkan kesimpulan bahwa kata terbanyak dalam sentimen positif adalah **'enak'** sebanyak **3557 kata, sedangkan** dalam Sentimen negatif kata terbanyak adalah **'orang'**, yaitu sebanyak **440 kata,**
2. Berdasarkan hasil evaluasi dari 2 model yang digunakan yaitu model *Neural Network* dan *LSTM*, dihasilkan model terbaik adalah *LSTM* dengan evaluasi menggunakan *F1 Score*, menghasilkan akurasi, precision dan recall sebesar 86.9%.



THANK YOU

