

# MODEL LSTM & NEURAL NETWORK

DALAM ANALISA SENTIMEN

Muhammad Nur Faza  
Puspita Laras  
Putri Oktaviani  
Raden Fachry Azwar

Kelompok 3



# LATAR BELAKANG

Pada era digitalisasi sekarang, orang bisa mengutarakan opini-opini tentang apapun dengan mudah, opini tersebut merupakan informasi yang dapat digunakan sebagai dasar pengambilan keputusan. Sehingga diperlukan sistem yang dapat mengolah data berbentuk opini dalam bentuk analisis sentimen.

Analisis sentimen merupakan salah satu teknik Natural Language Processing (NLP) yang menganalisis pendapat, sikap, dan emosi terhadap suatu entitas yang berupa teks.

NLP yang digunakan dalam analisis ini adalah teknik neural network dan LSTM (Long-Short Term Memory).



# RUMUSAN MASALAH

**1**

Bagaimana model neural network yang tepat untuk analisis sentimen data teks?

**2**

Bagaimana model LSTM yang tepat untuk analisis sentimen data teks?

**3**

Manakah model yang lebih baik, di antara neural network dan LSTM, untuk analisis sentimen data teks?

# TUJUAN

**1**

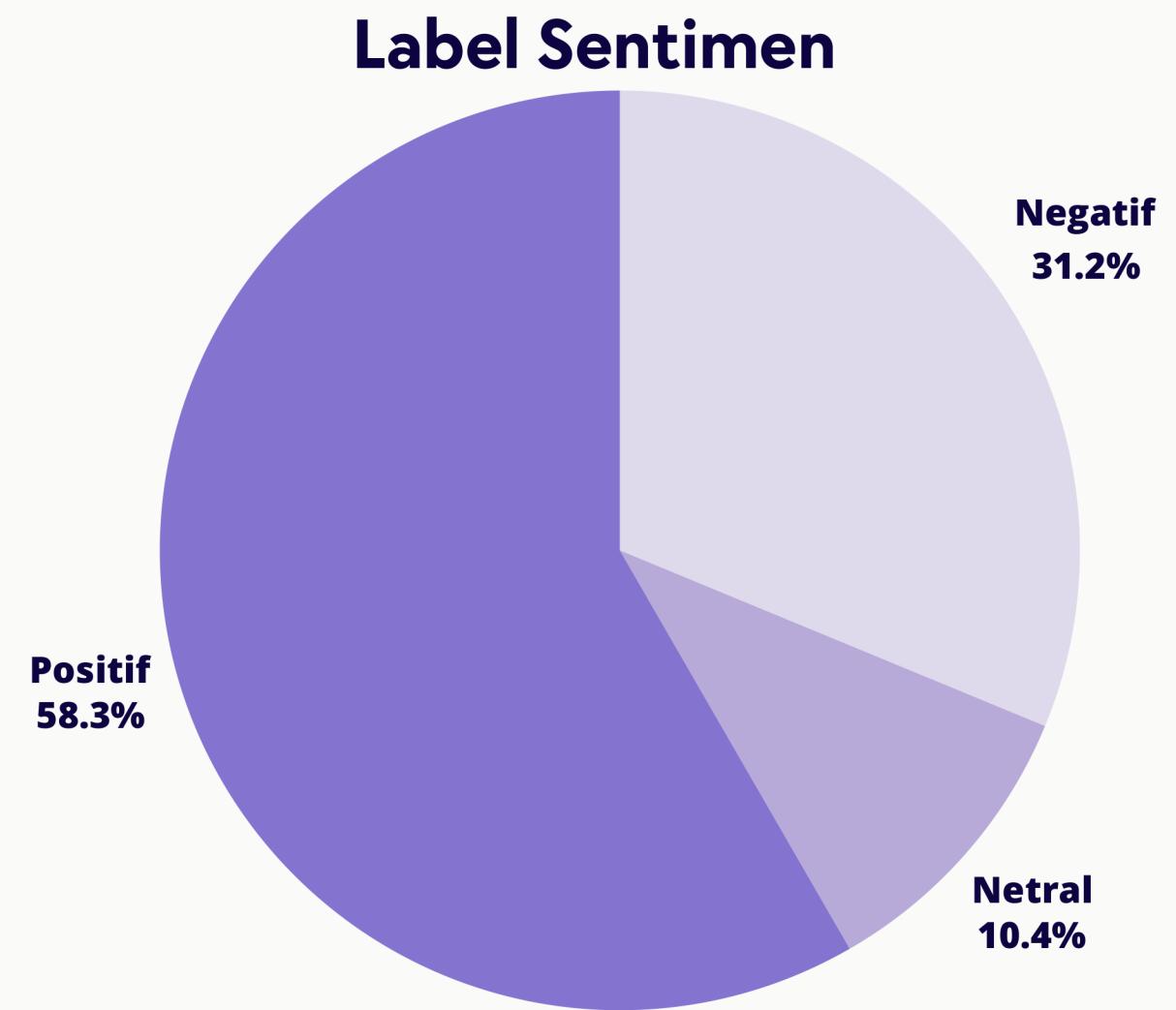
Membuat model neural network untuk analisis sentimen data teks.

**2**

Membuat model LSTM untuk analisis sentimen data teks.

**3**

Menentukan model yang lebih baik untuk analisa sentimen data teks.



# DATA

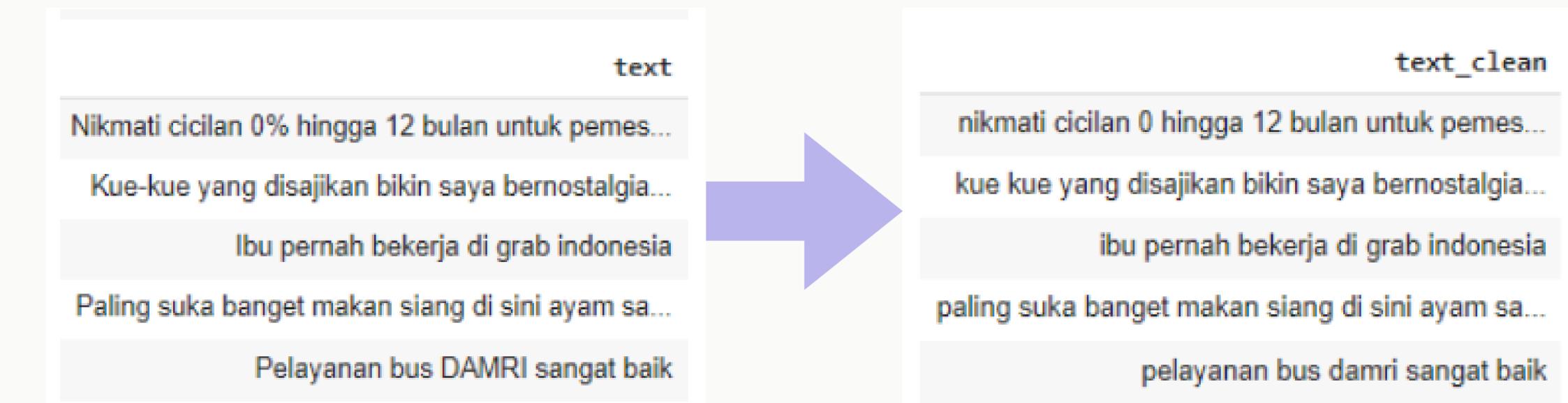
Data berasal dari Binar Academy yang berisi 2 kolom dan 11000 rows. Data terdiri dari 3 sentimen, yaitu Positif, Netral, dan Negatif.

	label	negative	neutral	positive
teks	count	3436	1148	6416
	unique	3412	1138	6383
	top	kesal diminta luhut stop penenggelaman kapal , begin... jangan pernah kecewa dengan apa yang diberikan...		
	freq	4	2	4

# PREPROCESSING

## CLEANSING

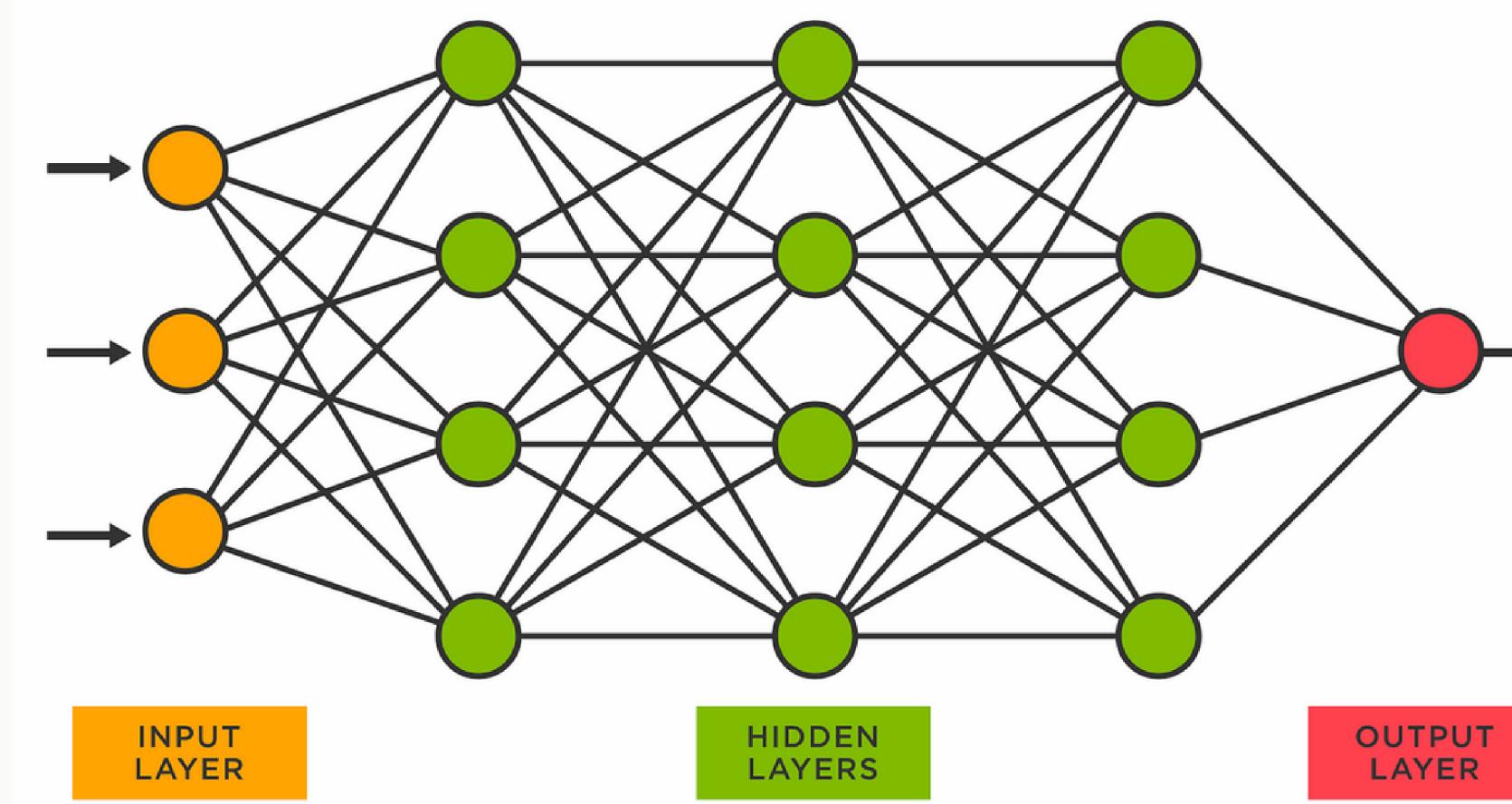
Menggunakan RegEx : Lowercase & menghilangkan karakter dan angka.  
Penggunaan huruf besar dan huruf kecil maupun karakter pada kalimat tidak mempengaruhi sentimen sehingga huruf diubah menjadi kecil semua dan karakter dihilangkan agar memudahkan mesin dalam membaca data.



# NEURAL NETWORK



# NEURAL NETWORK



Neuron terdiri dari tiga bagian yaitu;

- Input layer: input sinyal atau data
- Middle layer/hidden layer: memproses data dari inputan tadi
- Output layer: output dari data yang sudah diproses

# FEATURE EXTRACTION

## TF-IDF

(Term Frequency-Inverse Document Frequency)

Metode pembobotan dalam bentuk integrasi antara term frequency (banyaknya term) dengan inverse document frequency (mengukur bobot informasi).

Menggunakan modul TfidfVectorizer dari Sklearn. Modul ini dibangun dari rumus untuk menghitung TF-IDF.

## BAG OF WORDS

Bag of words menghitung frekuensi kemunculan kata dalam dokumen.

Menggunakan modul CountVectorizer dari Sklearn.

# SPLITTING DATA

X = teks ulasan

cleaned\_review  
warung ini dimiliki oleh pengusaha pabrik tahu...  
mohon ulama lurus dan k mmbri hujjah partai ap...  
lokasi strategis di jalan sumatera bandung tem...  
betapa bahagia nya diri ini saat unboxing pake...  
duh jadi mahasiswa jangan sombong dong kasih k...

Y = sentimen

sentiment  
positive  
neutral  
positive  
positive  
negative

Dataset terdiri dari teks ulasan dan sentimen yang dipisah menjadi dua bagian, yaitu data training dan data testing dengan persentase masing-masing 80% dan 20%.

Split dataset menggunakan library Sklearn dengan modul: modul\_selection dan sublibrary train\_test\_split.

Hasilnya berupa 4 buah file yaitu X\_train, X\_test, y\_train, dan y\_test.

# TRAINING

Menggunakan library Sklearn neural\_network dan modul MLPClassifier.

Lalu menyesuaikan model dengan data pelatihan menggunakan metode fit().



# EVALUASI

Setelah training, tahap selanjutnya adalah membuat prediksi menggunakan data pengujian dengan model predict().

Setelah melakukan prediksi, langkah terakhir adalah mengevaluasi kinerja model menggunakan sklearn metrics dengan dua modul yaitu classification\_report dan accuracy\_score.

TF IDF	Bag of Word
0,844	0,8400

# EVALUASI

Hasil klasifikasi yang didapatkan adalah classification report dan accuracy\_score berupa Akurasi, Presisi, Recall, dan F1-score

## **TF IDF**

	precision	recall	f1-score	support
0	0.78	0.79	0.79	680
1	0.88	0.90	0.89	1281
2	0.79	0.64	0.71	239
accuracy			0.84	2200
macro avg	0.82	0.78	0.80	2200
weighted avg	0.84	0.84	0.84	2200

Accuracy: 0.8409090909090909  
Data has been cleaned and exported to /mnt/data/cleaned\_data.csv

## **BAG OF WORDS**

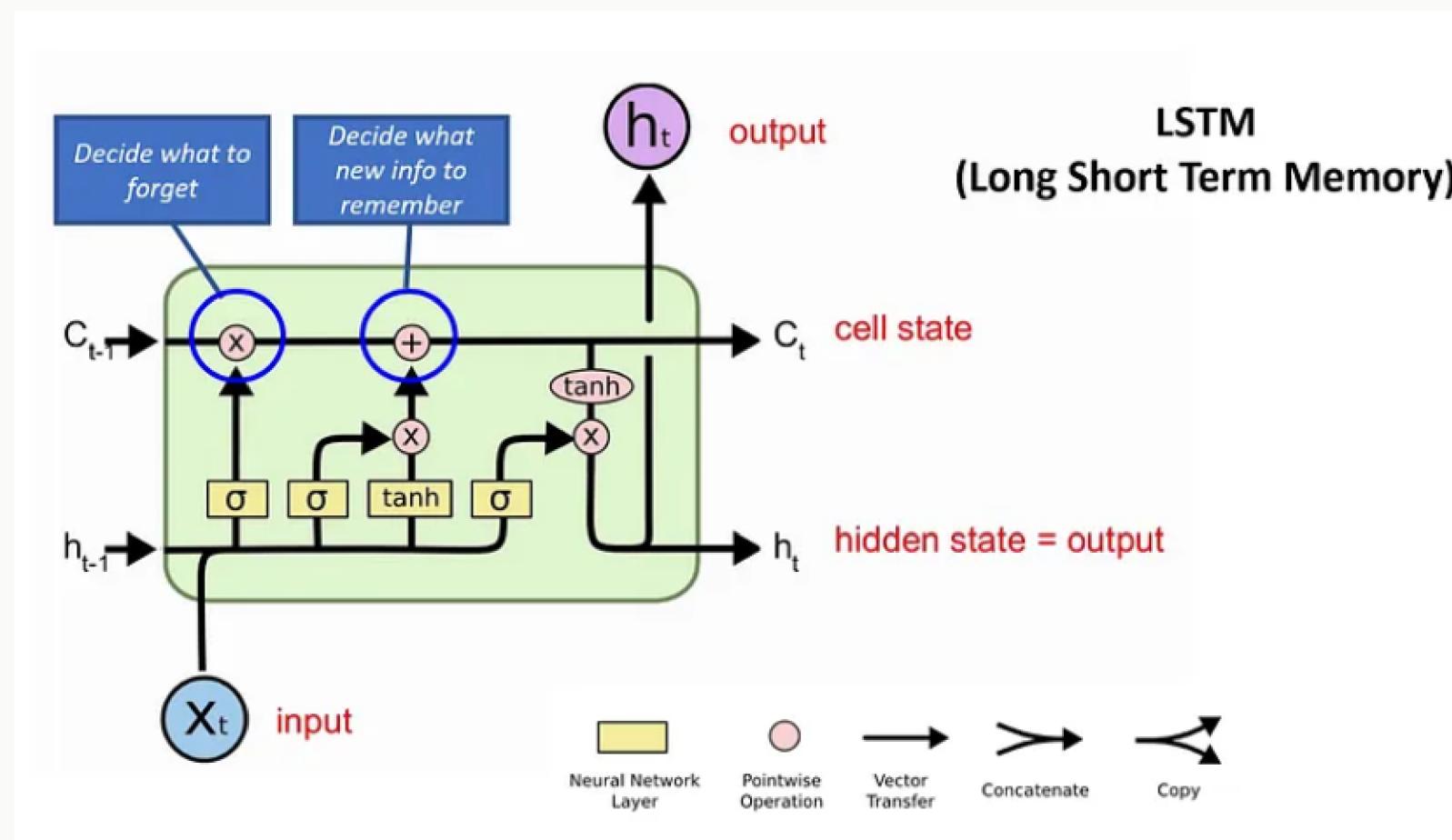
	precision	recall	f1-score	support
0	0.79	0.79	0.79	680
1	0.88	0.90	0.89	1281
2	0.82	0.72	0.77	239
accuracy			0.84	2200
macro avg	0.83	0.80	0.81	2200
weighted avg	0.84	0.84	0.84	2200

Accuracy: 0.8440909090909091  
Data has been cleaned and exported to /mnt/data/cleaned\_data.csv

# LONG-SHORT TERM MEMORY



# LSTM



LSTM adalah model yang didesain untuk mengatasi masalah pada RNN dengan cara menyimpan data pada memori. LSTM mempunyai cell state yang memiliki berperan seperti long-short memory dan dapat menghasilkan gradien yang lebih stabil.

# FEATURE EXTRACTION

Pada LSTM menggunakan Tokenizer dan pad sequences dari TensorFlow Keras.

Tokenizer akan membagi kalimat menjadi kata-kata dan menyimpan dalam bentuk integer. Hasil dari tokenizer didapatkan bahwa total kata adalah 17272.

Lalu kata akan disimpan dalam bentuk deretan angka (text to sequences) dan disusun dalam panjang deretan yang sama (pad sequences).

Split data yang dilakukan untuk model LSTM sama seperti pada model neural network.

# TRAINING

Model menggunakan jenis sequential yaitu menambahkan layer secara berurutan.  
Tidak menggunakan earlystopping agar dapat ditentukan secara manual.

## Layer Embedding

Memetakan angka hasil feature extraction pada vektor.

## Layer LSTM

## Layer Dropout

Untuk mencegah overfitting.

## Layer Output

Berupa 3 output dan menggunakan aktivasi softmax.

## Compile

Menggunakan loss binary/categorical crossentropy dan optimizer adam.

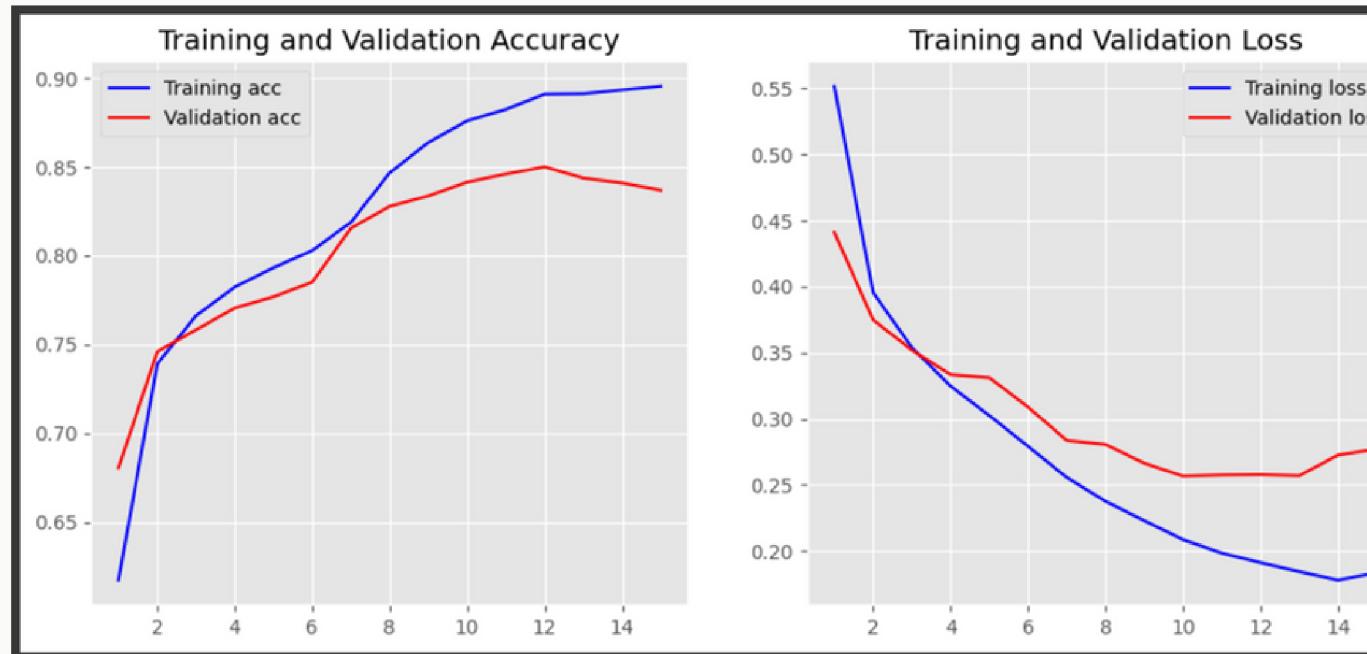
# PARAMETER TUNING

***neuron 128; dropout 0.2; epoch 15; batch 256***

MODEL	MAX FEATURES	EMBED DIMENTION	LOSS	OPTIMIZER	AKURASI
1	1000	16	binary crossentropy	adam default	0.84
2	1000	16	categorical crossentropy	adam lr 0.001	0.85
3	10000	100	categorical crossentropy	adam lr 0.001	0.85
4	10000	100	categorical crossentropy	adam lr 0.3	0.61

# EVALUASI

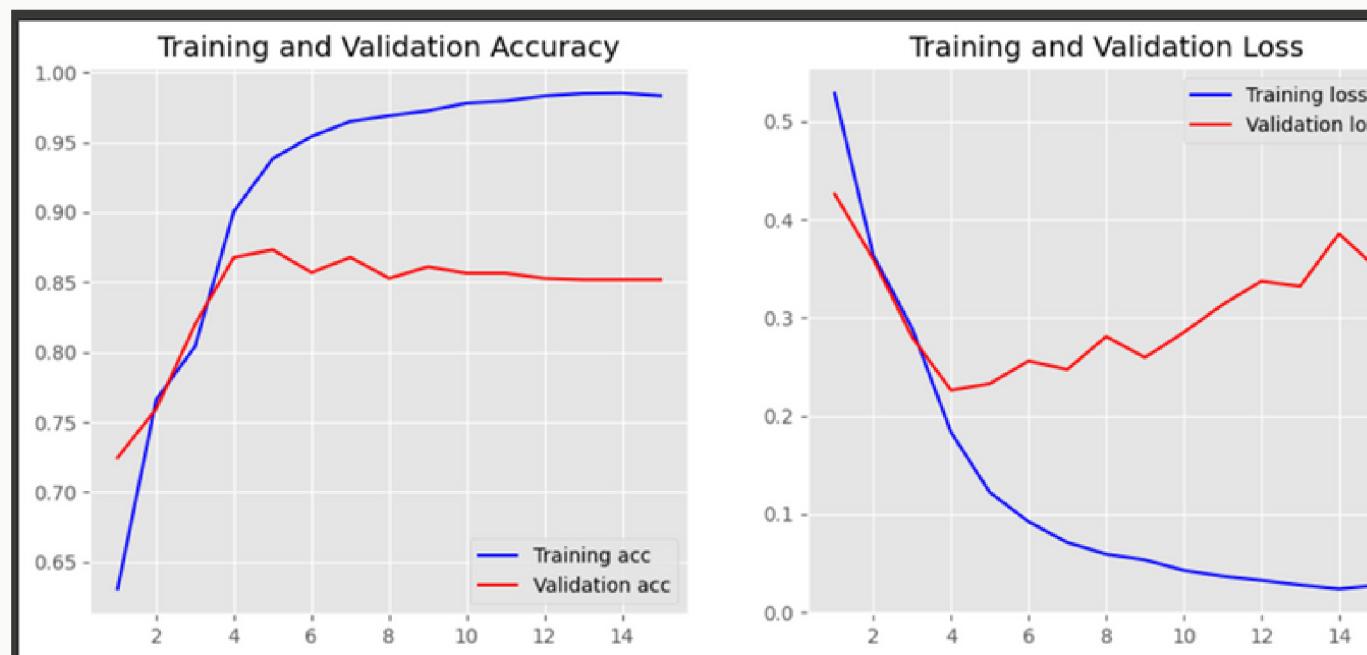
**MODEL 1**



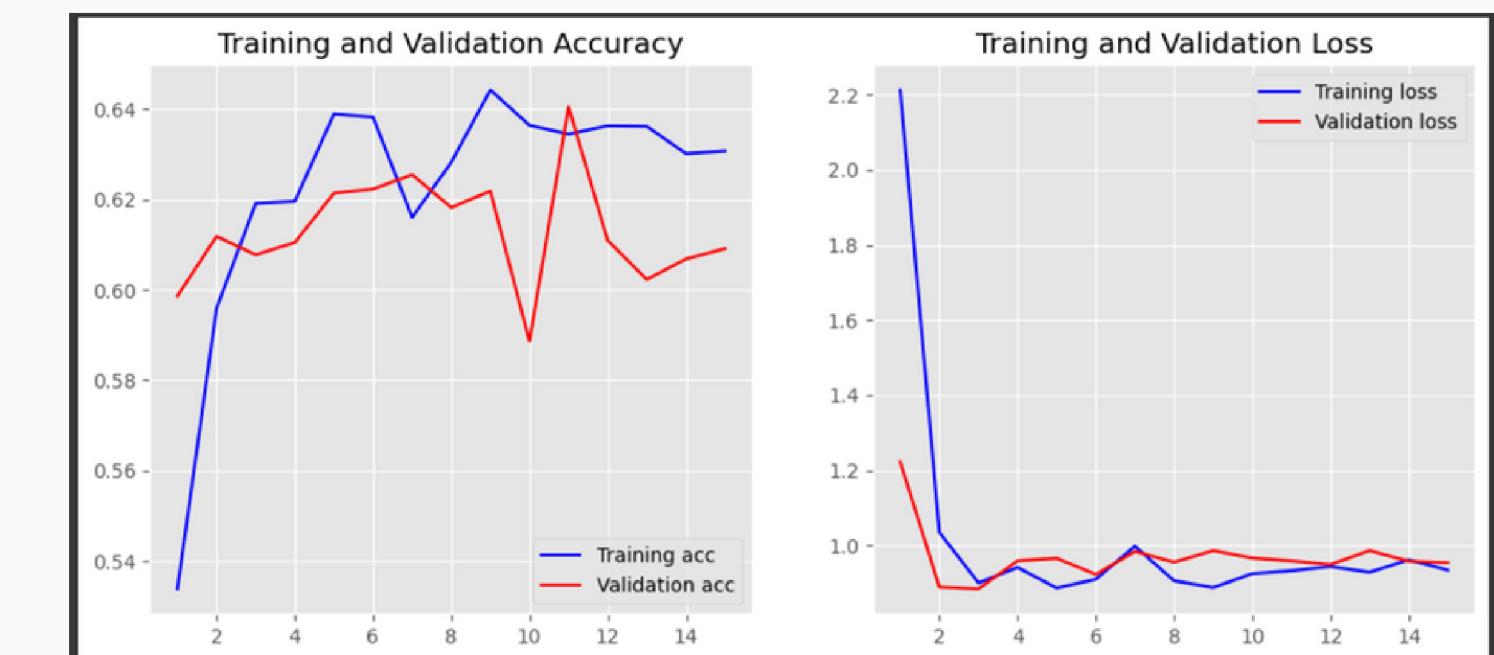
**MODEL 2**



**MODEL 3**

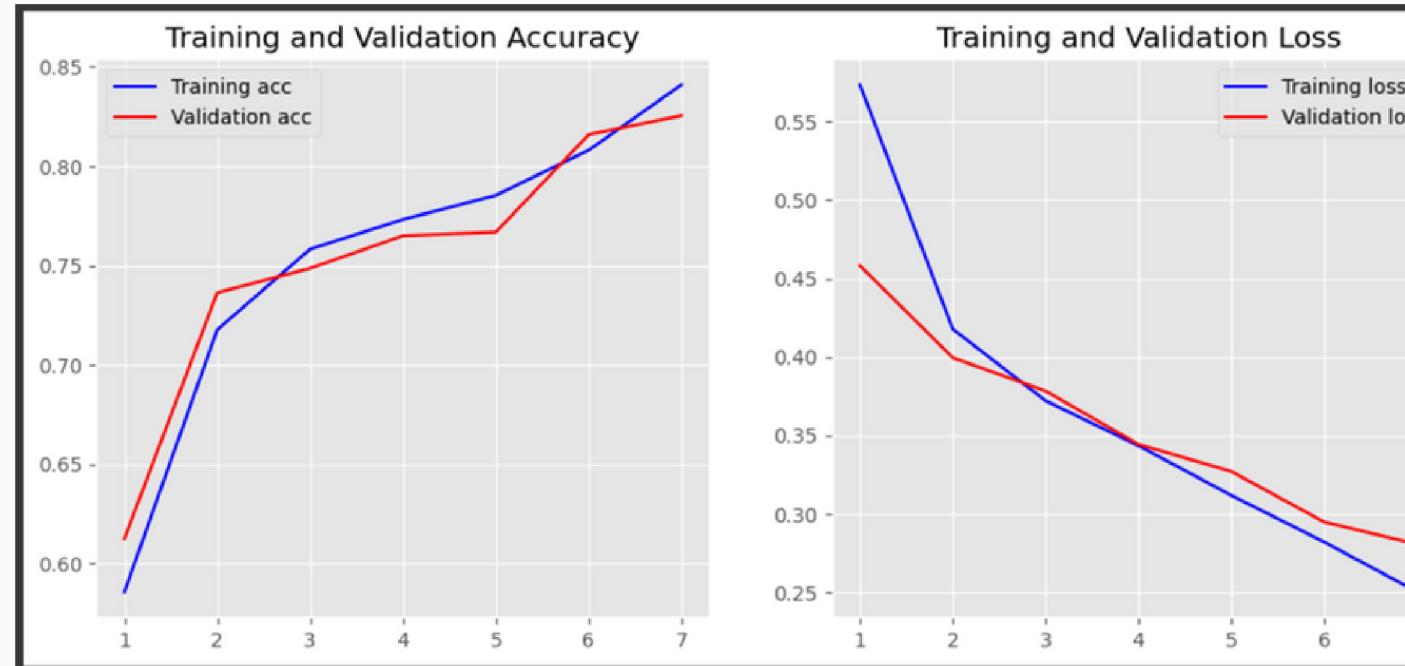


**MODEL 4**



# EVALUASI

## MODEL 1



TESTING SELESAI				
	precision	recall	f1-score	support
0	0.77	0.79	0.78	704
1	0.59	0.63	0.61	222
2	0.91	0.88	0.89	1274
accuracy			0.83	2200
macro avg	0.75	0.77	0.76	2200
weighted avg	0.83	0.83	0.83	2200

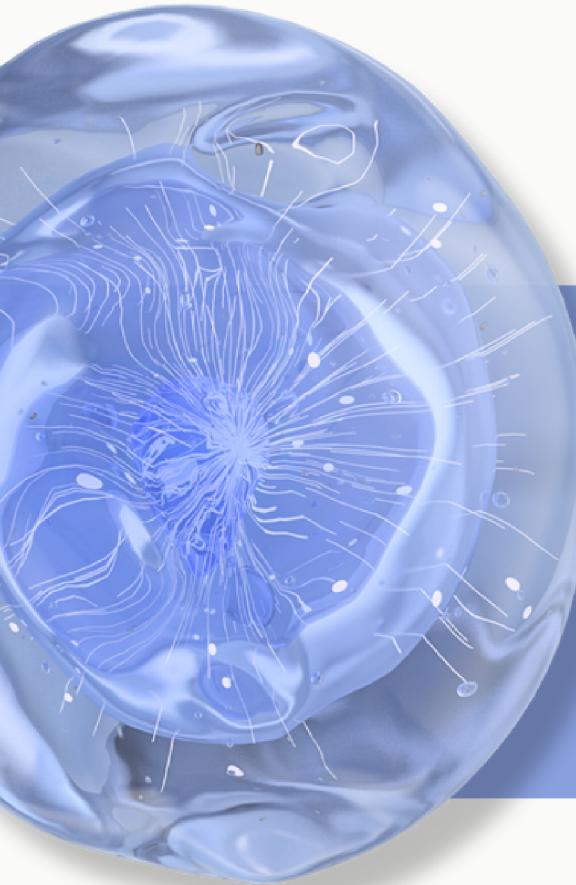
Rata-rata Accuracy: 0.8202727272727273

## MODEL 2



TESTING SELESAI				
	precision	recall	f1-score	support
0	0.74	0.82	0.78	704
1	0.66	0.59	0.63	222
2	0.91	0.87	0.89	1274
accuracy			0.83	2200
macro avg	0.77	0.76	0.77	2200
weighted avg	0.83	0.83	0.83	2200

Rata-rata Accuracy: 0.8285454545454545



# PERBANDINGAN

- ***Neural Network***

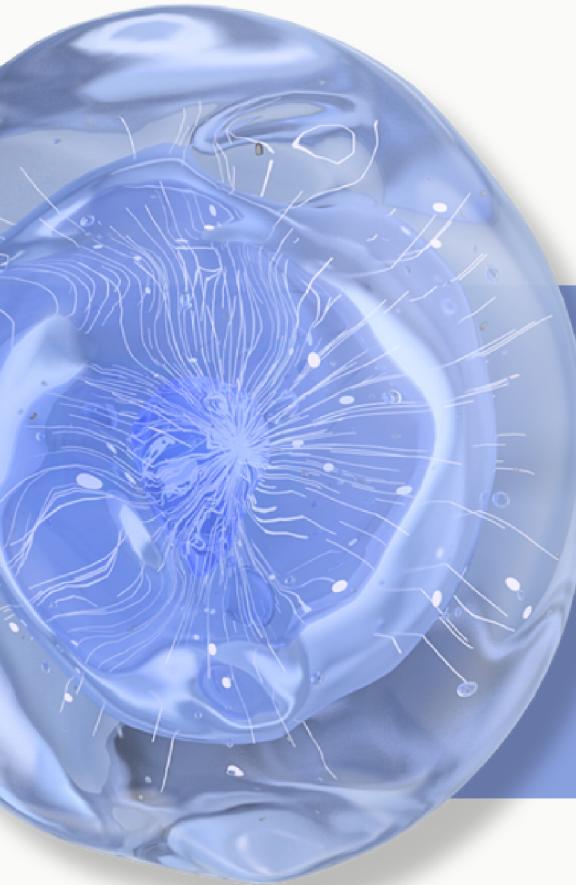
	precision	recall	f1-score	support
0	0.79	0.79	0.79	680
1	0.88	0.90	0.89	1281
2	0.82	0.72	0.77	239
accuracy			0.84	2200
macro avg	0.83	0.80	0.81	2200
weighted avg	0.84	0.84	0.84	2200

Accuracy: 0.8440909090909091  
Data has been cleaned and exported to /mnt/data/cleaned\_data.csv

- ***LSTM***

	TESTING SELESAI	precision	recall	f1-score	support
0		0.74	0.82	0.78	704
1		0.66	0.59	0.63	222
2		0.91	0.87	0.89	1274
accuracy				0.83	2200
macro avg		0.77	0.76	0.77	2200
weighted avg		0.83	0.83	0.83	2200

Rata-rata Accuracy: 0.8285454545454545



# PERBANDINGAN

KALIMAT	NEURAL NETWORK	LSTM
jangan kecewa pada Tuhan	positif	negatif
pengemis itu santun sekali	positif	negatif
Jakarta selalu macet	negatif	netral
makanan enak lokasi strategis	positif	positif
kok ngga jelas banget sih	positif	negatif

# KESIMPULAN

## *Kesimpulan 1*

Model Neural Network yang digunakan memiliki feature extraction Bag of Word dan training modul MLPClassifier default yang memiliki akurasi 0.84.

## *Kesimpulan 2*

Model LSTM yang digunakan memiliki feature extraction Tokenizer dengan maksimum kata 1000, neuron 128, dropout, dan loss categorical crossentropy yang memiliki akurasi 0.82.

## *Kesimpulan 3*

Model Neural Network memiliki kemampuan yang lebih baik untuk prediksi sentimen teks dari data yang digunakan.

# SARAN

Dilakukan penyeimbangan untuk jumlah data train agar jumlah data sentimen lebih sama.

