Perbandingan Analisis Sentimen Penanganan Banjir Dengan Metode SVM dan RNN Tipe LSTM

Pembimbing Utama : Dr. Herfina, M.Kom

Pembimbing Pendamping : Mulyati, M.Kom

Nama

Herry Wijaya

065116076

Bab I Pendahuluan

1. Latar Belakang

- Deep learning merupakan machine learning yang memiliki kemampuan dan spesifikasi yang lebih luas dan memiliki potensial yang dapat melebihi model machine learning tradisional
- Analisis Sentimen dilakukan untuk mengetahui pendapat masyarakat tentang fenomena banjir yang terjadi di Jakarta dan menggunakannya sebagai dataset yang diolah oleh model algoritma machine learning

2. Manfaat

- Mengetahui perbedaan performa antara Deep Learning dengan metode machine learning tradisional.

- Mengetahui pengaruh hyperparameter RNN tipe LSTM kepada akurasi yang dihasilkan.

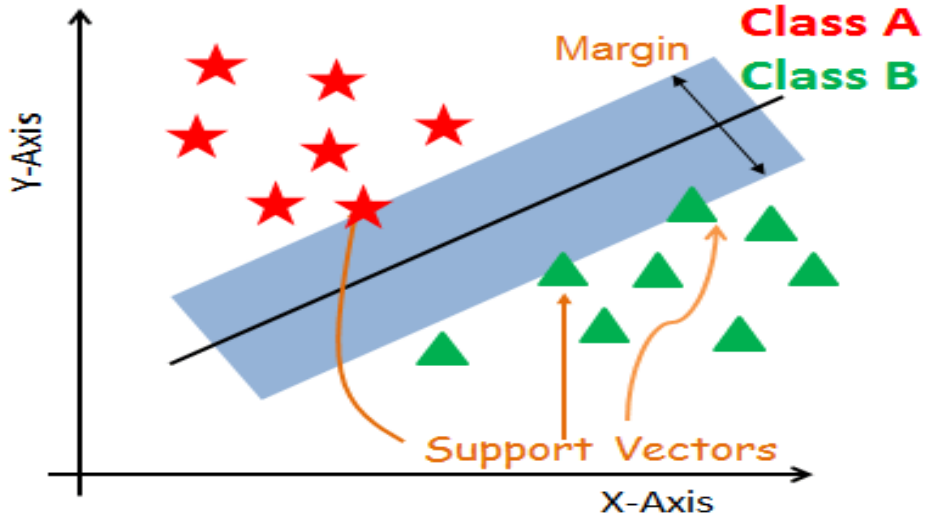
- Mengetahui pengaruh hyperparameter VSM kepada akurasi yang dihasilkan.

- Mengetahui perbandingan akurasi yang dihasilkan antara metode RNN tipe LSTM dengan SVM.

3. Ruang Lingkup

- Software yang digunakan adalah Jupyter Notebook dan RStudio
- Bahasa pemrograman yang digunakan adalah Python dan R
- Penelitian yang dilakukan merupakan analisis sentimen menggunakan dataset komentar Instagram tentang penanganan banjir
- Hasil penelitian adalah perbandingan performa prediksi yang dihasilkan oleh model algoritma machine learning Support Vector Machine dan Recurrent Neural Network tipe Long Short Term Memory

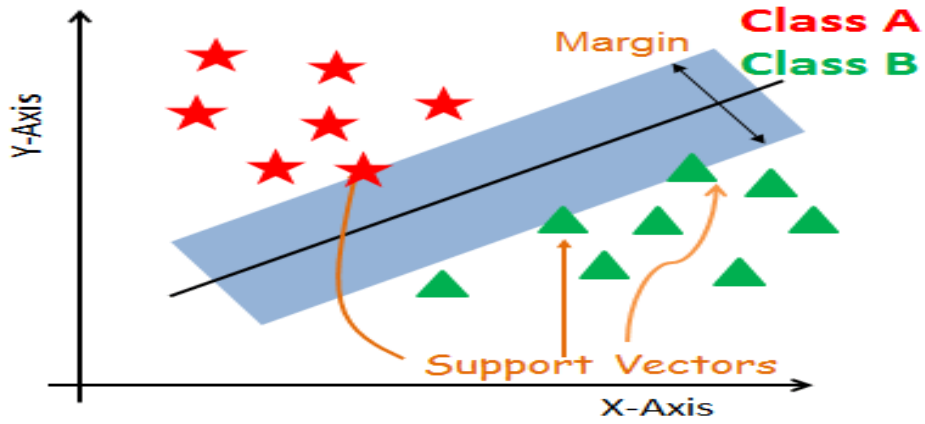
Bab II Tinjauan Pustaka



1. Support Vector Machine

Karakteristik :

- Support Vector = data point terdekat kepada hyperplane
- Hyperplane = wilayah yang memisahkan antara beberapa objek yang memilki kelas berbeda
- Margin = jarak antara dua garis di class point terdekat



Konfigurasi hyperparameter :

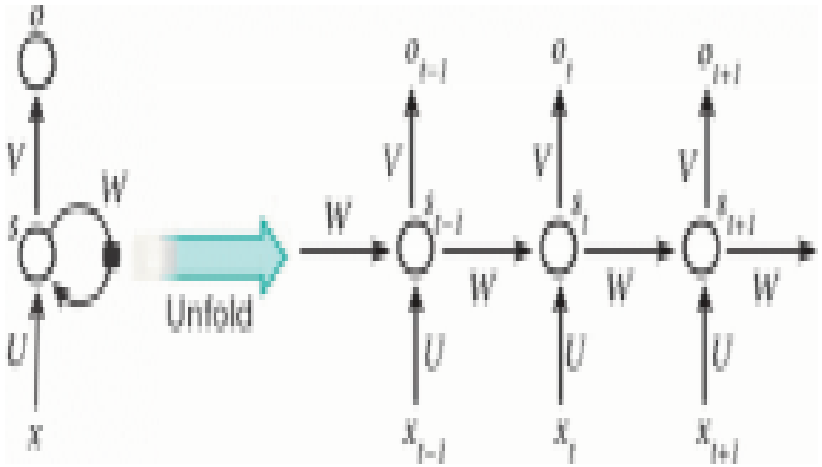
- Kernel = mengubah input dataset ke dimensi yang lebih tinggi
- Regularisasi = semakin kecil akan membuat margin hyperplane kecil dan semakin besar akan membuat margin hyperplane besar
- Gamma = semakin kecil hanya akan menentukan data point terdekat untuk garis pemisah dan semakin besar menentukan lebih banyak data point untuk perhitungan garis pemisah

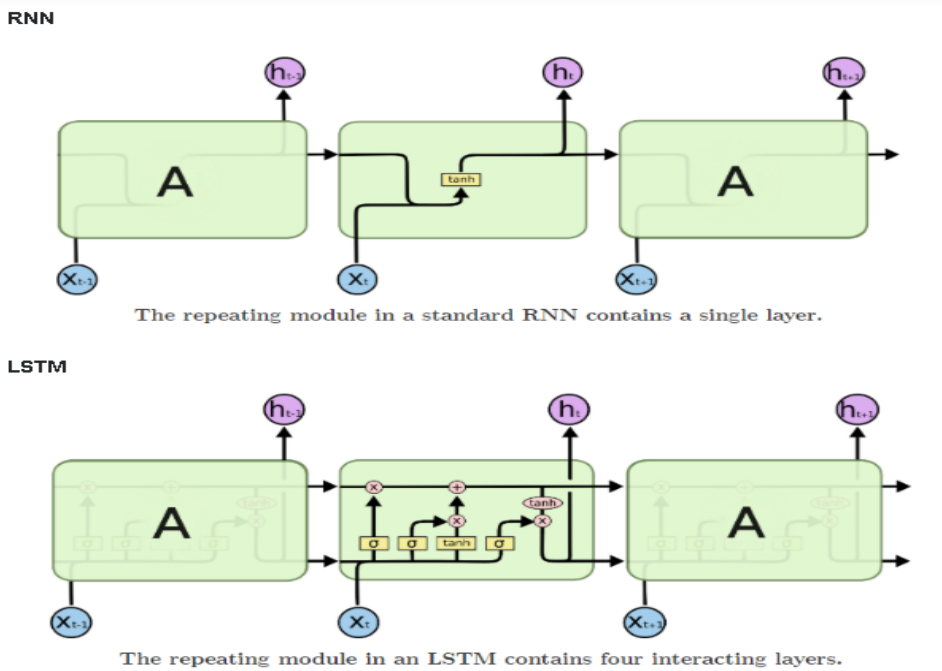
Linear Kernel

K (x, xi) = sum (x * xi)

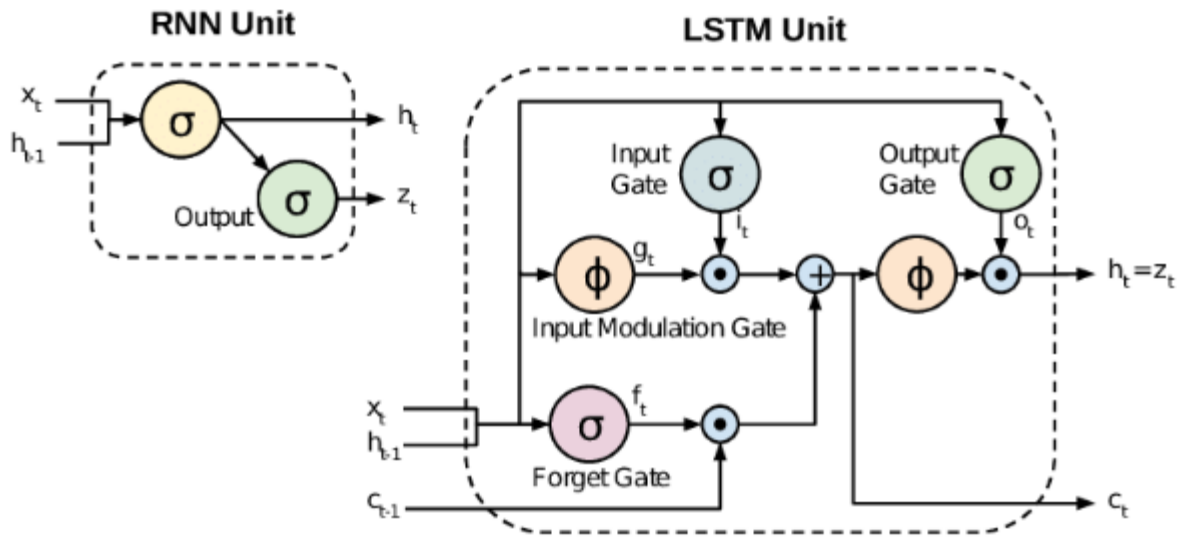
2. Recurrent Neural Network Tipe Long Short Term Memory

- RNN kurang cocok untuk memecahkan masalah long term temporal dependencies
- Mudah terjadi permalasahan vanishing / exploding gradient





- Cell = penyebaran informasi yang dibagikan di dalam jaringan dalam time step tertentu
- Gerbang input = menentukan informasi di time step selanjutnya dan hidden state sebelumnya dalam time step tertentu
- Gerbang lupa = menentukan apakah informasi baru diupdate atau dilupakan di hidden state
- Gerbang output = menentukan informasi di hidden state untuk time step selanjutnya



3. Penelitian terdahulu

- Judul : Sentiment Analysis using Recurrent Neural Network (Thomas, 2018)

Deskripsi : Analisis sentimen yang dilakukan menggunakan recurrent neural network bertipe long short term memory. Penelitian menggunakan bahasa Malayalam (India Selatan) sebagai data teks yang sudah diberi label polaritas. Hasil penelitian tersebut menghasilkan 80% rata-rata akurasi.

- Judul : Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Believe Network (Zulfa, 2017)

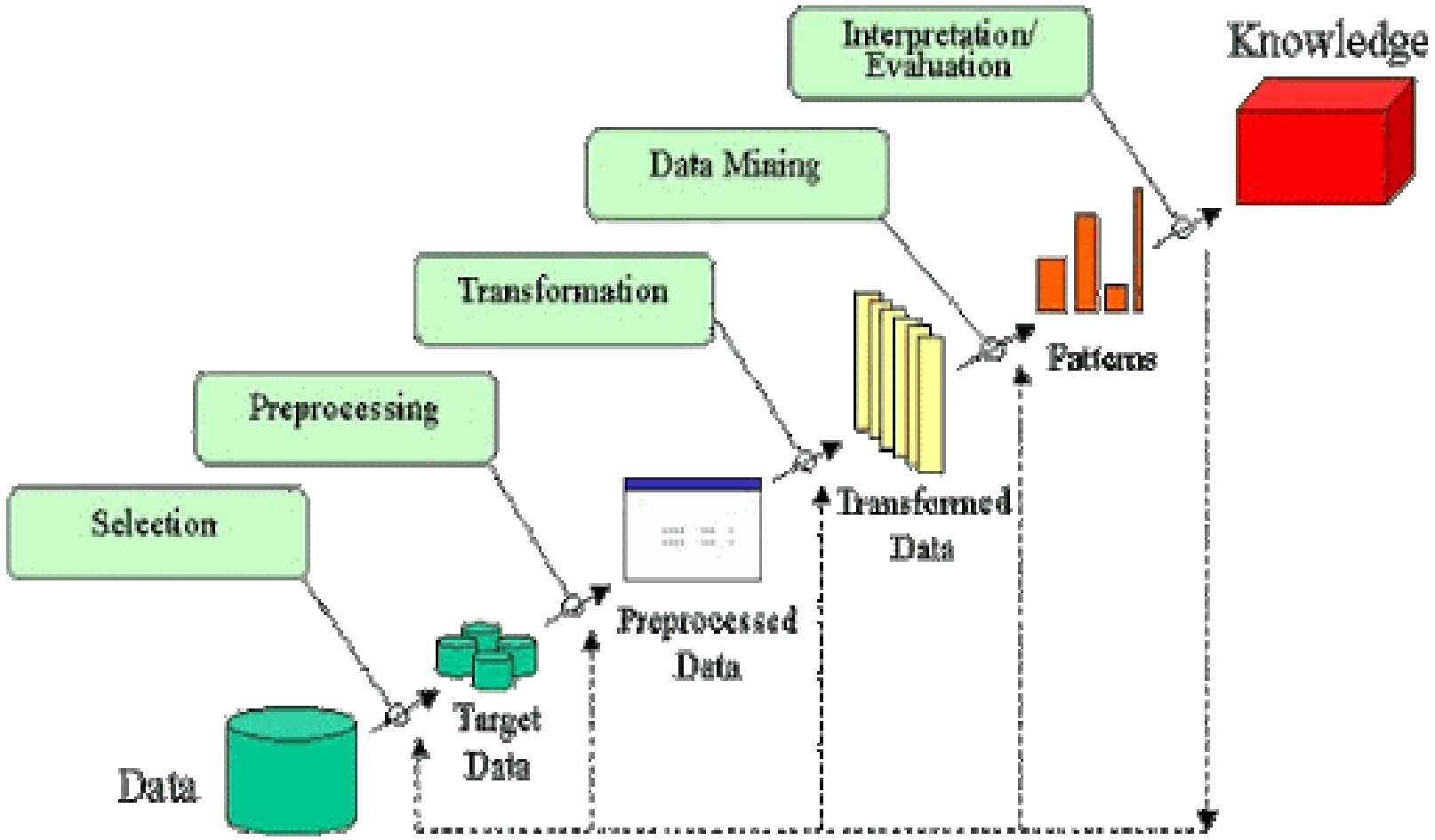
Deskripsi : Analisis sentiment yang dilakukan menggunakan deep believe network. Penelitian menggunakan komentar tweet dari twitter sebagai data teks yang sudah diberi label polaritas. Hasil penelitian tersebut menghasilkan 93,31 % akurasi.

- Judul : Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naïve Bayes (Gunawan, 2018)

Deskripsi : Analisis sentiment yang dilakukan menggunakan algoritma naïve bayes. Penelitian menggunakan review customer kosmetik sebagai data teks yang sudah diberi label polaritas. Hasil penelitian tersebut menghasilkan 77,78 % akurasi tertinggi.

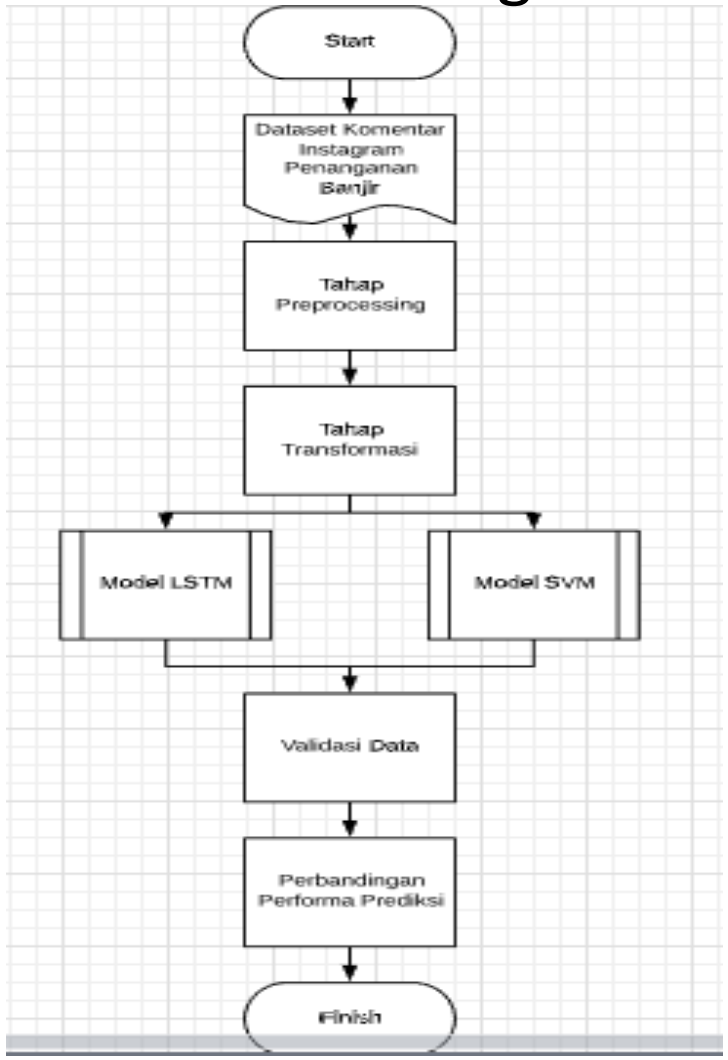
No.	Judul Penelitian	Machine Learning		Deep Learning		Dataset			
		Naïve Bayes	VSM	RNN	DBN	Instagram	Twitter	Review Customer	Teks Bahasa
1.	Sentiment Analysis using Recurrent Neural Network			✓					✓
2.	Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Believe Network				✓		✓		
3.	Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naïve Bayes	✓	✓					✓	
4.	Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode RNN dan VSM		✓	✓		✓			

Bab III Metode Penelitian

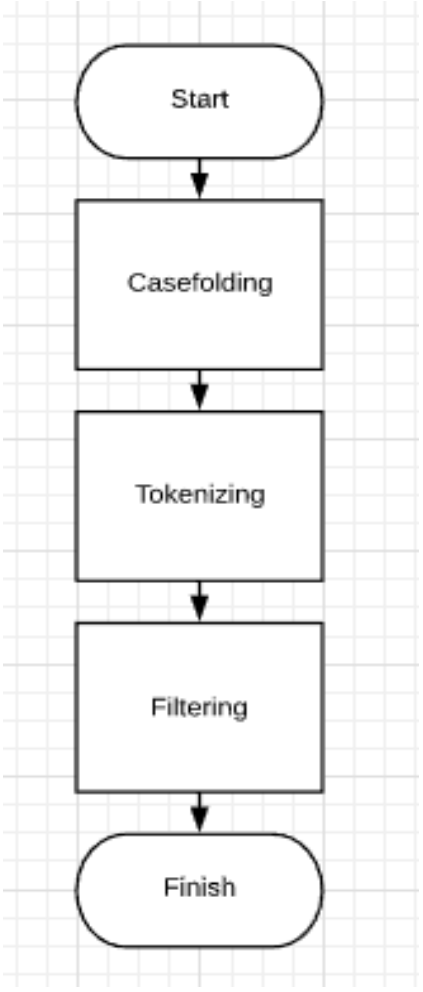


Bab IV Perancangan dan Implementasi

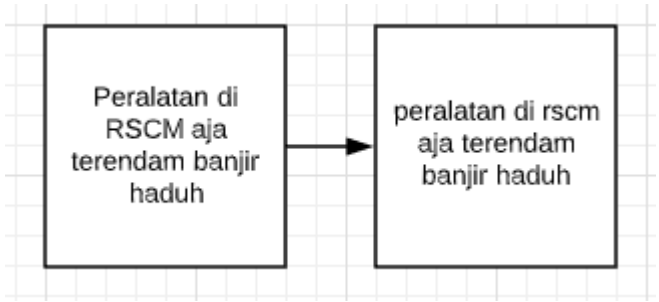
Flowchart Program



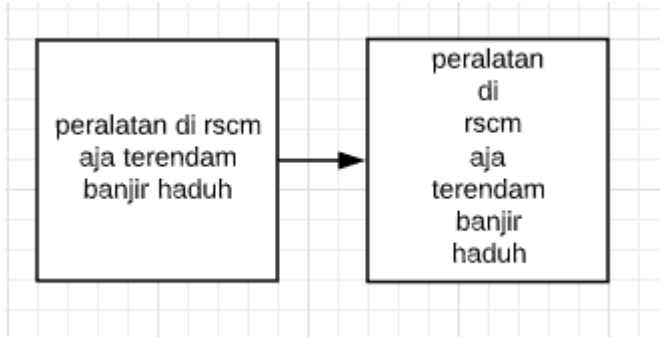
Tahap Preprocessing



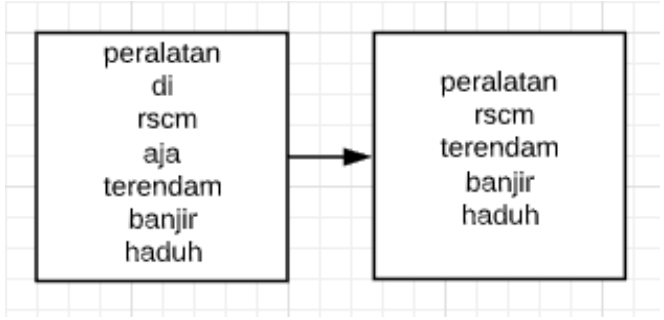
Case Folding = mengubah huruf capital menjadi lower case



Tokenizing = mengubah setiap kata menjadi token



Filtering = menghilangkan stopword



Tahap Transformasi Model VSM Menggunakan Metode TF-IDF

Tabel 4 Tabel TF-IDF

A, B	Nilai TF-IDF
(2, 2550)	0.4786
(2, 1123)	0.8781

Tabel 5 Tabel TF-IDF 2

Kata	1	2	3	4
'anjing'	0.6582	0.4226	0.5599	0.6793

- Komentar 'sekda idiot' yang berisi seluruh kata unik di dataset ditunjukkan bahwa kata 'sekda' berada di baris 2549 dan kata 'idiot' berada di baris 1122
- Kata 'anjing' yang berisi seluruh kata unik di dataset terkandung di empat komentar

Tahap Transformasi Model RNN Tipe LSTM Menggunakan Word Embedding

Tabel 6 Sampel Data One Hot Encoding

Dataset Komentar Instagram	One Hot Encoding	
	sekda	idiot
sekda idiot	1	0
	0	1

$$W = (-\frac{1}{\sqrt{h}}, \frac{1}{\sqrt{h}})$$
$$W = (-\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}})$$
$$W = (-0{,}477,\ 0{,}477)$$
$$X_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} X_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$W_f = \begin{bmatrix} 0{,}2 \\ 0{,}1 \end{bmatrix} W_c = \begin{bmatrix} 0{,}1 \\ 0{,}2 \end{bmatrix} B_f = [0{,}2]\ B_c = [0{,}2]$$
$$W_i = \begin{bmatrix} 0{,}1 \\ 0{,}1 \end{bmatrix} W_o = \begin{bmatrix} 0{,}4 \\ 0{,}3 \end{bmatrix} B_i = [0{,}2]\ B_0 = [0{,}3]$$

$$f_t = \sigma(W_f \,.\, [s_{t-1}, x_t] + b_f)$$
$$f_t = \sigma(\begin{bmatrix} 0{,}2 \\ 0{,}1 \end{bmatrix} \,.\, [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0{,}2])$$
$$f_t = \sigma(\begin{bmatrix} 0{,}2 \\ 0{,}1 \end{bmatrix} \,.\, 0 + \begin{bmatrix} 0{,}2 \\ 0{,}1 \end{bmatrix} \,.\, \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0{,}2])$$
$$f_t = \sigma(\begin{bmatrix} 0{,}4 \\ 0{,}2 \end{bmatrix})$$
$$f_t = \begin{bmatrix} 0{,}5987 \\ 0{,}5498 \end{bmatrix}$$

$$i_t = \sigma(W_i \cdot [s_{t-1}, x_t] + b_i)$$
$$i_t = \sigma(\begin{bmatrix} 0{,}1 \\ 0{,}1 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0{,}2])$$
$$i_t = \sigma(\begin{bmatrix} 0{,}1 \\ 0{,}1 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0{,}1 \\ 0{,}1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0{,}2])$$
$$i_t = \sigma(\begin{bmatrix} 0{,}3 \\ 0{,}2 \end{bmatrix})$$
$$i_t = \begin{bmatrix} 0{,}5744 \\ 0{,}5498 \end{bmatrix}$$
$$\hat{C}_t = \tanh(W_c \cdot [s_{t-1}, x_t] + b_c)$$
$$\hat{C}_t = \tanh(\begin{bmatrix} 0{,}1 \\ 0{,}2 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0{,}2])$$
$$\hat{C}_t = \tanh(\begin{bmatrix} 0{,}1 \\ 0{,}2 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0{,}1 \\ 0{,}2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0{,}2])$$
$$\hat{C}_t = \tanh(\begin{bmatrix} 0{,}3 \\ 0{,}2 \end{bmatrix})$$
$$\hat{C}_t = \begin{bmatrix} 0{,}2913 \\ 0{,}1974 \end{bmatrix}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$
$$C_t = \begin{bmatrix} 0{,}5987 \\ 0{,}5498 \end{bmatrix} * 0 + \begin{bmatrix} 0{,}5744 \\ 0{,}5498 \end{bmatrix} * \begin{bmatrix} 0{,}2913 \\ 0{,}1974 \end{bmatrix}$$
$$C_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0{,}1673 \\ 0{,}1085 \end{bmatrix}$$
$$C_t = \begin{bmatrix} 0{,}1673 \\ 0{,}1085 \end{bmatrix}$$

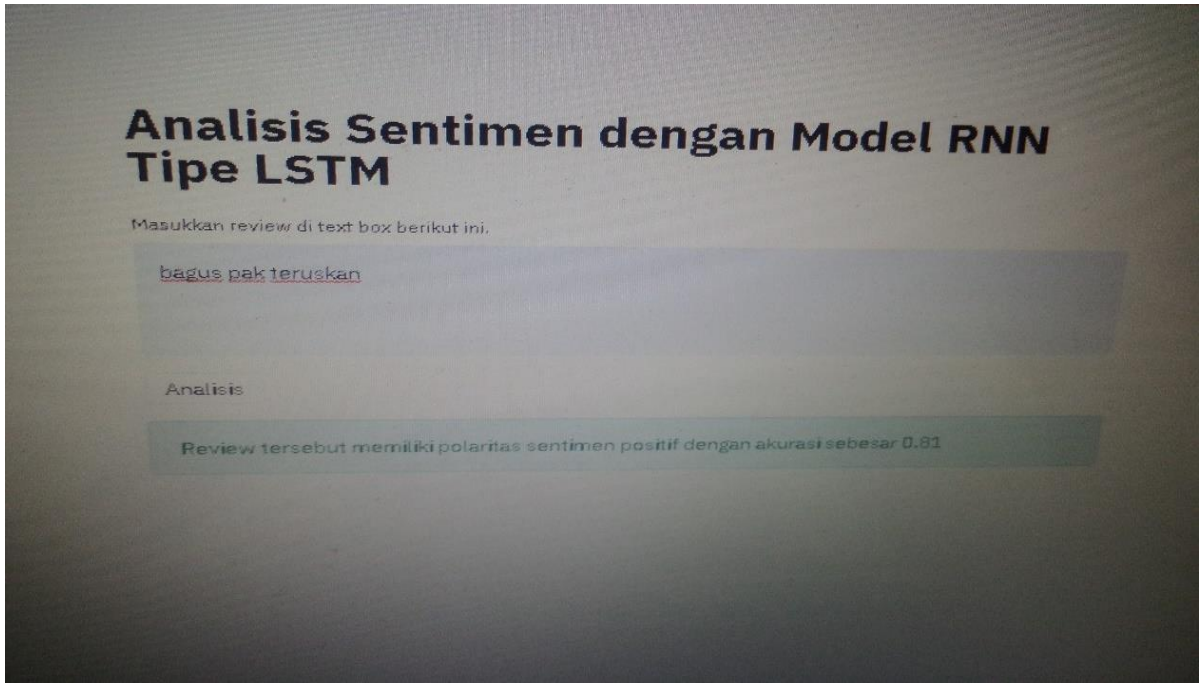
$$o_t = \sigma(W_o \cdot [s_{t-1}, x_t] + b_o)$$
$$o_t = \sigma(\begin{bmatrix} 0{,}4 \\ 0{,}3 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0{,}3])$$
$$o_t = \sigma(\begin{bmatrix} 0{,}4 \\ 0{,}3 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0{,}4 \\ 0{,}3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0{,}3])$$
$$o_t = \sigma(\begin{bmatrix} 0{,}7 \\ 0{,}3 \end{bmatrix})$$
$$o_t = \begin{bmatrix} 0{,}6682 \\ 0{,}5744 \end{bmatrix}$$
$$s_t = o_t * \tanh(C_t)$$
$$s_t = \begin{bmatrix} 0{,}6682 \\ 0{,}5744 \end{bmatrix} * \tanh(\begin{bmatrix} 0{,}1673 \\ 0{,}1085 \end{bmatrix})$$
$$s_t = \begin{bmatrix} 0{,}6682 \\ 0{,}5744 \end{bmatrix} * \begin{bmatrix} 0{,}1658 \\ 0{,}1081 \end{bmatrix}$$
$$s_t = \begin{bmatrix} 0{,}1101 \\ 0{,}0621 \end{bmatrix}$$

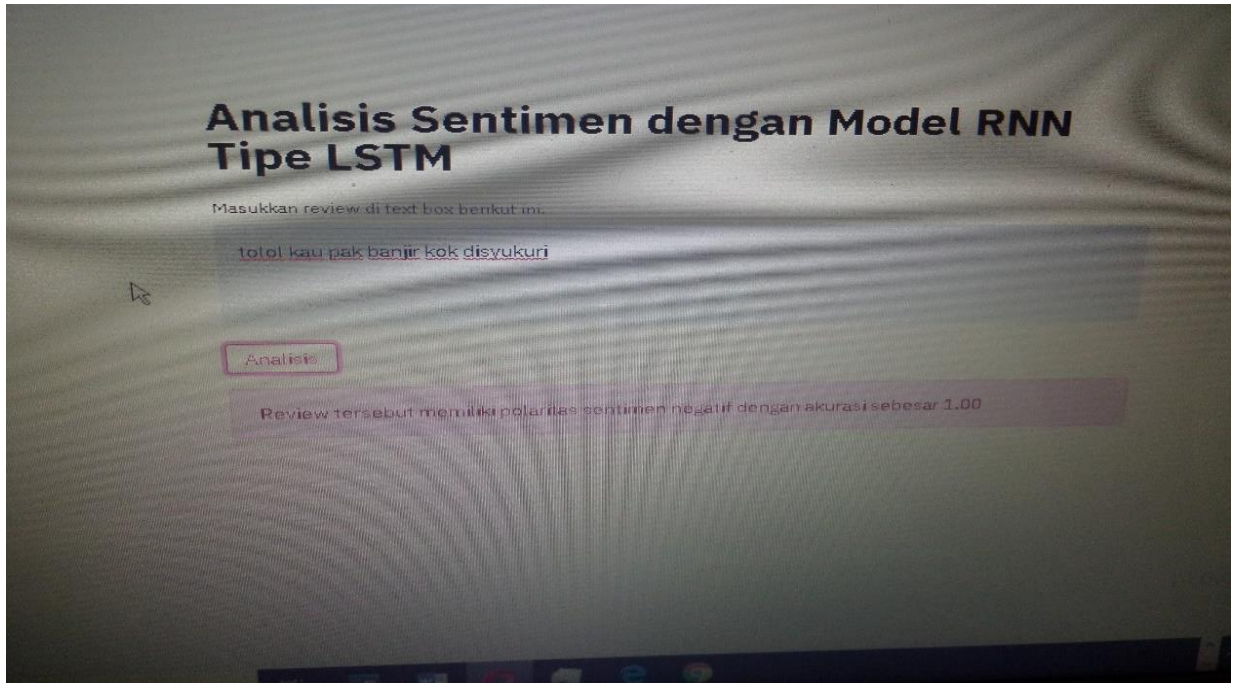
Tabel 7 Hasil Perhitungan LSTM

	f_t	i_t	$\hat{C}_t$	C_t	o_t	s_t
X1	$\begin{bmatrix} 0{,}5987 \\ 0{,}5498 \end{bmatrix}$	$\begin{bmatrix} 0{,}5744 \\ 0{,}5498 \end{bmatrix}$	$\begin{bmatrix} 0{,}2913 \\ 0{,}1974 \end{bmatrix}$	$\begin{bmatrix} 0{,}1673 \\ 0{,}1085 \end{bmatrix}$	$\begin{bmatrix} 0{,}6682 \\ 0{,}5744 \end{bmatrix}$	$\begin{bmatrix} 0{,}1101 \\ 0{,}0621 \end{bmatrix}$
X2	$\begin{bmatrix} 0{,}5553 \\ 0{,}5760 \end{bmatrix}$	$\begin{bmatrix} 0{,}5526 \\ 0{,}5760 \end{bmatrix}$	$\begin{bmatrix} 0{,}2079 \\ 0{,}3905 \end{bmatrix}$	$\begin{bmatrix} 0{,}2078 \\ 0{,}2874 \end{bmatrix}$	$\begin{bmatrix} 0{,}5852 \\ 0{,}6499 \end{bmatrix}$	$\begin{bmatrix} 0{,}1199 \\ 0{,}1818 \end{bmatrix}$

$$y_t = \text{softmax}(\begin{bmatrix} 0{,}6682 \\ 0{,}5744 \end{bmatrix})$$
$$y(0{,}6682) = (\frac{e^{0{,}6682}}{e^{0{,}6682} + e^{0{,}5744}}) = 0{,}5234$$
$$y(0{,}5744) = (\frac{e^{0{,}5744}}{e^{0{,}6682} + e^{0{,}5744}}) = 0{,}4766$$

Implementasi program





Video Demonstrasi Alat

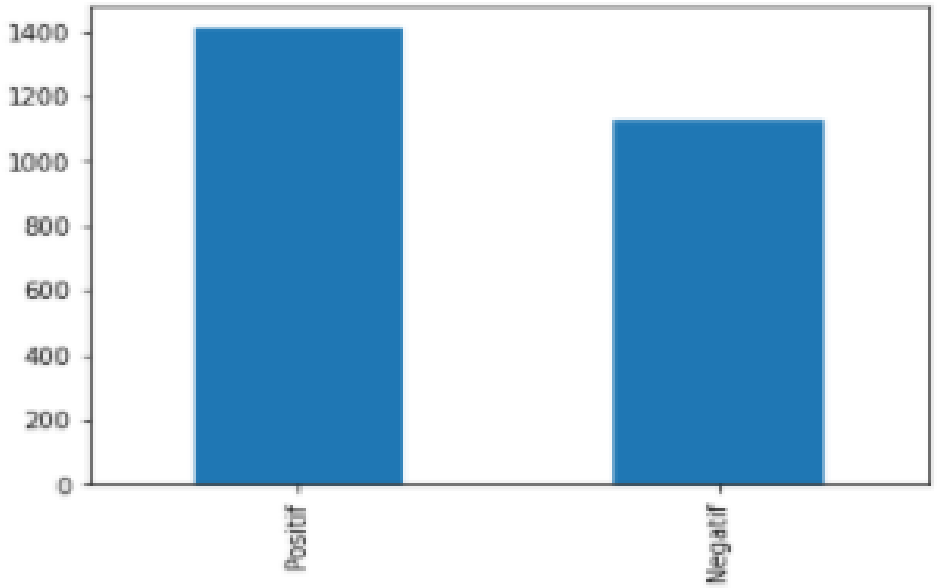
Bab V Pembahasan

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 500, 50)           282800
_________________________________________________________________
lstm_1 (LSTM)                (None, 25)                7600
_________________________________________________________________
dropout_1 (Dropout)          (None, 25)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 26
=================================================================
Total params: 290,426
Trainable params: 290,426
Non-trainable params: 0
_________________________________________________________________
```

-Layer Embedding sebagai input yang mengubah ukuran komentar sebesar 500 menjadi vector one hot encoding berukuran 50 menggunakan pad sequence
-1 Layer LSTM dengan 25 neuron sebagai hidden layer
-Layer Dropout dengan nilai 0,5
-Layer Output dengan fungsi aktivasi sigmoid dan 1 output (0 atau 1)

Sumber Dataset Komentar Instagram

```
Positif    1409
Negatif    1127
Name: sentiment, dtype: int64
```



Filter

	X	text	sentiment
2144	2144	ttp semangat hiraukan hinaan cacian dll bersamamu	2
2145	2145	moga sehat gubernurku	2
2146	2146	sehat sehat ya menciptakan	2
2147	2147	pemimpin ngurusin kemakmuran urusan rakyat bkn sib...	1
2148	2148	kota bekasi caplok	1
2149	2149	sehat aniesbpk gubernur yg tabah keras utk dki terbukti...	2
2150	2150	loh airnya ndak masukan tanah awas ntar dosa loh mela...	1
2151	2151	dr jabar	2
2152	2152	mata air keruh habibie ainun	2
2153	2153	smoga sehat slalu anies gubernur	2
2154	2154	hujan anugrah allah semoga allah menganugerahkan k...	2
2155	2155	semangat anies	2
2156	2156	gubernur keren gini bilang memalukan tsamara psianeh	2

Showing 2,146 to 2,159 of 2,544 entries, 3 total columns

- Dataset bersih berjumlah 2536
- Data yang dilatih berjumlah 1775 sampel dan data yang divalidasi berjumlah 761 sampel

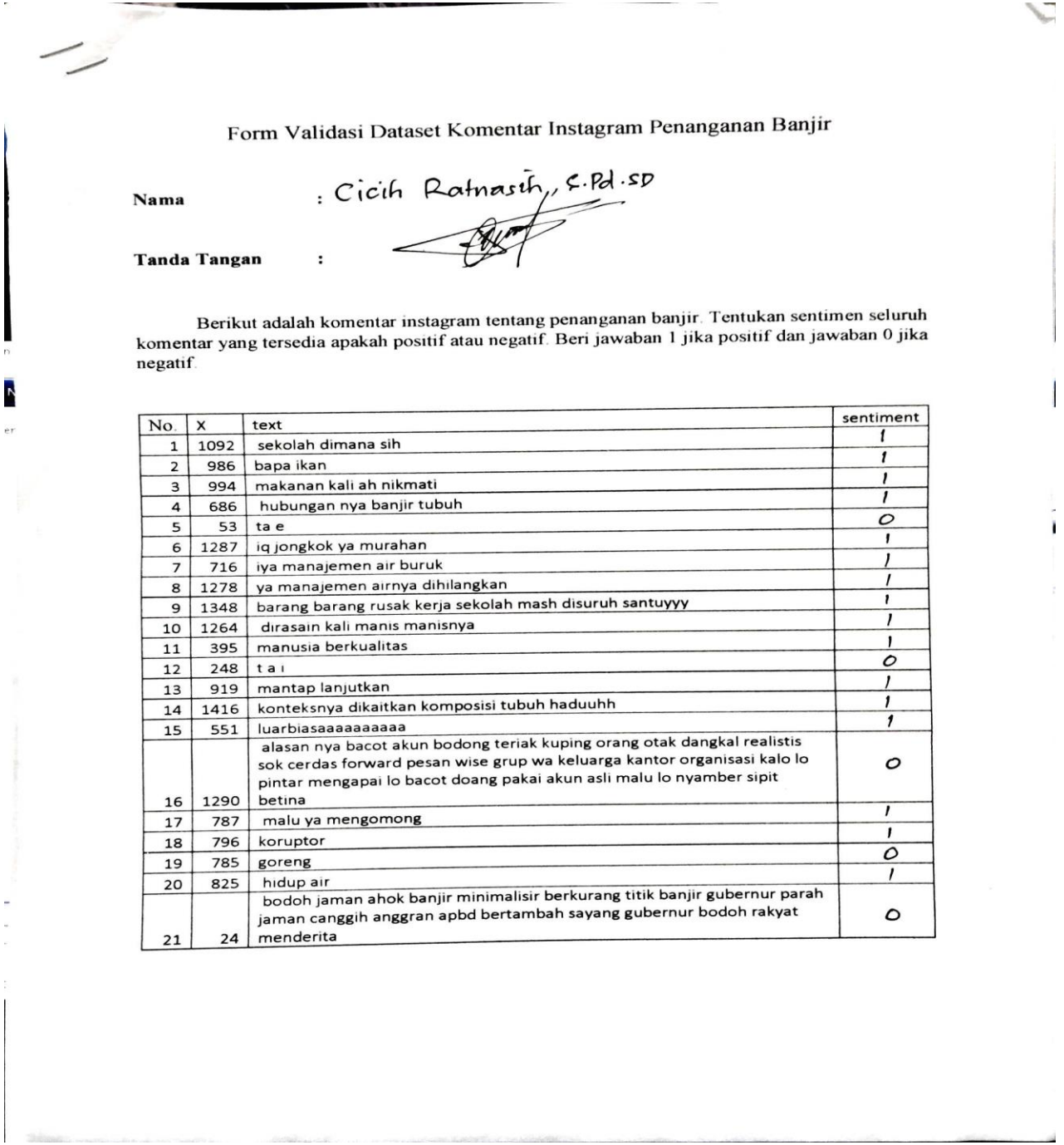
Proses Validasi

- K-Fold Cross Validation

```
------------------------------------------------------------------------
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.7882636196509923 - Accuracy: 75.55847764015198%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.7025580568319865 - Accuracy: 75.29566287994385%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.6940582425146316 - Accuracy: 75.55847764015198%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.6720041238682029 - Accuracy: 73.98160099983215%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.6425556579651250 - Accuracy: 75.68988204002380%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 75.21682024002075 (+- 0.6307503084348237)
> Loss: 0.6998879401661876
------------------------------------------------------------------------
```

- Wawancara dengan Ahli Bahasa

Komentar instagram yang digunakan dibandingkan oleh hasil wawancara ahli pakar bahasa Indonesia bernama Cicih Ratnasih, S.Pd.SD.



Form Validasi Dataset Komentar Instagram Penanganan Banjir

Nama : Cicih Ratnasih, S.Pd.SD

Tanda Tangan :

Berikut adalah komentar instagram tentang penanganan banjir. Tentukan sentimen seluruh komentar yang tersedia apakah positif atau negatif. Beri jawaban 1 jika positif dan jawaban 0 jika negatif.

No.	X	text	sentiment
1	1092	sekolah dimana sih	1
2	986	bapa ikan	1
3	994	makanan kali ah nikmati	1
4	686	hubungan nya banjir tubuh	1
5	53	ta e	0
6	1287	iq jongkok ya murahan	1
7	716	iya manajemen air buruk	1
8	1278	ya manajemen airnya dihilangkan	1
9	1348	barang barang rusak kerja sekolah mash disuruh santuyyy	1
10	1264	dirasain kali manis manisnya	1
11	395	manusia berkualitas	1
12	248	t a i	0
13	919	mantap lanjutkan	1
14	1416	konteksnya dikaitkan komposisi tubuh haduuhh	1
15	551	luarbiasaaaaaaaaaa	1
16	1290	alasan nya bacot akun bodong teriak kuping orang otak dangkal realistis sok cerdas forward pesan wise grup wa keluarga kantor organisasi kalo lo pintar mengapai lo bacot doang pakai akun asli malu lo nyamber sipit betina	0
17	787	malu ya mengomong	1
18	796	koruptor	1
19	785	goreng	0
20	825	hidup air	1
21	24	bodoh jaman ahok banjir minimalisir berkurang titik banjir gubernur parah jaman canggih anggran apbd bertambah sayang gubernur bodoh rakyat menderita	0

Confusion Matrix Hasil Wawancara Ahli Bahasa dan Model Terlatih

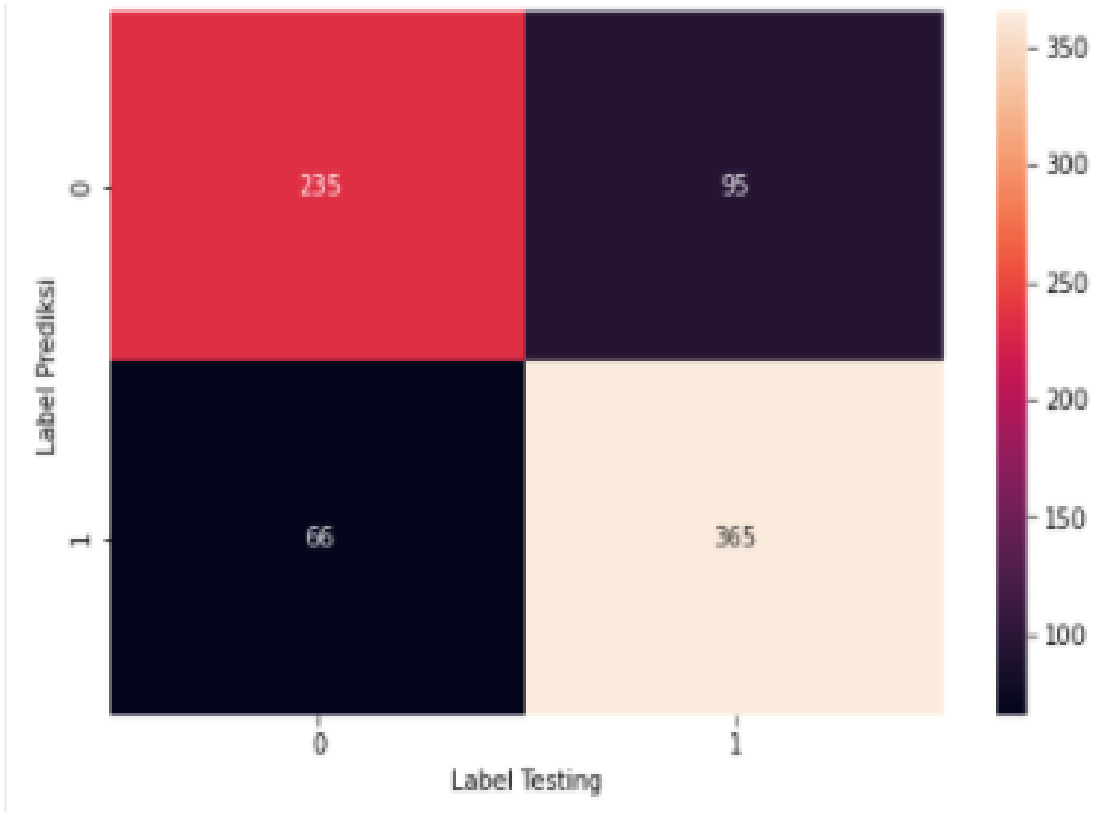
Hasil Wawancara Ahli Bahasa

```
Confusion Matrix :
[[ 65 238]
 [ 10 144]]
Accuracy Score : 0.4573304157549234
Report :
              precision    recall  f1-score   support

           0       0.87      0.21      0.34       303
           1       0.38      0.94      0.54       154

   micro avg       0.46      0.46      0.46       457
   macro avg       0.62      0.57      0.44       457
weighted avg       0.70      0.46      0.41       457
```

Model Terlatih



Perbandingan Besar C pada VSM tipe Linear

C sebesar 0,01

```
train accuracy=  93.34582942830365
test accuracy=  69.43231441048034
```

C sebesar 1

```
train accuracy=  95.43661971830986
test accuracy=  73.98160315374507
```

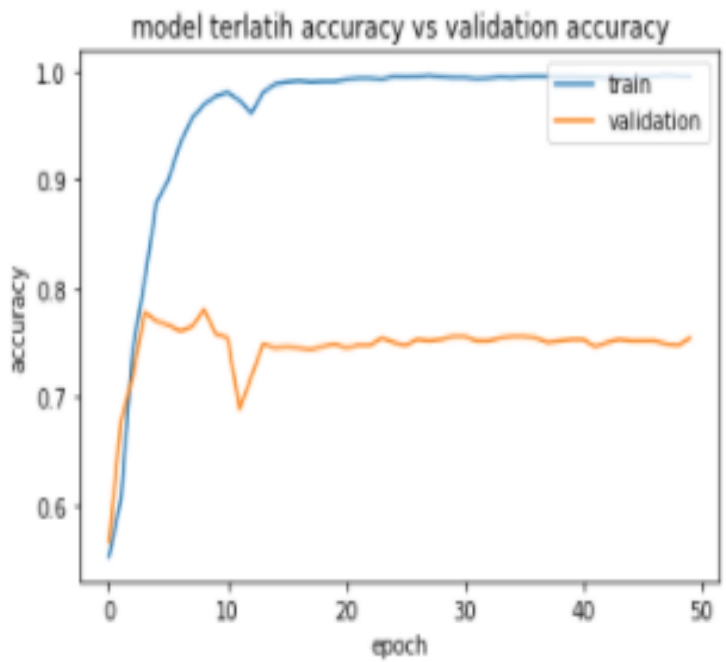
C sebesar 3

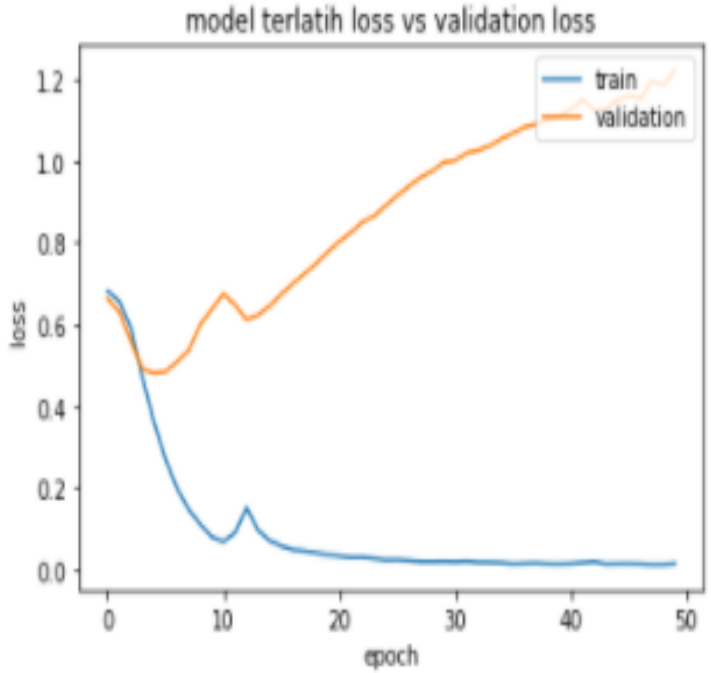
```
train accuracy=  93.34582942830365
test accuracy=  69.43231441048034
```

C sebesar 5

```
train accuracy=  92.68978444236177
test accuracy=  70.96069868995633
```

Performa RNN Tipe LSTM

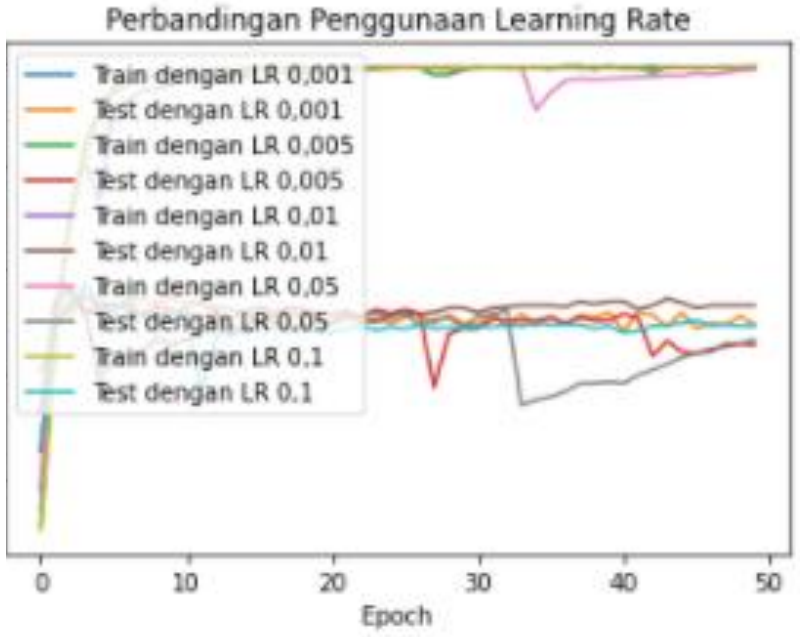





```
y_pred = model.predict(X_test)
accuracy = accuracy_score(ytest, y_pred.round())
print('Accuracy: %f' % accuracy)
# precision tp / (tp + fp)
precision = precision_score(ytest, y_pred.round())
print('Precision: %f' % precision)
# recall: tp / (tp + fn)
recall = recall_score(ytest, y_pred.round())
print('Recall: %f' % recall)
# f1: 2 tp / (2 tp + fp + fn)
f1 = f1_score(ytest, y_pred.round())
print('F1 score: %f' % f1)
```

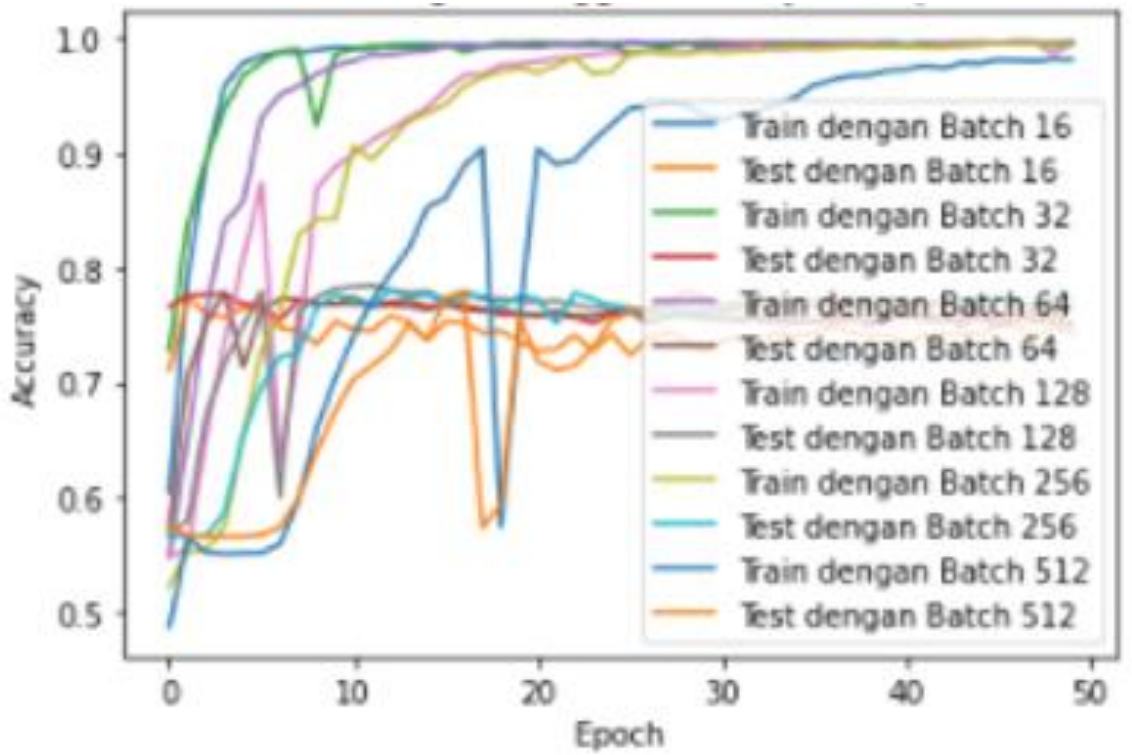
```
Accuracy: 0.788436
Precision: 0.793478
Recall: 0.846868
F1 score: 0.819304
```

Perbandingan Learning Rate



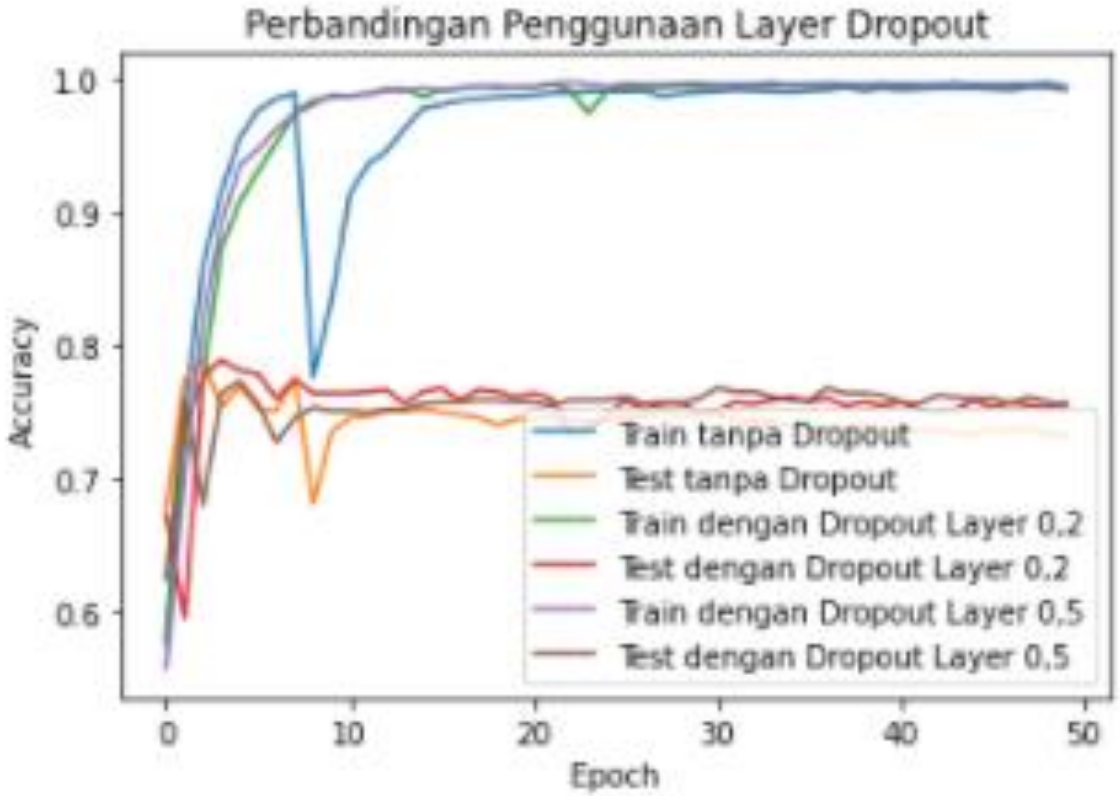
- Learning rate 0,05 menghasilkan performa terburuk pada fase training dan validasi
- Learning rate 0,01 menghasilkan performa terbaik pada fase validasi

Perbandingan Ukuran Batch



- Performa terburuk dihasilkan oleh ukuran batch sebesar 16
- Ukuran batch 32 dan 512 memiiliki performa terbaik pada fase training
- Ukuran batch 256 memiliki performa terbaik pada fase validasi

Perbandingan Dropout



- Penggunaan Dropout layer sebesar 0,2 memberikan performa terbaik pada epoch awal
- Penggunaan Dropout layer sebesar 0,5 memberikan performa terbaik pada epoch selanjutnya
- Performa terburuk dihasilkan oleh model algoritma tanpa menggunakan Dropout layer

Bab VI Kesimpulan

- Model *LSTM* memiliki akurasi sebesar 78,84%, presisi sebesar 79,34%, *recall* sebesar 84,68%, dan skor F1 sebesar 81,93%. Sedangkan model *SVM* hanya memiliki akurasi sebesar 73,98%, presisi sebesar 69,5%, recall sebesar 57,5%, dan skor F1 sebesar 53%.
- Model *LSTM* mampu mencapai 99%, sedangkan model *SVM* hanya dapat mencapai 93%.

Terimakasih