

**SKRIPSI**

**PERBANDINGAN ANALISIS SENTIMEN INSTAGRAM  
TENTANG PENANGANAN BANJIR DENGAN METODE  
SVM DAN RNN TIPE LSTM**

**Oleh**  
**Herry Wijaya**  
**065116076**



**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS PAKUAN**  
**BOGOR**  
**2019**

# **SKRIPSI**

## **PERBANDINGAN ANALISIS SENTIMEN INSTAGRAM TENTANG PENANGANAN BANJIR DENGAN METODE SVM DAN RNN TIPE LSTM**

Diajukan sebagai salah satu syarat untuk memperoleh  
Gelar Sarjana Komputer Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam

**Oleh**  
**Herry Wijaya**  
**065116076**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PAKUAN  
BOGOR  
2019**

## **HALAMAN PERSEMBAHAN**

Puji syukur kehadiran Tuhan YME yang senantiasa memberikan rahmatNya. Skripsi ini dipersembahkan kepada pihak-pihak yang telah memberikan dukungan sehingga Skripsi ini selesai. Dengan rasa syukur yang mendalam, dengan telah diselesaikannya Skripsi ini Penulis mempersembahkannya kepada:

1. Orang tua Penulis yang telah senantiasa membantu menyelesaikan Skripsi ini.
2. Pembimbing pendamping Ibu Mulyati, M.Kom dan pembimbing utama Ibu Dr. Herfina, M.Kom yang telah memberi masukan, semangat, dan arahan hingga akhirnya dapat terselesaikan Skripsi ini.

## **HALAMAN PENGESAHAN**

Judul : Perbandingan Analisis Sentimen Penanganan Banjir dengan Metode SVM  
dan RNN Tipe LSTM

Nama : Herry Wijaya

NPM : 065116076

Mengesahkan,

Pembimbing Pendamping  
Program Studi Ilmu Komputer  
FMIPA-UNPAK

Pembimbing Utama  
Program Studi Ilmu Komputer  
FMIPA-UNPAK

Mulyati, M.Kom

Dr. Herfina, M.Kom

Mengetahui,

Ketua Program Studi Ilmu Komputer  
FMIPA-UNPAK

Dekan  
FMIPA-UNPAK

Lita Karlitasari, S. Kom., MMSI.

Asep Denih, S.Kom.,  
M.Sc., Ph.D.

## **PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI**

Dengan ini saya menyatakan bahwa: Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian- bagian di mana sumber informasinya dicantumkan dengan cara referensi yang semestinya. Demikian pernyataan ini saya buat dengan sebenarbenarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, Januari 2021

Herry Wijaya

## RIWAYAT HIDUP



Penulis dilahirkan di Bogor pada tanggal 8 April 1998 dari pasangan Bapak Thio Bun Wee dan Ibu Surjani sebagai anak kedua dari dua bersaudara.

Penulis memulai pendidikan di Sekolah Dasar yang bertempat di SD Mardi Waluya, kemudian tahun 2010 masuk SLTP Mardi Waluya di Bogor dan Penulis adalah Alumni dari SMA Mardi Yuana Bogor.

Pada tahun 2016 penulis meneruskan pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Pada bulan Januari tahun 2021 penulis menyelesaikan penelitian dengan judul Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode *RNN* dan *SVM*.

## **PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

---

Saya yang bertandatangan di bawah ini :

Nama	: Herry Wijaya
NPM	: 065116076
Judul Skripsi	: Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode <i>RNN dan SVM</i>

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, Januari 2021

Herry Wijaya  
065116076

## RINGKASAN

**Herry Wijaya (065116076).** Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode *RNN* dan *SVM*. Di bawah bimbingan **Ibu Mulyati, M.Kom** dan **Ibu Dr. Herfina, M.Kom**.

Instagram adalah salah satu media sosial yang paling populer di Indonesia. Pemerintah menggunakannya sebagai media penyampaian informasi terkait aspek umum ataupun permasalahan-permasalahan untuk masyarakat. Salah satu contohnya adalah memberikan informasi kegiatan yang sedang berlangsung atau akan direncanakan terkait penanganan banjir yang dilakukan di Jakarta. Dengan hal ini, maka masyarakat dapat mengetahui informasi penting dan juga memberikan komentar terkait kegiatan-kegiatan tersebut. Penelitian ini merupakan analisis sentimen yang membandingkan performa model algoritma SVM (Support Vector Machine) dan RNN tipe LSTM (Recurrent Neural Network) menggunakan dataset komentar Instagram tentang penanganan banjir yang terjadi di Jakarta. Analisis sentimen ini bertipe klasifikasi biner dengan dua fitur, yaitu data teks komentar sebagai input observasi dan polaritas sentimen positif dan negatif sebagai label kelas yang akan diprediksi. Dataset komentar Instagram yang dilatih menggunakan kedua model algoritma akan dievaluasi agar diketahui seberapa baik model tersebut dalam memprediksi apakah suatu komentar berpolaritas sentimen positif atau negatif. Pengukuran evaluasi yang dihasilkan kedua model algoritma dibandingkan menggunakan confusion matrix. Hasil penelitian membuktikan bahwa model algoritma RNN tipe LSTM memiliki performa yang lebih baik daripada model algoritma SVM. Model LSTM memiliki akurasi sebesar 78,84%.

**Kata kunci :** Analisis Sentimen, Long Short Term Memory (LSTM), konfigurasi hyperparameter, confusion matrix, klasifikasi biner.



## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Tuhan YME karena atas rahmat dan hidayah serta izin-Nya penulis dapat menyelesaikan penulisan skripsi dengan judul **“Perbandingan Analisis Sentimen Instagram Tentang Penanganan Banjir dengan Metode SVM dan RNN Tipe LSTM”**.

Penyusunan laporan ini tidak dapat terselesaikan dengan baik tanpa adanya bantuan moral dan motivasi dari berbagai pihak. Oleh karena itu penulis dengan sepenuh hati ingin mengucapkan terimakasih kepada:

1. Dr. Herfina, M.Kom., selaku pembimbing utama yang telah berkenan meluangkan waktunya, memberikan petunjuk, serta saran dalam penyusunan laporan penelitian ini.
2. Ibu Mulyati, M.Kom., selaku pembimbing pendamping yang telah meluangkan waktunya dan memberikan arahan serta saran dalam penyusunan laporan penelitian ini.
3. Ibu Lita Karlitasari, S. Kom, MMSI., selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Pakuan Bogor.
4. Kedua orang tua yang telah memberikan perhatian dan senantiasa memberikan dorongan moral, material, dan motivasi kepada penulis.

Penulis menyadari bahwa penulisan laporan ini masih jauh dari sempurna. Untuk itu penulis mengharapkan kritik dan saran agar dapat memperbaiki laporan ini menjadi lebih baik. Penulis berharap laporan ini dapat bermanfaat bagi penulis dan masyarakat sebagai sumber pustaka dan menambah ilmu pengetahuan.

Bogor, Januari 2021

Penulis

## DAFTAR ISI

<b>HALAMAN COVER</b> .....	ii
<b>HALAMAN PERSEMBAHAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	iv
<b>PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI</b> .....	v
<b>RIWAYAT HIDUP</b> .....	vi
<b>PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA</b> .....	vii
<b>RINGKASAN</b> .....	viii
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiii
<b>DAFTAR LAMPIRAN</b> .....	xiv
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	2
1.3 Ruang Lingkup.....	2
1.4 Manfaat .....	2
<b>BAB II TINJAUAN PUSTAKA</b> .....	3
2.1. Dasar Teori.....	3
2.1.1 Analisis Sentimen.....	3
2.1.2 <i>Support Vector Machine</i> .....	3
2.1.3 <i>Recurrent Neural Network</i> .....	5
2.1.4 <i>Long Short Term Memory</i> .....	6
2.2. Penelitian Terdahulu .....	7
<b>BAB III METODE PENELITIAN</b> .....	9
3.1 Metode Penelitian .....	9
3.2 Tahapan Penelitian.....	9
3.2.1 <i>Selection</i> .....	9
3.2.2 <i>Preprocessing</i> .....	9

3.2.3 <i>Transformation</i> .....	9
3.2.4 <i>Data Mining</i> .....	9
3.2.5 Interpretasi dan Evaluasi.....	10
<b>BAB IV PERANCANGAN DAN IMPLEMENTASI</b> .....	11
4.1 Analisis.....	11
4.2 Perancangan .....	11
4.2.1 <i>Design Sistem</i> .....	11
4.2.2 Tahap <i>Preprocessing</i> .....	12
4.2.3 <i>Labeling</i> Polaritas Sentimen .....	13
4.2.4 Tahap Transformasi .....	14
4.2.5 Penerapan Model Algoritma <i>RNN</i> Tipe <i>LSTM</i> dan <i>VSM</i> .....	15
4.3 Implementasi .....	18
<b>BAB V HASIL DAN PEMBAHASAN</b> .....	19
5.1 Hasil.....	19
5.1.1 Penggunaan <i>Library</i> Python.....	19
5.1.2 Proses <i>Preprocessing</i> .....	19
5.1.3 Struktur <i>LSTM</i> .....	20
5.1.4 Proses Validasi .....	20
5.1.5 Proses Hasil Pengujian Model .....	21
5.2 Pembahasan .....	21
5.2.1 Evaluasi Data Uji .....	22
5.2.2 Hasil Pengujian dengan Ahli Bahasa .....	23
<b>BAB VI KESIMPULAN DAN SARAN</b> .....	27
6.1. Kesimpulan.....	27
6.2. Saran.....	27
<b>DAFTAR PUSTAKA</b> .....	28
<b>LAMPIRAN</b> .....	29

## DAFTAR GAMBAR

<b>Gambar 1.</b> <i>Margin Hyperplane Maksimum SVM</i> .....	4
<b>Gambar 2.</b> <i>Framework RNN</i> .....	5
<b>Gambar 3.</b> <i>Flowchart LSTM</i> .....	6
<b>Gambar 4.</b> <i>Proses Knowledge Discovery and Data Mining</i> .....	8
<b>Gambar 5.</b> <i>Flowchart Program</i> .....	11
<b>Gambar 6.</b> <i>Flowchart Tahap Preprocessing</i> .....	12
<b>Gambar 7.</b> <i>Proses Case Folding</i> .....	13
<b>Gambar 8.</b> <i>Proses Tokenizing</i> .....	13
<b>Gambar 9.</b> <i>Proses Filtering</i> .....	13
<b>Gambar 10.</b> <i>Source Code Implementasi Analisis Sentimen</i>	
<i>Menggunakan Streamlit</i> .....	18
<b>Gambar 11.</b> <i>Implementasi Analisis Sentimen</i>	
<i>Menggunakan Streamlit 1</i> .....	18
<b>Gambar 12.</b> <i>Implementasi Analisis Sentimen</i>	
<i>Menggunakan Streamlit 2</i> .....	18
<b>Gambar 13.</b> <i>Pemanggilan Library Modul Python</i> .....	19
<b>Gambar 14.</b> <i>Proses Processing Dataset Komentar Instagram</i> .....	19
<b>Gambar 15.</b> <i>Struktur Model RNN Tipe LSTM</i> .....	20
<b>Gambar 16.</b> <i>Hasil Validasi Menggunakan K-Fold Cross Validation</i> .....	21
<b>Gambar 17.</b> <i>Proses Hasil Pengujian Model</i> .....	21
<b>Gambar 18.</b> <i>Grafik Jumlah Sentimen</i> .....	22
<b>Gambar 19.</b> <i>Nilai Loss dan Akurasi Training dan Validasi</i> .....	22
<b>Gambar 20.</b> <i>Metric Model LSTM</i> .....	23
<b>Gambar 21.</b> <i>Metric Model VSM</i> .....	23
<b>Gambar 22.</b> <i>Confusion Matrix Hasil Perbandingan Ahli Pakar</i> .....	24
<b>Gambar 23.</b> <i>Perbandingan Perbedaan Ukuran Batch dan Layer Dropout</i> .....	25
<b>Gambar 24.</b> <i>Perbandingan Penggunaan Learning Rate Berbeda</i> .....	25

## DAFTAR TABEL

<b>Tabel 1.</b> Perbandingan Penelitian Sebelumnya dengan Penelitian yang akan Dikembangkan .....	8
<b>Tabel 2.</b> Sampel <i>Dataset</i> Komentar Instagram Penanganan Banjir .....	12
<b>Tabel 3.</b> Tabel Polaritas Sentimen .....	14
<b>Tabel 4.</b> Tabel <i>TF-IDF</i> .....	15
<b>Tabel 5.</b> Tabel <i>TF-IDF 2</i> .....	15
<b>Tabel 6.</b> Sampel Data <i>One Hot Encoding</i> .....	15
<b>Tabel 7.</b> Hasil Perhitungan <i>LSTM</i> .....	17
<b>Tabel 8.</b> Tabel <i>Confusion Matrix</i> .....	23
<b>Tabel 9.</b> Tabel Akurasi Validasi di Titik <i>Global Maximum</i> .....	25
<b>Tabel 10.</b> Tabel Akurasi Validasi di Titik <i>Global Maximum 2</i> .....	26

## DAFTAR LAMPIRAN

<b>Lampiran 1.</b> Data Hasil Sentimen Bahasa .....	30
<b>Lampiran 2.</b> Data Hasil Prediksi Model.....	31
<b>Lampiran 3.</b> Tahap Seleksi dan Preprocessing .....	33
<b>Lampiran 4.</b> Tahap Transformasi.....	33
<b>Lampiran 5.</b> Tahap Data Mining .....	35
<b>Lampiran 6.</b> Tahap Interpretasi dan Evaluasi .....	37

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sejak awal 1 Januari 2020 sudah sering terjadi musibah banjir di ibukota Jakarta dan area metropolitan sekitarnya. Musibah ini merugikan harta benda dan merusak fasilitas umum milik masyarakat serta memakan korban jiwa yang tidak sedikit jumlahnya. Banjir yang terjadi disebabkan oleh curah hujan yang deras sehingga sungai Ciliwung dan Cisadane tidak mampu menjaganya melebihi kapasitas. Alasan lainnya adalah jumlah penduduk yang tidak terkontrol dan kurangnya pemahaman tentang perencanaan lahan.

Analisis Sentimen adalah proses otomatis yang menentukan apakah bagian suatu teks mengandung konten objektif atau berupa opini yang dapat dianalisis lebih lanjut dengan polaritas sentimen teks tersebut (Jianqiang, 2018). Analisis sentimen dapat digunakan untuk menganalisis sentimen masyarakat terhadap permasalahan banjir yang terjadi. Hal ini memerlukan data teks yang mewakili komentar masyarakat sebagai bahan analisis, sehingga sosial media merupakan salah satu *channel* yang dapat diambil untuk menggali data yang diperlukan dalam penelitian. Beberapa metode klasifikasi yang digunakan dalam analisis sentimen adalah *RNN* dan *SVM*. *RNN* merupakan bagian dari *Deep Learning* dimana nilai akurasi yang dihasilkan dapat lebih baik dibandingkan dengan jaringan syaraf tiruan sederhana karena bobot yang dihitung akan lebih akurat mendekati persamaan setiap kata. Jenis *RNN* yang digunakan adalah *LSTM* (*Long Short Term Memory*) untuk menutupi kekurangan pada *RNN* yang tidak dapat menyimpan memori untuk dipilah dan menambahkan mekanisme *attention* agar setiap kata dapat lebih fokus pada konteks (Ivanedra, 2019). Sedangkan *SVM* (*Support Vector Machine*) merupakan algoritma *machine learning* yang sudah sering digunakan dalam mendeteksi polaritas suatu teks (Ahmad, 2017). Kelebihan *SVM* dalam sentiment analysis adalah memiliki performa yang baik jika jumlah sampel lebih banyak daripada jumlah *feature* dan dataset memiliki margin pemisahan yang jelas dalam setiap kelas.

Beberapa penelitian terdahulu yang sudah dilakukan mengenai analisis sentimen menggunakan metode *RNN* dan *SVM* diantaranya oleh Pusphita A., dkk. (2014) yang berjudul Penerapan Metode Klasifikasi *Support Vector Machine* (*SVM*) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. Metode yang digunakan adalah *SVM*. Hasil penelitiannya adalah klasifikasi dengan fungsi *kernel RBF* menghasilkan 100% akurasi pada data training dan 93,902% pada data latih. Sedangkan klasifikasi dengan fungsi *kernel polynomial* menghasilkan 98,810% akurasi pada data training dan 92,683% pada data latih. Penelitian selanjutnya dilakukan oleh Munir A., dkk. (2017) yang berjudul *Sentiment Analysis of Tweets using SVM*. Metode yang digunakan adalah *SVM*. Hasil penelitiannya adalah pada kedua *dataset* yang diteliti (*dataset review* pelanggan tentang *self-driving cars* dan *Apple*), *dataset self-driving cars* menghasilkan akurasi 59,91% dan *dataset Apple* menghasilkan akurasi 71,2%. Penelitian selanjutnya dilakukan oleh Nurrohmat MA & Azhari SN (2019) yang berjudul *Sentiment Analysis of Novel Review using Long Short Term Memory Method*. Metode yang digunakan adalah *RNN* bertipe *LSTM*. Hasil

penelitiannya adalah metode *LSTM* menghasilkan 72,85% akurasi jika menggunakan 2 layer *LSTM* dan 71,20% akurasi jika menggunakan 1 layer *LSTM*.

Berdasarkan permasalahan dan penelitian terkait, maka dibuatlah perbandingan analisis sentimen instagram tentang penanganan banjir dengan menggunakan metode *RNN* dan *SVM*. Diharapkan penelitian ini dapat bermanfaat untuk mengetahui perbedaan antara *deep learning* dengan *machine learning* tradisional. Perbedaan ini meliputi arsitektur model yang lebih rumit dan adanya beragam *hyperparameter* unik dalam setiap layer jaringan syaraf tiruan. Hal ini memberikan kesempatan untuk beragam konfigurasi yang berbeda dalam pelatihan model sehingga diperoleh output yang berbeda, apakah lebih baik atau lebih buruk. Dengan demikian, *deep learning* memberikan kesempatan yang lebih luas untuk membangun model yang cocok dan optimal.

## 1.2 Tujuan

Tujuan dilakukannya penelitian ini adalah untuk melakukan Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode *RNN* dan *SVM*.

## 1.3 Ruang Lingkup

Adapun ruang lingkup dari penelitian ini adalah:

1. Penelitian ini menganalisis sentimen instagram tentang penanganan banjir dengan Metode *RNN* dan *SVM* sebagai studi kasus klasifikasi biner.
2. Analisis hasil performa dari kedua metode tersebut dilakukan berdasarkan 2536 komentar instagram yang sudah dibersihkan dan diberi label polaritas sentimen.
3. Bahasa pemrograman yang digunakan adalah Python dan R. *Software* yang digunakan adalah *Jupyter Notebook* dan Rstudio.

## 1.4 Manfaat

Manfaat dari pengembangan Perbandingan Analisis Sentimen Instagram tentang Penanganan Banjir dengan Metode *RNN* dan *SVM* adalah sebagai berikut.

1. Mengetahui perbedaan performa antara *deep learning* dengan metode *machine learning* tradisional.
2. Mengetahui pengaruh *hyperparameter RNN* tipe *LSTM* kepada akurasi yang dihasilkan.
3. Mengetahui pengaruh *hyperparameter SVM* kepada akurasi yang dihasilkan.
4. Mengetahui perbandingan akurasi yang dihasilkan antara metode *RNN* tipe *LSTM* dengan *SVM*.



## BAB II TINJAUAN PUSTAKA

### 2.1. Dasar Teori

#### 2.1.1 Analisis Sentimen

Analisis sentimen adalah proses otomatis yang menentukan apakah bagian suatu teks mengandung konten objektif atau berupa opini yang dapat dianalisis lebih lanjut dengan polaritas sentimen teks tersebut (Jianqiang, 2018). Analisis sentimen dapat dibagi menjadi dua, yaitu metode berbasis *lexicon* dan metode *machine learning*. Metode berbasis *lexicon* menggunakan *lexicon* untuk memberi *tag* kepada kata-kata di kalimat yang muncul menjadi label positif atau negatif, kemudian sentimen dari seluruh dokumen disimpulkan berdasarkan kata-kata yang diberi tag. Metode berbasis *lexicon* merupakan pembelajaran tidak tersupervisi sehingga tidak memerlukan label polaritas. Sedangkan metode *machine learning* digunakan untuk permasalahan klasifikasi teks. Dalam metode ini, fitur yang tersedia seperti *unigram*, *bigram*, atau *word embeddings* diekstraksi dari teks, kemudian diberikan kepada model klasifikasi seperti *SVM* dan *RNN*. Metode *machine learning* merupakan pembelajaran tersupervisi sehingga diperlukan banyak data latih yang disertai label polaritas (Zou, 2018).

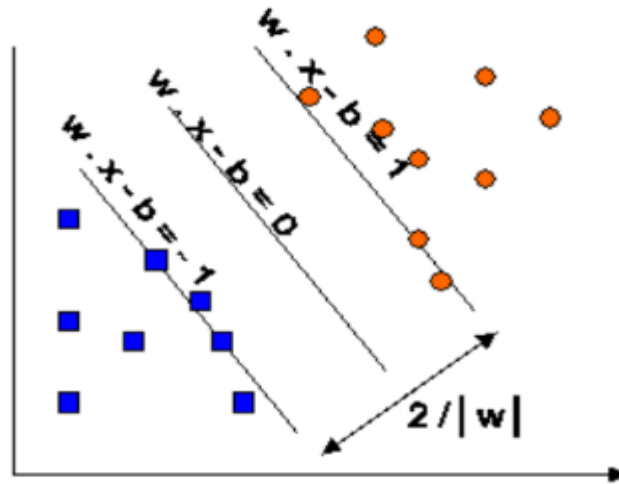
#### 2.1.2 Support Vector Machine

Algoritma *Support Vector Machine* adalah salah satu *classifier* untuk proses klasifikasi yang digunakan di pendekatan *machine learning*. Teknik ini berusaha untuk mencari fungsi pemisahan yang paling optimum untuk memisahkan data-data dari kelas yang berbeda (Lutfi, 2018). Pemisahan tersebut berupa *hyperplane*, yaitu daerah yang memisahkan dua bagian di mana setiap kelas berada di sisi masing-masing. Parameter-parameter yang terdapat di algoritma *Support Vector Machine* adalah *kernel*, *regularization*, *gamma*, dan *margin*. *Kernel* adalah parameter yang mengubah permasalahan ke dalam aljabar linear pada saat pembelajaran *hyperplane*. Hasil perhitungan operasi aljabar tersebut adalah objek vektor yang mewakili setiap data latih di daerah *n*-dimensi. *Regularization* adalah parameter yang mengoptimasi agar data latih dapat terhindar dari salah klasifikasi. Semakin besar nilai *regularization* maka optimasi akan memilih *hyperplane* dengan margin yang lebih kecil. Sedangkan jika semakin kecil nilai *regularization* maka optimasi akan memilih *hyperplane* dengan margin yang lebih besar. *Gamma* adalah parameter yang mendefinisikan bagaimana pentingnya setiap data latih. Semakin besar nilai *gamma* maka hanya poin yang dekat dengan margin akan diperhitungkan untuk garis pemisah. Sedangkan jika semakin kecil nilai *gamma* maka poin yang jauh dengan margin akan diperhitungkan juga untuk garis pemisah. *Margin* adalah parameter yang berupa garis pemisah poin kelas terdekat. *Margin* yang baik memiliki pemisahan yang besar kepada kedua kelas. Sedangkan margin yang buruk memiliki pemisahan yang kecil di mana ada poin kelas yang terletak dekat dengan garis pemisah.

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), \dots, (x_n, y_n)\} \quad (1)$$

$$\begin{aligned} w \cdot x + b &= 0 \\ w \cdot x_i - b &= 1 \quad \text{atau} \quad w \cdot x_i - b = -1 \end{aligned} \quad (2)$$

Algoritma *SVM* memetakan input vektor ke ruang dimensi lebih besar dimana terbentuk *hyperplane* pemisah. Dua *hyperplane* paralel terbentuk pada kedua sisi yang memisahkan data. Semakin besar *margin* atau besar jarak antara kedua *hyperplane* maka akan semakin baik generalisasi error yang dihasilkan *classifier* (Srivastava, 2010). Data point memiliki bentuk  $(x,y)$  yang mewakili titik lokasi di ruang dimensi.  $Y_n = 1 / -1$  merupakan konstanta yang menentukan kelas dimana poin  $x_n$  berada.  $N$  adalah jumlah sampel. Setiap  $x_n$  adalah  $p$ -dimensional vektor real.  $B$  adalah skalar dan  $w$  adalah  $p$ -dimensional vector. Vektor  $W$  menunjukkan *perpendicular* kepada *hyperplane* paralel. Jarak antara *hyperplane* adalah  $2 / |w|$ .



**Gambar 1** Margin Hyperplane Maksimum SVM

Sampel yang berada di *hyperplane* disebut *support vector*. *Hyperplane* yang memisahkan dengan *margin* terbesar  $M = 2 / |w|$  merupakan *vector machine* yaitu data point latih terdekat ke *hyperplane*.

$$y_j [w^T \cdot x_j + b] = 1, i=1 \quad (3)$$

*Optimal Canonical Hyperplane* adalah *canonical hyperplane* yang memiliki *margin* maksimum.  $N$  adalah jumlah data point latih. Agar ditemukan *hyperplane* pemisah optimal yang memiliki *margin* maksimum, maka harus dilakukan hal berikut.

$$Y_i [w^T \cdot x_i + b] \geq 1 \quad ; \quad i = 1, 2, \dots, n \quad (4)$$

Permasalahan optimasi diselesaikan dengan fungsi *Lagrange*.  $A_i$  adalah *lagrange multiplier*. Pencarian optimal *saddle point* penting dikarenakan *Lagrange* harus minimalkan besar  $w$  dan  $b$  serta harus memaksimalkan *nonnegative*  $A_i$  ( $A_i \geq 0$ ). Permasalahan dapat diselesaikan dengan *primal form* (bentuk  $w$  dan  $b$ ) atau *dual form* (bentuk  $A_i$ ).

$$\begin{aligned} L_p = L(w, b, \alpha) &= 1/2 \|w\|^2 - \sum_{i=1} \alpha_i (y_i (w^T x_i + b) - 1) \\ &= 1/2 w^T w - \sum_{i=1} A_i (y_i (w^T x_i + b) - 1) \end{aligned} \quad (5)$$

*Primary form* diubah menjadi dual form dengan substitusi  $w_0$  dan  $b_0$  kepada  $L_P$ . Untuk menentukan *hyperplane* optimal, *dual lagrangian* ( $L_d$ ) dimaksimalkan dengan *nonnegative*  $a_i$  dan *equality constraint*.

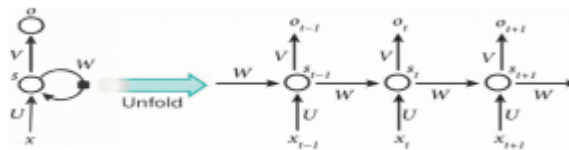
$$\begin{aligned}
\partial L / \partial w_0 &= 0 \\
w_0 &= \sum a_i y_i x_i \\
\partial L / \partial b_0 &= 0 \\
\sum a_i y_i &= 0 \\
L_d(\alpha) &= \sum a_i - 1/2 \sum a_i a_j y_i y_j x_i^T x_j \\
a_i &\geq 0, i = 1, 2, \dots, n \\
\sum_{i=1} a_i y_i &= 0
\end{aligned} \tag{6}$$

Vektor latih  $x_1$  dialokasikan ke ruang dimensional oleh fungsi *lagrange*. Algoritma SVM memiliki fungsi kernel yang berguna untuk mencari *hyperplane linear* pemisah dengan *margin* maksimal di ruang dimensional ini.  $\gamma$ ,  $r$ , dan  $d$  adalah parameter *kernel*. Terdapat beberapa fungsi *kernel*, yaitu sebagai berikut.

- *Linear kernel* :  $K(x_i, x_j) = x_i^T x_j$
- *Polynomial kernel* :  $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- *RBF kernel* :  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- *Sigmoid kernel* :  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

### 2.1.3 Recurrent Neural Network

*Recurrent Neural Network* adalah salah satu model jaringan saraf tiruan, di mana ia mampu menyimpan seluruh nilai input yang sebelumnya telah diproses di jaringan dan yang sedang diproses. Nilai yang tersembunyi pada setiap lapisan jaringan tergantung pada seluruh input yang telah dilihat (Thomas, 2018). Berbeda dengan jaringan saraf biasa, *recurrent neural network* dapat menerima input secara series tanpa adanya batasan ukuran yang telah ditentukan terlebih dahulu. Sehingga satu input dari series memiliki relasi dengan input lainnya dan akan memiliki pengaruh kepada input tetangganya. Relasi dan pengaruh tersebut diwakili sebagai *hidden state vector*, yang merepresentasikan konteks input berdasarkan input atau output sebelumnya. Setelah output dihasilkan, *hidden state* ini akan terupdate berdasarkan input dan akan digunakan lagi untuk pemrosesan berikutnya. Dengan saling berbagi *parameter*, *weight* dari setiap input juga berubah sesuai *hidden state* yang dihasilkan.



**Gambar 2** Framework RNN

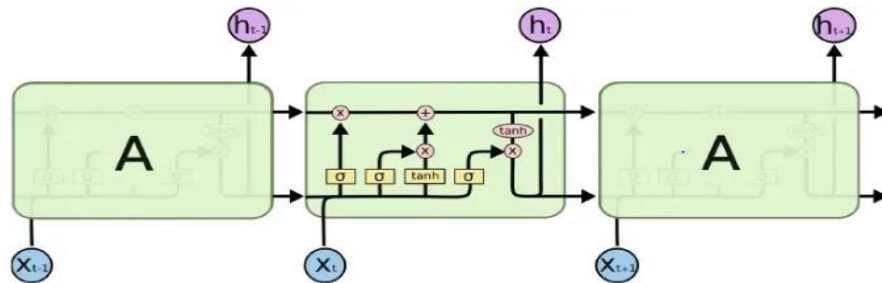
Gambar 2 merupakan *framework* RNN tradisional, *framework* ini membentuk jaringan beberapa node yang terorganisir menjadi beberapa layer. Setiap node dalam satu layer terhubung kepada node lain dalam layer selanjutnya (Patel, 2019).  $U$  merupakan node input yang memperoleh data dari luar jaringan,  $V$  merupakan node output yang menghasilkan output,  $W$  merupakan node hidden yang memodifikasi data dari input ke output berdasarkan *hidden state*. Persamaan  $A_t$  merupakan output dari

node sebelumnya,  $\theta$  menyatakan fungsi aktivasi yang digunakan,  $h_{t-1}$  menyatakan langkah waktu pada layer sebelumnya, dan  $x_t$  menyatakan baris input ( $x_0, x_1, x_2, \dots, x_n$ ).

$$A_t = \theta (h_{t-1}, x_t) \quad (7)$$

#### 2.1.4 Long Short Term Memory

Jaringan *LSTM* adalah tipe jaringan *RNN* yang sudah diperbaiki untuk mencegah permasalahan vanishing gradient. Permasalahan *vanishing gradient* terjadi jika *weight* di dalam jaringan memiliki fungsi *error* dalam kegiatan *update* kepada *weight* yang dihasilkan dalam setiap iterasi latihan. Hal ini akan menyebabkan gradien yang menghilang sedikit demi sedikit, sehingga akan menghalangi *weight* untuk berubah nilai. Gradien yang dimaksud adalah nilai yang mewakili parameter yang dimiliki jaringan.



Gambar 3 Flowchart LSTM

Cara kerja *LSTM* adalah menggantikan arsitektur *neuron* tersembunyi oleh fungsi aktivasi dari sel memori saat beberapa gerbang mengatur input dan outputnya masing-masing. Pada persamaan di bawah,  $C_t$  merupakan unit sel memori di *LSTM*,  $i_t$  merupakan gerbang input,  $F_t$  adalah gerbang lupa,  $b_f$  adalah bias gerbang lupa,  $U_f$  adalah *weight* koneksi kepada *hidden state* sebelumnya, dan  $O_t$  merupakan gerbang output. Persamaan ini menentukan gerbang lupa yang akan memilih informasi mana yang akan dibuang di *cell state*.

$$F_t = \theta (U_f \cdot h_{t-1} + w_f x_t + b_f) \quad (8)$$

Informasi baru yang dapat disimpan di *cell state*. Proses ini dilakukan di layer input dan layer output.  $I_t$  merupakan layer input,  $u_t$  merupakan *weight* koneksi kepada *hidden state* sebelumnya,  $w_t$  adalah koneksi *weight* kepada alur input *pattern*, dan  $b_i$  adalah bias layer input.

$$I_t = \theta (u_i \cdot h_{t-1} + w_i x_t + b_i) \quad (9)$$

$C_t$  adalah layer keadaan sementara,  $U_c$  adalah *weight* koneksi kepada *hidden state* sebelumnya,  $w_c$  adalah *weight* koneksi kepada input *pattern*, dan  $b_c$  adalah bias layer keadaan sementara. Layer keadaan sementara termodifikasi dari keadaan sebelumnya dengan persamaan kedua.

$$C_t = \theta (U_c \cdot h_{t-1} + w_c x_t + b_c) \quad (10)$$

$$C_t = f_t * c_{t-1} + i_t * C_t \quad (11)$$

$O_t$  merupakan layer output,  $u_o$  adalah *weight* koneksi kepada *hidden state* sebelumnya,  $w_o$  adalah *weight* koneksi kepada input *pattern*, dan  $b_o$  adalah bias layer output. Pada persamaan kedua,  $H_t$  merupakan output LSTM.

$$O_t = \theta (u_o \cdot H_{t-1} + w_o x_t + b_o) \quad (12)$$

$$H_t = a_t * \theta (C_t) \quad (13)$$

## 2.2. Penelitian Terdahulu

Berikut ini adalah beberapa jurnal yang berkaitan dengan penelitian yang akan dilakukan.

1. 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.
2. Judul : Sentimen Analisis Tweet Berbahasa Indonesia dengan *Deep Believe Network* (Zulfa, 2017)  
 Deskripsi : Analisis sentimen 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.
3. Judul : Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode *Naive Bayes* (Gunawan, 2018)  
 Deskripsi : Analisis sentimen yang dilakukan menggunakan algoritma *naive bayes*. Penelitian menggunakan review customer kosmetik sebagai data teks yang sudah diberi label polaritas. Hasil penelitian tersebut menghasilkan 77,78 % akurasi tertinggi.

## 2.3 Tabel Perbandingan

Tabel 1 menunjukkan perbandingan penelitian sebelumnya dengan penelitian yang akan dikembangkan.

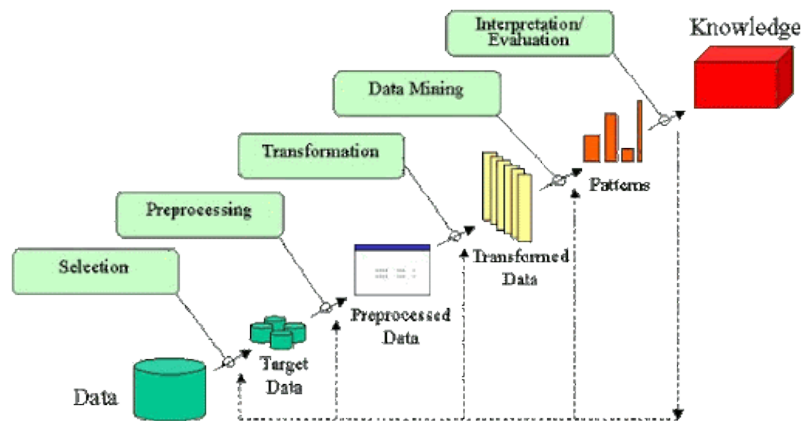
**Tabel 1** Perbandingan Penelitian Sebelumnya dengan Penelitian yang akan Dikembangkan

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

## BAB III METODE PENELITIAN

### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metodologi *Knowledge Discovery and Data Mining*. Gambar 4 menunjukkan tahapan-tahapan KDD.



Gambar 4 Proses *Knowledge Discovery and Data Mining*

### 3.2 Tahapan Penelitian

#### 3.2.1 Selection

Tahap seleksi dilakukan untuk memperoleh data target yang relevan untuk proses penelitian. Data target yang dimaksud adalah seluruh feature dan observasi yang akan dianalisis. Dalam analisis sentimen, *feature* data target dapat berupa teks dan polaritas sentimen terkait dari teks tersebut, sedangkan observasi data target adalah seluruh data teks yang akan dianalisis.

#### 3.2.2 Preprocessing

Tahap *preprocessing* dilakukan agar data target yang telah diperoleh siap untuk dianalisis. Hal ini dilakukan untuk memperbaiki representasi data teks dan kualitasnya dalam analisis sentimen.

#### 3.2.3 Transformation

Tahap transformasi dilakukan untuk mengubah target data menjadi bentuk yang dapat diproses oleh model algoritma *machine learning*. Data teks yang sudah mengalami proses *preprocessing* akan diubah menjadi representasi vektor agar dapat dilatih menggunakan model algoritma machine learning.

#### 3.2.4 Data Mining

Tahap data mining bertujuan untuk melatih dan menguji target data yang sudah ditransformasi menggunakan model algoritma *machine learning*, sehingga menghasilkan *output pattern* berupa performa akurasi. Performa akurasi mengukur seberapa baik model algoritma yang digunakan untuk memprediksi sentimen yang telah diberikan kepada setiap komentar.

### 3.2.5 Interpretasi dan Evaluasi

Tahap interpretasi dan evaluasi bertujuan untuk mengukur seberapa baik performa model algoritma yang digunakan dalam memprediksi klasifikasi sentimen dengan benar. Output yang dihasilkan sebagai pengukuran evaluasi adalah sebagai berikut.

1. *Confusion matrix*
2. Akurasi
3. *Recall*
4. Presisi
5. Skor F1



## BAB IV PERANCANGAN DAN IMPLEMENTASI

### 4.1 Analisis

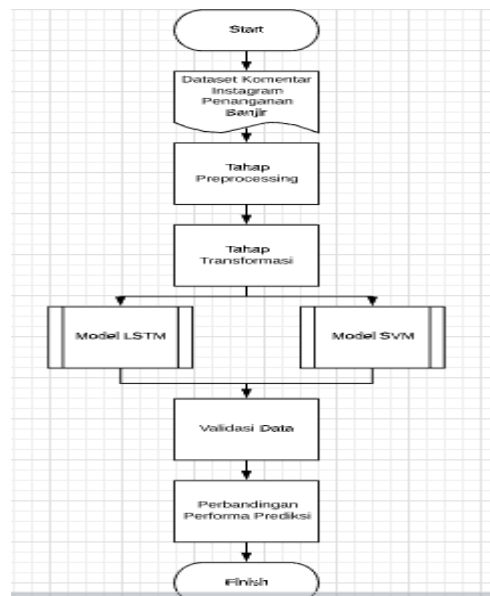
Penelitian yang dilakukan adalah menganalisis perbedaan metode *SVM* dan *RNN* tipe *LSTM* dengan menggunakan dataset komentar instagram tentang penanganan banjir. Tujuan penelitian ini adalah untuk mengetahui perbedaan antara kedua metode tersebut dalam hal performa dan proses kerja. Hasil penelitian ini adalah menentukan performa prediksi dalam analisis sentimen dari kedua metode yang digunakan. Performa prediksi dianalisis dan diuji berdasarkan penggunaan *hyperparameter* yang berbeda dari kedua metode. Hal tersebut diukur berdasarkan nilai *metric* yang dihasilkan.

Dalam penelitian yang dilakukan, dataset komentar yang digunakan memiliki 1532 observasi dan 2 *feature*, yaitu teks komentar dan polaritas sentimen. Teks komentar tersebut sudah dilakukan tahap *preprocessing* sehingga data teks siap untuk digunakan sebagai data latih dan data uji ke dalam model *machine learning*. Studi kasus analisis sentimen yang dilakukan adalah klasifikasi biner, yaitu klasifikasi sentimen yang dipisahkan menjadi dua bagian. Bagian tersebut adalah sentimen positif dan sentimen negatif yang dilambangkan oleh nilai 1 untuk positif dan 0 untuk negatif di *feature* polaritas sentimen. Klasifikasi dilakukan dengan cara metode *supervised machine learning* dimana polaritas sentimen sudah diberikan kepada masing-masing observasi di dataset.

### 4.2 Perancangan

#### 4.2.1 Design Sistem

Perancangan design sistem yang digunakan adalah menggunakan diagram *flowchart*. Gambar 5 menunjukkan *flowchart* program analisis sentimen yang akan dikembangkan.



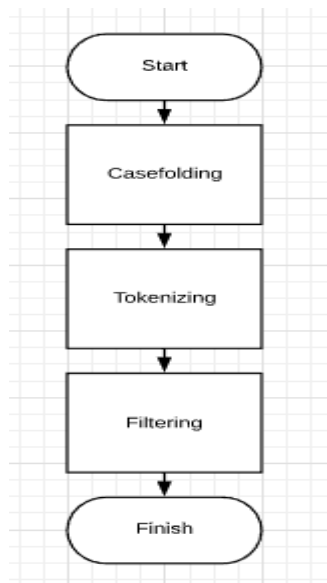
**Gambar 5** *Flowchart* Program

#### 4.2.2 Tahap *Preprocessing*

Tahap *preprocessing* dilakukan kepada seluruh observasi yang ada di dataset. Hasil akhir dari tahap *preprocessing* merupakan input teks yang akan dilakukan tahapan transformasi menjadi output vektor. Tabel 2 menunjukkan sampel dataset komentar instagram penanganan banjir yang belum dibersihkan. Sedangkan Gambar 6 menunjukkan *flowchart* tahap *preprocessing*.

**Tabel 2** Sampel Dataset Komentar Instagram Penanganan Banjir

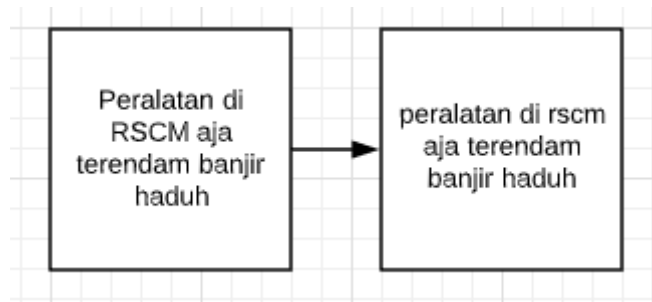
Data ke -	Teks
1	Memang mantap pak gubernur yg satu ini ❤️
2	Parah, mdh2n yg lolos cpns 2019 tdk kya gini jln pikiran dan omonganya kedepan. Parahhhhh
3	Management air yang gagal
4	Jurusan baru nih .....manajemen air, yang akuntansi air ada gak ya . Wkwkwkwk
5	Apa hubungannya bambang, dipikir kita avatar
6	Peralatan di RSCM aja terendam banjir haduh



**Gambar 6** *Flowchart* Tahap *Preprocessing*

##### A. *Case Folding*

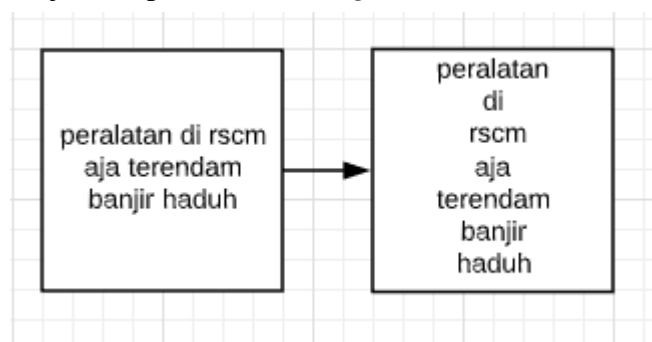
*Case Folding* adalah proses yang mengubah seluruh huruf besar menjadi huruf kecil. Gambar 7 menunjukkan contoh proses *case folding*.



**Gambar 7** Proses *Case Folding*

#### B. Tokenizing

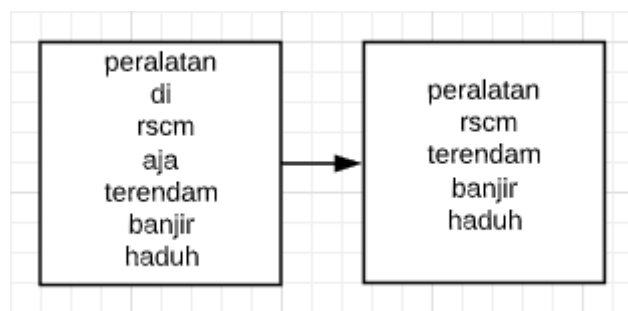
*Tokenizing* adalah proses memilah kata-kata yang ada di dokumen menjadi token. Gambar 8 menunjukkan proses *tokenizing*.



**Gambar 8** Proses *Tokenizing*

#### C. Filtering

*Filtering* adalah proses penyaringan kata-kata yang ada di dokumen. Analisis yang dilakukan menggunakan algoritma *stopword* sehingga kata-kata yang tidak penting akan dihilangkan. Algoritma *stopword* juga menghilangkan tanda baca, angka, dan url. Gambar 9 menunjukkan proses *filtering*.



**Gambar 9** Proses *Filtering*

### 4.2.3 Labeling Polaritas Sentimen

Pelabelan polaritas sentimen dilakukan kepada seluruh observasi di *dataset* menurut makna teks komentar. Makna yang terdapat di *dataset* komentar yang diperoleh merupakan pendapat dan pandangan masyarakat terhadap penanganan banjir yang dilakukan oleh sekda DKI. Polaritas sentimen positif yang diberikan mengarah

kepada lelucon atau ungkapan semangat kepada sekda DKI, sedangkan polaritas sentimen negatif yang diberikan mengarah kepada sindiran, ungkapan tidak setuju, dan ungkapan negatif. Pemakai bahasa terkadang menggunakan berbagai ungkapan untuk mengekspresikan kemarahan, kekesalan, kekecewaan, ketidaksenangan atau bahkan kebencian terhadap suatu hal atau kejadian yang menimpanya. Ungkapan tersebut sering disebut atau dikategorikan makian (Zulfa, 2017). Tabel 3 menunjukkan labeling polaritas sentimen kepada *dataset* komentar instagram penanganan banjir.

**Tabel 3** Tabel Polaritas Sentimen

Data ke -	Sentiment
1	0
2	0
3	0
4	1
5	1
6	0

#### 4.2.4 Tahap Transformasi

Tahap transformasi dilakukan untuk mengubah target data menjadi bentuk yang dapat diproses oleh model algoritma *machine learning*. Hal ini dilakukan dengan menggunakan *word embedding*, yaitu metode *TF-IDF* untuk *SVM* dan metode *Tokenizer* dari tensorflow keras untuk *RNN* tipe *LSTM*. *Word embedding* berguna untuk mengubah *feature* yang diperoleh dari input teks komentar menjadi representasi bentuk vektor. Analisis yang dilakukan adalah klasifikasi biner, sehingga output *feature* yang dihasilkan akan menentukan apakah kata-kata tersebut merupakan sentimen positif atau negatif.

*TF-IDF* adalah metode untuk mengetahui kata penting yang muncul dalam dokumen. Nilai *TF-IDF* dapat diketahui dengan menghitung *term frequency* dan *inverse-document frequency* dari kata yang akan dicari. Semakin mendekati nilai 1 maka nilai *TF-IDF* dari kata tersebut menyatakan bahwa kata tersebut penting dan merupakan *feature* yang baik untuk analisis. Berikut adalah contoh studi kasus pada metode *TF-IDF* terhadap *dataset* komentar yang digunakan pada tabel 4 dimana A adalah indeks lokasi komentar di *dataset* dan B adalah lokasi kata di kamus. Komentar 'sekda idiot' berada di baris ketiga, kemudian berdasarkan kamus yang berisi seluruh kata unik di *dataset* ditunjukkan bahwa kata 'sekda' berada di baris 2549 dan kata 'idiot' berada di baris 1122. Nilai *TF-IDF* yang dihasilkan menunjukkan kata 'idiot' merupakan *feature* yang baik dikarenakan kata tersebut memiliki nilai *TF-IDF* yang besar dan memiliki arti kasar. Oleh karena itu, keberadaan kata tersebut di suatu komentar akan selalu menghasilkan sentimen negatif. Contoh kata kasar lainnya adalah kata 'anjing' yang akan selalu menghasilkan sentimen negatif, terkandung di empat komentar, sedangkan kata 'idiot' hanya terkandung di satu komentar. Hal ini ditunjukkan pada tabel 5.

**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

*Tokenizer* adalah metode yang memproses teks dengan mengubahnya ke dalam bentuk vektor. Input yang diterima adalah *corpus* seluruh teks dan outputnya adalah vektor *feature* yang mempresentasikan kata-kata di dalam *corpus*. Input yang diterima dipecah menjadi *token* dan dilakukan *pad sequence* agar seluruh input memiliki ukuran yang sama.

#### 4.2.5 Penerapan Model Algoritma *RNN* Tipe *LSTM* dan *SVM*

Input teks tersebut dikonversi menjadi *sequence* yang akan diproses menggunakan *one hot encoding*. *One hot encoding* merepresentasikan angka menjadi nilai vektor dimana semua elemen bernilai 0 kecuali hanya satu elemen yang bernilai 1 yaitu posisi elemen yang sama dengan kata pada kamus. Contoh perhitungan menggunakan komentar *dataset* instagram 'sekda idiot' yang bernilai negatif atau 0 di dataset komentar instagram yang sudah dilakukan proses *preprocessing*.

**Tabel 6** Sampel Data *One Hot Encoding*

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

Bobot awal dan nilai *weight* diinisialisasikan secara random dengan jangkauan dari (-0,477, 0,477). Jangkauan ini menentukan nilai *weight* dan bias yang akan dihasilkan di gerbang *cell*, *forget*, input, dan output.

$$W = \left(-\frac{1}{\sqrt{h}}, \frac{1}{\sqrt{h}}\right)$$

$$W = \left(-\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)$$

$$W = (-0,477, 0,477)$$

$$X_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$W_f = \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} \quad W_c = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix} \quad B_f = [0,2] \quad B_c = [0,2]$$

$$W_i = \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix} \quad W_o = \begin{bmatrix} 0,4 \\ 0,3 \end{bmatrix} \quad B_i = [0,2] \quad B_o = [0,3]$$

Dalam pembelajaran yang dilakukan, setiap input *one hot encoding* akan diproses secara sekuensial menurut *time step* yang sedang berlangsung. Untuk *time step* pertama, nilai *hidden state* bernilai 0. Langkah pertama di perhitungan *LSTM* terdapat di gerbang *forget*, yaitu memutuskan informasi apa yang akan dihapus dari *cell state*. Gerbang *forget* akan memproses  $s_{t-1}$  dan  $x_t$  sebagai input serta menggunakan fungsi aktivasi *sigmoid* agar output yang dihasilkan bernilai antara 0 sampai 1. Nilai weight di gerbang lupa diperoleh berdasarkan persamaan 8

$$f_t = \sigma \left( \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0,2] \right)$$

$$f_t = \left( \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0,2] \right)$$

$$f_t = \sigma \left( \begin{bmatrix} 0,4 \\ 0,2 \end{bmatrix} \right)$$

$$f_t = \begin{bmatrix} 0,5987 \\ 0,5498 \end{bmatrix}$$

Langkah kedua di perhitungan *LSTM* terdapat di gerbang input, yaitu memutuskan untuk nilai mana yang akan diperbaharui dan menentukan nilai regularisasi  $\hat{C}_t$  yang akan ditambahkan ke *cell state*. Gerbang input akan memproses  $s_{t-1}$  dan  $x_t$  sebagai input serta menggunakan fungsi aktivasi *sigmoid* agar output yang dihasilkan bernilai antara 0 sampai 1 dan menggunakan fungsi aktivasi *tanh* agar output yang dihasilkan antara -1 sampai 1. Nilai weight di gerbang input diperoleh berdasarkan persamaan 9.

$$i_t = \left( \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0,2] \right)$$

$$i_t = \left( \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0,2] \right)$$

$$i_t = \left( \begin{bmatrix} 0,3 \\ 0,2 \end{bmatrix} \right)$$

$$i_t = \begin{bmatrix} 0,5744 \\ 0,5498 \end{bmatrix}$$

$$\hat{C}_t = \tanh(W_c \cdot [s_{t-1}, x_t] + b_c)$$

$$\hat{C}_t = \tanh \left( \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix} \cdot [0, \begin{bmatrix} 1 \\ 0 \end{bmatrix}] + [0,2] \right)$$

$$\hat{C}_t = \tanh \left( \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0,2] \right)$$

$$\hat{C}_t = \tanh \left( \begin{bmatrix} 0,3 \\ 0,2 \end{bmatrix} \right)$$

$$\hat{C}_t = \begin{bmatrix} 0,2913 \\ 0,1974 \end{bmatrix}$$

Langkah ketiga di perhitungan *LSTM* adalah memperbarui *cell state* dengan cara menghapus informasi sebelumnya. Hal ini dilakukan dengan cara mengkalikan nilai di gerbang *forget* dengan nilai *cell state* di *time step* sebelumnya, kemudian menjumlahkannya dengan hasil kali antara nilai di gerbang input dengan nilai regularisasi di *cell state*. Nilai weight di *cell state* diperoleh berdasarkan persamaan 11

$$C_t = \begin{bmatrix} 0,5987 \\ 0,5498 \end{bmatrix} * 0 + \begin{bmatrix} 0,5744 \\ 0,5498 \end{bmatrix} * \begin{bmatrix} 0,2913 \\ 0,1974 \end{bmatrix}$$

$$C_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0,1673 \\ 0,1085 \end{bmatrix}$$

$$C_t = \begin{bmatrix} 0,1673 \\ 0,1085 \end{bmatrix}$$

Langkah keempat di perhitungan *LSTM* terdapat di gerbang output, yaitu menentukan nilai di gerbang output dan nilai *hidden state* yang akan digunakan untuk *time step* selanjutnya. Gerbang output menggunakan fungsi aktivasi *sigmoid* agar output yang dihasilkan bernilai antara 0 sampai 1. Nilai *hidden state* dihasilkan dengan cara mengkalikan nilai di gerbang output dengan nilai di *cell state* yang telah diregularisasi. Nilai weight di gerbang output diperoleh berdasarkan persamaan 12.

$$o_t = \left( \begin{bmatrix} 0,4 \\ 0,3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0,3] \right)$$

$$o_t = \left( \begin{bmatrix} 0,4 \\ 0,3 \end{bmatrix} \cdot 0 + \begin{bmatrix} 0,4 \\ 0,3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [0,3] \right)$$

$$o_t = \begin{bmatrix} 0,7 \\ 0,3 \end{bmatrix}$$

$$o_t = \begin{bmatrix} 0,6682 \\ 0,5744 \end{bmatrix}$$

$$s_t = o_t * \tanh(C_t)$$

$$s_t = \begin{bmatrix} 0,6682 \\ 0,5744 \end{bmatrix} * \tanh \left( \begin{bmatrix} 0,1673 \\ 0,1085 \end{bmatrix} \right)$$

$$s_t = \begin{bmatrix} 0,6682 \\ 0,5744 \end{bmatrix} * \begin{bmatrix} 0,1658 \\ 0,1081 \end{bmatrix}$$

$$s_t = \begin{bmatrix} 0,1101 \\ 0,0621 \end{bmatrix}$$

Perhitungan dilanjutkan hingga seluruh input terproses, dalam contoh observasi yang digunakan terdapat dua input kata. Dalam *time step* selanjutnya, nilai *cell state* dan *hidden state* yang telah diperoleh akan digunakan lagi. Tabel 7 menunjukkan hasil perhitungan *LSTM*. Nilai *hidden state* terakhir digunakan untuk memprediksi polaritas sentimen. Nilai vektor output dengan probabilitas tertinggi akan terpilih sebagai hasil prediksi. Perhitungan tersebut menggunakan fungsi aktivasi *softmax*.

**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	$\begin{bmatrix} 0,5553 \\ 0,5760 \end{bmatrix}$	$\begin{bmatrix} 0,5526 \\ 0,5760 \end{bmatrix}$	$\begin{bmatrix} 0,2079 \\ 0,3905 \end{bmatrix}$	$\begin{bmatrix} 0,2078 \\ 0,2874 \end{bmatrix}$	$\begin{bmatrix} 0,5852 \\ 0,6499 \end{bmatrix}$	$\begin{bmatrix} 0,1199 \\ 0,1818 \end{bmatrix}$

$$y_t = \text{softmax} \left( \begin{bmatrix} 0,6682 \\ 0,5744 \end{bmatrix} \right)$$

$$y(0,6682) = \left( \frac{e^{0,6682}}{e^{0,6682} + e^{0,5744}} \right) = 0,5234$$

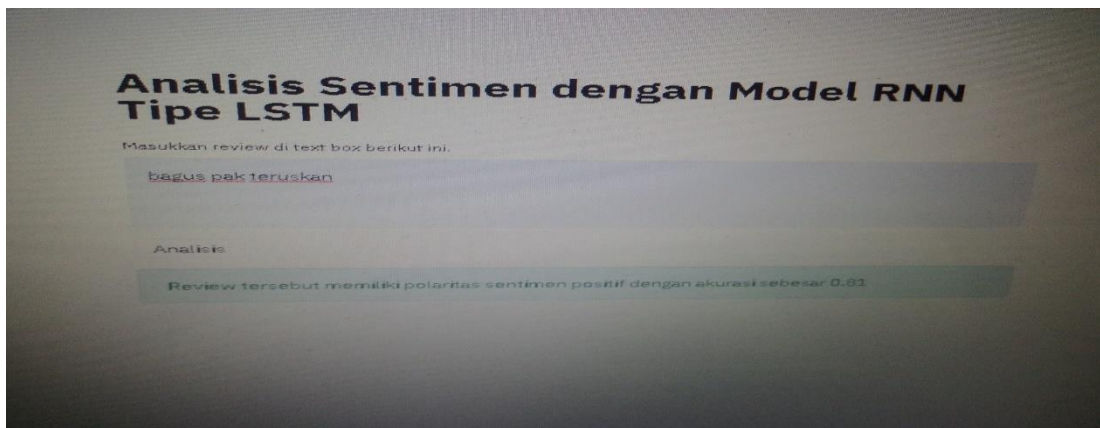
$$y(0,5744) = \left( \frac{e^{0,5744}}{e^{0,6682} + e^{0,5744}} \right) = 0,4766$$

### 4.3 Implementasi

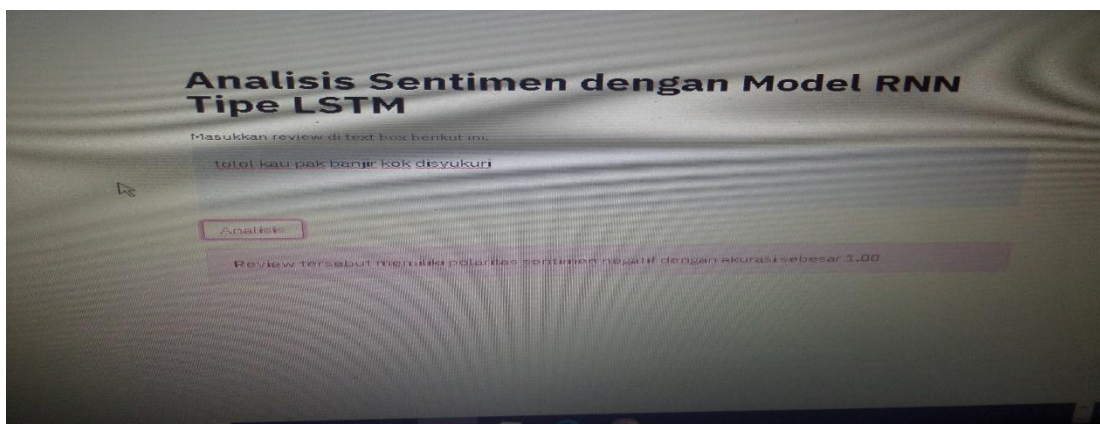
Implementasi analisis sentimen dilakukan menggunakan streamlit. Streamlit sebagai *framework* aplikasi *machine learning* bahasa python digunakan agar model yang dilatih dapat dijalankan menjadi aplikasi web. Gambar 10 menunjukkan *source code* implementasi analisis sentimen menggunakan streamlit sedangkan gambar 11 dan 12 menunjukkan implementasi analisis sentimen menggunakan streamlit.

```
soso.py
1 import streamlit as st
2 import pickle
3 from tensorflow.keras.models import load_model
4 from tensorflow.keras.preprocessing.text import Tokenizer
5 from tensorflow.keras.preprocessing.sequence import pad_sequences
6 def predict(message):
7     model=load_model('movie_sent.h5')
8     with open('tokenizer.pickle', 'rb') as handle:
9         tokenizer = pickle.load(handle)
10        x_1 = tokenizer.texts_to_sequences([message])
11        x_1 = pad_sequences(x_1, maxlen=500)
12        predictions = model.predict(x_1)[0][0]
13        return predictions
14 st.title("Analisis Sentimen dengan Model RNN Tipe LSTM")
15 message = st.text_area("Masukkan review di text box berikut ini.")
16 if st.button("Analisis"):
17     with st.spinner("Menganalisis teks review"):
18         prediction=predict(message)
19         if prediction > 0.6:
20             st.success("Review tersebut memiliki polaritas sentimen positif dengan akurasi sebesar {:.2f} ".format(prediction))
21             st.balloons()
22         else:
23             st.error("Review tersebut memiliki polaritas sentimen negatif dengan akurasi sebesar {:.2f} ".format(1-prediction))
24
```

**Gambar 10** Source Code Implementasi Analisis Sentimen Menggunakan Streamlit



**Gambar 11** Implementasi Analisis Sentimen Menggunakan Streamlit 1



**Gambar 12** Implementasi Analisis Sentimen Menggunakan Streamlit 2



## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Hasil

##### 5.1.1 Penggunaan *Library* Python

Sebelum memulai penelitian analisis sentimen, modul *library* python yang diperlukan selama penelitian harus dilakukan *import* terlebih dahulu. Hal ini penting, terutama modul *tensorflow keras* sebagai *library deep learning* berupa model RNN tipe LSTM. Program dijalankan menggunakan Jupyter Notebook yang menggunakan *virtual environment*. Gambar 13 menunjukkan pemanggilan *library* modul python.

```
import numpy as np
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential, load_model
from keras.layers import Dense, Embedding, LSTM, Bidirectional, Flatten, Dropout
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split
import os
import collections
import re
import string

Using TensorFlow backend.
```

Gambar 13 Pemanggilan *Library* Modul Python

##### 5.1.2 Proses *Preprocessing*

*Dataset* komentar instagram tentang penanganan banjir yang digunakan dalam penelitian dibersihkan terlebih dahulu. Proses *preprocessing* yang dilakukan adalah mengubah seluruh huruf kapital menjadi huruf kecil, menghilangkan simbol dan angka, mengubah seluruh kata slang menjadi bentuk umumnya, dan menghilangkan kata-kata tidak penting. Proses *processing* dilakukan dengan bahasa R di RStudio. Gambar 14 menunjukkan proses *processing dataset* komentar instagram.

```
tweets=gsub("\n", " ", tweets)
tweets=tweets %>% replace_html() %>% replace_url()
tweets=tweets %>% replace_emoji(.) %>% replace_html(.)
tweets=tweets %>% replace_tag(tweets, pattern="@([A-Za-z0-9_]+)", replacement="") %>%
  replace_hash(tweets, pattern="#([A-Za-z0-9_]+)", replacement="")
spell.lex=read.csv("colloquial-indonesian-lexicon.csv")
tweets=replace_internet_slang(tweets, slang=paste0("\b",
  spell.lex$slang, "\b"), replacement=spell.lex$formal, ignore.case=TRUE)
tweets=strip(tweets)
tweets= tweets %>% as.data.frame() %>% distinct()
nrow(tweets)
tweets=as.character(tweets$.)
stemming=function(x){paste(apply(x,katadasar),collapse=" ")}

```

Gambar 14 *Processing Dataset* Komentar Instagram.

### 5.1.3 Struktur LSTM

Dalam struktur model algoritma *RNN* tipe *LSTM* diamati perubahan konfigurasi jumlah *neuron* pada setiap layer yang berbeda ataupun besar nilai *hyperparameter* agar diketahui dampaknya kepada performa prediksi. Layer-layer yang diamati adalah layer *LSTM*, layer *Dense*, dan layer *Dropout*. Untuk proses kompilasi yang dilakukan model pada layer output digunakan fungsi aktivasi *sigmoid*, fungsi loss *Binary Cross-Entropy*, optimasi tipe *adam*, dan *metrics accuracy*. *Binary Cross-Entropy Loss* adalah teknik yang mengukur seberapa jauh performa prediksi yang dilakukan model dalam nilai 0 dan 1. Optimasi tipe *adam* adalah algoritma optimasi dimana *learning rate* yang dilakukan model *deep learning* berubah secara adaptif sehingga setiap parameter memiliki *learning rate* individual. *Metrics accuracy* digunakan sebagai fungsi yang bertanggung jawab sebagai performa model, dimana pada penelitian yang dilakukan adalah akurasi prediksi model. Gambar 15 menunjukkan struktur model *RNN* tipe *LSTM* yang digunakan saat melakukan penelitian. Gambar tersebut menunjukkan salah satu observasi penelitian yang dilakukan menggunakan 25 *neuron* pada 1 layer *LSTM*, nilai layer *Dropout* sebesar 0,5, dan 1 *neuron* pada layer *Dense* sebagai layer output.

```
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
```

**Gambar 15** Struktur Model RNN Tipe LSTM

### 5.1.4 Proses Validasi

Proses validasi data dalam penelitian yang dilakukan menggunakan *K-Fold Cross Validation* dengan jumlah *fold* sebanyak 5 dan 10 *epoch* dalam setiap fold. Gambar 16 menunjukkan hasil validasi menggunakan *K-Fold Cross Validation*. Gambar tersebut menunjukkan bahwa rata-rata akurasi yang dihasilkan oleh seluruh *fold* adalah sebesar 75,21%.

```

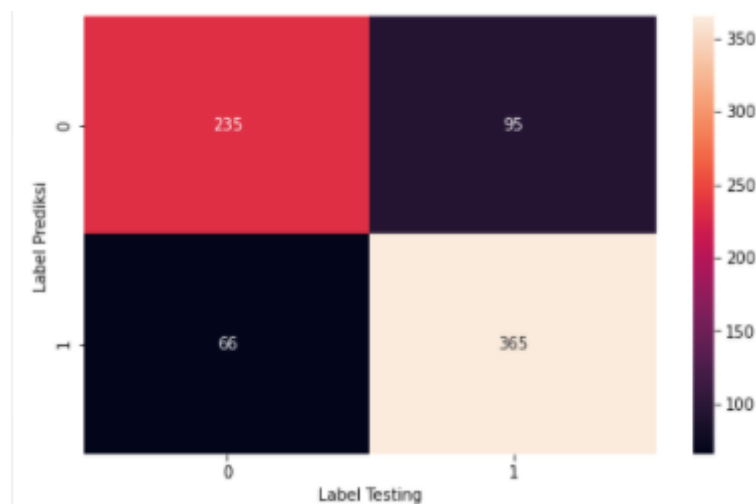
-----
Score per fold
-----
> Fold 1 - Loss: 0.7882636196509923 - Accuracy: 75.55847764015198%
-----
> Fold 2 - Loss: 0.7025580568319865 - Accuracy: 75.29566287994385%
-----
> Fold 3 - Loss: 0.6940582425146316 - Accuracy: 75.55847764015198%
-----
> Fold 4 - Loss: 0.6720041238682029 - Accuracy: 73.98160099983215%
-----
> Fold 5 - Loss: 0.642555657965125 - Accuracy: 75.6898820400238%
-----
Average scores for all folds:
> Accuracy: 75.21682024002075 (+- 0.6307503084348237)
> Loss: 0.6998879401661876
-----

```

**Gambar 16** Hasil Validasi Menggunakan *K-Fold Cross Validation*

### 5.1.5 Proses Hasil Pengujian Model

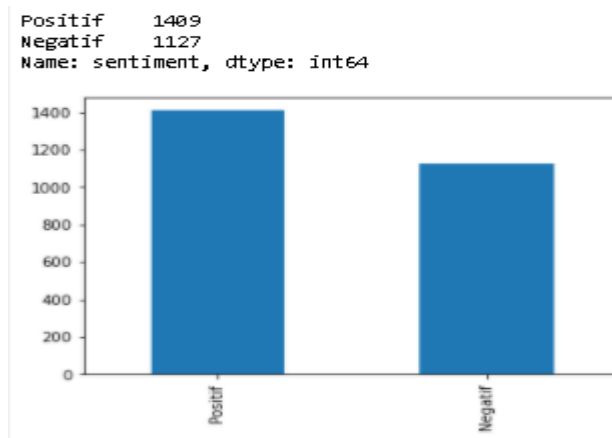
Proses hasil pengujian model yang dibuat menggunakan *confusion matrix*. *Confusion matrix* digunakan untuk mengetahui seberapa dapat dipercaya model dalam memprediksi kelas sentimen. *Confusion matrix* membandingkan data uji untuk mencari hasil dari *metric* yang dimiliki model. Gambar 17 menunjukkan proses hasil pengujian model.



**Gambar 17** Proses Hasil Pengujian Model.

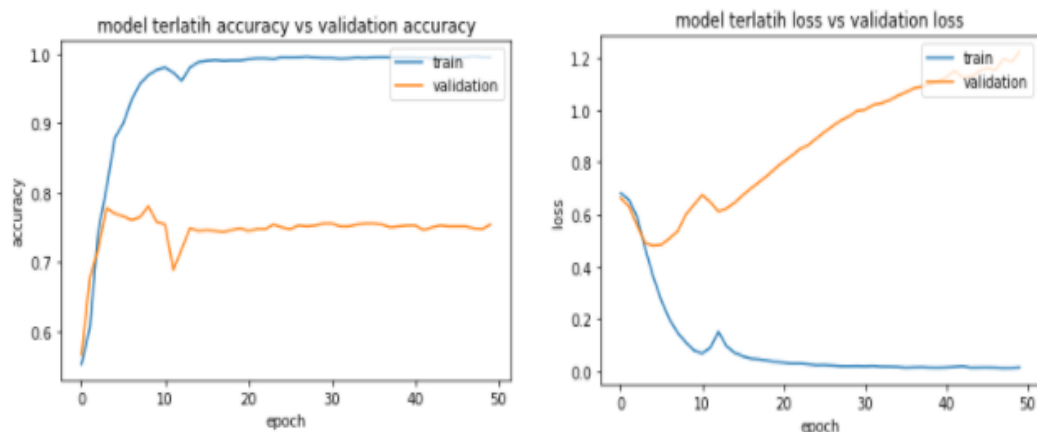
## 5.2 Pembahasan

Dalam penelitian yang dilakukan, dataset komentar instagram tentang penanganan banjir yang digunakan memiliki 1127 komentar negatif dan 1409 komentar positif. Hal ini ditunjukkan pada gambar 18. Sumbu X pada grafik batang tersebut menunjukkan jenis komentar instagram yang digunakan sebagai *feature*, yakni komentar negatif bernilai 0 dan komentar positif bernilai 1. Sedangkan sumbu Y menunjukkan jumlah komentar masing-masing polaritas sentimen yang ada di *dataset*. Dari seluruh komentar yang ada, 70% keseluruhan data digunakan sebagai data latih yaitu sebanyak 1775 dan 30% sisanya sebanyak 761 sebagai data *testing*.



**Gambar 18** Grafik Jumlah Sentimen

Percobaan yang dilakukan menggunakan 50 *epoch*, dimana *hyperparameter* yang digunakan adalah 1 layer *LSTM* dengan 25 *neuron*, layer *Dropout* sebesar 0,5, dan menggunakan ukuran *batch* sebesar 32. Dalam percobaan tersebut dihasilkan akurasi validasi tertinggi sebesar 78,84%. Gambar 19 menunjukkan nilai akurasi dan *loss training* dan validasi.



**Gambar 19** Nilai Akurasi dan *Loss Training* dan Validasi

### 5.2.1 Evaluasi Data Uji

Proses evaluasi data uji yang dilakukan menggunakan 458 data komentar. Evaluasi dilakukan menggunakan *confusion matrix*. Pada tabel 8 ditunjukkan bahwa terdapat 81 komentar positif dan 249 komentar negatif yang diprediksi secara benar. Namun terdapat 53 komentar negatif yang diprediksi merupakan komentar positif dan juga 75 komentar positif yang diprediksi merupakan komentar negatif.

**Tabel 8** Tabel *Confusion Matrix*

Data Test	Prediksi	
	Positif	Negatif
Positif	81	75
Negatif	53	249

Model tersebut kemudian dibandingkan dengan algoritma SVM dari besar metric yang dihasilkan. Perbandingan yang diamati adalah hasil *presisi*, *recall*, dan skor F1. Gambar 20 menunjukkan metric model *LSTM* dan gambar 21 menunjukkan *metric* model SVM.

```
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
```

**Gambar 20** *Metric Model LSTM*

```
import numpy as np
#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentiment'])
#lin_svm.fit(test_vectors, data_test['sentiment'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print( "train accuracy= ", np.mean(y_train_hat == data_train['sentiment'])*100)
print( "test accuracy= ", np.mean(y_test_hat == data_test['sentiment'])*100)
```

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

**Gambar 21** *Metric Model SVM*

### 5.2.2 Hasil Pengujian dengan Ahli Bahasa

Data validasi komentar instagram yang digunakan kemudian dibandingkan oleh ahli pakar bahasa Indonesia bernama Cicih Ratnasih, S.Pd.SD. Beliau memberikan polaritas sentimen kepada seluruh data validasi yang digunakan pada saat proses evaluasi. Dalam perbandingan yang dilakukan terdapat 144 komentar positif dan 65 komentar negatif yang diprediksi secara benar. Namun terdapat 238 komentar negatif yang diprediksi merupakan komentar positif dan juga 10 komentar positif yang diprediksi merupakan komentar negatif. Hasil perbandingan memiliki akurasi yang

kecil, yaitu hanya sebesar 45,73%. Perbandingan ini memiliki nilai *false positive* yang besar dikarenakan beliau menganggap bahwa meskipun komentar bersifat negatif namun ia memilih positif jika bertema penanganan banjir ataupun sesuai dengan permasalahan banjir. Hal ini berhubungan dengan sarkasme dan ambiguitas yang terkandung dalam teks komentar yang tersedia, serta pemahaman yang berbeda mengenai topik yang berada di teks. Gambar 22 menunjukkan *confusion matrix* hasil perbandingan yang dilakukan oleh ahli pakar kepada data validasi yang digunakan.

```
Confusion Matrix :
[[ 65 238]
 [ 10 144]]
Accuracy Score : 0.4573304157549234
Report :
```

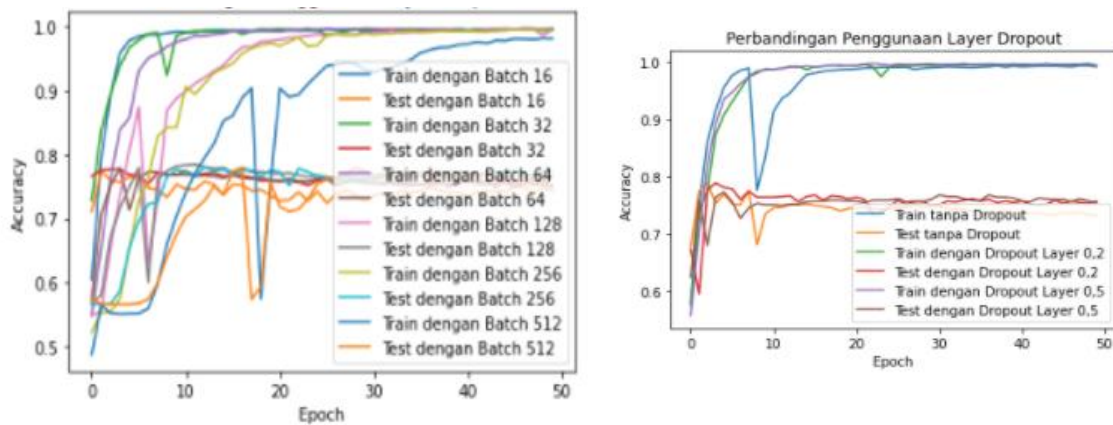
	precision	recall	f1-score	support
0	0.87	0.21	0.34	303
1	0.38	0.94	0.54	154
micro avg	0.46	0.46	0.46	457
macro avg	0.62	0.57	0.44	457
weighted avg	0.70	0.46	0.41	457

**Gambar 22** *Confusion Matrix* Hasil Perbandingan Ahli Pakar

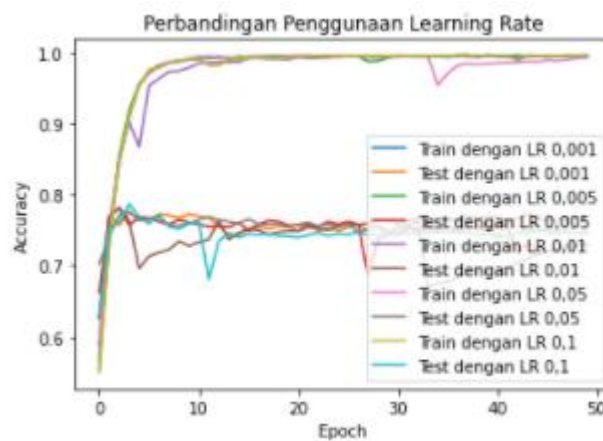
Pada penelitian yang dilakukan, terdapat beberapa kendala yang dihadapi, salah satunya adalah data yang digunakan tidak seimbang dan juga polaritas sentimen yang diberikan kepada masing-masing komentar belum akurat. Hal ini mempengaruhi performa model yang dilatih dari segi *loss* dan akurasi. Pada percobaan yang telah dilakukan dapat diamati bahwa model memiliki performa baik dalam fase *training* namun memiliki performa yang buruk dan bahkan menurun dalam fase validasi. Model yang dilatih memiliki akurasi validasi yang *stagnan* dan *loss* validasi yang selalu menaik secara eksponensial. Tabel 9 menunjukkan akurasi validasi di titik global maksimum yang dihasilkan oleh berbagai konfigurasi *hyperparameter*. Penelitian yang dilakukan adalah mengamati pengaruh jumlah layer, jumlah *neuron* pada layer LSTM yang digunakan, dan besar *learning rate* yang digunakan. Hasil penelitian yang diperoleh adalah dapat diketahui bahwa menggunakan layer yang banyak, jumlah *neuron* yang besar, dan *learning rate* yang terlalu besar tidak menjamin bahwa model algoritma yang dilatih akan memiliki performa yang baik. Dibuktikan pada percobaan 2 layer LSTM dengan masing-masing 200 *neuron* hanya menghasilkan akurasi validasi 66,10% di titik gradien *global maximum*. Pada penelitian yang dilakukan tidak dicantumkan perbedaan besar nilai layer *Dropout* karena efek yang terdampak di model algoritma yang dilatih tidak cukup signifikan. Perbedaan besar ukuran batch juga tidak dicantumkan dikarenakan ia hanya mempengaruhi performa yang dimiliki model algoritma pada *epoch* awal. Tabel 10 menunjukkan akurasi validasi di titik global maksimum yang dihasilkan oleh jumlah data *testing* yang berbeda.

**Tabel 9** Tabel Akurasi Validasi di Titik *Global Maximum*

Jumlah Layer	Jumlah Neuron	Besar <i>Learning Rate</i>		
		0,000001	0,001	1
2	200	57,82%	66,10%	56,64%
2	100	57,16%	75,56%	56,64%
2	50	57,69%	75,82%	56,64%
2	25	57,29%	77,40%	56,64%
1	200	60,05%	75,03%	60,32%
1	100	58,74%	75,30%	60,71%
1	50	61,37%	77,00%	55,72%
1	25	61,37%	78,84%	58,87%



**Gambar 23** Perbandingan Perbedaan Ukuran Batch dan Layer Dropout



**Gambar 24** Perbandingan Penggunaan Learning Rate Berbeda

**Tabel 10** Tabel Akurasi Validasi di Titik *Global Maximum 2*

Persentase Data Testing	Jumlah Sampel Training	Jumlah Sampel Testing	Akurasi Validasi
10%	2282	254	80,31%
20%	2028	508	78,15%
30%	1775	761	78,84%
40%	1521	1015	77,33%
50%	1268	1268	75,78%



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa model algoritma *RNN* tipe *LSTM* memiliki performa yang lebih baik daripada model algoritma *SVM*. 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%. Performa yang dimiliki model *LSTM* juga lebih baik pada fase *training*. Model *LSTM* mampu mencapai 99%, sedangkan model *SVM* hanya dapat mencapai 93%.

Penelitian dilakukan menggunakan metode penelitian *Knowledge Discovery and Data Mining* dimana hasil akhir yang diperoleh adalah performa prediksi klasifikasi biner. Analisis sentimen yang dilakukan menggunakan *dataset* komentar instagram tentang penanganan banjir yang diolah dan diproses menggunakan bahasa R. Kemudian data yang sudah diproses dilatih dan dievaluasi oleh model algoritma *machine learning* menggunakan *framework deep learning* yaitu tensorflow dan keras dalam bahasa python. Hasil evaluasi yang dilakukan oleh model algoritma yang dilatih tersimpan dan diluncurkan menggunakan *framework* aplikasi python yaitu streamlit. Hal ini dilakukan agar model yang dilatih dapat digunakan secara langsung lewat internet oleh pengguna umum.

#### 6.2 Saran

Penelitian yang dilakukan masih memiliki banyak kekurangan. Model yang dilatih terjadi *overfit* sehingga performa prediksi terhadap data baru yang tidak berasal dari data *training* akan menghasilkan prediksi yang kurang akurat. Hal ini dibuktikan dengan nilai akurasi yang meningkat dan nilai *loss* yang menurun pada fase *training*, namun nilai akurasi *stagnan* dan nilai *loss* semakin menaik pada fase validasi.

Pada penelitian ini terdapat beberapa hal yang dapat dikembangkan untuk penelitian selanjutnya agar menghasilkan performa prediksi yang lebih baik. Salah satunya adalah dengan melakukan interpolasi data sehingga *dataset* komentar yang digunakan menghasilkan gradien yang stabil. Hal lainnya adalah melakukan proses *stemming* dan *speech tagging* dalam tahap *preprocessing*. Dari segi implementasi, penelitian ini merupakan klasifikasi biner, sehingga penelitian klasifikasi *multi-class* dengan tiga jenis polaritas sentimen atau lebih dapat dilakukan sebagai perbandingan penelitian.

## DAFTAR PUSTAKA

- Ahmad, Munir. Shabib Aftab., dan Iftikhar Ali. 2017. Sentiment Analysis of Tweets using SVM. *International Journal of Computer Applications* Vol. 177 No. 5: 25-29.
- Aldi, Muhammad Wildan Putra. Jondri., dan Annisa Aditsania. 2018. Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *Proceeding Engineering* Vol.5 No.2: 3548-3555.
- Colace, Francesco. Massimo De Santo., dan Luca Greco. 2014. SAFE: A Sentiment Analysis Framework for E-Learning. *Jurnal IJET* Vol. 9 No. 6: 37-41.
- Faadilah A. 2020. Analisis Sentimen pada Ulasan Aplikasi Tokopedia di Google Play Store Menggunakan Metode Long Short Term Memory. Skripsi. Jurusan Matematika, Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta.
- Gunawan, Billy. Helen Sasty Pratiwi., dan Enda Esyudha Pratama. 2018. Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naïve Bayes. *Jurnal Edukasi dan Penelitian Informatika* Vol. 4 No. 2: 113-118.
- Ivanedra, Kasyfi dan Metty Mustikasari. 2019. Implementasi Metode Recurrent Neural Netowrk pada Text Summarization dengan Teknik Abstraktif. *Jurnal Teknologi Informasi dan Ilmu Komputer* Vol. 6 No. 4: 377-382.
- Jianqiang, Zhao. Gui Xiaolin., dan Zhang Xuejun. 2018. Deep Convolution Neural Networks for Twitter Sentiment Analysis. *Jurnal IEEE* Vol. 6: 23253-23260.
- Lutfi, Anang Anggono. Adhistya Erna Permanasari., dan Silmi Fauziati. 2018. Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine. *Jurnal JISEBI* Vol. 4 No. 1: 57-64.
- Nurrohmat, Muh Amin dan Azhari SN. 2019. Sentiment Analysis of Novel Review using Long Short Term Memory Method. *Jurnal IJCCS* Vol. 13 No. 3: 209-218.
- Octaviani, Pusphita Anna. Yuciana Wilandari., dan Dwi Ispriyanti. 2014. Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. *Jurnal Gaussian* Vol. 3 No. 4: 811-820.
- Patel, Alpna dan Arvind Kumar Tiwari. 2019. Sentiment Analysis by using Recurrent Neural Network. *Proceeding 2nd International Conference on Advanced Computing and Software Engineering*.
- Rashid, Tarik A. Polla Fattah, dan Delan K. Awla. 2018. Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms. *Jurnal Procedia Computer Science* 140: 324-333.
- Thomas, Merin dan Latha C.A. 2014. Sentiment Analysis using Recurrent Neural Network. *Jurnal IJET* Vol. 7 No. 2,27: 88-92.
- Zulfa, Ira dan Edi Winarko. 2017. Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Belief Network. *Indonesian Journal of Computing and Cybernetics Systems* Vol. 11 No. 2: 187-198.


# **LAMPIRAN**

## Lampiran 1. Data Sentimen Ahli Bahasa

### 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	luarbiasaaaaaaa	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

## Lampiran 2. Data Hasil Prediksi Model

No.	Text	Label	Prediksi	Hasil
1	sekolah dimana sih	Negatif	Positif	Tidak Sesuai
2	bapa ikan	Negatif	Positif	Tidak Sesuai
3	makanan kali ah nikmati	Positif	Positif	Sesuai
4	hubungan nya banjir tubuh	Positif	Positif	Sesuai
5	ta e	Negatif	Negatif	Sesuai
6	iq jongkok ya murahan	Negatif	Positif	Tidak Sesuai
7	iya manajemen air buruk	Negatif	Positif	Tidak Sesuai
8	ya manajemen airnya dihilangkan	Positif	Positif	Sesuai
9	barang barang rusak kerja sekolah mash disuruh santuyyy	Negatif	Positif	Tidak Sesuai
10	dirasain kali manis manisnya	Positif	Positif	Sesuai
11	manusia berkualitas	Negatif	Positif	Tidak Sesuai
12	t a i	Negatif	Negatif	Sesuai
13	mantap lanjutkan	Positif	Positif	Sesuai
14	konteksnya dikaitkan komposisi tubuh haduuhh	Positif	Positif	Sesuai
15	luarbiasaaaaaaa	Positif	Positif	Sesuai
16	alasan nya bacot akun bodong teriak kuping orang otak dangkal realistik sok cerdas forward pesan wise grup wa keluarga kantor organisasi kalo lo pintar mengapa lo bacot doang pakai akun asli malu lo nyamber sipit betina	Negatif	Negatif	Sesuai
17	malu ya mengomong	Negatif	Positif	Tidak Sesuai
18	koruptor	Negatif	Positif	Tidak Sesuai
19	goreng	Positif	Negatif	Tidak Sesuai
20	hidup air	Negatif	Positif	Tidak Sesuai
21	bodoh jaman ahok banjir minimalisir berkurang titik banjir gubernur parah jaman canggih anggran apbd bertambah sayang gubernur bodoh rakyat menderita	Negatif	Negatif	Sesuai
22	bijak komentar wawancara kebanyakan ngelem jdinya gue tenggelem kepalanya dibanjir seprti	Negatif	Positif	Tidak Sesuai
23	kemana banjir parah program jalankan prokasih dll diadukan kena banjir kena banjir aneh	Negatif	Positif	Tidak Sesuai
24	peliharaan gubernur dilepas awas galak goblok	Negatif	Negatif	Sesuai
25	dinikmati mundur enteng mulut disuruh menikmati banjir	Negatif	Negatif	Sesuai

26	dad jokes gone wrong	Negatif	Negatif	Sesuai
27	kasih kesempatan surutnya hemmm suruh menikmati tuh ikan duyung warga jakarta semoga lekas surut kesehatan warga jakarta	Negatif	Positif	Tidak Sesuai
28	sekda terbaik	Positif	Positif	Sesuai
29	cocok twwa nikmati banjir jekardah nya	Positif	Negatif	Tidak Sesuai
30	dasar badut dki	Negatif	Negatif	Sesuai
31	agam	Negatif	Positif	Tidak Sesuai
32	kalo jaman ahok deh tamat tuh pejabat ngebacot	Negatif	Positif	Tidak Sesuai
33	komen pro kontra menyolot	Negatif	Positif	Tidak Sesuai
34	menteri nya pinokio bicaranya ngawur banyakpura tau ya bong	Negatif	Negatif	Sesuai
35	wis podo edane podo kocloke podo pengunge	Negatif	Positif	Tidak Sesuai
36	president disyukurin enak korupsi viralnya angetnya teralihkan harun masiku bersyukur masak indak	Negatif	Positif	Tidak Sesuai
37	mantaabbbb ilmunan	Negatif	Positif	Tidak Sesuai
38	manajemen sumberdaya perairan	Positif	Positif	Sesuai
39	ditunggu alasan ngeles ajaib nya butuh hiburan	Negatif	Positif	Tidak Sesuai
40	otak nya isi isi	Negatif	Positif	Tidak Sesuai
41	tenggelamkan	Positif	Negatif	Tidak Sesuai
42	manajemen air maksdnyafucku	Negatif	Positif	Tidak Sesuai
43	tag warga kelapa gading disuruh menikmati cid ci	Negatif	Negatif	Sesuai
44	pajak bayar orang orang ya	Negatif	Negatif	Sesuai
45	mulut sang sekda dimanagement	Negatif	Positif	Tidak Sesuai
46	syuoggg	Negatif	Positif	Tidak Sesuai
47	saudara meninggal banjir nikmati nikmati biji matamu meletup evaluasi ente blok	Positif	Positif	Sesuai
48	suruh ahok pejabat dinegaranya china	Negatif	Positif	Tidak Sesuai
49	avatar keleus management air kalo bilang dinikmati lu anies becus nanganin banjir	Negatif	Negatif	Sesuai
50	tenang ojo ngegass	Negatif	Positif	Tidak Sesuai

### Lampiran 3 Tahap Seleksi dan Preprocessing

Dataset komentar instagram yang digunakan berasal dari instagram Anies Baswedan dengan link <https://www.instagram.com/aniesbaswedan/> . Instagram tentang penanganan banjir diperoleh menggunakan aplikasi *instagram comment scraper* yang dapat diakses di link <https://github.com/AgiMaulana/Instagram-Comments-Scraper> . Output web scaping yang dihasilkan adalah dokumen dengan format .csv. Dokumen tersebut dilakukan tahap preprocessing menggunakan RStudio. Gambar di bawah menunjukkan dataset komentar instagram sebelum dan sesudah dilakukan tahap *preprocessing*. Tahap preprocessing yang dilakukan adalah menghilangkan emoji, simbol, lambang, tanda baca, url, dan mengubah seluruh kata menjadi huruf kecil.

text	x1
1 Ayo doakan pak Anis dan jajarannya, yg berkerja denga...	ayo doakan pak anis dan jajarannya yg berkerja dengan ...
2 Nico devandri jaya	
3 Harusnya kembali kemasing2 induvidu. Bukan gubernur...	alhamdulillah semoga jakarta bebas banjir dan bpk cepe...
4 Pak Anies tolong keluarkan PerGub kalau buang sampa...	alhamdulillah smoga kami yg di kemayoran tdk kena ba...
5 Alhamdulillah pak anies, 🍌🍌❤️	sampe kapn psbb pak
6 Semangat pak!!! Semoga lekas pulihh❤️ Negara ini butu...	kami warga jakarta yg merasakan kerja bapak selalu men...
7 semoga lekas sembuh dan beraktivitas kembali pak gub...	ok boss banyak sholat malam bossancamanbanyak dari ...
8 Sehat2 Bpk2....	semangat dan sehat selalu pak anis salam dari medan
9 Terpaksa bener2 Terpaksa beberapa Pohon di Tebang.....	apakahbenar akan di bangun waduk di cipinang melayu...
10 @wayanayung @putubayus	psbb jngan diperpanjang lgi pak bngung dirumahkan tr...
11 Sehat terus ya pak barokalloh semoga Alloh selalu melin...	banyak cerita ttg mal fatih hampir putus asa menaklukka...
12 🤔🤔🤔	wahh langgagganan sya pulau mas rawamgun kl hujan ...
13 إشك الحمد لله	iya gitu kerjain rakyat tahu kok mana hasil kerja mana hs...

## Lampiran 4 Tahap Transformasi

Dataset komentar instagram kemudian diubah menjadi vektor agar dapat diproses oleh model algoritma *machine learning*. *Pad sequence* dilakukan agar seluruh kata memiliki ukuran yang sama, jumlah yang digunakan adalah sebesar 500. Gambar di bawah menunjukkan komentar yang akan dilatih dan dilakukan *testing*. X adalah teks komentar dan Y adalah polaritas sentimen. Gambar di bawah adalah contoh komentar ‘sekda idiot’ yang telah diubah menjadi bentuk vektor.

```
x_train
array([[ 0,  0,  0, ..., 617, 278, 18],
       [ 0,  0,  0, ...,  0,  0, 188],
       [ 0,  0,  0, ..., 4016, 728, 1608],
       ...,
       [ 0,  0,  0, ..., 18,  2, 1053],
       [ 0,  0,  0, ..., 2970, 11, 2971],
       [ 0,  0,  0, ...,  0, 16, 239]])

y_train
array([1, 1, 0, ..., 1, 0, 0])

x_test
array([[ 0,  0,  0, ...,  0,  0, 2105],
       [ 0,  0,  0, ..., 28, 344, 169],
       [ 0,  0,  0, ...,  0, 4829, 87],
       ...,
       [ 0,  0,  0, ...,  2, 1623, 566],
       [ 0,  0,  0, ..., 1468, 323,  3],
       [ 0,  0,  0, ...,  5, 199, 4]])

y_test[:5]
array([1, 1, 1, 0, 1])
```



## Lampiran 5 Tahap Data Mining

Model algoritma deep learning yang digunakan adalah RNN tipe LSTM. Gambar di bawah menunjukkan struktur model algoritma RNN tipe LSTM yang digunakan. *Embedding layer* sebagai *layer* input memiliki dimensi output sebesar 50 dan ukuran input yang digunakan adalah 500. 1 *Layer LSTM* digunakan dengan 25 neuron dan *layer dropout* yang digunakan adalah sebesar 50%. *Layer* output memiliki menggunakan fungsi aktivasi *sigmoid* dan model ini menggunakan *optimizer* tipe *adam* dan fungsi *loss binary cross entropy*. Gambar berikutnya menunjukkan perbandingan perubahan nilai *weight* pada *layer* input, *layer LSTM*, dan *layer* output sebelum dan sesudah dilakukan pelatihan.

```
model = Sequential()
model.add(Embedding(vocab_size, 50, input_length=500))
#model.add(LSTM(128, return_sequences=True))
model.add(LSTM(25))
#model.add(LSTM(25, return_sequences=False))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.summary()
```

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		

```
model.layers[0].get_weights()[0] #embedding Layer
```

```
array([[ -0.00667881,  0.0060389 , -0.00618096, ..., -0.03659378,
         0.03374885,  0.0249132 ],
       [ 0.04907945,  0.02305199,  0.00699021, ..., -0.02326611,
        -0.02138522, -0.01486439],
       [ 0.03419683, -0.04393194, -0.03779797, ...,  0.03725043,
        -0.00868268,  0.01333943],
       ...,
       [-0.00842016, -0.00979613,  0.01434166, ..., -0.03152382,
        -0.00129258,  0.00033991],
       [-0.02263725,  0.00121761,  0.00315341, ...,  0.00417855,
        -0.02265002,  0.04954734],
       [ 0.00424999,  0.00274809, -0.04052993, ..., -0.02204845,
         0.04062733,  0.04293218]], dtype=float32)
```

```
model.layers[0].get_weights()[0] #embedding Layer setelah dilakukan pelatihan
```

```
array([[ -0.01663748, -0.00226307, -0.0101633 , ..., -0.03285335,
         0.02270668,  0.01842277],
       [ 0.04097559, -0.01280921,  0.00836992, ..., -0.12473793,
        -0.01846766, -0.06629265],
       [ 0.04671336, -0.04834731, -0.04704616, ...,  0.01431771,
        -0.03782011, -0.01017635],
       ...,
       [-0.00842016, -0.00979613,  0.01434166, ..., -0.03152382,
        -0.00129258,  0.00033991],
       [-0.02263725,  0.00121761,  0.00315341, ...,  0.00417855,
        -0.02265002,  0.04954734],
       [ 0.00424999,  0.00274809, -0.04052993, ..., -0.02204845,
         0.04062733,  0.04293218]], dtype=float32)
```

```
model.layers[1].get_weights()[1][:1] #Lstm Layer
```

```
array([[ 0.2014539 , -0.0555664 , -0.04489231, -0.03168269, -0.07396004,
        -0.05362966,  0.09017994, -0.13079324,  0.08588401,  0.03594792,
        -0.0191135 ,  0.1772574 ,  0.03455441, -0.22193721, -0.11679181,
        -0.09946696, -0.10873184, -0.05731406,  0.18291552,  0.2562521 ,
        -0.02517279, -0.01644193, -0.14494847, -0.1413549 ,  0.05462877,
         0.13195619,  0.06950865, -0.14721037, -0.0051882 ,  0.11798716,
         0.03284625,  0.02036392,  0.03308409,  0.00681077,  0.05626589,
        -0.08621456, -0.04122331, -0.00955835, -0.0958387 ,  0.00849803,
         0.07979112, -0.01772661,  0.03376376, -0.02281306, -0.07928587,
        -0.09341914,  0.0444522 ,  0.08512637,  0.00284381, -0.21302995,
        -0.20203617, -0.0256984 , -0.02056118, -0.20384642, -0.03725058,
         0.15239322,  0.00079836, -0.02944683, -0.09746803, -0.07456066,
         0.03464906,  0.06914802,  0.08611368,  0.18067944,  0.08671355,
        -0.02168464, -0.19669464,  0.14481576, -0.09258109,  0.07027893,
         0.11096929,  0.02469852,  0.07980117, -0.01211315,  0.22491786,
        -0.01444968,  0.11519668,  0.02388313, -0.03800178, -0.13268192,
        -0.1173096 ,  0.05050849, -0.11307062, -0.02848834, -0.05657708,
         0.07489671,  0.01233876,  0.03712245, -0.08248938,  0.1098187 ,
        -0.05152405, -0.0723884 ,  0.13067147, -0.06008657, -0.00059279,
         0.05015608, -0.06918631, -0.1122506 , -0.13084455, -0.08381535]],
      dtype=float32)
```

```
model.layers[1].get_weights()[1][:1] #Lstm Layer setelah pelatihan
```

```
array([[ 0.11996683, -0.05027745, -0.09610961, -0.12872604, -0.2513147 ,
        -0.08493607, -0.02464038, -0.26736856,  0.13886811,  0.05620999,
        -0.11818276,  0.10135693, -0.04459219, -0.26933253, -0.1787128 ,
        -0.18052918, -0.2355965 , -0.16611065,  0.08892497,  0.21093741,
        -0.03454354, -0.13073544, -0.27993858, -0.25409603, -0.03795198,
         0.11570705,  0.09420282, -0.18127477, -0.07773811,  0.1053028 ,
         0.03551818,  0.01907948, -0.0359938 ,  0.08152837,  0.10397405,
        -0.10490259, -0.08058972, -0.01460762, -0.10019637,  0.00380887,
         0.09067342, -0.05032586,  0.00141751, -0.02911337, -0.08213402,
        -0.09440634,  0.02047694,  0.04321177, -0.07547098, -0.2432588 ,
        -0.18698272, -0.024773 ,  0.00928442, -0.11356439, -0.00574839,
         0.17219588, -0.02170449,  0.08733696, -0.03612214, -0.06645945,
        -0.0760816 ,  0.16534598,  0.11807124,  0.2227116 ,  0.07975227,
        -0.00230249, -0.1712768 ,  0.01902525, -0.04175143,  0.04399374,
         0.22883765,  0.0615534 ,  0.19995582, -0.12283132,  0.27415365,
        -0.13272941,  0.10098962, -0.01280456, -0.1396868 , -0.2948271 ,
        -0.16334875,  0.02411721, -0.27336258,  0.00675621, -0.03823594,
        -0.07638792, -0.13699177, -0.05477246, -0.08069743,  0.1118181 ,
        -0.10787532, -0.19022343, -0.02068666, -0.15669997, -0.07667562,
        -0.01940657, -0.21029353, -0.2669364 , -0.28094232, -0.1915469 ]],
      dtype=float32)
```

```
model.layers[3].get_weights()[0] #Layer output
```

```
array([[ -0.02657959],
       [ -0.14117044],
       [ -0.12223363],
       [ -0.44898617],
       [ -0.17976141],
       [ -0.36204517],
       [  0.12541157],
       [ -0.44881827],
       [ -0.20225021],
       [  0.06883556],
       [  0.19045895],
       [ -0.08747381],
       [ -0.24498016],
       [ -0.08221871],
       [ -0.12900674],
       [ -0.24926172],
       [ -0.43057653],
       [  0.3188334 ],
       [ -0.23685132],
       [  0.27139354],
       [ -0.19642681],
       [ -0.29059196],
       [ -0.39250833],
       [  0.09351653],
       [ -0.30957502]], dtype=float32)
```

```
model.layers[3].get_weights()[0] #Layer output setelah pelatihan
```

```
array([[ -0.71782655],
       [ -0.6593245 ],
       [ -0.09857243],
       [ -0.7700062 ],
       [ -0.80658275],
       [ -0.92625237],
       [  0.5987508 ],
       [ -0.9775569 ],
       [  0.27963942],
       [  0.5816576 ],
       [  0.7678734 ],
       [ -0.61353993],
       [ -0.7862112 ],
       [  0.2492619 ],
       [ -0.445686 ],
       [ -0.8437419 ],
       [ -0.8735155 ],
       [  0.8249715 ],
       [ -0.8419405 ],
       [  0.8561869 ],
       [ -0.6810518 ],
       [ -0.9169638 ],
       [ -0.9227627 ],
       [  0.74612176],
       [ -0.9551249 ]], dtype=float32)
```

## Lampiran 6 Tahap Interpretasi dan Evaluasi

Model yang dilatih kemudian diuji menggunakan sampel data *testing*. Gambar di bawah adalah contoh prediksi yang dilakukan oleh model terhadap komentar yang berasal dari input *user*. Proses ini dilakukan dalam implementasi menggunakan aplikasi streamlit. Semakin mendekati nilai 1 maka hasil prediksi yang dihasilkan adalah komentar positif, sedangkan semakin mendekati nilai 0 maka hasil prediksi yang dihasilkan adalah komentar negatif.

```
string11="ter gg emang pak saya setuju banget kalau begitu"  
x_1=tokenizer.texts_to_sequences([string11])  
x_1 = pad_sequences(x_1,maxlen=500)  
model.predict(x_1)
```

```
array([[0.99984443]], dtype=float32)
```

```
string11="sialan kau pak"  
x_1=tokenizer.texts_to_sequences([string11])  
x_1 = pad_sequences(x_1,maxlen=500)  
model.predict(x_1)
```

```
array([[0.12112696]], dtype=float32)
```

```
string11="tolol banget pak komen dipikir dulu emangnya dikira enak ya banjir"  
x_1=tokenizer.texts_to_sequences([string11])  
x_1 = pad_sequences(x_1,maxlen=500)  
model.predict(x_1)
```

```
array([[0.23203415]], dtype=float32)
```

```
string11="bagus saya setuju pak ayo lanjutkan"  
x_1=tokenizer.texts_to_sequences([string11])  
x_1 = pad_sequences(x_1,maxlen=500)  
model.predict(x_1)
```

```
array([[0.99831855]], dtype=float32)
```