

# Analisis Sentimen pada Media Sosial Twitter Berbahasa Indonesia dengan Metode GloVe

Dicky Wahyu Hariyanto<sup>1</sup>, Warih Maharani, S.T., M.T.<sup>2</sup>

<sup>1,2</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>dickyhariyantowahyu@students.telkomuniversity.ac.id, <sup>2</sup>wmaharani@telkomuniversity.ac.id

---

## Abstrak

Seiring dengan perkembangan media sosial, warganet Indonesia ramai menggunakan media sosial untuk berbagi informasi. Salah satunya menggunakan fitur cuitan pada media sosial Twitter, untuk membahas suatu topik tertentu. Bencana merupakan salah satu topik yang ramai dibahas, mulai dari kenapa terjadinya bencana dan bagaimana penanganannya oleh pihak berwenang. Analisa sentimen dapat dilakukan untuk menganalisa cuitan dengan topik bencana ini, agar dapat digunakan sebagai tolak ukur bagaimana penanganan bencana dan kenapa bencana itu terjadi menurut pendapat warganet. Pada penelitian ini dibuat analisa sentimen menggunakan *word embedding Global Vector (GloVe)* yang bertujuan untuk meningkat performa analisa sentimen. Model *Global Vector* dibentuk dari korpus Wikipedia Indonesia, dengan dataset cuitan dengan topik bencana yang berjumlah 1500 data cuitan. Klasifikasi sentimen yang digunakan adalah metode *deep learning* model *Long Short-Term Memory (LSTM)*. Yang mana model *Global Vector* diembedd ke dalam *layernya*. Dalam penelitian ini akan dilakukan dua scenario pengujian dengan data cuitan dengan label data sentimen seimbang dan pengujian dengan label data sentimen tidak seimbang. Dari hasil pengujian dengan data seimbang didapatkan akurasi sebesar 73% dan pada data dengan label tidak seimbang didapatkan presisi 74,5% dan *recall* 74,5% dengan akurasi 75%.

**Kata kunci :** analisa sentimen, GloVe, LSTM, Twitter, word embedding

---

## Abstract

Along with the development of social media, Indonesian citizens are often using social media to share information. One of them uses twitter's social media tweeting feature, to discuss a particular topic. Disasters are one of the topics that are discussed, ranging from why disasters occur and how they are handled by the authorities. Sentiment analysis can be done to analyze tweets on the topic of this disaster, so that it can be used as a benchmark for how disaster management and why disasters occur in the opinion of citizens. In this study, sentiment analysis was made using the word embedding Global Vector (GloVe) which aims to improve the performance of sentiment analysis. The Global Vector model was formed from the Corpus Wikipedia Indonesia, with a dataset of tweets with disaster topics totaling 1500 tweet data. The sentiment classification used is the deep learning method of the Long Short-Term Memory (LSTM) model. Which is where the Global Vector model is dimbedded into its layers. In this study, two test scenarios were conducted with tweet data with balanced sentiment data labels and tests with disproportionate sentiment data labels. From the test results with balanced data obtained 73% accuracy and in data with unbalanced labels obtained 74.5% precision and 74.5% recall with 75% accuracy.

**Keywords:** GloVe, LSTM, sentiment analysis, word embedding

---

## 1. Pendahuluan

### Latar Belakang

Analisis sentimen pada media sosial saat ini menjadi suatu topik menarik untuk diteliti, hal ini dikarenakan perkembangan penggunaan media sosial yang pesat terutama di Indonesia[1]. Salah satunya adalah media sosial yang bernama Twitter. Dengan mengirimkan cuitan yang merupakan pesan singkat, pengguna Twitter dapat menyampaikan pandangannya terhadap suatu bahasan tertentu yang terjadi di sekitarnya. Cuitan ini dapat dianalisis sentimennya untuk berbagai kebutuhan, misalnya untuk melihat kepribadian seorang individu atau ketertarikan orang terhadap suatu hal dan banyak hal lainnya [2][3].

Bencana pada beberapa tahun terakhir banyak terjadi di mana-mana, seperti banjir, gempa bumi, gunung meletus. Topik tentang bencana ini tidak luput dari perhatian warganet untuk membahas atau membagikan pandangannya terhadap bencana yang terjadi di Indonesia. Warganet biasanya membicarakan banyak hal tentang topik bencana ini, mulai dari penyebab bencana dan bagaimana penanganan dari pihak yang berwenang dalam menangani bencana yang terjadi. Cuitan mengenai bencana ini dapat dianalisis sentimennya menggunakan sistem analisa sentimen dan bisa menjadi rekomendasi kepada pihak yang berwenang untuk mengambil tindakan atau melihat penilaian warganet terhadap kinerjanya.

Untuk membangun sistem analisa sentimen ini akan digunakan metode *word embedding*. *Word embedding* adalah metode yang berguna untuk merepresentasikan kata ke dalam bentuk vektor. Metode ini dapat meningkatkan performa analisa sentimen, oleh karena hal tersebut tersebut *word embedding* banyak digunakan pada penelitian yang membahas analisa sentimen. Dalam penelitian ini akan digunakan model *word embedding*

yang bernama *GloVe (Global Vector)*. Model *Global Vector* dipilih karena memiliki tingkat akurasi yang baik dibandingkan dengan model *word embedding* lainnya seperti *word2vec (Continuous Bag of Words & Skip-gram)* dan *doc2vec*[4].

*Word embedding* dan data cuitan kemudian akan dianalisis untuk diklasifikasikan sentimennya. Pada penelitian ini akan digunakan metode *deep learning* untuk mengklasifikasikan sentimen. Model yang akan digunakan adalah *Long Short-Term Memory (LSTM)*. *LSTM* ini memiliki beberapa layer yang salah satu layer-nya berfungsi untuk *word embedding* dan memiliki performa yang baik untuk mengklasifikasikan sentimen, jika digunakan dengan *word embedding* model *Global Vector* [4].

Pada penelitian ini, data yang digunakan adalah data cuitan yang bertemakan bencana dengan jumlah label data seimbang dan dengan label data sentimen tidak seimbang. Data dari Wikipedia Indonesia digunakan sebagai korpus, karena belum adanya model *pre-trained Global Vector* dalam Bahasa Indonesia. Data dari Wikipedia Indonesia akan dibentuk menjadi model *Global Vector* yang digunakan sebagai *word embedding*. Penjelasan lebih lengkap mengenai pelatihan ini akan dijelaskan pada bagian selanjutnya.

### Topik dan Batasannya

Topik dan batasan pada penelitian adalah menganalisa cuitan tentang bencana pada di media sosial Twitter berbahasa Indonesia. Jumlah data cuitan yang digunakan pada penelitian ini berjumlah 514, 1150 dan 1506 data cuitan, yang mana telah dilabeli dengan label positif dan negatif dengan jumlah yang sama. Selain itu, pada penelitian ini akan digunakan data dengan label yang tidak seimbang dengan total 1282 data cuitan. Dengan 751 data label positif dan 531 data label negatif. Model *word embedding* yang digunakan adalah model yang dibentuk dari kumpulan data Wikipedia Indonesia dengan 364998 kosakata, dikarenakan belum ada model *Global Vector* Bahasa Indonesia yang tersedia. Model *Global Vector* yang dibentuk menggunakan dimensi 50, 100, 150, 200, 250, 300 dan 350.

### Permasalahan

Bagaimana penerapan *word embedding Global Vector* dan algoritma *deep learning Long Short-Term Memory* untuk membentuk model analisa sentimen.

### Tujuan

Menerapkan dan menganalisa performa model analisa sentimen yang menggunakan *word embedding Global Vector* dan model *deep learning Long Short-Term Memory* untuk mengklasifikasikan sentimen cuitan bencana.

## 2. Tinjauan Pustaka

### 2.1 Studi Terkait

Banyak metode yang dapat digunakan dalam membentuk analisis sentimen, seperti yang dilakukan Imaduddin et. al yang mana menggunakan beberapa model *word embedding word2vec Continuous Bag of Word (CBOW)*, *word2vec Skip-gram* dan *doc2vec*. Dari hasil penelitian tersebut ditemukan bahwa *Global Vector* lebih baik dibandingkan dengan model *word embedding* lainnya[4].

*Global Vector* memiliki kelebihan akurasi yang lebih baik dibandingkan dengan model *word embedding word2vec Continuous Bag of Word (CBOW)* dan *word2vec Skip-gram* karena Pennington et.al dan Sharma et. al membentuk model *Global Vector* dari pendekatan *CBOW* dan *Skip-gram* dalam membentuk *word embedding*. Sehingga memiliki kelebihan dari sisi akurasi dan komputasi yang lebih cepat[5][6].

Pengimplementasian *Global Vector* sebagai *word embedding* untuk analisa sentimen pada penelitian yang dilakukan oleh Imaduddin et. al menggunakan *Long Short-Term Memory* untuk melakukan proses klasifikasi sentimen untuk ulasan hotel mendapatkan hasil yang paling baik dibandingkan dengan implementasi *word embedding* lainnya[4].

Selain itu alasannya dipilihnya *Long Short-Term Memory* untuk mengklasifikasikan sentimen adalah karena *Long Short-Term Memory* dapat mengatasi masalah *vanishin/exploding gradient* yang merupakan pengembangan dari model *Recurrent Neural Network (RNN)*, hal ini disampaikan oleh Li et. al pada penelitiannya yang menganalisa sentimen dari teks. Oleh karena dasar tersebut pada penelitian ini akan diterapkan model *word embedding Global Vector (GloVe)* dan *Long Short-Term Memory* untuk analisa sentimen cuitan bencana [7].

### 2.2 Tinjauan Pustaka

#### 2.2.1 Word2Vec

*Word2Vec* adalah metode untuk merepresentasikan kata ke dalam bentuk vektor atau *word embedding* yang diusulkan oleh Mikolov et. al yang memiliki dua model arsitektur [8]. Dua model arsitektur *Word2vec* yang diusulkan untuk membentuk representasi kata adalah *Continuous Bag-of-Words (CBOW)* dan *Skip-gram* model.

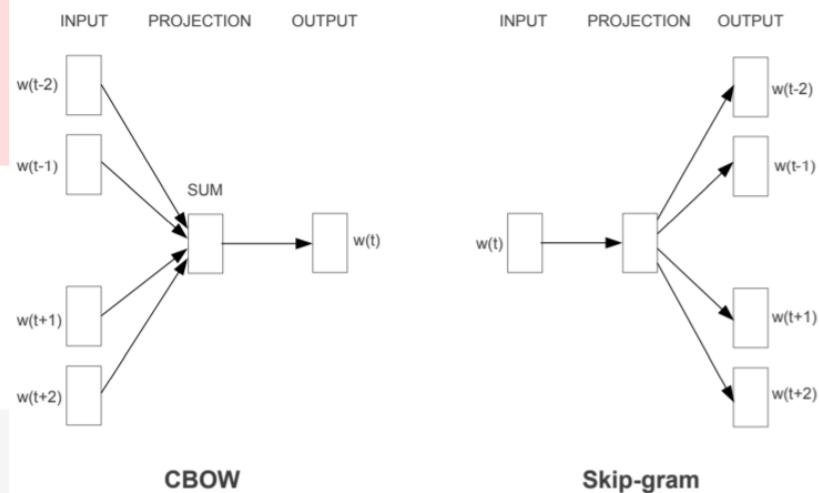
##### 2.2.1.1 Continuous Bag-of-Words (CBOW)

CBOW adalah arsitektur model dari *Word2vec* yang basisnya menggunakan model *Log-linear*. Arsitektur model CBOW bekerja dengan cara membagikan layer proyeksi pada semua kata, sehingga vektor pada setiap kata sama rata (diproyeksikan dalam posisi yang

sama). Untuk memprediksi kata, pada model CBOW kata diprediksi berdasarkan konteksnya.

### 2.2.1.2 Skip-gram

*Skip-gram* adalah model yang kedua yang diusulkan oleh Mikolov et. al pada metode *Word2vec*. Perbedaan mendasar dari model *Skip-gram* ini dibandingkan dengan *CBOW* adalah jika *CBOW* memprediksi kata atau mengklasifikasikannya berdasarkan konteksnya, sedangkan pada *Skip-gram* prediksi dilakukan dengan melihat kata lain yang ada di sekitar kata tersebut dalam kalimat yang sama [8]. Arsitektur dari *CBOW* dan *Skip-gram* dapat dilihat pada gambar berikut ini



<sup>1</sup>Gambar 1. 1 Arsitektur CBOW dan Skip-gram

### 2.2.2 Global Vector (GloVe)

*Global Vector* adalah sebuah metode yang diusulkan oleh para peneliti di Stanford University untuk membentuk representasi kata ke dalam bentuk vektor. Metode ini menggabungkan kelebihan dari model pendahulunya yaitu *Word2vec* yang menggunakan model *Continuous Bag-of-Words* dan *Skip-gram*, sehingga memiliki akurasi dan waktu komputasi yang baik. Hal ini sudah dibuktikan oleh para peneliti di Stanford University [5].

Dengan ukuran vektor dan korpora yang kecil, performa GloVe lebih baik dari model pendahulunya. *GloVe* merepresentasikan kata ke dalam vektor dengan membentuk matriks dari korpus. Yang setiap sel dari matriks nilai  $x$  menggambarkan berapa sering suatu kata muncul dalam konteks tertentu. Probabilitas kemunculan dalam konteks kata tertentu ini yang kemudian akan dinormalisasi dan dihitung hubungan probabilitas antar kata.

### 2.2.3 Recurrent Neural Network (RNN)

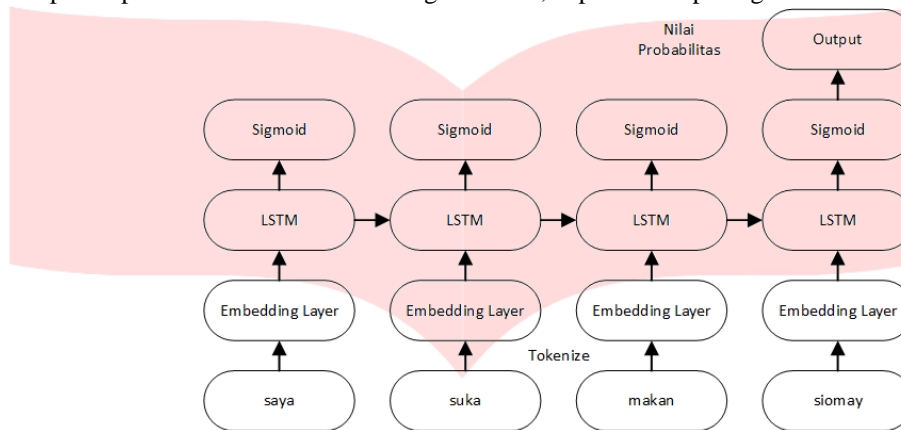
*RNN* merupakan metode *deep learning* dengan arsitektur *neural networks* yang memiliki kemampuan untuk merepresentasikan data yang berbentuk sekuensial. *RNN* memiliki kemampuan untuk menyimpan informasi pada keadaan sebelumnya untuk dapat mengetahui kemungkinan atau menghasilkan keluaran atas dasar keadaan sebelumnya.

Namun *RNN* memiliki kelemahan di mana saat terlalu banyak sekuens yang dikerjakan akan membuat *RNN* mengalami *vanishing gradient/exploding gradient*. *Vanishing/exploding gradient* adalah kondisi di mana nilai *gradient* bisa sangat kecil atau sama dengan nol dan bisa juga sangat besar[7]. Oleh karena hal ini dibangun metode yang dapat mengatasi kelemahan tersebut dengan penambahan sistem gerbang, yang disebut *LSTM (Long Short-Term Memory)*.

### 2.2.4 Long Short-Term Memory

<sup>1</sup> Sumber : Mikolov et. al "Efficient Estimation of Word Representations in Vector Space Tomas"

LSTM adalah pengembangan dari RNN untuk mengatasi masalah *vanishing/exploding gradient*. Pada arsitektur LSTM ditambahkan *gate* atau gerbang yang berfungsi untuk mengatur informasi apa yang harus diingat. Ada penambahan tiga gerbang, yang masing-masing berfungsi untuk gerbang input, gerbang untuk menghapus informasi sebelumnya dan gerbang keluaran. Dengan penambahan tiga gerbang ini LSTM dapat mengatur informasi yang disimpan dengan lebih baik sehingga tidak terjadi *vanishing/exploding gradient* [7]. Berikut adalah contoh dari model penerapan klasifikasi sentimen dengan LSTM, dapat dilihat pada gambar di bawah ini :



<sup>2</sup>Gambar 1. 2 Contoh Model LSTM

### 2.2.5 Pre-Processing

*Pre-processing* adalah proses yang harus dilakukan sebelum data dapat digunakan untuk dianalisis. Karena data berbentuk teks yang didapat dari Twitter biasanya mengandung banyak kesalahan seperti kata tidak terstruktur, kesalahan penulisan, karakter yang tidak diperlukan, singkatan, dan hal lain yang dapat membuat proses ekstraksi kata dan analisis sentimen tidak bisa memberikan performa yang baik [9].

Oleh karena hal tersebut perlu dilakukan langkah *pre-processing*, untuk meminimalisir atau menghilangkan kesalahan-kesalahan pada data. Sehingga saat data diproses akan menghasilkan hasil yang maksimal baik akurasi maupun proses klasifikasi. Ada beberapa skema yang biasa diterapkan untuk *pre-processing* pada Twitter [2], yaitu :

- *Case Folding*, mensekualiskan karakter, sehingga kata dalam kalimat berbentuk huruf kecil atau besar semua. Dalam penelitian ini huruf akan dirubah ke dalam bentuk huruf kecil.
- *Stop Word Removal*, menghilangkan kata yang memiliki pengaruh atau makna yang lemah dalam suatu kalimat, seperti kata “yang”.
- *Symbol Removal*, menghilangkan simbol-simbol yang tidak perlu seperti URL, @, # atau spasi berlebih yang banyak terpadat pada cuitan.
- *Tweet Tokenization*, pada tahap ini cuitan (tweet) yang berupa kalimat akan dipotong setiap katanya menjadi bentuk token.

### 2.2.6 Confusion Matrix

Salah satu cara untuk mengevaluasi hasil dari analisis sentimen adalah dengan menggunakan evaluasi matriks yang biasa disebut *confusion matrix*. Dalam *confusion matrix* ada beberapa hal yang dapat dievaluasi dari hasil klasifikasi analisis sentimen, yaitu akurasi, presisi, sensitivitas dan *f-score* yang merupakan gabungan dari dua evaluasi sensitivitas dan presisi [2].

Tabel 1. 1 Confusion Matrix

|         | Terprediksi Positif | Terprediksi Negatif |
|---------|---------------------|---------------------|
| Positif | TP                  | FN                  |
| Negatif | FP                  | TN                  |

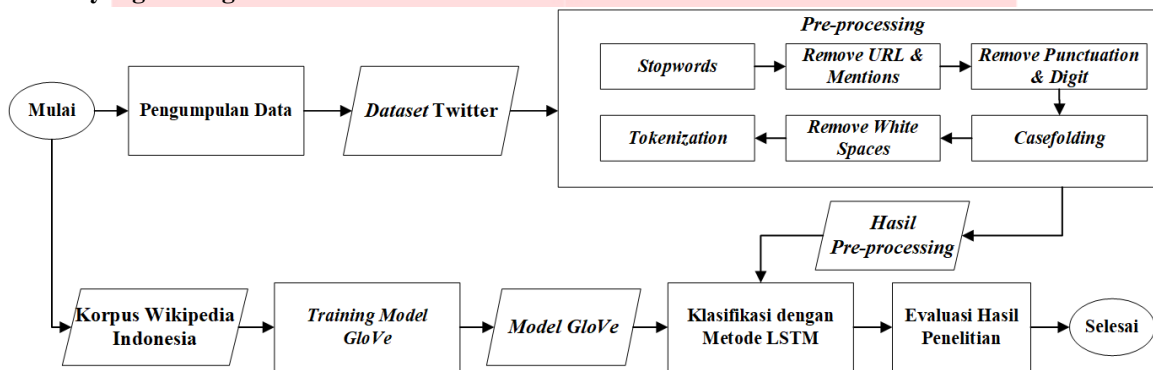
<sup>2</sup> Sumber : <https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step-50d074f09948>

Dapat dilihat pada Tabel 1.1 contoh tabel confusion matrix di mana hasil dari klasifikasi mempunyai nilai masing-masing yaitu TP (*True Positive*), FN (*False Negative*), FP (*False Positive*) dan TN (*True Negative*). Untuk menghitung nilai dari setiap evaluasinya dapat dimasukkan ke dalam persamaan sebagai berikut :

$$\bullet \text{ Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.1) \quad \bullet \text{ Sensitivitas} = \frac{TP}{TP+FN} \quad (2.3)$$

$$\bullet \text{ Presisi} = \frac{TP}{TP+FP} \quad (2.2) \quad \bullet \text{ F-Score} = 2 \cdot (\text{presisi} \cdot \text{sensitivitas}) / (\text{presisi} + \text{sensitivitas}) \quad (2.4)$$

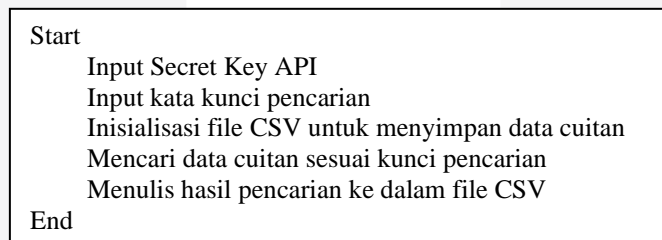
### 3. Sistem yang Dibangun



Gambar 1.3 Alur Pembangunan Sistem

#### 3.1 Pengumpulan Data

Proses pengumpulan data dilakukan dengan menggunakan bahasa pemrograman python versi 3 dengan meminta akses dari API yang disediakan oleh Twitter. Untuk mendapat ijin mengakses API Twitter, sebelumnya harus mendaftar dengan akun Twitter yang kita miliki pada halaman *Developers Twitter*. Setelah terdaftar, kita akan mendapat kunci dan token untuk mengakses API Twitter. Algoritma *crawling* di bawah ini berfungsi untuk mendapatkan data cuitan dari kata kunci tertentu, yang hasilnya akan dituliskan dalam file CSV berisikan daftar cuitan dari kata kunci yang dicari.



Hasil pencarian data cuitan dari algoritma di atas yang di simpan dalam file csv, contoh bentuk tweet dapat dilihat pada tabel 1.2 :

Tabel 1.2 Contoh Hasil Crawling

| Username  | Cuitan  |
|-----------|---|
| @infoBMKG | Kepala BMKG turun langsung ke Kulonprogo untuk memastikan peralatan monitoring cuaca dan sensor gempabumi dipasang<br><a href="https://t.co/W6tpzn5YeJ">https://t.co/W6tpzn5YeJ</a> |

#### 3.2 Pre-Processing

Hasil dari pengumpulan data tidak akan langsung digunakan, namun akan dilakukan pre-processing terlebih dahulu. Hal ini dikarenakan data yang dikumpulkan di Twitter masih banyak karakter-karakter yang tidak diperlukan dan kesalahan dalam penulisan, adanya singkatan dan hal lainnya yang bias

menurunkan performa akurasi dan klasifikasi sentiment. Berikut merupakan tahapan dari *pre-processing* yang akan dilakukan pada penelitian ini dapat dilihat pada Tabel 1.1 *Pre-Processing* di bawah ini :

**Tabel 1. 3 Langkah Pre-processing**

| Langkah Pre-Processing     | Hasil  |
|----------------------------|--|
| Cuitan Asli                | Kepala BMKG turun langsung ke Kulonprogo untuk memastikan peralatan monitoring cuaca dan sensor gempabumi dipasang <a href="https://t.co/W6tpzn5YeJ">https://t.co/W6tpzn5YeJ</a> |
| Stopwords                  | Kepala BMKG turun langsung ke Kulonprogo memastikan peralatan monitoring cuaca sensor gempabumi dipasang <a href="https://t.co/W6tpzn5YeJ">https://t.co/W6tpzn5YeJ</a>           |
| Remove URL & Mention       | Kepala BMKG turun langsung ke Kulonprogo memastikan peralatan monitoring cuaca sensor gempabumi dipasang   |
| Remove Punctuation & Digit | Kepala BMKG turun langsung ke Kulonprogo memastikan peralatan monitoring cuaca sensor gempabumi dipasang   |
| Casefolding                | kepala bmgk turun langsung ke kulonprogo memastikan peralatan monitoring cuaca sensor gempabumi dipasang   |
| Remove White Space         | kepala bmgk turun langsung ke kulonprogo memastikan peralatan monitoring cuaca sensor gempabumi dipasang   |
| Tokenization               | [kepala, bmgk, turun, langsung, ke, kulonprogo, memastikan, peralatan, monitoring, cuaca, sensor, gempabumi, dipasang]   |

### 3.3 Training Model GloVe

Pada tahap ini dilakukan training korpus Wikipedia Indonesia yang akan dibentuk menjadi model *Global Vector*. Dalam penelitian ini akan menggunakan beberapa dimensi vektor yaitu dimensi 50, 100, 150, 200, 250, 300 dan 350. Korpus Wikipedia dirubah direpresentasikan kedalam vektor dengan cara membentuk matriks kemunculan kata dalam suatu konteks tertentu, dapat dilihat dalam tabel berikut.

**Tabel 1. 4 Tabel Co-occurrence**

|        | Saya | Suka | Makan | Siomay | Tapi | Tidak | Pare |
|--------|------|------|-------|--------|------|-------|------|
| Saya   | 0    | 2    | 0     | 0      | 0    | 0     | 0    |
| Suka   | 2    | 0    | 1     | 1      | 0    | 1     | 1    |
| Makan  | 0    | 1    | 0     | 0      | 0    | 0     | 0    |
| Siomay | 0    | 1    | 0     | 0      | 0    | 0     | 0    |
| Tapi   | 0    | 0    | 0     | 0      | 0    | 1     | 0    |
| Tidak  | 0    | 1    | 0     | 0      | 0    | 1     | 0    |
| Pare   | 0    | 1    | 0     | 0      | 0    | 0     | 0    |

Yang kemudian dibuat perhitungan probabilitas kemunculan kata tadi ke dalam bentuk seperti pada persamaan (3.1) di bawah ini. Kemudian hasil yang didapat kemudian akan dihitung nilai *cost function* dengan menggunakan persamaan (3.2)

$$wTi+w^{\rightarrow} +bi+bk=log(XXik) \quad (3.1)$$

$$J=\sum_{i,k=1}^V f(Xik)(wTiw^{\rightarrow} j+bi+bk-logXik)2 \quad (3.2)$$

$$f(X_{i_k}) = \begin{cases} \left(\frac{X_{i_k}}{x_{max}}\right)^{\alpha} & ; \text{if } X_{i_k} < x_{max} \\ 1 & ; \text{lainnya} \end{cases} \quad (3.3)$$

Untuk pembobotan kata, yang dalam persamaan di atas dilambangkan dengan  $f(X_{i_k})$  digunakan persamaan (3.3) yang merupakan hasil dari tabel matriks di atas. Pada penelitian ini parameter *GloVe* seperti nilai  $X_{max}$  dan  $\alpha$  akan menggunakan sesuai dengan yang telah dilakukan oleh Pennington et. al dimana  $X_{max}=100$ ,  $\alpha=0.75$  dan iterasi yang disesuaikan dengan besarnya vektor, agar mendapatkan performansi yang baik. Setelah model terbentuk, model *GloVe* akan diembedd ke dalam layer LSTM untuk dilakukan klasifikasi sentimen.

### 3.4 Klasifikasi Sentimen dengan LSTM

Pada proses klasifikasi sentimen pada model LSTM kita perlu menentukan bentuk input dan besarnya. Pada penelitian ini akan menggunakan bentuk input *sequence*. Kemudian pada *layer* berikutnya model *word embedding* akan diembedd pada *layer embedding* LSTM. Dengan menyesuaikan besar vektor yang diembedd dan juga pada *layer* ini akan dilakukan ekstraksi fitur di mana setiap kata pada dataset akan dicari bobot vektornya yang kemudian akan diklasifikasikan negatif dan positif sentimennya. Pada *layer* LSTM pada penelitian ini menggunakan 64 unit LSTM dan *dropout* 0.5 dengan lapisan menggunakan aktivasi ReLU dengan 250 unit dan 1 unit aktivasi sigmoid setelahnya, yang dapat dilihat pada gambar 1.4 di bawah ini :

```

Penjelasan model.....
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
embedding (Embedding)       (None, 50, 50)             149100
-----
lstm (LSTM)                  (None, 64)                  29440
-----
dense (Dense)                (None, 250)                 16250
-----
dense_1 (Dense)              (None, 1)                   251
-----
Total params: 195,041
Trainable params: 195,041
Non-trainable params: 0

```

**Gambar 1. 4 Model LSTM**

Proses data diklasifikasikan dengan LSTM, dimulai dengan data yang telah *dipre-processing* dicari bentuk angka atau representasinya dalam bentuk vektor yang telah dibuat menggunakan *word embedding Global Vector*[10]. Kemudian masuk satu persatu secara berurutan dalam satu set ke LSTM, hasil dari setiap *hidden layer* akan disebarkan ke *hidden layer* lainnya bersama dengan masukan berikutnya. Setelah hasil akhir keluar, kemudian akan diteruskan ke *layer dense* untuk merubah hasil keluarannya sesuai dengan aktivasi ReLU dan sigmoid.

### 3.5 Evaluasi Hasil Klasifikasi

Setelah model mengklasifikasikan analisis sentimen, langkah selanjutnya adalah melakukan evaluasi dari hasil tersebut. Evaluasi menggunakan metode *Confusion Matrix* di mana nilai dari hasil klasifikasi dimasukkan ke dalam persamaan untuk menghitung tingkat akurasi yang mana seberapa akurat model dapat melakukan klasifikasi dengan benar (2.1). Kemudian tingkat presisi, seberapa menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model (2.2). *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi (2.3).

## 4. Evaluasi

Pada penelitian ini dilakukan pengujian menggunakan data dengan jumlah label negatif dan positif yang seimbang dengan jumlah total data 514, 1150 dan 1506 data cuitan. Juga dilakukan pengujian menggunakan data dengan jumlah label negatif dan positif yang tidak seimbang dengan jumlah label negatif 351 dan label positif 751 data cuitan. Model *Global Vector* yang digunakan dalam pengujian menggunakan dimensi 50, 100, 150, 200, 250, 300 dan 350 dengan menggunakan 25 *epoch* dan jumlah urutan 50. Untuk melihat keberhasilan dari pengujian, pada penelitian ini digunakan *Confusion Matrix* untuk melihat hasil dari pengujian berupa nilai akurasi, presisi, *recall* dan *f-score*.

#### 4.1 Analisis Sentimen Menggunakan Model *GloVe* dengan Jumlah Data Seimbang

Pada pengujian ini dilakukan pengujian dengan data cuitan yang memiliki label sentimen seimbang dengan dimensi vektor yang berbeda. Data cuitan dibagi menjadi data train dan tes, dengan presentase data tes sebanyak 20% dari setiap total jumlah data cuitan. Hasil pengujian dapat dilihat pada tabel 1.5 di bawah ini :

**Tabel 1. 5 Pengujian Data Seimbang**

| Jumlah Data | Dimensi <i>GloVe</i> | Akurasi    | Presisi    | <i>Recall</i> | <i>F-Score</i> |
|-------------|----------------------|------------|------------|---------------|----------------|
| 514         | 50                   | 61%        | 63%        | 61%           | 60%            |
| <b>514</b>  | <b>100</b>           | <b>67%</b> | <b>67%</b> | <b>67%</b>    | <b>67%</b>     |
| 514         | 150                  | 63%        | 63%        | 63%           | 62,5%          |
| 514         | 200                  | 60%        | 60%        | 60%           | 60%            |
| 514         | 250                  | 61%        | 62%        | 61,5%         | 61%            |
| 514         | 300                  | 64%        | 64,5%      | 64%           | 64%            |
| 514         | 350                  | 63%        | 63%        | 63%           | 63%            |
| 1150        | 50                   | 65%        | 65,5%      | 65,5%         | 65%            |
| 1150        | 100                  | 68%        | 69%        | 68%           | 68%            |
| 1150        | 150                  | 69%        | 69%        | 69%           | 69%            |
| 1150        | 200                  | 73%        | 73%        | 73%           | 72,5%          |
| 1150        | 250                  | 72%        | 72%        | 72%           | 72,5%          |
| <b>1150</b> | <b>300</b>           | <b>73%</b> | <b>73%</b> | <b>73%</b>    | <b>73%</b>     |
| 1150        | 350                  | 72%        | 72%        | 71,5%         | 72%            |
| 1506        | 50                   | 70%        | 70%        | 70%           | 70%            |
| 1506        | 100                  | 71%        | 71%        | 70,5%         | 70,5%          |
| 1506        | 150                  | 72%        | 71%        | 71%           | 71,5%          |
| 1506        | 200                  | 70%        | 70%        | 69,5%         | 69,5%          |
| 1506        | 250                  | 71%        | 70,5%      | 71%           | 71%            |
| 1506        | 300                  | 70%        | 70,5%      | 70,5%         | 70,5%          |
| <b>1506</b> | <b>350</b>           | <b>72%</b> | <b>72%</b> | <b>71%</b>    | <b>71%</b>     |

Berdasarkan tabel di atas, dapat dilihat tingkat performansi pada setiap jumlah data dan dimensi vektor yang digunakan hasilnya berbeda dan ditandai dengan angka *bold*. Untuk hasil terbaik pada pengujian ini, dihasilkan dari jumlah data cuitan 1000 dan dimensi vektor berjumlah 300 dengan tingkat akurasi 73%, presisi 73%, *recall* 73% dan *f-score* 73%.

#### 4.2 Analisis Sentimen Menggunakan Model *GloVe* dengan Jumlah Data Tidak Seimbang



Pengujian dengan jumlah sentimen negatif dan positif ini dilakukan untuk melihat bagaimana performansi yang dihasilkan. Data yang digunakan berjumlah 1282 data cuitan, dengan jumlah data positif sebanyak 751 cuitan. Sama dengan pengujian sebelumnya dimensi vektor yang digunakan yaitu 50, 100, 150, 200, 250, 300 dan 350. Hasil dari pengujian dapat dilihat pada tabel 1.6 di bawah ini :

**Tabel 1. 6 Tabel Pengujian Data Tidak Seimbang**

| <b>Dimensi GloVe</b> | <b>Akurasi</b> | <b>Presisi</b> | <b>Recall</b> | <b>F-Score</b> |
|----------------------|----------------|----------------|---------------|----------------|
| 50                   | 74%            | 73,5%          | 75%           | 73,5%          |
| 100                  | 73%            | 72%            | 70,5%         | 71%            |
| <b>150</b>           | <b>75%</b>     | <b>74,5%</b>   | <b>74,5%</b>  | <b>74,5%</b>   |
| 200                  | 74%            | 72,5%          | 71,5%         | 72%            |
| 250                  | 74%            | 73%            | 74%           | 73,5%          |
| 300                  | 75%            | 74%            | 74%           | 74%            |
| 350                  | 71%            | 69,5%          | 69%           | 69%            |

Dapat dilihat pada tabel di atas, untuk hasil terbaik dengan tingkat akurasi 75%, presisi 74,5%, *recall* 74,5%, dan *f-score* 74,5% yang dihasilkan oleh dimensi vektor 150 yang hasilnya tidak berbeda jauh dengan yang dihasilkan model *GloVe* dengan dimensi vektor 300.

## 5. Kesimpulan

Dari hasil pengujian yang telah dilakukan dengan menggunakan model *GloVe* sebagai *word embedding* pada LSTM untuk analisis sentimen cuitan pada Twitter berbahasa Indonesia, didapati hasil terbaik untuk performa klasifikasi yaitu dengan tingkat akurasi 73%. Tingkat akurasi tersebut diperoleh dengan menggunakan data cuitan 1150 dan dimensi vektor *GloVe* 300. Dan pada penggunaan data yang tidak seimbang dengan jumlah data dengan label positif lebih banyak, didapati hasil klasifikasi dengan tingkat presisi 74,5% dan *recall* 74,5% dengan akurasi 75% dan *f-score* 74,5%.

**Daftar Pustaka**

- [1] WebSindo, "Indonesia Digital 2019 : Media Sosial," 2019. [Online]. Available: <https://websindo.com/indonesia-digital-2019-media-sosial/>. [Accessed: 01-Sep-2019].
- [2] A. Giachanou and F. Crestani, "Like It or Not : A Survey of Twitter Sentiment Analysis Methods," vol. 49, no. 2, 2016.
- [3] J. Li and L. Qiu, "A Sentiment Analysis Method of Short Texts in Microblog," 2017.
- [4] H. Imaduddin, "Word Embedding Comparison for Indonesian Language Sentiment Analysis," *2019 Int. Conf. Artif. Intell. Inf. Technol.*, pp. 426–430, 2019.
- [5] J. Pennington, R. Socher, and C. D. Manning, "GloVe : Global Vectors for Word Representation."
- [6] Y. Sharma, G. Agrawal, P. Jain, T. Kumar, and S. Member, "Vector Representation of Words for Sentiment Analysis Using GloVe," pp. 279–284, 2017.
- [7] D. Li and J. Qian, "Text Sentiment Analysis Based on Long Short-Term Memory," pp. 471–475, 2016.
- [8] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Vector Space," pp. 1–12.
- [9] Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," vol. 3536, no. c, 2017.
- [10] J. Wang, "An LSTM Approach to Short Text Sentiment Classification with Word Embeddings," pp. 214–223, 2018.