



## Analisis Sentimen Penanganan Covid-19 Menggunakan Metode *Long Short-Term Memory* Pada Media Sosial Twitter

Jasman Pardede<sup>a</sup>, Ivan Pakpahan<sup>b</sup>

<sup>a</sup> Fakultas Teknologi Industri / Teknik Informatika, jasman@itenas.ac.id, Institut Teknologi Nasional Bandung

<sup>b</sup> Fakultas Teknologi Industri / Teknik Informatika, ivanpakpahangn@gmail.com, Institut Teknologi Nasional Bandung

### ABSTRACT

*Social media can be used to convey people's aspirations for government policies. Several government policies regarding the regulation of the handling of Covid-19 often elicit responses and criticism from the public, especially on Twitter social media. The aspirations conveyed can contain positive or negative responses. To find out the representation of public sentiment based on these responses, it is necessary to do a sentiment analysis technique. The Long Short-Term Memory (LSTM) method is a deep learning method that can be used in sentiment analysis. LSTM is used because it has the advantage of being able to store large amounts of information in memory cells. Before carrying out classification modeling, the dataset must go through the process of case folding, punctuation removal, normalization, and stopword removal. This aims to ease the training process by eliminating characters or words that are not needed. Next, the word is vectorized using FastText, the goal is to change the string data type to an array vector, so that the word can be processed in the LSTM. The final performance of the model is measured based on the value of precision, recall, accuracy, and f Measure. Based on testing the dropout layer parameters on the hidden layer against 10 fold cross validation, the average accuracy of model testing resulting from all k folds is 72.4%. with the maximum model performance achieved at k fold = 9, when using a dropout layer of 0.4, the values achieved are precision, recall, accuracy, and f measure respectively: 76.74%, 80.49%, 78.31%, 78.57%.*

**Keywords:** Sentiment Analysis, LSTM, FastText

### ABSTRAK

Media sosial dapat dimanfaatkan untuk menyampaikan aspirasi masyarakat terhadap kebijakan pemerintah. Beberapa kebijakan pemerintah yang menyangkut regulasi penanganan covid-19 kerap kali menuai tanggapan dan kritik masyarakat terutama di media sosial twitter. Aspirasi yang disampaikan dapat memuat respon positif atau negatif. Untuk mengetahui representasi sentimen masyarakat berdasarkan respon tersebut, maka perlu dilakukan suatu teknik analisis sentimen. Metode *Long Short-Term Memory* (LSTM) merupakan metode *Deep learning* yang dapat digunakan dalam melakukan analisis sentimen. LSTM digunakan karena memiliki kelebihan dapat menyimpan informasi yang besar di dalam sel memori. Sebelum melakukan pemodelan klasifikasi, dataset harus melewati proses *case folding*, *punctuation removal*, *normalization*, dan *stopword removal*. Hal ini bertujuan untuk meringankan proses pelatihan dengan menghilangkan karakter atau kata yang tidak dibutuhkan. Selanjutnya kata divektorkan menggunakan FastText, tujuannya untuk mengubah tipe data string menjadi vektor array, sehingga kata dapat diproses di dalam LSTM. Performa akhir model diukur berdasarkan nilai *precision*, *recall*, *accuracy*, dan *f Measure*. Berdasarkan pengujian parameter *dropout* layer pada *hidden* layer terhadap 10 *fold cross validation*, rata-rata akurasi pengujian model yang dihasilkan dari keseluruhan k fold adalah sebesar 72.4%. dengan performa model maksimum yang dicapai terdapat pada k fold = 9, saat menggunakan *dropout* layer sebesar 0.4, dengan nilai yang dicapai adalah *precision*, *recall*, *accuracy*, dan *f measure* secara berurutan: 76.74%, 80.49%, 78.31%, 78.57%.

**Kata Kunci:** Analisis Sentimen, LSTM, FastText

## 1. PENDAHULUAN

Media sosial merupakan platform digital yang dimanfaatkan penggunaannya untuk saling berinteraksi dan juga menyebarkan konten. Selain itu, media sosial juga dapat menjadi wadah demokrasi bagi masyarakat terhadap sistem pemerintahan, hal ini berkaitan dengan peningkatan kebijakan secara transparans dan juga partisipasi masyarakat dalam mengevaluasi kinerja pemerintah. Pada saat kondisi krisis kesehatan publik, seperti saat terjadinya pandemi, pemerintah harus berupaya melakukan penanganan ekstra dan sigap untuk memberikan informasi secara efektif juga efisien kepada masyarakat. Gagalnya komunikasi yang dilakukan pemerintah akan menyebabkan masyarakat menjadi takut, khawatir, dan panik di masa kritis [1].

Topik yang kerap menjadi *trending* di media sosial Twitter pada bulan Maret hingga Mei 2020 adalah cuitan-cuitan masyarakat mengenai kebijakan pemerintah terhadap penanganan pandemi Covid-19 di Indonesia [2]. Kebijakan yang ditetapkan pemerintah untuk mengatasi Covid-19 banyak menuai reaksi pro dan kontra dari kalangan masyarakat [3]. Sebagian masyarakat menganggap bahwa pemerintah sudah bekerja maksimal dalam menangani dan mencegah penyebaran pandemi Covid-19, sebagian lainnya bertanggapan bahwa sikap pemerintah masih tidak tegas dan kurang maksimal dalam memutus rantai penyebaran Covid-19 di Indonesia. Respon dan tanggapan masyarakat tersebut dapat dijadikan sebagai tolak ukur untuk mengevaluasi kebijakan pemerintah.

Namun berdasarkan pengamatan penulis, beberapa cuitan yang disampaikan masyarakat di media sosial Twitter sering menggunakan pemilihan istilah kata yang tidak tepat dan tidak baku. Dari banyaknya cuitan yang tidak baku tersebut bisa saja beberapa memuat sentimen yang sama. Maka dari itu, perlu dilakukan suatu analisis terhadap tanggapan masyarakat tersebut guna mengetahui model umum sentimen masyarakat terkait kebijakan pemerintah dalam menangani pandemi.

Analisis sentimen adalah teknik yang digunakan untuk mengidentifikasi, mengekstraksi opini, dan memproses data tekstual secara otomatis untuk mendapatkan sentimen yang terkandung dalam opini. Tujuan dari analisis sentimen adalah untuk menemukan pola dari suatu tulisan apakah memuat makna positif atau negatif. Analisis sentimen idealnya membutuhkan data dan sumber daya yang besar, terlebih jika murni dilakukan dengan pendekatan berbasis pembelajaran mesin [4]. Teknik dan metode dalam analisa sentimen sangat banyak. Oleh karena itu, dalam pemilihan teknik atau algoritma akan sangat dipengaruhi pada tujuan yang hendak dicapai.

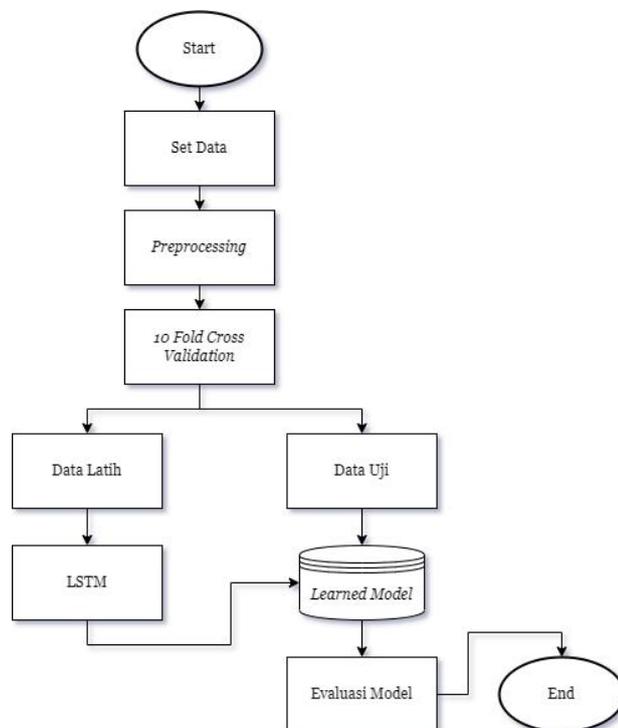
Salah satu teknik yang dapat digunakan untuk analisis sentimen adalah dengan memanfaatkan metode *Deep learning* khususnya *Natural Language Processing* (NLP). Walaupun pada awalnya, *Deep learning* dirancang untuk penelitian di bidang pemrosesan citra digital dan pengenalan sinyal suara, *Deep learning* saat ini dapat dimanfaatkan untuk memproses data tekstual untuk keperluan analisis sentimen.

*Deep learning* memiliki beberapa jenis variasi algoritma, seperti *Convolutional Neural Network* (CNN), *Self Organizing Maps* (SOM), *Recurrent Neural Network* (RNN), dan lainnya. Metode RNN kemudian terbagi lagi menjadi beberapa variasi, salah satu yang populer yaitu *Long Short-Term Memory* (LSTM). LSTM merupakan salah satu metode dalam *Deep learning* yang dapat digunakan untuk *Natural Language Processing* (NLP) seperti pengenalan suara, translasi teks, dan juga analisis sentimen. LSTM merupakan pengembangan dari metode sebelumnya, yaitu *Recurrent Neural Network* (RNN). LSTM memiliki kemampuan untuk menyimpan atau mengingat informasi pada sel memori dengan ruang yang sangat besar. Metode LSTM juga dirancang untuk menyelesaikan permasalahan *vanishing gradient* yang terjadi pada RNN.

Beberapa penelitian mengenai klasifikasi teks menggunakan metode LSTM sudah pernah dilakukan sebelumnya. Seperti pada penelitian yang dilakukan [5] mengenai klasifikasi SMS yang mengandung spam dan non-spam, LSTM mampu memberikan tingkat akurasi sebesar 96,09%, presisi sebesar 96,09%, *recall* sebesar 96,09% dan *f-measure* sebesar 96,09%. Selanjutnya pada penelitian [6], yang melakukan penelitian untuk mengklasifikasi judul berita *clickbait* berbahasa Indonesia, LSTM memberikan tingkat akurasi sebesar 82% dengan *loss* sebesar 38%. Seluruh penelitian terdahulu tersebut menjadi referensi utama untuk dilakukannya penelitian pada studi ini yang juga menggunakan LSTM. Studi yang dilakukan saat ini akan menambahkan fitur *word embedding* menggunakan fasttext, karena penulis menduga akan meningkatkan performa model klasifikasi.

## 2. METODOLOGI PENELITIAN

Dalam pembuatan model klasifikasi teks menggunakan LSTM, terdapat beberapa tahapan yang harus dilakukan. Tahapan tersebut antara lain: (1) pengumpulan data, (2) *preprocessing*, (3) *splitting* set data, (4) pelatihan set data, dan (5) evaluasi model. Tahapan pembentukan model LSTM ditunjukkan pada Gambar 1.



Gambar 1 Diagram Alir Metode Penelitian

### 2.1 Pengumpulan Data

Dalam penelitian yang dilakukan, set data yang digunakan adalah *Indonesian Tweets COVID-19 Handling* (2020). Set data tersebut sudah divalidasi oleh penelitian yang dilakukan [7]. Set data berisi cuitan twitter mengenai aspek umum terhadap penanganan Covid-19 oleh pemerintah Indonesia. Cuitan Twitter tersebut diaggregasi dengan kata kunci “#COVID-19indonesia, COVID-19 di Indonesia, penanganan pemerintah terhadap COVID-19” dimulai dari tanggal 23 Maret 2020–14 Mei 2020.

Teknik pengumpulan data dan informasi yang digunakan dalam penelitian yang dilakukan yaitu dengan melakukan studi literatur yang akan menghasilkan data primer. Set data merupakan data yang bersifat publik dapat diakses di website Kaggle. Data terdiri dari 3 label, yaitu positif, netral, dan negatif. Data tersebut kemudian dimodifikasi, dipilih yang hanya memuat label positif dan negatif, serta menghapus data yang memiliki duplikat.

### 2.2 Text Preprocessing

Sebelum diproses ke dalam pembentukan model klasifikasi menggunakan LSTM, dilakukan tahap *preprocessing* terhadap teks di dalam set data, yaitu *case folding*, *punctuation removal*, *normalization*, *stopword removal*, dan *word embedding*. Tahap *preprocessing* yang dilakukan dijelaskan pada tahapan sebagai berikut:

#### - Case Folding

Teknik *case folding* dilakukan dengan mengubah semua huruf dalam kata menjadi format *lower case* atau huruf kecil. *Case Folding* bertujuan untuk menyeragamkan karakter yang ada pada teks set data sehingga dapat memudahkan dalam proses penghapusan karakter atau istilah-istilah eksklusif yang tidak diinginkan pada penelitian.

Tabel 1 Contoh Case Folding

Sebelum <i>Case Folding</i>	Sesudah <i>Case Folding</i>
di Indonesia yg gratis hanya wifi, bodo amat pemerintah dg Covid 19 ....lbh pnting ekonomi	di indonesia yg gratis hanya wifi, bodo amat pemerintah dg covid 19 ....lbh pnting ekonomi

- *Punctuation Removal*

*Punctuation removal* adalah teknik untuk menghapus tanda baca. Tanda baca yang dimaksud merupakan tanda baca seperti @? ! , / = + - \ > < ; “ ( ) { } [ ] . : | dan lainnya. Menghapus tanda baca dilakukan karena tanda baca tidak diperhatikan selama proses *training* sehingga dengan melakukan penghapusan tanda baca membuat proses *training* akan menjadi lebih ringan.

Tabel 2 Contoh Punctuation Removal

Sebelum <i>Punctuation Removal</i>	Sesudah <i>Punctuation Removal</i>
di indonesia yg gratis hanya <b>wifi</b> , bodo amat pemerintah dg <b>covid 19</b> ....lbh pnting ekonomi	di indonesia yg gratis hanya wifi bodo amat pemerintah dg covid lbh pnting ekonomi

- *Normalization*

Melakukan normalisasi berguna agar penggunaan kata di dalam kalimat menjadi kata yang baku sesuai dengan ejaan bahasa Indonesia. Proses ini dilakukan apabila terdapat kata tidak baku ditemukan di dalam kamus *slang word*, kata tersebut kemudian diubah dengan terjemahannya.

Tabel 3 Contoh Normalization

Sebelum <i>Normalization</i>	Sesudah <i>Normalization</i>
di indonesia <b>yg</b> gratis hanya wifi <b>bodo</b> amat pemerintah <b>dg</b> covid <b>lbh pnting</b> ekonomi	di indonesia yang gratis hanya wifi bodoh amat pemerintah dengan covid lebih penting ekonomi

- *Stopword Removal*

Proses ini dilakukan dengan menyaring kata yang ada di dalam kalimat dengan cara menghilangkan kata-kata yang termasuk sebagai *stopword* bahasa Indonesia. Penghapusan *stopword* bertujuan untuk mengurangi jumlah kata yang tidak memiliki makna dalam sebuah dokumen.

Tabel 4 Contoh Stopword Removal

Sebelum <i>Stopword Removal</i>	Sesudah <i>Stopword Removal</i>
<b>di</b> indonesia <b>yang</b> gratis <b>hanya</b> wifi bodoh <b>amat</b> pemerintah <b>dengan</b> covid <b>lebih penting</b> ekonomi	indonesia gratis wifi bodoh pemerintah covid ekonomi

- *FastText Word Embedding*

FastText adalah jaringan saraf tiruan yang disusun dari masukan berupa nilai konversi dari kata. Terdapat 3 lapisan yang menyusun arsitektur jaringan FastText, yaitu lapisan input (*embedding layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). FastText merupakan *library* yang dikembangkan oleh Facebook, dengan mengembangkan jaringan Word2Vec. FastText dan Word2Vec sama-sama dapat dimanfaatkan untuk keperluan *word embedding*. FastText dan Word2Vec digunakan untuk mendeteksi kemiripan kata secara semantik maupun sintaksis di dalam set data. FastText memiliki keunggulan dibanding Word2Vec. Salah satunya adalah kemampuan FastText untuk menangani kata yang tidak pernah ditemukan sebelumnya

Terdapat dua jenis arsitektur pada jaringan saraf fastText yaitu *Contious Bag of Words* (CBOW) dan *Skip-gram* [8]. Perbedaan dari CBOW dan Skip-gram adalah cara CBOW yang bekerja cenderung memprediksi probabilitas satu kata sebagai target dari konteks sebagai *input*. Sedangkan Skip-gram adalah kebalikannya dari CBOW, yang berarti kata yang menjadi target adalah sebuah konteks terdiri dari kelompok kata.

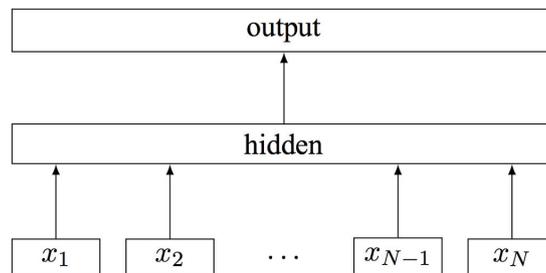
FastText menginisiasikan suatu vektor kata dengan menjumlahkan nilai dari n-gram kata dan nilai dari penyisipan kata itu sendiri. (1).

$$FastText_{word} = V_{word} + \sum g \in ngrams(word) \tag{1}$$

Jika kata yang disematkan tidak ada di dalam kamus/data *training*, keluar dari kosakata (OOV), maka vektor kata adalah total dari n-gram (2).

$$FastText_{oov\_word} = \sum g \in ngrams(oov\_word) \tag{2}$$

Selanjutnya, token kata tersebut menjadi *input* di dalam jaringan saraf Skip-gram untuk menghasilkan nilai probabilitas konteks. Arsitektur jaringan skip-gram dapat dilihat pada gambar 2. Contoh vektor kata yang dihasilkan oleh FastText dapat dilihat pada tabel 5.



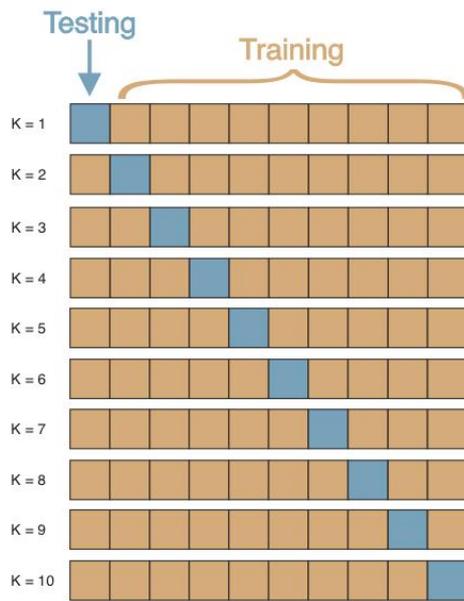
Gambar 2 Arsitektur FastText

Tabel 5 Contoh FastText *Embedding*

Sebelum <i>FastText Embedding</i>	Sesudah <i>FastText Embedding</i>
indonesia gratis wifi bodoh pemerintah covid ekonomi	[[[-0.04,0.21,0.04,0.32,0.27,0.58,0.21,...]]]

**2.3 K Fold Cross Validation**

*Cross-validation* (CV) adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma, dimana data dipisahkan menjadi dua subset yaitu data proses pembelajaran dan data validasi/evaluasi. Model atau algoritma dilatih oleh subset pembelajaran dan divalidasi oleh *subset* validasi. Selanjutnya pemilihan jenis CV dapat didasarkan pada ukuran *dataset*. Menurut Antoni Wibowo, pakar *data mining*, CV *K-fold* digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi.

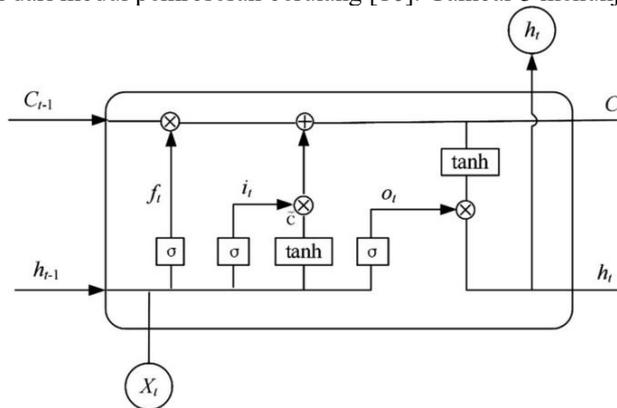


Gambar 3 Skema 10 Fold Cross Validation

Menurut Jason Brownlee, pakar *Deep Learning*, 10 fold CV adalah nilai K fold CV yang direkomendasikan untuk pemilihan model terbaik. 10 Fold telah diuji melalui beberapa eksperimen untuk menghasilkan perkiraan keterampilan model dengan bias rendah dan varians sederhana.. Dalam 10 fold CV, data dibagi menjadi 10 partisi berukuran kira-kira sama, sehingga tersedia 10 variasi subset data untuk mengevaluasi kinerja model atau algoritma. Untuk masing-masing dari 10 subset data tersebut, CV akan menggunakan 9 fold untuk pelatihan dan 1 fold untuk pengujian, seperti yang ditunjukkan pada Gambar 2.

**2.4 Long Short-Term Memory (LSTM)**

*Long Short-Term Memory* adalah salah satu variasi RNN yang dibuat untuk menghindari masalah ketergantungan jangka panjang pada RNN. LSTM dapat mengingat informasi jangka panjang. Sama seperti RNN, LSTM juga terdiri dari modul pemrosesan berulang [10]. Gambar 3 menunjukkan arsitektur LSTM.



Gambar 4 Arsitektur LSTM

Ide dari LSTM adalah dibuatnya jalur yang menghubungkan konteks lama ( $ct-1$ ) ke konteks baru ( $ct$ ) yang disebut juga *cell state*, *memory cell* atau jalur memori. Dengan adanya jalur tersebut, suatu nilai pada konteks yang lama akan dengan mudah dihubungkan ke konteks yang baru jika diperlukan dengan sedikit sekali modifikasi. LSTM memiliki kemampuan untuk menghapus atau menambahkan informasi ke keadaan

sel, dan diatur dengan cermat oleh fungsi *sigmoid*. Langkah-langkah pada LSTM memiliki 4 gerbang *layer* yaitu *forget gate* (1), *Input gate*(2)(3), *cell state* (4), dan *output gate*(5)(6).

- *Forget Gate*

Pada tahapan ini informasi yang kurang dibutuhkan atau tidak terlalu memiliki makna terhadap kasus yang diolah akan dihilangkan menggunakan fungsi *Sigmoid*.

$$ft = \sigma(Wxf \cdot xt + Whf \cdot (ht - 1) + bf) \quad (1)$$

Keterangan :

$ft$  = *forget gate*

$\sigma$  = fungsi *Sigmoid*

$Wxf$  = nilai *weight input* untuk *forget gate*

$Whf$  = nilai *weight output* untuk *forget gate*

$ht - 1$  = nilai *output/hidden* sebelum orde ke t

$xt$  = nilai *input* pada orde ke t

$bf$  = nilai bias pada *forget gate*

- *Input Gate*

Proses ini akan memilah dan menentukan informasi tertentu yang akan diperbarui ke bagian *cell state* dengan menggunakan fungsi aktivasi *Sigmoid*.

$$it = \sigma(Wxi \cdot xt + Whi \cdot (ht - 1) + bi) \quad (2)$$

$$\bar{Ct} = \tanh(WxC \cdot xt + WhC \cdot (ht - 1) + bC) \quad (3)$$

Keterangan :

$It$  = *input gate*

$\sigma$  = fungsi *Sigmoid*

$Wxi$  = nilai *weight input* untuk *input gate*

$Whi$  = nilai *weight output* untuk *input gate*

$ht - 1$  = nilai *output/hidden* sebelum orde ke t

$xt$  = nilai *input* pada orde ke t

$bi$  = nilai bias pada *input gate*

$\bar{Ct}$  = nilai kandidat baru ke *cell state*

$\tanh$  = fungsi *Tanh*

$WxC$  = nilai *weight input* untuk *cell state*

$WhC$  = nilai *weight output* untuk *cell state*

$bC$  = nilai bias untuk *cell state*

- *Cell State*

Selanjutnya memperbarui nilai *cell state* lama ( $ct-1$ ) menjadi *cell state* baru ( $ct$ ).

$$Ct = ft * (Ct - 1) + (it * \bar{Ct}) \quad (4)$$

Keterangan:

$Ct$  = *Cell state*

$ft$  = *forget gate*

$Ct - 1$  = *Cell state* sebelum orde ke t

$it$  = *input gate*

$\bar{Ct}$  = nilai kandidat baru ke *cell state*

- *Output Gate*

Tahapan akhir terjadi pada komponen *output gate*. Dilakukan dengan menjalankan aktivasi *Sigmoid* untuk menghasilkan nilai *output* pada *hidden state* dan menempatkan *cell state* pada *Tanh*. Setelah menghasilkan nilai *output Sigmoid* dan nilai *output Tanh* kedua hasil aktivasi tersebut dikali sebelum menuju langkah selanjutnya.

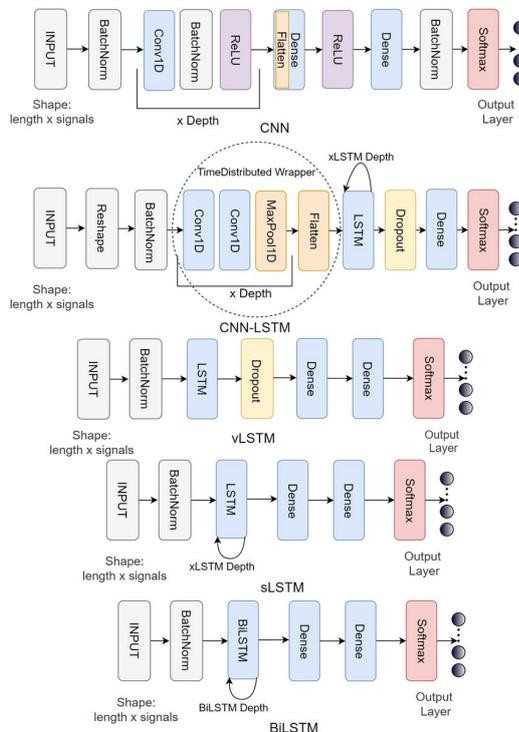
$$ot = \sigma(Wxo \cdot xt + Who \cdot (ht - 1) + bo) \quad (5)$$

$$ht = ot * \tanh(Ct) \quad (6)$$

Keterangan:

- $O_t$  = output gate
- $\sigma$  = fungsi Sigmoid
- $W_{xo}$  = nilai weight input untuk output gate
- $W_{ho}$  = nilai weight output untuk output gate
- $h_{t-1}$  = nilai output/hidden sebelum orde ke t
- $x_t$  = nilai input pada orde ke t
- $b_o$  = nilai bias pada output gate
- $h_t$  = nilai output orde t
- $\tanh$  = fungsi Tanh
- $C_t$  = Cell state

LSTM memiliki beberapa jenis variasi arsitektur, seperti *vanilla* LSTM, *Deep* LSTM, *Bidirectional* LSTM, dan lainnya, yang menjadi pembeda antar variasi LSTM adalah jumlah layer LSTM dan urutan prosesnya. Beberapa variasi LSTM dapat dilihat pada gambar 5.



Gambar 5 Variasi LSTM

### 2.5 Evaluasi Model

*Confusion matrix* digunakan sebagai metode pengujian dalam penelitian analisis sentimen penanganan Covid-19 menggunakan metode LSTM. Metode ini merepresentasikan hasil klasifikasi menggunakan matriks yang dapat dilihat pada Gambar 4.

Aktual	Prediksi	
	True	False
True	<b>TP</b>	<b>FP</b>
False	<b>FN</b>	<b>TN</b>

Gambar 6 Tabel *Confusion Matrix*

Penjelasan gambar diatas antara lain TP (*True Positive*) yaitu jumlah nilai positif yang berhasil diklasifikasikan sebagai positif. FP (*False positive*) yaitu nilai positif yang salah diklasifikasikan menjadi negatif. FN (*False negative*) yaitu nilai negatif yang salah diklasifikasikan sebagai positif. TN (*True negative*) yaitu nilai negatif yang berhasil diklasifikasikan sebagai nilai negatif.

Metode pengujian yang digunakan dalam penelitian yang dilakukan menghasilkan perhitungan yang diformulasikan sebagai berikut :

- *Accuracy*

*Accuracy* adalah perhitungan seberapa tepat klasifikasi yang telah dibangun, sesuai dengan target yang ada.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \tag{7}$$

- *Precision*

*Precision* adalah perhitungan keakuratan antara data target dengan hasil prediksi dari model.

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{8}$$

- *Recall*

*Recall* adalah perhitungan yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{9}$$

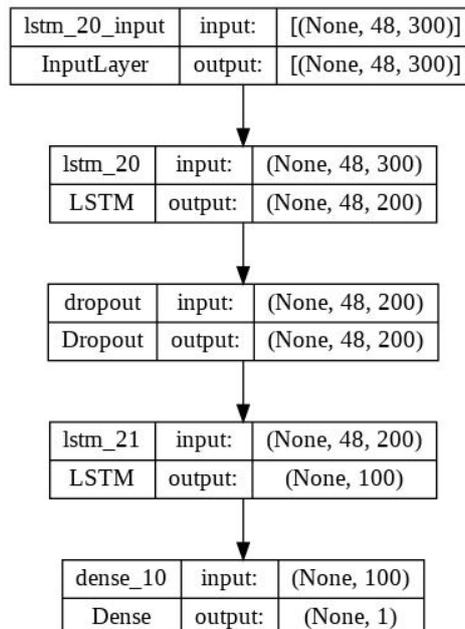
- *F Measure*

Parameter *F Measure* adalah perhitungan yang menggambarkan perbandingan antara *precision* dan *recall*. Jika nilai FN dan FP tidak mendekati sebaiknya digunakan *f measure* dibandingkan nilai akurasi.

$$F\ Measure = \frac{2 \times Precision \times Recall}{(Precision+Recall)} \times 100\% \tag{10}$$

### 2.6 Arsitektur Model

Pada penelitian yang dilakukan, untuk mendapatkan suatu model klasifikasi yang optimal dilakukan beberapa eksperimen pengujian arsitektur menggunakan variasi *dropout layer*. Setelah itu dilakukan proses *training* dilanjutkan proses validasi. Adapun rancangan dari kedua jaringan yang dimaksud ditampilkan pada Gambar 7.



Gambar 7 Arsitektur Jaringan

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Set Data

Setelah melakukan *preprocessing*, data *tweet* di dalam set data menjadi sejumlah 1660 baris kalimat (Gambar 8) dengan komposisi label positif sebanyak 760 dan label negatif sebanyak 900. Kemudian set data tersebut dibagi menggunakan *10 fold cross validation*, sehingga menghasilkan 10 variasi set data yang baru dengan indeks yang berbeda-beda, ditampilkan pada Tabel 6.



Gambar 8 Ukuran Set Data Hasil *Text Preprocessing*

Tabel 6 Indeks Pembagian Set Data

K	Indeks Data Latih	Indeks Data Uji
1	166-1659	0-165
2	0-165, 332-1659	166-331
3	0-331, 498-1659	332-497
4	0-497, 664-1659	498-663
5	0-663, 830-1659	664-829
6	0-829, 996-1659	830-997
7	0-995, 1162-1659	996-1161
8	0-1161, 1328-1659	1162-1327
9	0-1327, 1494-1659	1328-1493
10	0-1493	1494-1659

#### 3.2 Hasil Pelatihan

Pelatihan model dilakukan dengan menggunakan *10 fold* set data dan disesuaikan dengan arsitektur dan parameter *dropout*, sehingga setiap satu partisi diuji sebanyak lima kali. Berikut adalah Tabel 7, Tabel 8, Tabel 9, Tabel 10, dan Tabel 11 yang menampilkan hasil *training* dari model yang telah dibentuk sesuai dengan arsitektur jaringan:

Tabel 7 Hasil Pelatihan Tanpa Dropout Layer

K	Loss	Accuracy	Val Loss	Val Accuracy
1	0.010685	0.997768	1.014469	0.8
2	0.00826	0.997024	1.028974	0.806667
3	0.020064	0.985863	0.797635	0.786667
4	0.037845	0.99256	0.963316	0.76
5	0.03074	0.995536	1.176996	0.753333
6	0.020323	0.99628	0.984469	0.793333
7	0.032179	0.994792	0.985539	0.773333
8	0.000674	0.993304	0.835497	0.806667
9	0.015111	0.998512	1.046681	0.8
10	0.006551	0.998512	1.33849	0.74
<b>Rerata</b>	<b>0.018243</b>	<b>0.995015</b>	<b>1.017207</b>	<b>0.782</b>

Tabel 8 Hasil Pelatihan Menggunakan Dropout Layer 0.2

K	Loss	Accuracy	Val_Loss	Val_Accuracy
1	0.006199	0.994792	0.835882	0.813333
2	0.023032	0.991815	0.937743	0.786667
3	0.006333	0.99628	0.914735	0.793333
4	0.001515	0.994048	0.905467	0.793333
5	0.033345	0.985119	1.108917	0.733333
6	0.029737	0.997768	1.262695	0.746667
7	0.006336	0.997768	1.060438	0.8
8	0.040622	0.995536	0.981085	0.793333
9	0.001641	0.997024	1.155867	0.78
10	0.012381	0.998512	1.33638	0.746667
<b>Rerata</b>	<b>0.016114</b>	<b>0.994866</b>	<b>1.049921</b>	<b>0.778667</b>

Tabel 9 Hasil Pelatihan Menggunakan Dropout Layer 0.3

K	Loss	Accuracy	Val_Loss	Val_Accuracy
1	0.113808	0.964286	0.639176	0.78
2	0.11499	0.950149	0.705505	0.766667
3	0.116261	0.94122	0.679061	0.76
4	0.132471	0.943452	0.682942	0.726667
5	0.10993	0.966518	0.704245	0.786667
6	0.131366	0.958333	0.75682	0.786667
7	0.180477	0.954613	0.76453	0.753333
8	0.117906	0.946429	0.627979	0.786667
9	0.125428	0.958333	0.808828	0.72
10	0.137483	0.954613	0.774903	0.74
<b>Rerata</b>	<b>0.128012</b>	<b>0.953795</b>	<b>0.714399</b>	<b>0.760667</b>

Tabel 10 Hasil Pelatihan Menggunakan Dropout Layer 0.4

K	Loss	Accuracy	Val_Loss	Val_Accuracy
1	0.022099	0.997768	0.986536	0.813333
2	0.019297	0.994792	0.979556	0.82
3	0.011303	0.994048	0.936247	0.8
4	0.022702	0.997768	1.019994	0.793333
5	0.033317	0.997024	0.960426	0.793333
6	0.04905	0.998512	0.987091	0.82
7	0.000853	0.994792	0.931225	0.793333
8	0.006168	0.997768	1.122731	0.766667
9	0.006269	0.993304	0.939601	0.786667
10	0.028787	0.998512	1.387997	0.733333
<b>Rerata</b>	<b>0.019985</b>	<b>0.996429</b>	<b>1.02514</b>	<b>0.792</b>

Tabel 11 Hasil Pelatihan Menggunakan Dropout Layer 0.5

K	Loss	Accuracy	Val_Loss	Val_Accuracy
1	0.012508	0.997024	1.121432	0.786667
2	0.01495	0.994792	1.285536	0.766667
3	0.024149	0.997768	0.903783	0.813333
4	0.012026	0.99256	0.873234	0.806667
5	0.054092	0.997768	1.077871	0.806667

<b>6</b>	0.056111	0.991071	1.060283	0.753333
<b>7</b>	0.018982	0.997768	1.027617	0.806667
<b>8</b>	0.023829	0.997768	1.073954	0.766667
<b>9</b>	0.05344	0.997768	1.140104	0.76
<b>10</b>	0.040409	0.99628	1.284593	0.726667
<b>Rerata</b>	<b>0.03105</b>	<b>0.996057</b>	<b>1.084841</b>	<b>0.779334</b>

### 3.3 Hasil Evaluasi Model

Pada tahap evaluasi, dilakukan pengujian pada data uji menggunakan model yang sudah melewati proses pelatihan. Prediksi yang akan dihasilkan oleh sistem akan dibandingkan dengan *ground truth*, sehingga menghasilkan 4 parameter, yaitu presisi, *recall*, akurasi, dan *f-measure*. Berikut adalah Tabel 4 yang menampilkan hasil dari pengujian model:

Tabel 12 Hasil Pengujian Model

Partisi	Dropout	TP	FP	TN	FN	Prec	Rec	ACC	F Score
1	Tidak	65	21	54	26	75.58	71.43	71.69	73.45
2	Tidak	68	27	48	23	67.07	76.39	73.49	71.43
3	Tidak	64	25	50	27	83.64	58.97	75.30	69.17
4	Tidak	65	25	50	26	58.82	68.49	65.06	63.29
5	Tidak	72	30	45	19	71.83	56.67	64.46	63.35
6	Tidak	55	27	67	17	67.95	74.65	74.10	71.14
7	Tidak	49	22	72	23	69.84	69.84	77.11	69.84
8	Tidak	55	31	63	17	61.84	73.44	72.29	67.14
9	Tidak	57	29	65	15	72.84	71.95	72.89	72.39
10	Tidak	57	29	65	15	74.65	69.74	75.30	72.11
Rata-rata						<b>70.41</b>	<b>69.16</b>	<b>72.17</b>	<b>69.33</b>
Partisi	Dropout	TP	FP	TN	FN	Prec	Rec	ACC	F Score
1	0.2	46	9	79	32	71.58	74.73	69.88	73.12
2	0.2	47	16	72	31	69.01	68.06	72.89	68.53
3	0.2	43	12	76	35	74.60	60.26	71.69	66.67
4	0.2	46	13	75	32	64.38	64.38	68.67	64.38
5	0.2	50	25	63	28	83.93	52.22	68.67	64.38
6	0.2	50	35	58	23	67.53	73.24	73.49	70.27
7	0.2	47	26	67	26	64.18	68.25	73.49	66.15
8	0.2	49	29	64	24	65.79	78.13	75.90	71.43
9	0.2	45	17	76	28	70.59	73.17	71.69	71.86
10	0.2	44	19	74	29	79.41	71.05	78.31	75.00
Rata-rata						<b>71.10</b>	<b>68.35</b>	<b>72.47</b>	<b>69.18</b>
Partisi	Dropout	TP	FP	TN	FN	Prec	Rec	ACC	F Score
1	0.3	51	20	56	39	71.91	70.33	68.67	71.11
2	0.3	47	9	67	43	63.95	76.39	71.08	69.62
3	0.3	53	13	63	37	78.18	55.13	71.69	64.66
4	0.3	66	18	58	24	62.82	67.12	68.07	64.90
5	0.3	43	30	46	47	80.30	58.89	69.88	67.95
6	0.3	53	25	70	18	74.60	66.20	75.90	70.15
7	0.3	52	25	70	19	65.67	69.84	74.70	67.69
8	0.3	47	16	79	24	60.00	79.69	71.69	68.46
9	0.3	53	25	70	18	69.32	74.39	71.08	71.76
10	0.3	57	30	65	14	70.97	57.89	69.88	63.77
Rata-rata						<b>69.77</b>	<b>67.59</b>	<b>71.27</b>	<b>68.01</b>

Partisi	Dropout	TP	FP	TN	FN	Prec	Rec	ACC	F Score
1	0.4	44	19	84	19	72.22	71.43	69.28	71.82
2	0.4	43	24	79	20	66.28	79.17	73.49	72.15
3	0.4	44	23	80	19	77.97	58.97	72.89	67.15
4	0.4	44	23	80	19	72.58	61.64	72.89	66.67
5	0.4	49	29	74	14	78.57	73.33	74.70	75.86
6	0.4	47	29	73	17	67.95	74.65	74.10	71.14
7	0.4	50	26	76	14	65.67	69.84	74.70	67.69
8	0.4	51	34	68	13	67.69	68.75	75.30	68.22
9	0.4	44	21	81	20	76.74	80.49	78.31	78.57
10	0.4	47	22	80	17	75.71	69.74	75.90	72.60
Rata-rata						<b>72.14</b>	<b>70.80</b>	<b>74.16</b>	<b>71.19</b>
Partisi	Dropout	TP	FP	TN	FN	Prec	Rec	ACC	F Score
1	0.5	59	22	62	23	70.59	79.12	70.48	74.61
2	0.5	60	25	59	22	66.28	79.17	73.49	72.15
3	0.5	61	27	57	21	66.67	64.10	68.07	65.36
4	0.5	66	20	64	16	69.84	60.27	71.08	64.71
5	0.5	59	18	66	23	58.90	47.78	53.61	52.76
6	0.5	53	18	72	23	65.52	80.28	73.49	72.15
7	0.5	54	14	76	22	62.82	77.78	74.10	69.50
8	0.5	44	18	72	32	68.12	73.44	76.51	70.68
9	0.5	53	17	73	23	76.62	71.95	75.30	74.21
10	0.5	54	19	71	22	73.97	71.05	75.30	72.48
Rata-rata						<b>67.93</b>	<b>70.49</b>	<b>71.14</b>	<b>68.86</b>

Berdasarkan hasil uji coba yang telah dilakukan menggunakan data *test* sebanyak ±166 kalimat *Tweet*, menunjukkan bahwa model LSTM yang telah dirancang dalam penelitian yang dilakukan berjalan dengan nilai akurasi optimal saat menggunakan dropout layer 0.4. Nilai akurasi terbaik pengujian model dicapai pada partisi K = 9.

### 3.4 Visualisasi Sentimen



Gambar 9 Word Cloud pada Kelas Negatif

Gambar 9 merupakan hasil visualisasi pada kelas Negatif. Hasil visualisasi pada kelas negatif menunjukkan adanya ketidakpuasan/kecewaan masyarakat terhadap kinerja pemerintah dalam menangani pandemi Covid-19. Seperti pada cuitan berikut “*“*covid-19 menjangkiti lebih dari 100 negara. korsel hanya melakukan social distancing, bkn lockdown. krn, pemerintah mengurusinya dg melakukan rapid test, dan rakyatpun mematuhi. apa kabar indonesia? #negaraabaikanrakyat

#kesadaranmemicuperubahan #rezimcucitangan”. Visualisasi di kelas negatif ini dapat dijadikan acuan pemerintah agar lebih optimal lagi dalam menyikapi pandemi Covid-19 di Indonesia.

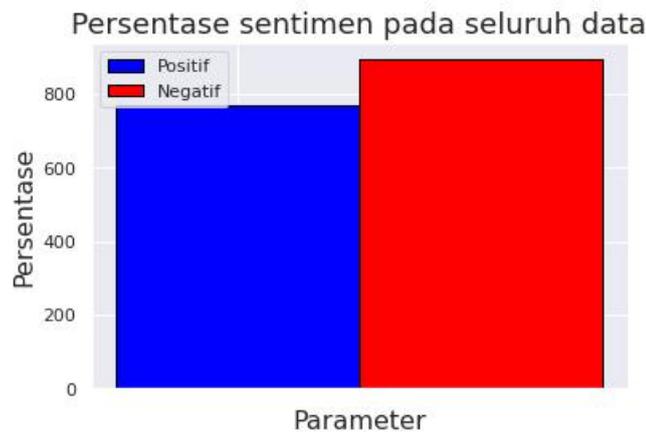


Gambar 10 Word Cloud pada Kelas Positif

Gambar 10 merupakan hasil visualisasi pada kelas Positif. Hasil visualisasi pada kelas positif menunjukkan adanya kepuasan masyarakat atas kinerja pemerintah dan juga berisi ajakan untuk berpartisipasi dalam penanganan Covid-19. Seperti pada cuitan berikut “dengan patuh pada himbuan pemerintah untuk #socialdistancing maka diri sendiri bisa jadi pahlawan, demi menyelamatkan masyarakat lainnya, dari meluasnya wabah #covid\_19 di indonesia. kalau bukan kita siapa lagi dan kalau bukan sekarang kapan lagi mengakhiri musibah ini”. Visualisasi di kelas positif ini dapat dijadikan acuan pemerintah untuk mempertahankan bahkan terus meningkatkan kinerjanya, selain itu, dapat juga menjadi pandangan bagi masyarakat lainnya agar ikut berpartisipasi dalam mencegah penyebaran wabah Covid-19.

### 3.5 Perbandingan Sentimen

Sub bab ini menjelaskan terkait perbandingan antara sentimen yang berhasil diprediksi oleh model dengan data *testing*. Tujuannya untuk menghitung persentase sentimen masyarakat baik itu positif atau negatif. Pada Gambar 11, ditampilkan grafik sentimen yang berhasil diekstrak dari cuitan twitter. Grafik menunjukkan sentimen negatif diperoleh sebanyak 100 dan sentimen positif sebanyak 66. Hal ini menunjukkan bahwa kurang lebih 54% dari dataset cuitan masyarakat masih merasa tidak puas atas hasil kinerja pemerintah dalam menangani pandemi covid-19.



Gambar 11 Persentase Sentimen

## 4. KESIMPULAN DAN SARAN

Sebelum melakukan pelatihan model klasifikasi, dataset terlebih dulu diolah dalam *preprocessing*, hal ini bertujuan untuk meringankan memori saat pelatihan dan juga mengkonversikan kata menjadi matriks. Dari hasil pengujian model yang sudah dilaksanakan, diperoleh kesimpulan bahwa pengujian model dengan LSTM dengan memiliki *dropout* 0.4 memiliki nilai akurasi yang optimal dibandingkan dengan parameter lainnya, model maksimum yang dihasilkan dari evaluasi model dicapai pada partisi 9. Dari perhitungan evaluasi yang ditampilkan, disimpulkan bahwa model mampu melakukan klasifikasi analisis sentimen penanganan Covid-19 dengan parameter yang dihasilkan adalah *precision*, *recall*, *accuracy*, dan *f measure* secara berurutan: 76.74%, 80.49%, 78.31%, 78.57%.

Penulis berharap penelitian selanjutnya dapat meningkatkan hasil klasifikasi yang lebih akurat dan lebih baik. Maka dari itu, penulis memiliki saran untuk pengembangan penelitian di masa yang akan datang, yaitu sebagai berikut :

- Menambahkan jumlah set datayang relevan, dikarenakan jumlah set data sangat mempengaruhi kinerja dan hasil akurasi model klasifikasi.
- Menguji metode *word embedding* yang lain seperti Word2Vec, Glove, Bert, Elmo, dan metode vektorisasi teks yang lain.
- Melakukan pembangunan model LSTM dengan arsitektur yang berbeda, seperti *Bidirectional*, LSTM-CNN, GRU, dan arsitektur lainnya.

#### DAFTAR PUSTAKA

- [1] Chen, Q., Min, C., Zhang, W., Wang, G., Ma, X., and Evans, R. (2020). Unpacking the black box: How to promote citizen engagement through government social media during the COVID-19 crisis. *ELSEVIER*, 1, doi: <https://doi.org/10.1016/j.chb.2020.106380>
- [2] Nurhajati, Lestari., Sukandar, Rudi., Oktaviani, Rani Chandra., and Wijayanto, Xenia Angelica. (2020). PERBINCANGAN ISU CORONA COVID-19 PADA MEDIA DARING DAN MEDIA SOSIAL DI INDONESIA. Lembaga Penelitian, Publikasi, dan Pengabdian Masyarakat.
- [3] Adpim, B. (2021, April 14). *TREN PERKEMBANGAN PENANGANAN COVID-19 DINILAI SEMAKIN BAIK*. Tersedia di biroadpim.kalteng.go.id: <https://biroadpim.kalteng.go.id/2021/04/tren-perkembangan-penanganan-covid-19-dinilai-semakin-baik/> [Accessed 7 Juli 2022]
- [4] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Cambridge: Morgan & Claypool Publishers.
- [5] Muslikah, A. N. (2021). *Sms Spam Detection Menggunakan Metode Long Short-Term Memory*.
- [6] Habibie, I. (2018). *Identifikasi Judul Berita Clickbait Berbahasa Indonesia Dengan Algoritma Long Short-Term Memory (Lstm) Recurrent Neural Network*.
- [7] Prastyo, P., Sumi, A., Dian, A., & Permanasari, A. (2020). Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel. *Journal of Information Systems Engineering and Business Intelligence*, 11. doi : <https://doi.org/10.20473/jisebi.6.2.112-122>
- [8] Saputra, H. D. (2021). Implementasi Metode Multilayer Perceptron Dengan Fasttext Word Embedding Untuk Klasifikasi Ujaran Kebencian.
- [9] Wibowo, A. (2017, November 24). *10 Fold-Cross Validation*. Tersedia di MTI Binus: <https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/> [Accessed 7 Juli 2022]
- [10] Dobilas, S. (2022, Februari 7). *LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past*. Tersedia di: Toward Data Science: <https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e> [Accessed 7 Juli 2022]