



SENTIMENT ANALYSIS UJARAN KEBENCIAN TWITTER MENGGUNAKAN LSTM DAN BIDIRECTIONAL LSTM

Senin, 12 Desember 2022

Adrianus Charlie Hadirria Agelsadewa
195314174





BAB I

PENDAHULUAN

SENTIMENT ANALYSIS UJARAN KEBENCIAN
TWITTER MENGGUNAKAN LSTM DAN
BIDIRECTIONAL LSTM

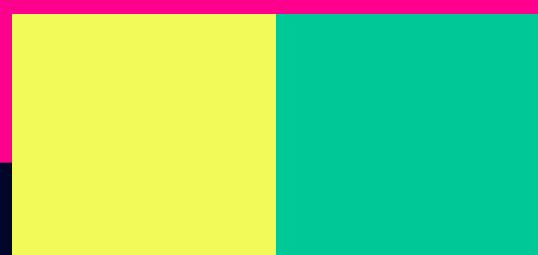
Adrianus Charlie Hadirria Agelsadewa
195314174





LATAR BELAKANG

- Indonesia memasuki industri 4.0
- Pengguna Twitter di Indonesia mencapai 18,45 Juta
- Terjadi fenomena ujaran kebencian di berbagai platform media sosial, tak terkecuali di Twitter
- Ujaran kebencian dapat memicu sebuah konflik baik individu maupun kelompok
- Sentiment analysis dapat menentukan status menentukan status kecenderungan tweet





RUMUSAN MASALAH

Diperlukan pendekatan untuk mengenali ujaran kebencian di Twitter

TUJUAN PENELITIAN

- 01 Melakukan sentiment analisis ujaran kebencian tweets pengguna Twitter di Indonesia menggunakan algoritma LSTM dan Bidirectional LSTM.
- 02 Menghitung akurasi optimal dari algoritma LSTM dan Bidirectional LSTM.
- 03 Mengetahui parameter deep learning yang mampu memberikan hasil paling optimal.



MANFAAT PENELITIAN

Dapat membuat sebuah model deep learning menggunakan algoritma LSTM dan Bidirectional LSTM yang dapat mendeteksi ujar kebencianan.



BATASAN MASALAH

- 01 Dataset yang digunakan merupakan data tweets dari pengguna Twitter di Indonesia
- 02 Dataset yang digunakan berbahasa Indonesia
- 03 Dataset merupakan tweets yang membahas tentang kejadian fenomenal hacker Bjorka





BAB II

LANDASAN TEORI

SENTIMENT ANALYSIS UJARAN KEBENCIAN
TWITTER MENGGUNAKAN LSTM DAN
BIDIRECTIONAL LSTM

Adrianus Charlie Hadirria Agelsadewa
195314174



TINJAUAN PUSTAKA

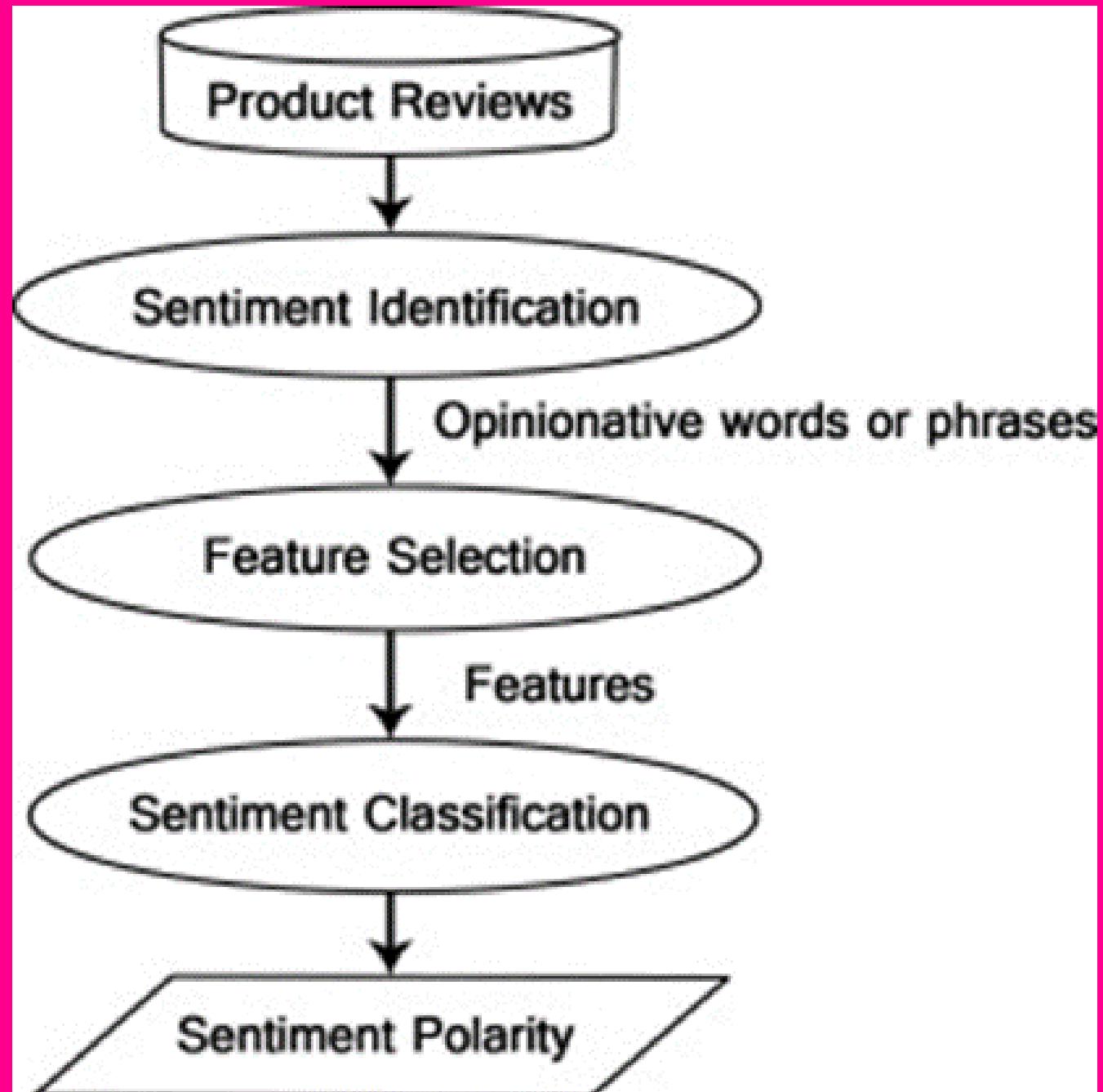
| Pengarang dan Tahun | Judul | Metode | Keterangan |
|--|---|--|--|
| Mohammed M.Abdelgwad, Taysir Hassan A Soliman, Ahmed I.Taloba, Mohamed Fawzy Farghaly (2021) | Arabic Aspect Based Sentiment Analysis Using Bidirectional GRU Based Models | Bidirectional GRU | Mendapatkan akurasi optimal sebesar 83,98% |
| Veny Amilia Fitri, Rachmadita Andreswari, Muhammad Azani Hasibuan (2019) | Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm | Naïve Bayes | Mendapatkan akurasi sebesar 83,43% |
| | | Decision Tree | Mendapatkan akurasi sebesar 82,91% |
| | | Random Forest | Mendapatkan akurasi sebesar 82,91% |
| Muhammad Ali Fauzi (2018) | Random Forest Approach for Sentiment Analysis in Indonesia Language | Random Forest | Melakukan nilai OOB/akurasi sebesar 82,9% |
| Zhao Jianqiang, Gui Xiaolin, Zhang Xuejun (2018) | Deep Convolution Neural Networks for Twitter Sentiment Analysis | Deep Learning CNN (Convolutional Neural Network) | Mendapatkan akurasi sebesar 87,62% |

TINJAUAN PUSTAKA

| Pengarang dan Tahun | Judul | Metode | Keterangan |
|--|--|------------------------------|--|
| Mohammad Rezwanul Huq, Ahmad Ali, Anika Rahman (2017) | Sentiment Analysis on Twitter Data using KNN and SVM | KNN Classifier | Mendapatkan akurasi terbaik 84,32% dengan 5-fold cross validation |
| | | Support Vector Machine (SVM) | Mendapatkan akurasi terbaik 77,97% dengan 5-fold cross validation |
| Munir Ahmad, Shabib Aftab, Iftikhar Ali (2017) | Sentiment Analysis of Tweets using SVM | Support Vector Machine (SVM) | Mendapatkan akurasi sebesar 85% |
| Shiyang Liao, Junbo Wang, Ruiyun Yua ,Koichi Satob ,Zixue Cheng | CNN for situations understanding based on sentiment analysis of twitter data | CNN | Mendapatkan akurasi tahap pengembangan sebesar 74,5% |

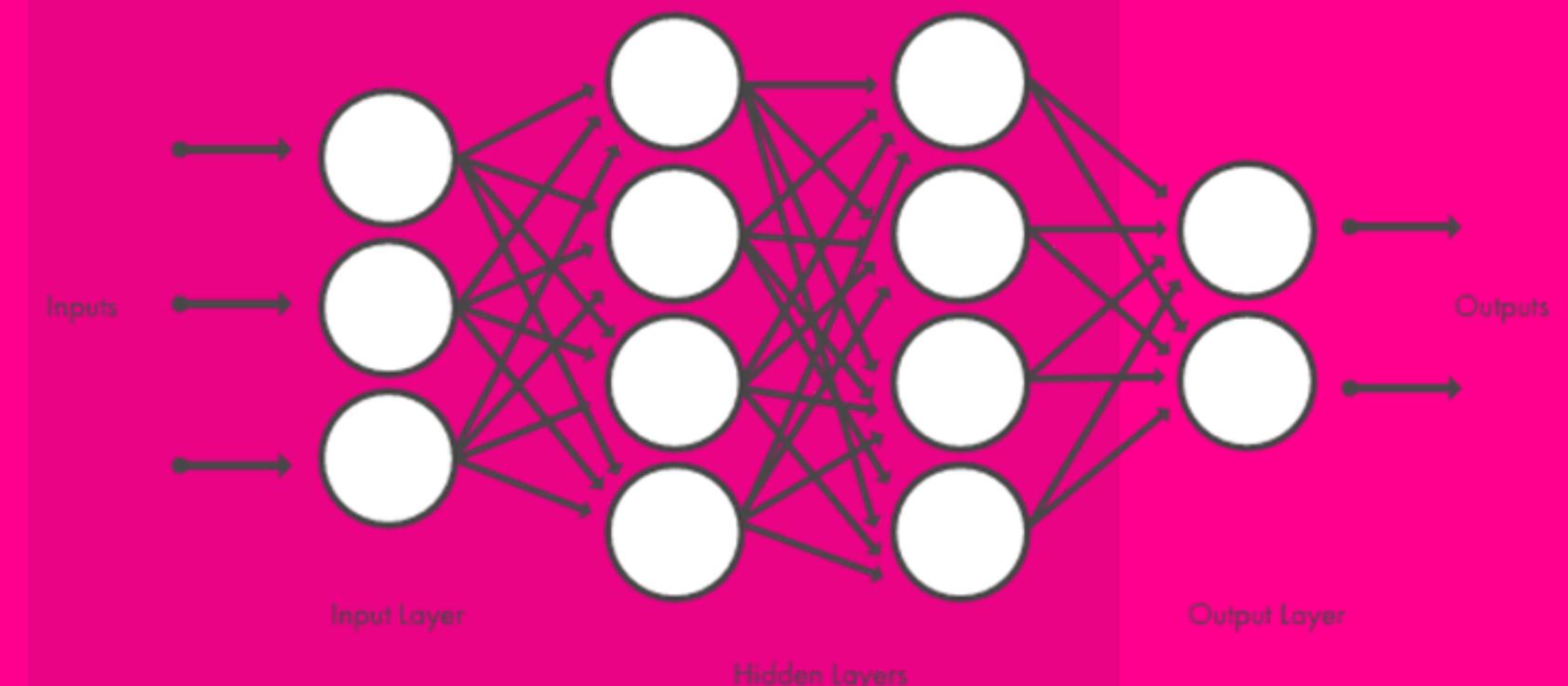
SENTIMENT ANALYSIS

- Sentiment analysis adalah studi berbasis komputer tentang pendapat, keyakinan, dan emosi orang tentang entitas tertentu.
- Sentiment analysis dengan kata lain digunakan untuk mengetahui keadaan suatu data, baik yang mengandung sentimen positif, netral, maupun negatif.
- Dimungkinkan juga untuk menggunakan sentiment analysis sebagai klasifikasi. (Hal.17)



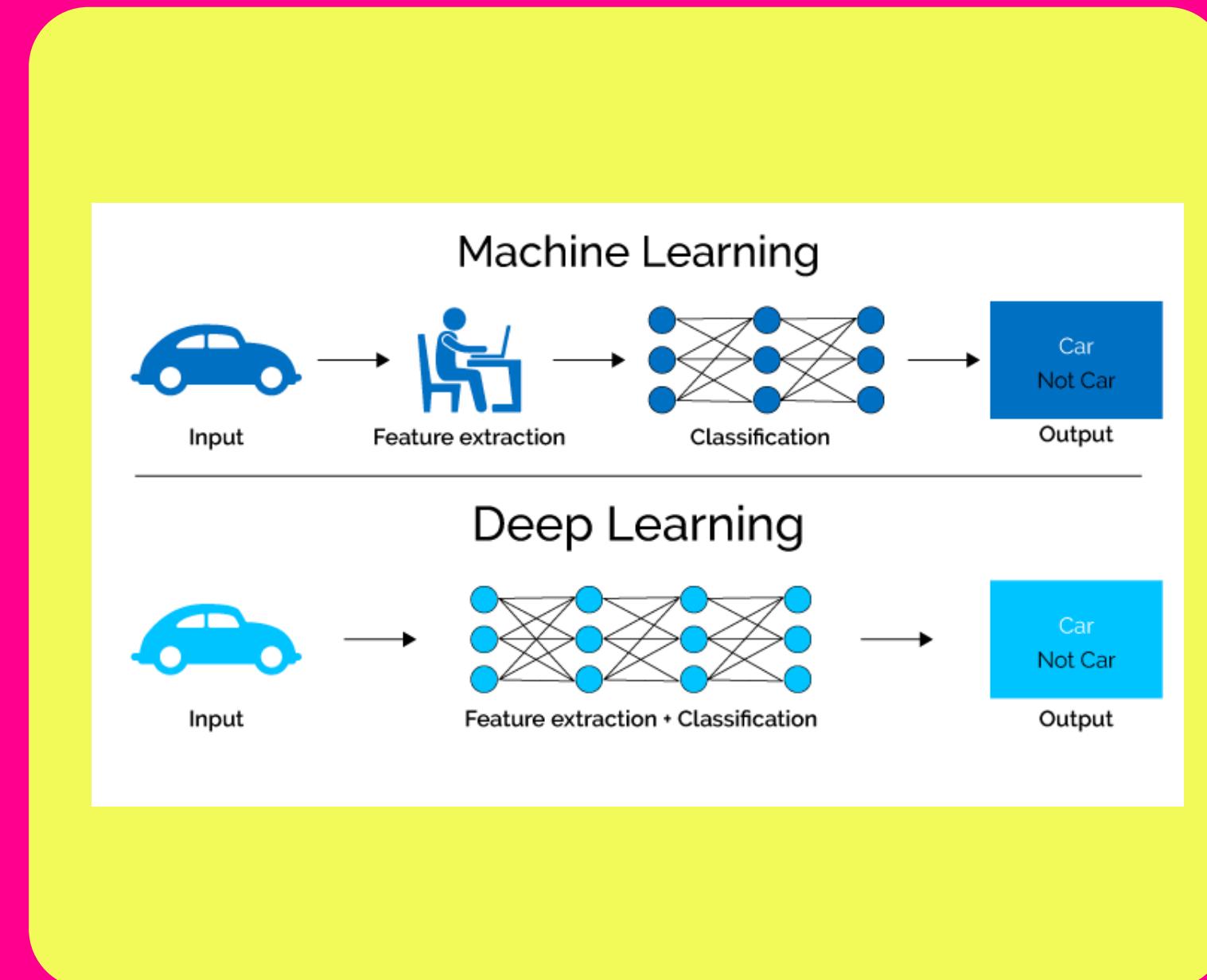
JARINGAN SYARAF TIRUAN

- Jaringan Syaraf Tiruan (JST) merupakan jaringan yang terdiri dari unit pemrosesan kecil yang dimodelkan seperti sistem syaraf manusia.
- JST merupakan alat pemodelan data statistik non-linear yang digunakan untuk memodelkan hubungan kompleks antara input dan output untuk menemukan pola dalam suatu data.
- Model JST pada dasarnya merupakan fungsi model matematika yang mendefinisikan fungsi dari x ke y .
- Secara umum, JST terdiri dari 3 komponen, yaitu input layer, hidden layer serta output layer.



DEEP LEARNING

- Deep learning merupakan salah satu bidang machine learning yang memungkinkan tanpa terjadinya sebuah feature extraction pada data
- Digunakan untuk klasifikasi gambar, text, audio dan video
- Model berdasarkan data yang besar dan berlabel serta jaringan syaraf multilayer
- Deep learning dapat menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma backpropagation untuk menunjukkan bagaimana mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya





KOMPONEN DEEP LEARNING

Deep learning memiliki beberapa komponen penting yang befungsi untuk mengoptimalkan pembelajaran model, diantaranya adalah:

- Optimizer
- Activation Function
- Regularization
- Loss Function

OPTIMIZER

Optimizer adalah algoritma atau metode yang digunakan untuk mengubah atribut jaringan saraf, seperti bobot tingkat pembelajaran, untuk mengurangi loss. Berikut beberapa macam optimizer yang ada dalam deep learning:

- RMSProp
- Adam
- SGD



ACTIVATION FUNCTION

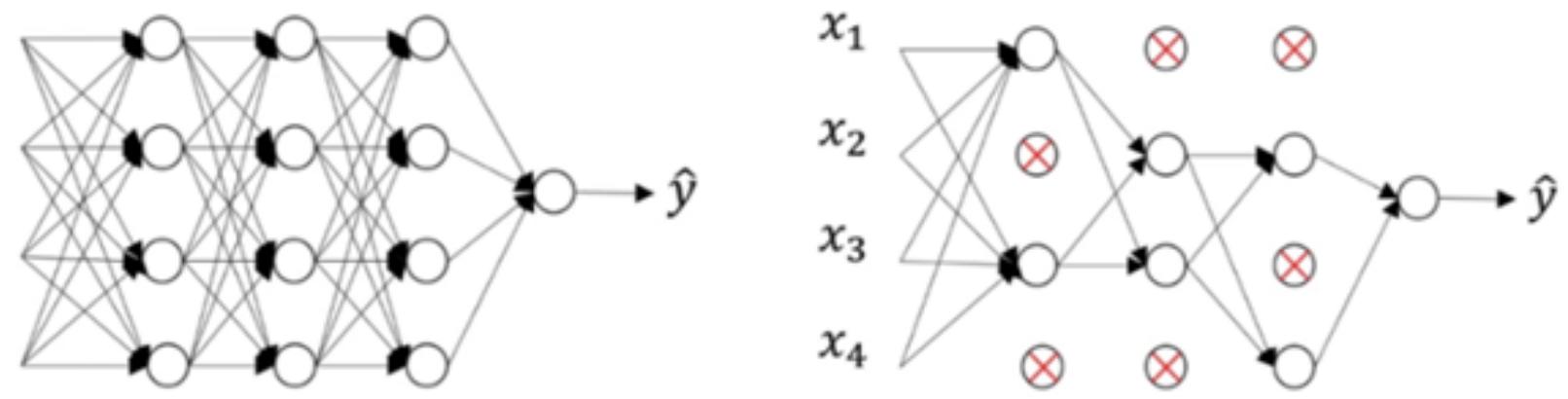
Activation function bertanggung jawab atas apakah neuron dalam jaringan saraf deep learning harus diaktifkan atau tidak. Artinya, ia menggunakan beberapa operasi matematika sederhana untuk menentukan apakah neuron input jaringan relevan atau tidak relevan dengan proses prediksi. Berikut contoh beberapa activation function yang ada di dalam deep learning:

- Sigmoid
- ReLu
- Softmax
- TanH



REGULARIZATION

Regularisasi adalah serangkaian teknik yang mengurangi kompleksitas model jaringan saraf selama pelatihan untuk menghindari overfitting. Salah satu metode regularisasi yang umum digunakan adalah Dropout. Istilah "dropout" mengacu pada putusnya node (input dan lapisan tersembunyi) dalam jaringan saraf. Semua koneksi maju dan mundur dengan node yang dijatuhkan dihapus sementara, sehingga menciptakan arsitektur jaringan baru dari jaringan induk. Dropout berarti selama proses training dengan beberapa probabilitas P neuron dari jaringan saraf dimatikan.



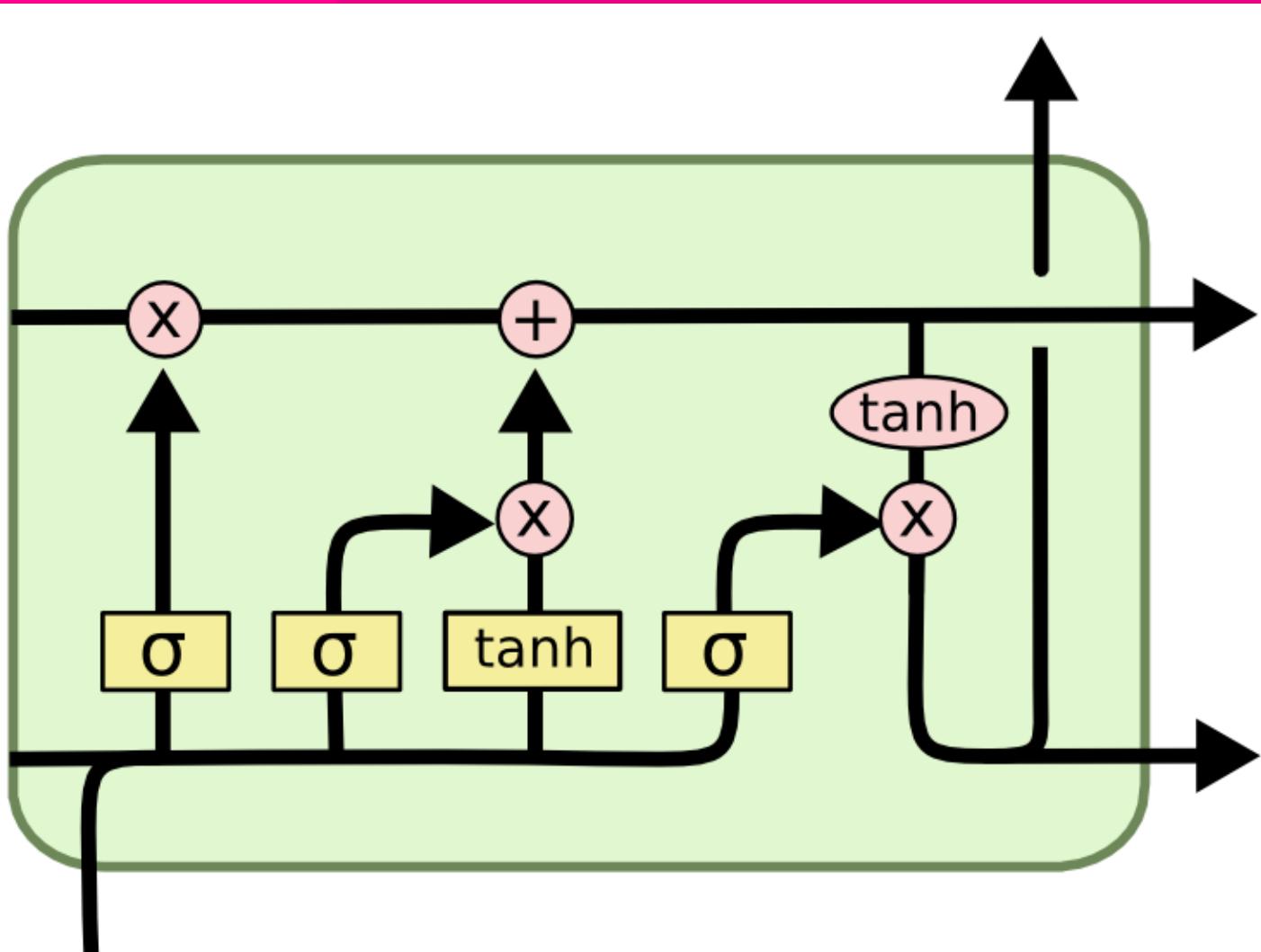
LOSS FUNCTION

Loss function adalah fungsi yang membandingkan target dan nilai keluaran yang diprediksi; mengukur seberapa baik jaringan saraf memodelkan data pelatihan. Selama pelatihan, tujuannya adalah untuk meminimalkan loss antara kekuatan yang diprediksi dan target. Berikut beberapa contoh loss function yang ada di dalam model deep learning:

- Mean Squared Error
- Mean Absolute Error
- Binary Crossentropy
- Categorical Crossentropy



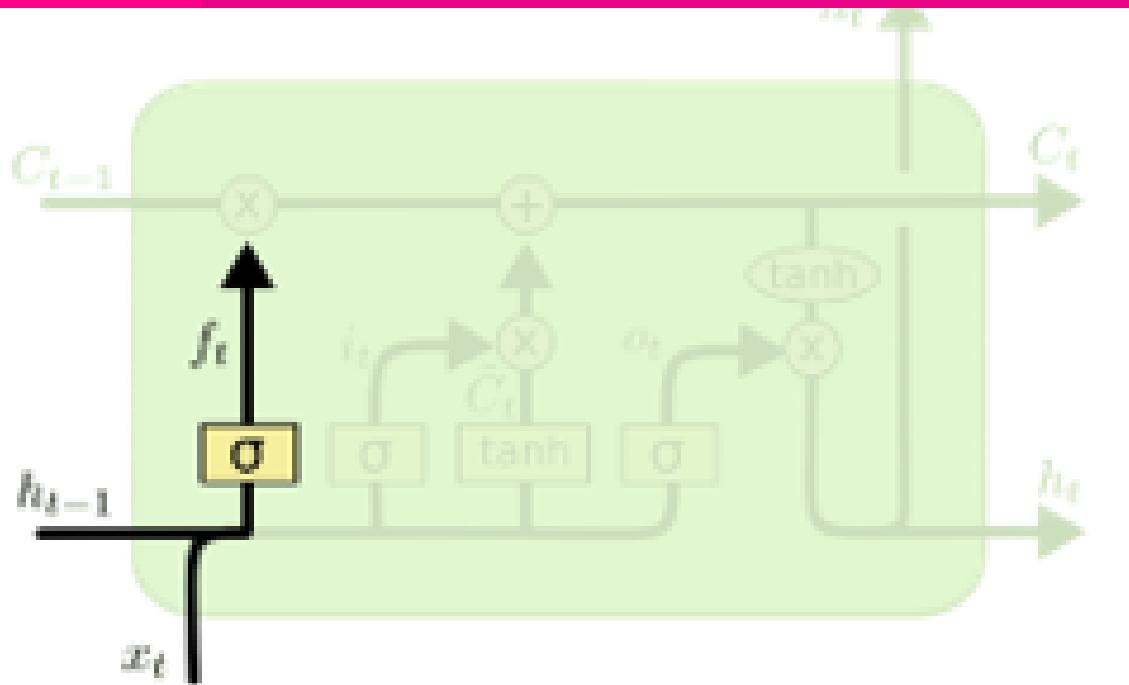
LONG-SHORT TERM MEMORY (LSTM)



Long Short-Term Memory atau biasa dikenal dengan LSTM merupakan sebuah RNN (jaringan syaraf berulang) yang dapat mengatasi permasalahan ketergantungan jangka Panjang . LSTM terdiri dari beberapa komponen utama sebagai berikut:

- Forget Gate
- Input Gate
- Cell State
- Output gate

FORGET GATE (LSTM)



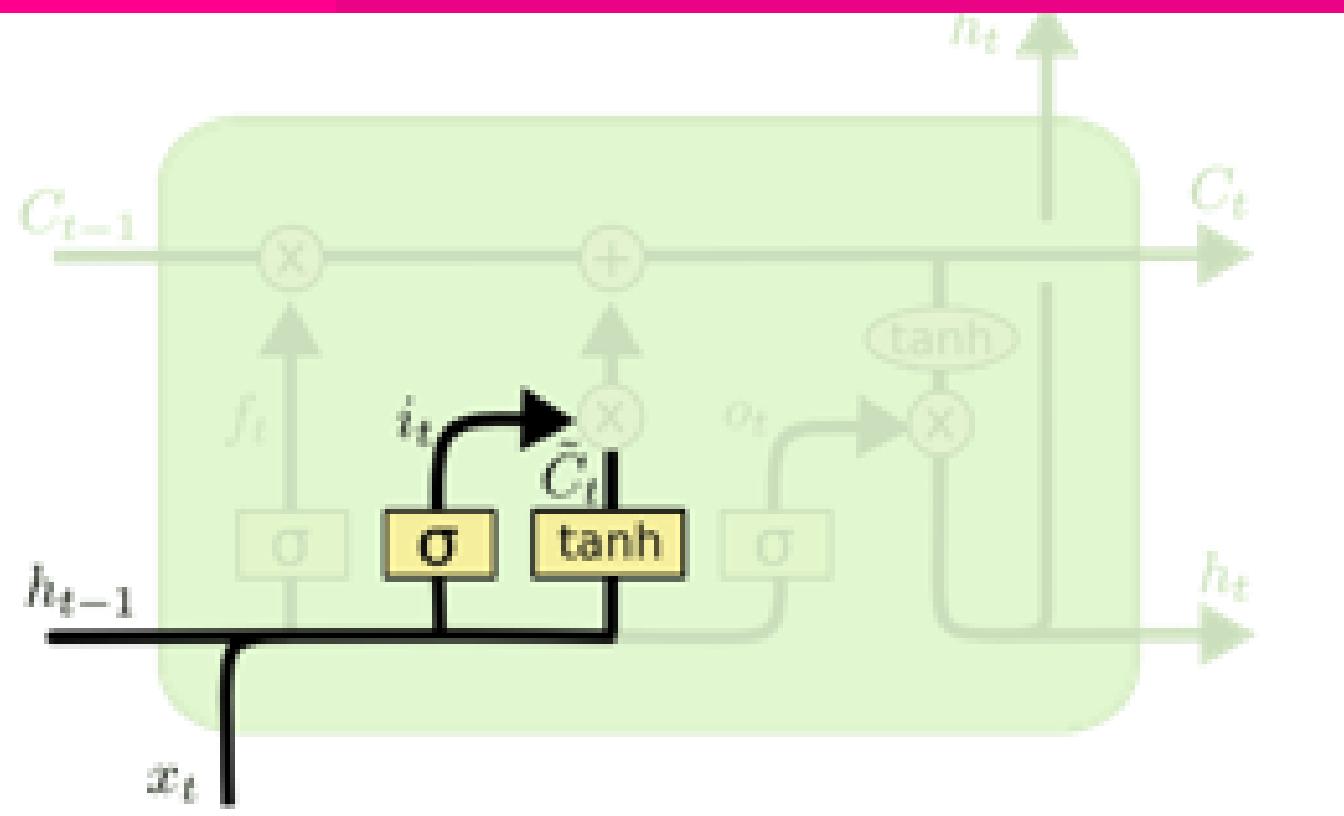
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Di dalam forget gate akan terjadi sebuah proses dimana akan menentukan informasi apa yang akan dibuang dari cell state

| | |
|----------|--------------------------------|
| ft | Nilai forget gate |
| wf | Nilai bobot untuk forget gate |
| $h(t-1)$ | Nilai output dari waktu ke t-1 |
| x_t | Nilai input pada waktu ke t |
| bf | Bias pada forget gate |
| σ | Fungsi sigmoid |

INPUT GATE (LSTM)

Di dalam input gate, akan menentukan informasi baru yang akan disimpan dalam cell state.



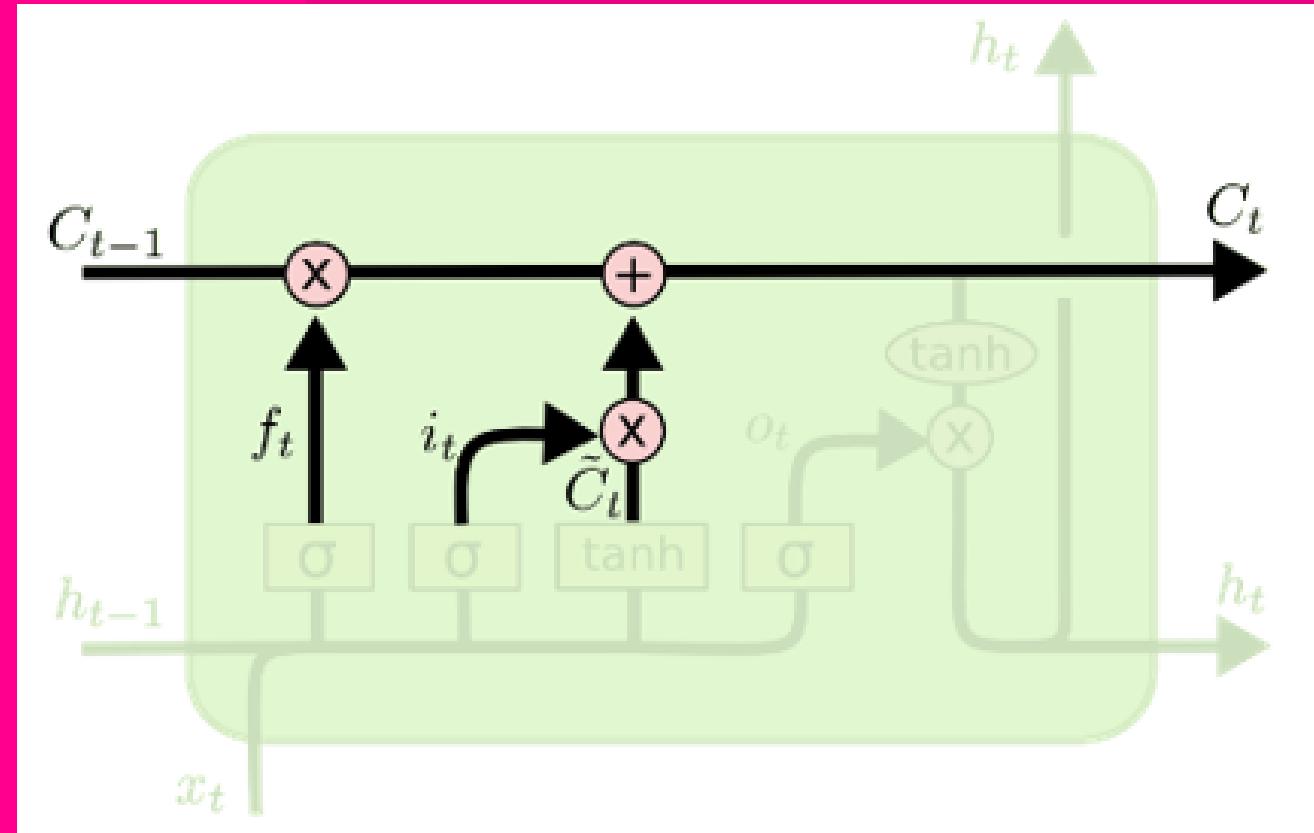
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

| | |
|-------------|--------------------------------|
| i_t | Nilai input gate |
| W_i | Nilai bobot untuk input gate |
| $h(t-1)$ | Nilai output dari waktu ke t-1 |
| x_t | Nilai input pada waktu ke t |
| b_i | Bias pada input gate |
| C_t | Nilai kandidat cell state |
| $\sigma(x)$ | Fungsi sigmoid |
| $\tanh(x)$ | Fungsi tanh |

CELL STATE (LSTM)

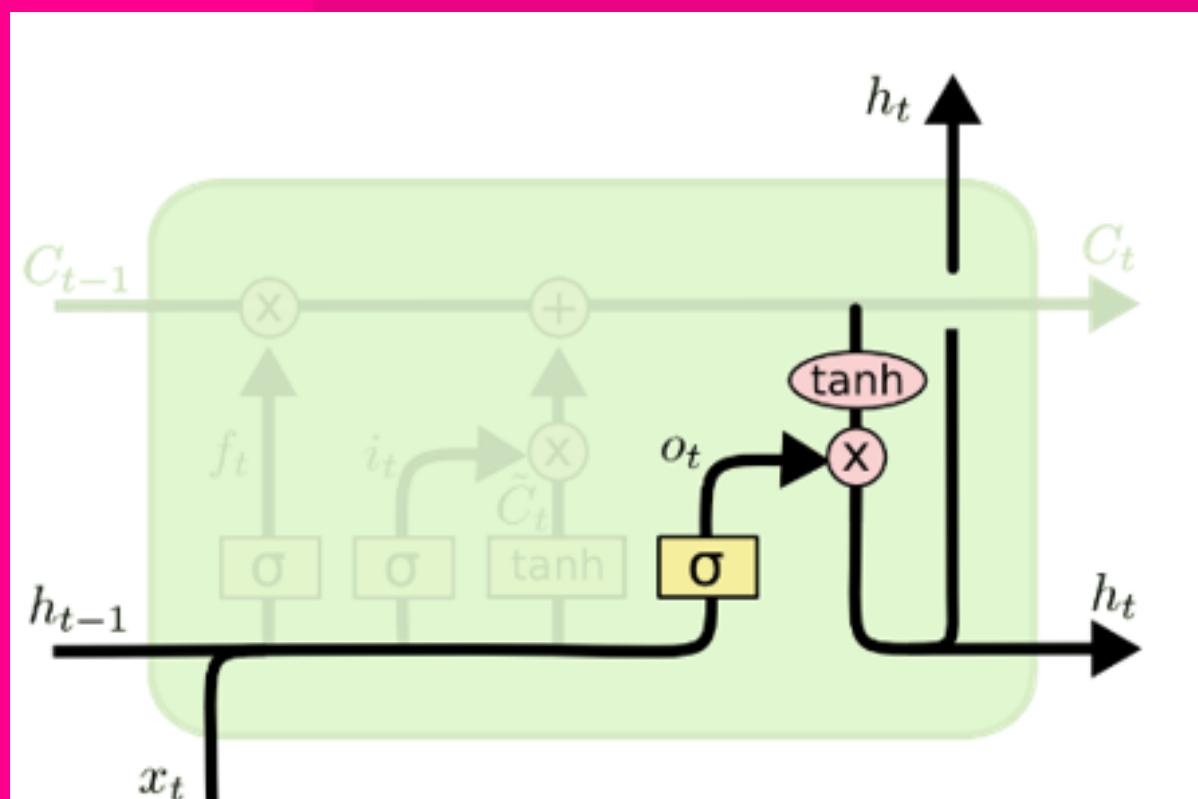
Menampung hasil operasi forget gate dan input gate yang nantinya akan digunakan untuk memperbarui nilai dari cell state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

| | |
|---------------|---|
| f_t | Nilai forget gate |
| i_t | Nilai input gate |
| C_t | Nilai memory cell state |
| $C(t-1)$ | Nilai memory pada cell state sebelumnya |
| \tilde{C}_t | Nilai kandidat cell state |

OUTPUT GATE (LSTM)



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

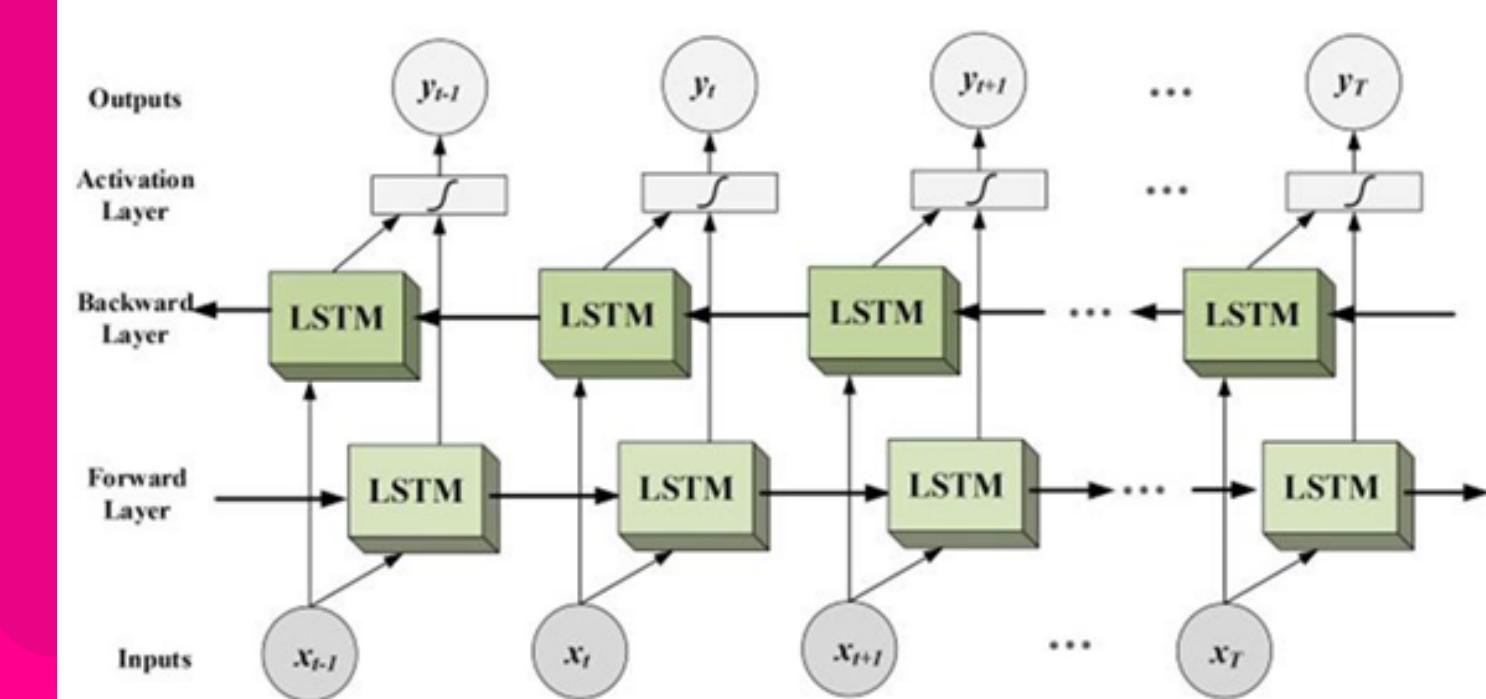
$$h_t = o_t * \tanh (C_t)$$

| | |
|-------------|--------------------------------|
| o_t | Nilai input gate |
| W_o | Nilai bobot untuk output gate |
| $h(t-1)$ | Nilai output dari waktu ke t-1 |
| x_t | Nilai input pada waktu ke t |
| b_o | Bias pada output gate |
| h_t | Nilai output gate |
| $\sigma(x)$ | Fungsi sigmoid |
| $\tanh(x)$ | Fungsi tanh |

Menentukan informasi apa yang akan dijadikan sebuah output dari layer tersebut (melakukan filter terhadap cell state)

BIDIRECTIONAL LSTM

- Bidirectional LSTM atau LSTM dua arah adalah perpanjangan dari LSTM tradisional yang dapat meningkatkan kinerja model untuk masalah klasifikasi urutan.
- BiLSTM melatih 2 urutan input, dimana 1 input dengan urutan aslinya, dan satu terbalik untuk memberikan konteks tambahan ke jaringan dan mengarah pada pembelajaran masalah yang lebih cepat dan lebih lengkap





BAB III

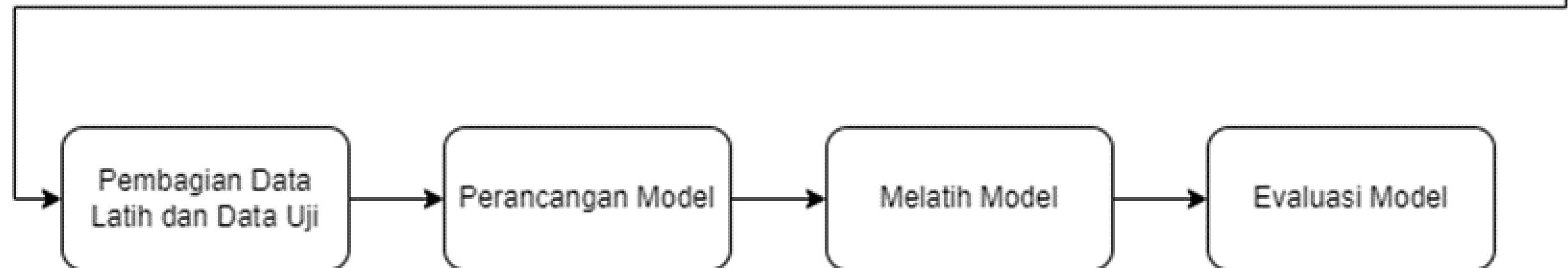
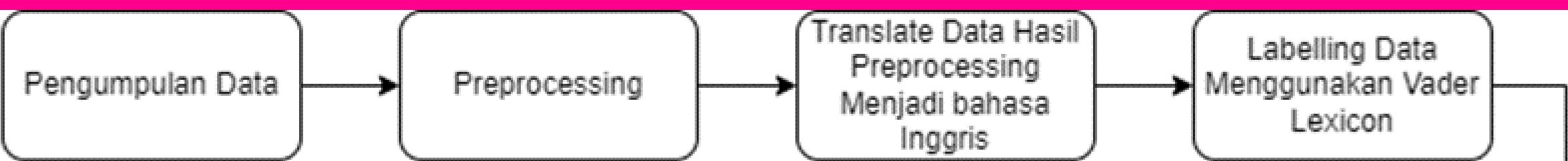
METODOLOGI PENELITIAN

SENTIMENT ANALYSIS UJARAN KEBENCIAN
TWITTER MENGGUNAKAN LSTM DAN
BIDIRECTIONAL LSTM

Adrianus Charlie Hadirria Agelsadewa
195314174



TAHAPAN PENELITIAN



PENGUMPULAN DATA

01 Data diperoleh dari hasil scrapping di Twitter dengan menggunakan beberapa filter diantaranya adalah hashtag Bjorka dan tweets berbahasa indonesia tanpa menyertakan retweet dan link

02 Data berjumlah 20.000

03 Data disimpan dalam format Excel file (.xlsx)

04 Scrapping dilakukan menggunakan libraries snscrepe libraries dari python

| created at | username | text |
|------------|---------------|--|
| 2022-11-02 | rudiartraa | yaah__, udah selesai nih Bjorka Bjorkaan nya ? |
| 2022-11-02 | oi_felix315 | Kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :(((|
| 2022-11-01 | falahsanjaya_ | Knp ga ada yg jadi Bjorka Halloween party kemaren? :((|

| Variable | Keterangan |
|------------|--|
| created at | Tanggal tweet diunggah |
| username | Nama pengguna akun Twitter |
| text | Tweet/teks yang diunggah oleh pengguna Twitter |

PREPROCESSING DATA

Tahapan ini bertujuan untuk membuat kualitas data yang akan digunakan dalam training/analisis menjadi baik. Dalam kasus sentiment analysis ujaran kebencian ini, untuk membuat kualitas dataset teks menjadi baik, maka perlu dilakukan beberapa tahapan preprocessing diantaranya :

- Lowercasing
- Pembersihan Karakter Non-Alphanumeric
- Pembersihan Karakter Khusus
- Normalisasi Kata-Kata “Alay”
- Stemming
- Stop word removal



LOWERCASING

- Lowercasing merupakan tahapan transformasi huruf menjadi huruf kecil.
- Input dari tahapan ini adalah sebuah kalimat yang terdiri dari huruf besar dan kecil yang akan diproses menjadi sebuah output kalimat yang terdiri dari huruf kecil saja

| Sebelum | Sesudah |
|---|---|
| yaah <u> </u> , udah selesai nih Bjorka Bjorkaan nya ? | yaah <u> </u> , udah selesai nih bjorka bjorkaan nya ? |
| Apa kabar Bjorka? | apa kabar bjorka? |
| Kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :(((| kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :(((|

PEMBERSIHAN NON ALPHANUMERIC

- Alphanumeric merupakan karakter yang terdiri dari huruf dan angka.
- Pembersihan Non-Alphanumeric bertujuan untuk membersihkan karakter selain huruf dan angka, seperti tanda baca dan karakter khusus lainnya.
- Tahapan ini dilakukan agar data yang akan digunakan dalam penelitian dapat bekerja secara maksimal karena dalam membuat model deep learning

| Sebelum | Sesudah |
|--|---|
| yaah <u>_</u> udah selesai <u>nih</u> bjorka <u>bjorkaan</u> nya ? | yaah udah selesai <u>nih</u> bjorka <u>bjorkaan</u> nya |
| apa <u>karab</u> bjorka? | apa kabar bjorka |
| kadang <u>gua</u> <u>mikir</u> <u>sekilas</u> pp <u>gua</u> <u>mirip</u> bjorka, <u>tapi</u> <u>kan</u> <u>gua</u> <u>udh</u> pake pp ini lama | kadang <u>gua</u> <u>mikir</u> <u>sekilas</u> pp <u>gua</u> <u>mirip</u> bjorka <u>tapi</u> <u>kan</u> <u>gua</u> <u>udh</u> pake pp ini lama |
| :(((| |

PEMBERSIHAN KARAKTER KHUSUS

- Pembersihan karakter khusus ini bertujuan untuk menghapus beberapa karakter khusus yang ada di dalam kalimat/teks, seperti kalimat yang mengandung kata "\n", url, serta emoji.
- Dilakukan untuk membuat kualitas data menjadi lebih baik agar model yang dilatih menjadi baik pula dalam mengenali kata-kata dan konteksnya

| Sebelum | Sesudah |
|--|--|
| rame rame pada hapus twit yg menuduh msf adalah sosok bjorka @@ | rame rame pada hapus twit yg menuduh msf adalah sosok bjorka |
| bjorka kembali tebar ancaman, kecam semua akun palsu http://inet.detik.com/security/d-6303238/bjorka-kembali-tebar-ancaman-kecam-semua-akun-palsu?bytedance=true | bjorka kembali tebar ancaman, kecam semua akun palsu |
| bjorka, aku padamu ☺ ❤ | bjorka, aku padamu |

NORMALISASI KATA ALAY

- Bertujuan untuk mengubah kata-kata yang dikategorikan menjadi kata-kata alay menjadi bentuk dasarnya.
- Bertujuan untuk mengurangi variansi kata yang ada di dalam dataset yang akan membuat kualitas dataset menjadi lebih baik dan optimal dalam klasifikasi nantinya.
- Tahapan ini akan menghilangkan kata alay yang sudah terdaftar pada dataset kata-kata alay yang sudah didapatkan melalui Kaggle.

| Sebelum | Sesudah |
|---|---|
| anjirrr kok bisa adek gw percaya kalo bjorka itu gus syamsudin | anjirrr kok bisa adik gue percaya kalau bjorka itu gus syamsudin |
| knp ga ada yg jadi bjorka halloween party kemaren | kenapa tidak ada yang jadi bjorka halloween party kemarin |
| pengen gitu bikin sih bjorka biar gak bikin orang gak gagal fokus rakyat karena saat banyak kasus yg gagal fokus karena hacker bjorka bayaran pemerintah | ingin begitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah |

STEMMING

- Stemming merupakan Teknik menguraikan bentuk suatu kata menjadi bentuk asalnya/dasarnya.
- Hal ini dilakukan untuk mengubah kata berimbahan menjadi bentuk dasarnya agar mengurangi variansi serta meningkatkan kualitas dataset menjadi lebih optimal.

| Sebelum | Sesudah |
|--|---|
| anjirrr kok bisa adik gue percaya kalau bjorka itu gus svamsudin | anjirrr kok bisa adik gue percaya kalo bjorka itu gus svamsudin |
| kenapa tidak ada yang jadi bjorka halloween party kemarin | kenapa tidak ada yang jadi bjorka halloween party kemarin |
| ingin begitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah | ingin gitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayar pemerintah |

STOP WORD REMOVAL

- Tahapan pembersihan stopword berfungsi untuk menghilangkan kata umum (common words) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna.
- Tujuan dari menghilangkan stop words adalah mengurangi jumlah kata dalam sebuah dokumen yang akan berpengaruh terhadap kecepatan dan performa model.
- Stop word yang akan dijadikan sebagai acuan didapatkan dari libraries python Sastrawi (Sastrawi, 2017).

| Sebelum | Sesudah |
|--|---|
| anjirrr kok bisa adik gue percaya kalo bjorka itu gus syamsudin | anjirrr kok bisa adik gue percaya bjorka gus syamsudin |
| kenapa tidak ada yang jadi bjorka halloween party kemarin | bjorka halloween party kemarin |
| ingin gitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah | gitu bikin sih bjorka biar bikin gagal fokus rakyat gagal fokus hacker bjorka bayaran pemerintah |

LABELLING DATA

- Pelabelan data berfungsi untuk memberikan label yang terdiri dari positif, negative maupun netral terhadap dataset.
- Hal ini perlu dilakukan karena model deep learning yang akan dibangun membutuhkan sebuah label untuk dapat berlatih dan mengklasifikasikan ujaran kebencian twitter.
- Untuk melakukan pelabelan terhadap dataset, menggunakan Vader Lexicon. Vader (Valence Aware Dictionary and Sentiment Reasoner) Lexicon merupakan sebuah tools/libraries yang berfungsi untuk melakukan sentiment analysis berbasis aturan yang secara khusus disesuaikan dengan sentiment yang diungkapkan di media sosial



LABELLING DATA

Dengan menggunakan Vader, data teks akan diproses dan dicari nilai sentimennya. Output yang dihasilkan oleh Vader merupakan sebuah probabilitas dari 3 kemungkinan yaitu positif, negative dan netral. Dari probabilitas tersebut, untuk melakukan labelling terhadap suatu teks di dalam dataset, maka diambil nilai compound dari ketiga probabilitas tersebut. Nilai compound adalah metric yang menghitung semua peringkat leksikon yang telah dinormalisasi antara -1 (paling negative) sampai dengan 1(paling positif).

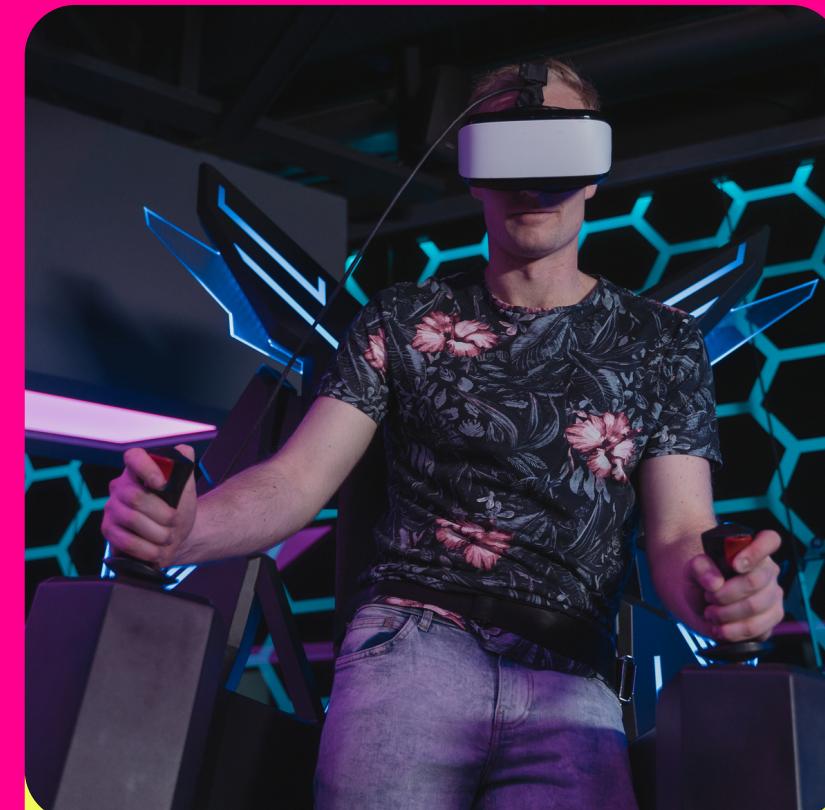
- Positive ($>0,05$)
- Neutral ($>=-0,05$ dan $<=0,05$)
- Negative ($<0,05$)

| Teks | Vader Output | Label |
|---|---|----------|
| Bjorka is fun | {'neg': 0.0, 'neu': 0.694, 'pos': 0.306, 'compound': 0.296} | Positive |
| Bjorka please leak the secret document of suspicious plot funds | {'neg': 0.189, 'neu': 0.775, 'pos': 0.036, 'compound': -0.6728} | Negative |
| Talk to Bjorka Drinking Coffee Together | {'neg': 0.0, 'neu': 0.954, 'pos': 0.046, 'compound': 0.0387} | Neutral |

PEMISAHAN DATASET

- Memisahkan data menjadi data latih dan data uji
- Rasinya latih dan uji adalah 0,8 : 0,2
- Data label ditransformasi dari label satuan menjadi array yang berisi 3 kemungkinan (ex. [1,0,0]).

| Label | Label One Hot Encoder |
|-------|-----------------------|
| 0 | [1,0,0] |
| 1 | [0,1,0] |
| 2 | [0,0,1] |



TOKENIZING

- Tahapan tokenizing bertujuan untuk mengubah kata menjadi sebuah token (angka) .
 - Terdapat sekitar 17.586 Variasi Kata
 - Tahapan ini akan menggunakan libraries pada python. Pertama, akan membuat sebuah index kata dan kemudian mentransformasikan kata menjadi index-index tersebut.
 - Menyamakan panjang dari kata dengan memberikan token tambahan di akhir kalimat

SKENARIO PENGUJIAN

- Preprocessing dataset
- Pembagian dataset menjadi data train dan test
- Membuat beberapa skenario pengujian, yang dibagi menjadi dua bagian besar yaitu skenario sederhana dan kompleks dengan data dan perlakuan model yang sama
- Skenario sederhana:
 - Arsitektur model
 - Activation Function
 - Optimizer
- Skenario Kompleks
 - Arsitektur model
 - Optimizer





BAB IV

IMPLEMENTASI, ANALISA

DAN PEMBAHASAN

SENTIMENT ANALYSIS UJARAN KEBENCIAN
TWITTER MENGGUNAKAN LSTM DAN
BIDIRECTIONAL LSTM

Adrianus Charlie Hadirria Agelsadewa
195314174



PENGUMPULAN DATA

- Data tweet yang dicari menggunakan keyword "Selena Gomez"
- Data tweet yang diambil di antara tanggal 1-2 Maret 2023
- Harus menggunakan Twitter Developer Account terlebih dahulu
- Dikumpulkan data sebanyak 11.646 tweets
- Menggunakan libraries tweepy
- Disimpan menjadi file excel

```
query='selena gomez' lang:en -has:links -is:replies'
tweets=getTweets(api=api,query=query,items=10000)
tweets
```

| | created_at | username | text |
|-------|--------------------------------|-----------------|---|
| 5 | Thu Mar 02 23:59:30 +0000 2023 | darkonsun | social media is a disease and this hailey bieb... |
| 25 | Thu Mar 02 23:56:18 +0000 2023 | babyyannieee | Justin Bieber & Selena Gomez followed each... |
| 28 | Thu Mar 02 23:55:57 +0000 2023 | likabeast101 | Selena Gomez drove by my house 🎉 |
| 29 | Thu Mar 02 23:55:36 +0000 2023 | whoismelanway | I understand hailey Bieber because if my husba... |
| 32 | Thu Mar 02 23:54:50 +0000 2023 | _mayyraa | but then i always realize because SELENA GOMEZ... |
| ... | ... | ... | ... |
| 11634 | Wed Mar 01 10:19:04 +0000 2023 | SAFFAOnAirRadio | @SAFFAOnAirRadio is now playing, Selena Gomez,... |
| 11635 | Wed Mar 01 10:19:01 +0000 2023 | tshepiixx | Go follow Selena Gomez on IG 😊 |
| 11638 | Wed Mar 01 10:17:59 +0000 2023 | Chief_I_H8_U | @Ricius496 @endlessbonerz Selena Gomez has a m... |
| 11642 | Wed Mar 01 10:15:42 +0000 2023 | vinalraswati | I still can taste it and i hate it. \n\nPeople... |
| 11644 | Wed Mar 01 10:13:24 +0000 2023 | TheSwaddle | Over the past few days, the alleged feud betwe... |

```
def getTweets(api,query,items):
    tweets=[]
    for tweet in tweepy.Cursor(api.search_30_day,'skripsi30day',fromDate='202303040000',toDate='202303050000',
                               query=query,maxResults=100).items(items):
        tweets.append(tweet)
    return tweets
```

PREPROCESSING

Tahapan preprocessing merupakan tahapan untuk membuat kualitas data yang akan digunakan menjadi optimal.

1. Selecting Data

```
data = data[data["text"].str.contains("RT") == False]  
data
```

2. Lowercasing

```
#Lower text  
def lowercase(text):  
    return text.lower()
```



social media is a disease and this hailey bieb...
justin bieber & selena gomez followed each...
selena gomez drove by my house 🌟
i understand hailey bieber because if my husba...
but then i always realize because selena gomez...
...
@saffaonairradio is now playing, selena gomez,...
go follow selena gomez on ig 😊
@ricius496 @endlessbonerz selena gomez has a m...
i still can taste it and i hate it. \n\people...
over the past few days, the alleged feud betwe...

```
#Remove non alphanumeric character  
def removeNonAlphanumeric(text):  
    text = re.sub('[^0-9a-zA-Z]+', ' ', text)  
    return text
```



3. Remove Unnecessary

```
#Remove unnecessary characters  
def removeUnnecessary(text):  
    text=re.sub('\n', '', text) # menghapus newline  
    text=re.sub('rt', '', text) # menghapus kata retweet  
    text=re.sub('user', '', text) # menghapus kata user  
    text=result = re.sub(r"http\S+", "", text)  
    text = re.sub(' +', ' ', text) # menghapus karakter ekstra  
    text=re.sub('x.{3} | x.{2}', '', text) # menghapus pola emoji  
    return text
```



social media is a disease and this hailey bieb...
justin bieber amp selena gomez followed each o...
selena gomez drove by my house
i understand hailey bieber because if my husba...
but then i always realize because selena gomez...
...
saffaonairradio is now playing selena gomez m...
go follow selena gomez on ig
ricius endlessbonerz selena gomez has a medic...
i still can taste it and i hate it people you ...
over the past few days the alleged feud betwee...

social media is a disease and this hailey bieb...
justin bieber amp selena gomez followed each o...
selena gomez drove by my house
i understand hailey bieber because if my husba...
but then i always realize because selena gomez...
...
saffaonairradio is now playing selena gomez m...
go follow selena gomez on ig
ricius endlessbonerz selena gomez has a medic...
i still can taste it and i hate it people you ...
over the past few days the alleged feud betwee...

4. Remove Non Alphanumeric

PREPROCESSING

Tahapan preprocessing merupakan tahapan untuk membuat kualitas data yang akan digunakan menjadi optimal.

5. Stopword Removal

```
def removeStopWord(text):
    stop_words = set(stopwords.words("english"))
    word_tokens = word_tokenize(text)
    filtered_text = [word for word in word_tokens if word not in stop_words]
    return ' '.join(filtered_text)
```



```
social media disease hailey bieber selena gome...
justin bieber amp selena gomez followed twitte...
selena gomez drove house
understand hailey bieber husbands ex selena go...
always realize selena gomez always looks good
...
saffaonairradio playing selena gomez marshmell...
go follow selena gomez ig
ricius endlessbonerz selena gomez medical issu...
still taste hate people know selena gomez
past days alleged feud selena gomez hailey bie...
```

6. Lemmatization

```
def lemmatizeWord(text):
    word_tokens = word_tokenize(text)
    # provide context i.e. part-of-speech
    lemmas = [lemmatizer.lemmatize(word, pos ='v') for word in word_tokens]
    return ' '.join(lemmas)
```



```
social media diseas hailey bieber selena gomez...
justin bieber amp selena gomez follow twitter ...
selena gomez drive hous
understand hailey bieber husband ex selena gom...
alway realiz selena gomez alway look good
...
saffaonairradio play selena gomez marshmello w...
go follow selena gomez ig
riciu endlessbonerz selena gomez medic issu st...
still tast hate peopl know selena gomez
past day alleg feud selena gomez hailey bieber...
```

LABELLING

- Tahapan ini berfungsi untuk memberikan label pada data tweets
- Menggunakan vader sentiment analysis
- Mempunyai 3 kutub kecenderungan
- Menggunakan vader sentiment analysis

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        return "Positive"

    elif sentiment_dict['compound'] <= - 0.05 :
        return "Negative"

    else :
        return "Neutral"
```



| | created_at | username | text | preprocessed | sentiment |
|-------|--------------------------------|------------------|--|---|-----------|
| 5 | Thu Mar 02 23:59:30 +0000 2023 | darkonsun | social media is a disease and this hailey bieb... | social media diseas hailey bieber selena gomez... | Neutral |
| 25 | Thu Mar 02 23:56:18 +0000 2023 | babyyanniee | Justin Bieber & Selena Gomez followed each... | justin bieber amp selena gomez follow twitter ... | Neutral |
| 28 | Thu Mar 02 23:55:57 +0000 2023 | likabeast101 | Selena Gomez drove by my house 🎉 | selena gomez drive hous | Neutral |
| 29 | Thu Mar 02 23:55:36 +0000 2023 | whoismelanway | I understand hailey Bieber because if my husba... | understand hailey bieber husband ex selena gom... | Neutral |
| 32 | Thu Mar 02 23:54:50 +0000 2023 | _mayyraa | but then i always realize because SELENA GOMEZ... | alway realiz selena gomez alway look good | Positive |
| ... | ... | ... | ... | ... | ... |
| 11634 | Wed Mar 01 10:19:04 +0000 2023 | SAFFAOOnAirRadio | @SAFFAOOnAirRadio is now playing. Selena Gomez,... | saffaonairradio play selena gomez marshmello w... | Positive |
| 11635 | Wed Mar 01 10:19:01 +0000 2023 | tshepiixx | Go follow Selena Gomez on IG 😊 | go follow selena gomez ig | Neutral |
| 11638 | Wed Mar 01 10:17:59 +0000 2023 | Chief_I_H8_U | @Ricius496 @endlessbonerz Selena Gomez has a m... | riciu endlessbonerz selena gomez medic issu st... | Neutral |
| 11642 | Wed Mar 01 10:15:42 +0000 2023 | vinalraswati | I still can taste it and i hate it. \n\nPeople... | still tast hate peopl know selena gomez | Negative |
| 11644 | Wed Mar 01 10:13:24 +0000 2023 | TheSwaddle | Over the past few days, the alleged feud betwe... | past day alleg feud selena gomez hailey bieber... | Negative |

TOKENIZING

- Mengubah data berupa kalimat menjadi kumpulan token
- panjang token maksimal adalah 15 kata
- Mengisi sisa element yang kosong dengan token 0
- Kata setelah kata urutan ke 15 akan dipotong

```
sequences=tokenizer.texts_to_sequences(data['preprocessed'])
padded=pad_sequences(sequences=sequences,padding=padding_type,truncating=trunc_type,maxlen=max_len)
temp=0
kamus={}
for item in sequences:
    coba=len(item)
    if coba not in kamus:
        kamus[coba]=1
    else:
        kamus[coba]+=1
```



```
max_len=15
trunc_type = 'post'
padding_type = 'post'
oov_tok = "<OOV>"
tokenizer=Tokenizer(oov_token=oov_tok)
tokenizer.fit_on_texts(data['preprocessed'])
```



```
array([ 183,  226, 1686,      7,      6,      2,      3,   219,   285,      0,      0,
       0,      0,      0,      0])
```

SPLITTING DATA

- Memisahkan data menjadi data latih dan data uji
- Memisahkan data dengan rasio 0.9 untuk data training dan 0.1 untuk data test
- Mengubah label menjadi *one hot encoded* label

```
y_train= tf.keras.utils.to_categorical(y_train, num_classes=3)
y_test= tf.keras.utils.to_categorical(y_test, num_classes=3)
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

✓ 0.0s

```
((16000, 20), (4000, 20), (16000, 3), (4000, 3))
```

MODELLING SEDERHANA

- Menyusun model deep learning pada skenario sederhana
- Membuat function untuk membuat model deep learning berdasarkan beberapa skenario
- Menyusun callback function early stopping
- Membuat list parameter untuk skenario sederhana
- Skenario ini berfokus pada arsitektur model, activation function dan optimizer dari model.

```
def createSimpleModel(skenario):  
    if skenario ==1:  
        model=tf.keras.models.Sequential([  
            tf.keras.layers.Embedding(len(tokenizer.word_index),16,input_length=max_len),  
            tf.keras.layers.LSTM(64,return_sequences=True,activation='tanh'),  
            tf.keras.layers.Flatten(),  
            tf.keras.layers.Dense(3, activation='softmax')  
        ])  
        model.compile(loss='categorical_crossentropy',optimizer=tf.keras.optimizers.RMSprop(),metrics=['accuracy',f1_m])  
    else:  
        model=tf.keras.models.Sequential([  
            tf.keras.layers.Embedding(len(tokenizer.word_index),1228,input_length=max_len),  
            tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64,return_sequences=True,activation='tanh')),  
            tf.keras.layers.Flatten(),  
            tf.keras.layers.Dense(3,activation='softmax')  
        ])  
        model.compile(loss='categorical_crossentropy',optimizer=tf.keras.optimizers.Adam(),metrics=['accuracy',f1_m])  
    model.summary()  
    return model
```

```
def createSimpleModel(activation):  
    model=tf.keras.models.Sequential([  
        tf.keras.layers.Embedding(len(tokenizer.word_index),16,input_length=max_len),  
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64,return_sequences=True,activation=activation['hidden'])),  
        tf.keras.layers.Flatten(),  
        tf.keras.layers.Dense(3, activation=activation['output'])  
    ])  
    model.compile(loss='categorical_crossentropy',optimizer=tf.keras.optimizers.Adam(),metrics=['accuracy',f1_m])  
    model.summary()  
    return model
```

```
result_complex=[]  
models_complex=[]  
for i in range(1,3):  
    model=createComplexModel(i)  
    history=model.fit(x_train,y_train,epochs=20,validation_split=0.1,callbacks=[callback])  
    models_complex.append(model)  
    result_complex.append(history)
```

MODELLING SEDERHANA

- Menyusun model deep learning pada skenario sederhana
- Membuat function untuk membuat model deep learning berdasarkan beberapa skenario
- Menyusun callback function early stopping
- Membuat list parameter untuk skenario sederhana
- Skenario ini berfokus pada arsitektur model, activation function dan optimizer dari model.

```
activation=[{'hidden':'relu','output':'sigmoid'},{'hidden':'relu','output':'softmax'},{'hidden':'sigmoid','output':'sigmoid'},{'hidden':'sigmoid','output':'softmax'},{'hidden':'tanh','output':'sigmoid'},{'hidden':'tanh','output':'softmax'}]
result=[]
models=[]
for item in activation:
    print(item)
    model=createSimpleModel(activation=item)
    history=model.fit(x_train,y_train,epochs=20,validation_split=0.1,callbacks=[callback])
    models.append(model)
    result.append(history)
```

```
def createSimpleModel(optimizer):
    model=tf.keras.models.Sequential([
        tf.keras.layers.Embedding(len(tokenizer.word_index),16,input_length=max_len),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64,return_sequences=True,activation='relu')),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(3, activation='softmax')
    ])
    model.compile(loss='categorical_crossentropy',optimizer=optimizer,metrics=['accuracy',f1_m])
    model.summary()
    return model
```

```
optimizer=['adam','rmsprop']
result=[]
models=[]
for item in optimizer:
    print('Train model with {} optimizer'.format(item))
    model=createSimpleModel(optimizer=item)
    history=model.fit(x_train,y_train,epochs=50,validation_split=0.1)
    models.append(model)
    result.append(history)
```

MODELLING KOMPLEKS

- Menyusun model deep learning pada skenario Kompleks
- Membuat function untuk membuat model deep learning berdasarkan beberapa skenario
- Menyusun callback function early stopping
- Membuat list parameter untuk skenario sederhana
- Skenario ini berfokus pada arsitektur model dan optimizer

```
def createComplexModel(skenario):
    if skenario ==1:
        model=tf.keras.models.Sequential([
            tf.keras.layers.Embedding(len(tokenizer.word_index),128,input_length=max_len),
            tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2),
            tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2),
            tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(512,activation='relu'),
            tf.keras.layers.Dropout(0.5),
            tf.keras.layers.Dense(3, activation='softmax')
        ])
        model.compile(loss='mse',optimizer=tf.keras.optimizers.RMSprop(),metrics=['accuracy',f1_m])
    else:
        model=tf.keras.models.Sequential([
            tf.keras.layers.Embedding(len(tokenizer.word_index),128,input_length=max_len),
            tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2)),
            tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2)),
            tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='tanh',dropout=0.2)),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(512,activation='relu'),
            tf.keras.layers.Dropout(0.5),
            tf.keras.layers.Dense(3,activation='softmax')
        ])
        model.compile(loss='mse',optimizer=tf.keras.optimizers.RMSprop(),metrics=['accuracy',f1_m])
    model.summary()
    return model
```

```
result=[]
models=[]
for i in range(1,3):
    model=createComplexModel(i)
    history=model.fit(x_train,y_train,epochs=20,validation_split=0.1,callbacks=[callback])
    models.append(model)
    result.append(history)
```

MODELLING KOMPLEKS

- Menyusun model deep learning pada skenario Kompleks
- Membuat function untuk membuat model deep learning berdasarkan beberapa skenario
- Menyusun callback function early stopping
- Membuat list parameter untuk skenario sederhana
- Skenario ini berfokus pada arsitektur model dan optimizer

```
def createComplexModel(optimizer):
    model=tf.keras.models.Sequential([
        tf.keras.layers.Embedding(len(tokenizer.word_index),128,input_length=max_len),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='relu',dropout=0.2)),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='relu',dropout=0.2)),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128,return_sequences=True,activation='relu',dropout=0.2)),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(512,activation='relu'),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(3,activation='softmax')
    ])
    model.compile(loss='mse',optimizer=optimizer,metrics=['accuracy',f1_m])
    model.summary()
    return model
```

```
optimizer=[ 'adam', 'rmsprop' ]
result=[]
models=[]
for item in optimizer:
    print('Train model with {} optimizer'.format(item))
    model=createComplexModel(optimizer=item)
    history=model.fit(x_train,y_train,epochs=50,validation_split=0.1,callbacks=[callback])
    models.append(model)
    result.append(history)
```

```
callback = tf.keras.callbacks.EarlyStopping(
    monitor="val_accuracy",
    min_delta=0,
    patience=5,
    verbose=1,
    mode="auto",
    baseline=None,
    restore_best_weights=True,
```

EVALUASI

- Model akan dievaluasi dengan tiga metrics evaluation diantaranya:
 - Loss
 - Akurasi
 - F1 score
- Model akan dievaluasi menggunakan data uji yang telah dipisahkan
- Akan divisualisasikan

```
for item in models:  
    temp=item.evaluate(x_test,y_test)  
    temp  
  
17/17 [=====] - 0s 15ms/step - loss: 0.0572 - accuracy: 0.8887 - f1_m: 0.8929  
17/17 [=====] - 0s 28ms/step - loss: 0.0504 - accuracy: 0.9098 - f1_m: 0.9113
```

```
def compareAccuracyArchitecture(result):  
    plt.plot(result[0].history['accuracy'],color='r',label='LSTM accuracy')  
    plt.plot(result[1].history['accuracy'],color='g',label='BiLSTM accuracy')  
    plt.plot(result[0].history['val_accuracy'],color='y',label='LSTM Validation accuracy')  
    plt.plot(result[1].history['val_accuracy'],color='b',label='BiLSTM Validation accuracy')  
    plt.xlabel("Epochs")  
    plt.ylabel("Nilai")  
    plt.title("Akurasi LSTM dan BiLSTM")  
    plt.legend()  
    plt.show()  
compareAccuracyArchitecture(result)  
def comparef1scoreArchitecture(result):  
    plt.plot(result[0].history['f1_m'],color='r',label='LSTM F1 Score')  
    plt.plot(result[1].history['f1_m'],color='g',label='BiLSTM F1 Score')  
    plt.plot(result[0].history['val_f1_m'],color='y',label='LSTM Validation F1 Score')  
    plt.plot(result[1].history['val_f1_m'],color='b',label='BiLSTM Validation F1 Score')  
    plt.xlabel("Epochs")  
    plt.ylabel("Nilai")  
    plt.title("F1 Score LSTM dan BiLSTM")  
    plt.legend()  
    plt.show()  
comparef1scoreArchitecture(result)  
def compareLossArchitecture(result):  
    plt.plot(result[0].history['loss'],color='r',label='LSTM Loss')  
    plt.plot(result[1].history['loss'],color='g',label='BiLSTM Loss')  
    plt.plot(result[0].history['val_loss'],color='y',label='LSTM Validation Loss')  
    plt.plot(result[1].history['val_loss'],color='b',label='BiLSTM Validation Loss')  
    plt.xlabel("Epochs")  
    plt.ylabel("Nilai")  
    plt.title("Loss LSTM dan BiLSTM")  
    plt.legend()  
    plt.show()  
compareLossArchitecture(result)
```

ANALISA DAN PEMBAHASAN

Analisa dan pembahasan akan berfokus pada

- Skenario pengujian sederhana dan kompleks
- Hyper parameter tuning pada setiap skenario pengujian
- Model telah dilatih dengan 20 epochs, dengan validation split sebesar 0.1 dengan callback function early stopping



CONTOH INPUT OUTPUT TIAP LAYER

Data Input

```
array([1120, 2973, 1430,    96,   518, 1178,   125,     2,     3,   137,   135,
       489,   493,     0,     0])
```

Embedding Layer

Output Embedding

```
(array([[[ -1.85020231e-02,   1.21905589e-02,   6.10311106e-02,
          -1.94835272e-02,  -8.83538797e-02,  -4.75622807e-03,
          3.10477186e-02,  -7.07646715e-04,   9.96988639e-03,
         -1.52977491e-02,   1.93516742e-02,   1.18370874e-04,
         3.59906629e-02,   4.03568372e-02,  -2.42998861e-02,
        -4.68846224e-02],
       [-1.91136599e-02,  -1.93392765e-02,  -3.57645750e-03,
        2.34253667e-02,  -3.85785922e-02,  -2.11429000e-02,
       -4.47412394e-02,  -1.42867491e-03,   4.34010066e-02,
       -1.67119876e-02,   5.36322594e-03,  -1.62796378e-02,
       6.15905970e-04,  -4.47611921e-02,   1.88140310e-02,
      4.81216647e-02],
      [-5.26615046e-02,   3.31416093e-02,   1.28477737e-02,
       2.85221264e-02,   9.16281864e-02,   1.69648863e-02,
      -6.30381480e-02,   1.39041971e-02,   1.03058936e-02,
       6.50004968e-02,   6.98561743e-02,   6.68175742e-02,
       2.77879089e-02,  -7.43802264e-02,  -3.43307649e-04,
      -3.93239670e-02],
      [ 4.04064283e-02,   1.84509102e-02,   6.96756467e-02,
       1.25931231e-02,  -6.49561360e-02,   4.76747714e-02,
       4.15527821e-02,  -2.76437663e-02,  -4.97472705e-03,
      -1.84718799e-02,   7.08500203e-03,  -3.81316878e-02,
      -6.90287650e-02,   3.49530205e-02,  -3.04530375e-03,
      -2.29139123e-02],
      [ 3.16500925e-02,   8.61200541e-02,   2.34175324e-02,
       5.32880351e-02,  -5.35190031e-02,  -4.33027819e-02,
      -7.01153874e-02,  -1.51166518e-03,   1.93319276e-01,
      -9.16124284e-02,   6.21530674e-02,   4.50230055e-02,
     -7.95278326e-02]], dtype=float32),
(1, 15, 16))
```

CONTOH INPUT OUTPUT TIAP LAYER

Data Input

```
(array([[[ -1.85020231e-02,  1.21905589e-02,  6.10311106e-02,
       -1.94835272e-02, -8.83538797e-02, -4.75622807e-03,
        3.10477186e-02, -7.07646715e-04,  9.96988639e-03,
       -1.52977491e-02,  1.93516742e-02,  1.18370874e-04,
        3.59906629e-02,  4.03568372e-02, -2.42998861e-02,
       -4.68846224e-02],
      [-1.91136599e-02, -1.93392765e-02, -3.57645750e-03,
       2.34253667e-02, -3.85785922e-02, -2.11429000e-02,
      -4.47412394e-02, -1.42867491e-03,  4.34010066e-02,
      -1.67119876e-02,  5.36322594e-03, -1.62796378e-02,
       6.15905970e-04, -4.47611921e-02,  1.88140310e-02,
       4.81216647e-02],
      [-5.26615046e-02,  3.31416093e-02,  1.28477737e-02,
       2.85221264e-02,  9.16281864e-02,  1.69648863e-02,
      -6.30381480e-02,  1.39041971e-02,  1.03058936e-02,
       6.50004968e-02,  6.98561743e-02,  6.68175742e-02,
      -2.77879089e-02, -7.43802264e-02, -3.43307649e-04,
      -3.93239670e-02],
      [ 4.04064283e-02,  1.84509102e-02,  6.96756467e-02,
       1.25931231e-02, -6.49561360e-02,  4.76747714e-02,
      -4.15527821e-02, -2.76437663e-02, -4.97472705e-03,
      -1.84718799e-02,  7.08500203e-03, -3.81316878e-02,
      -6.90287650e-02,  3.49530205e-02, -3.04530375e-03,
      -2.29139123e-02],
      [ 3.16500925e-02,  8.61200541e-02,  2.34175324e-02,
       ...
       5.32880351e-02, -5.35190031e-02, -4.33027819e-02,
      -7.01153874e-02, -1.51166518e-03,  1.93319276e-01,
      -9.16124284e-02,  6.21530674e-02,  4.50230055e-02,
      -7.95278326e-02]]], dtype=float32),
(1, 15, 16))
```

LSTM Layer

Output LSTM

```
(array([[-0.48588306, -0.23743902,  0.25821865,  0.10533518,  0.11363746,
       -0.35054055,  0.19668896,  0.42732814,  0.2516474 , -0.17203362,
        0.52830374, -0.29109183,  0.07346579,  0.22638859,  0.49995556,
       0.3923472 ]], dtype=float32),
(1, 16))
```

CONTOH INPUT OUTPUT TIAP LAYER

Data Input

```
(array([[-0.48588306, -0.23743902,  0.25821865,  0.10533518,  0.11363746,
       -0.35054055,  0.19668896,  0.42732814,  0.2516474 , -0.17203362,
        0.52830374, -0.29109183,  0.07346579,  0.22638859,  0.49995556,
        0.3923472 ]], dtype=float32),
 (1, 16))
```

Dense Layer

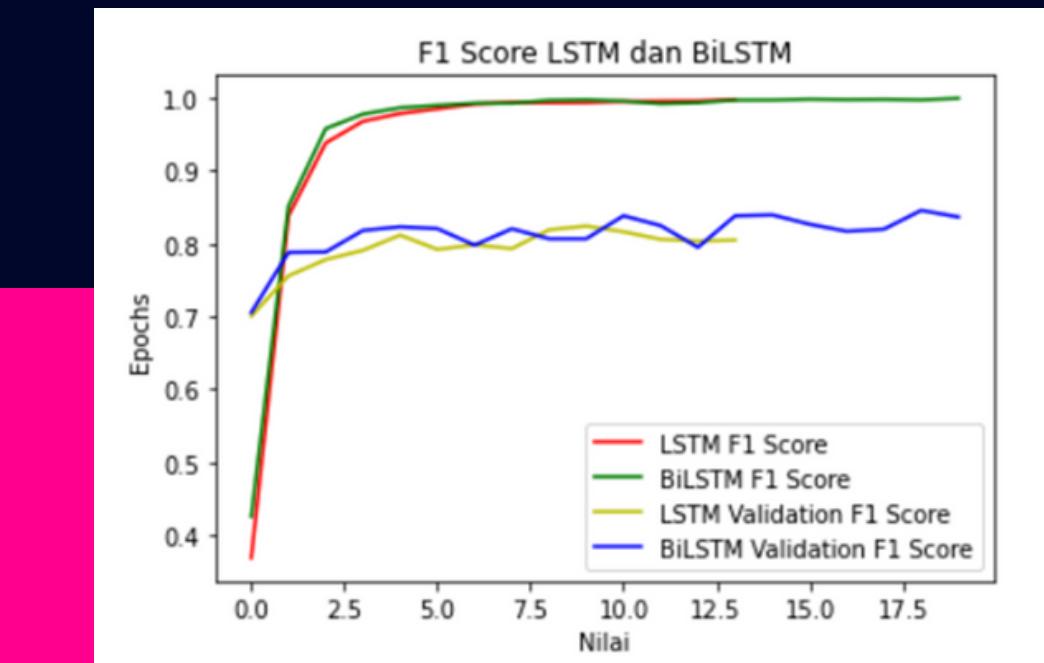
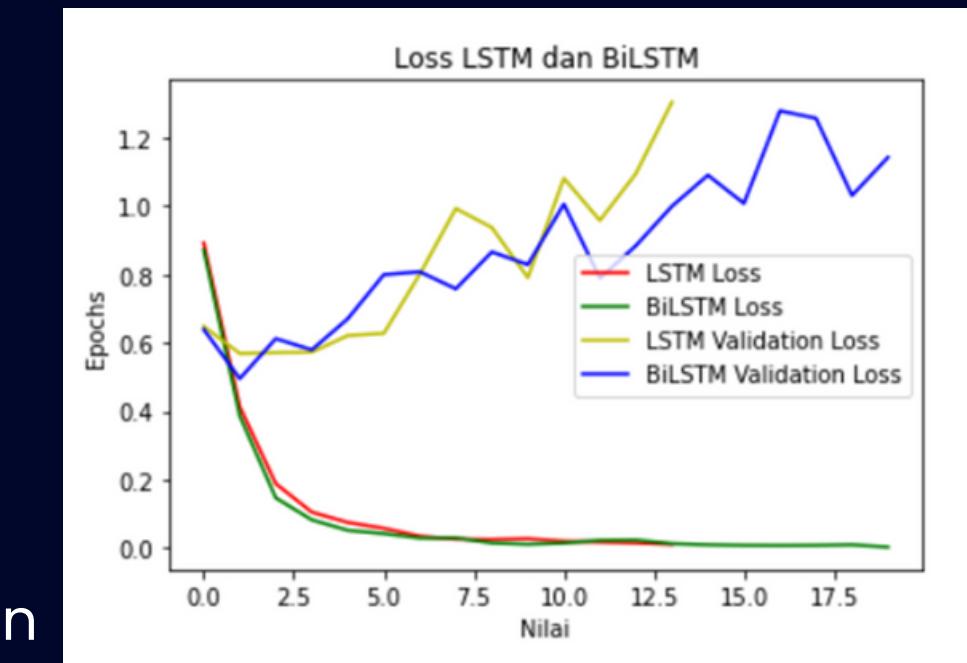
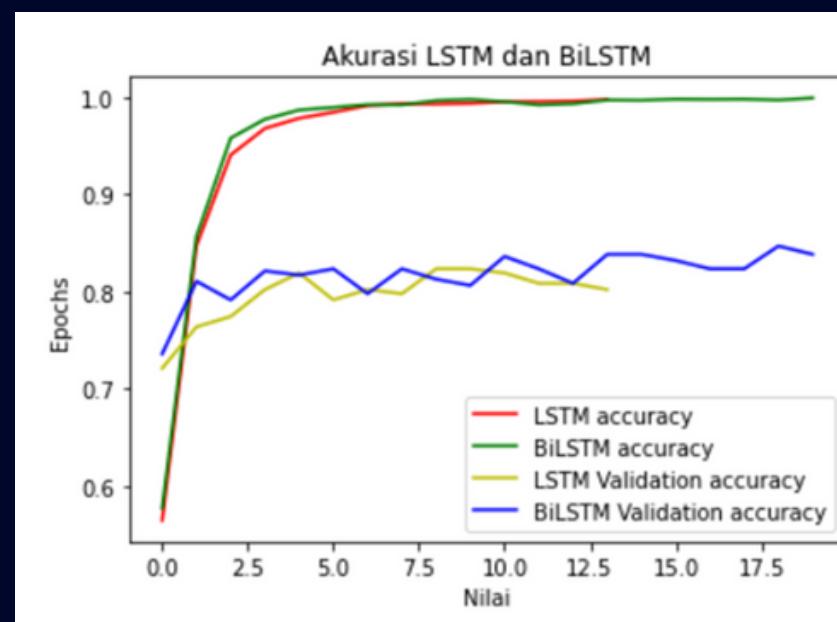
```
(array([[0.01305054, 0.01796896, 0.96898055]], dtype=float32), (1, 3))
```

Output Dense



SKENARIO SEDERHANA: ARSITEKTUR MODEL

- Model BiLSTM lebih unggul dibandingkan dengan model LSTM dari segi akurasi, loss dan f1 score
- Hal ini dikarenakan BiLSTM dapat bekerja secara dua arah untuk mengetahui konteks kalimat secara lebih luar
- Model mengalami overfitting

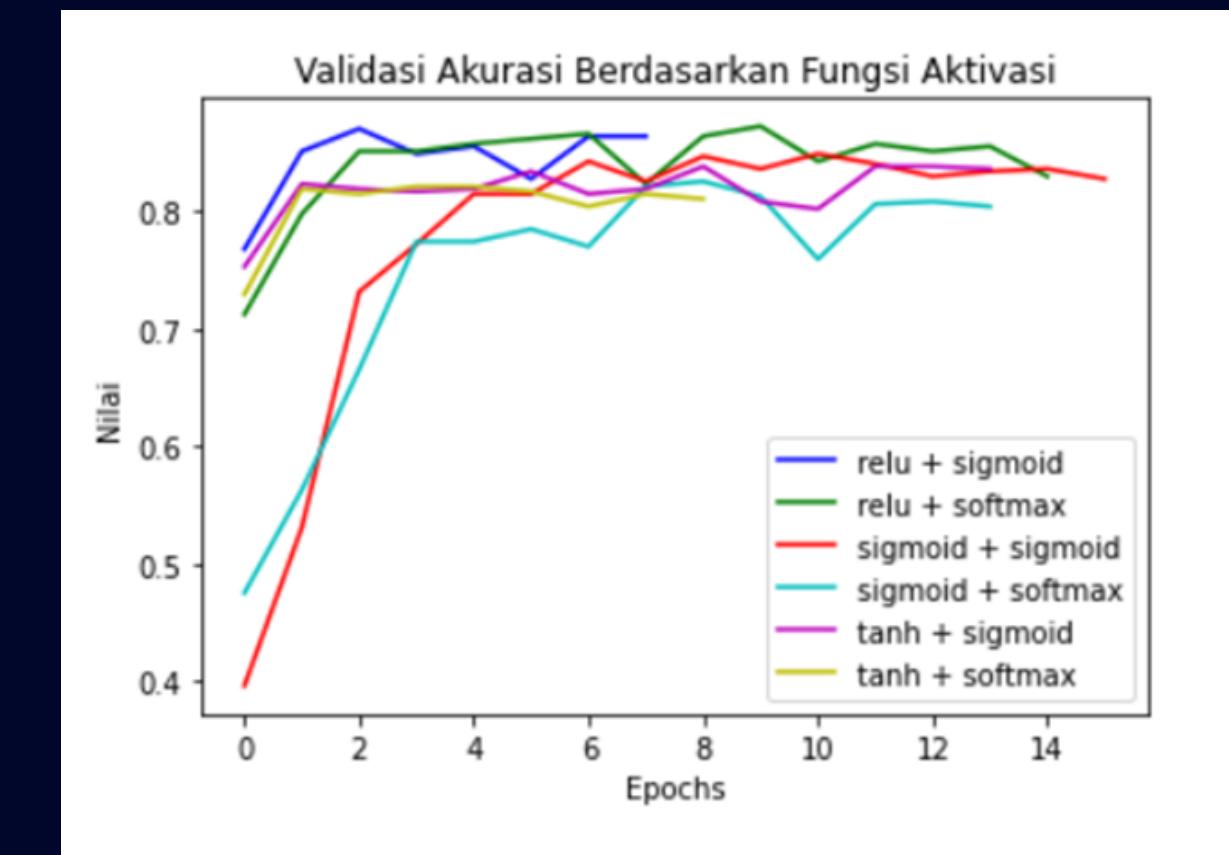
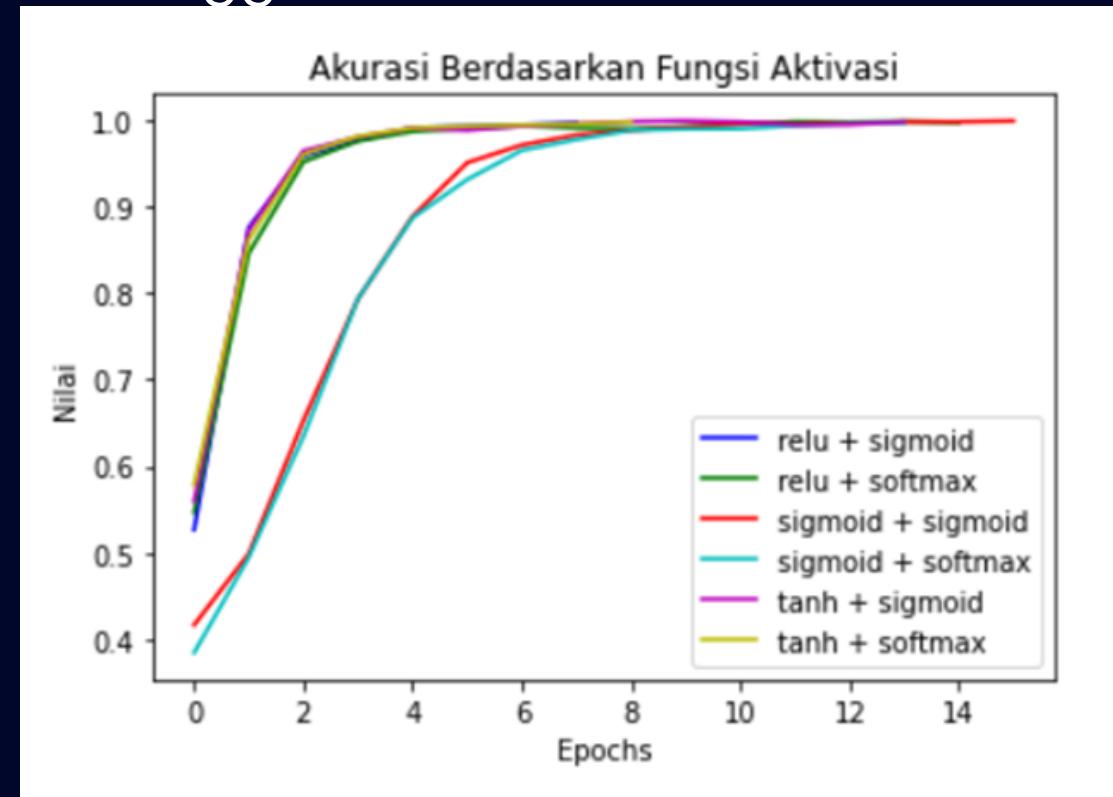


| No | Layer/Aktifasi | Optimizer | Fungsi loss | Early stopping | Loss | Accuracy | F1 Score |
|----|--|-----------|-------------|----------------|--------|----------|----------|
| 1 | <ul style="list-style-type: none"> • Embedding (16) • LSTM (64/tanh) • Flatten • Dense (3/softmax) | Adam | Categorical | 9 | 0.5386 | 0.8157 | 0.8007 |
| 2 | <ul style="list-style-type: none"> • Embedding (16) • BiLSTM(64/tanh) • Flatten • Dense(3/softmax) | Adam | Categorical | 8 | 0.4712 | 0.8522 | 0.8427 |

Tabel 4. 2 Hasil evaluasi training perbandingan arsitektur *layer* model.

SKENARIO SEDERHANA: FUNGSI AKTIVASI LAYER

- Activation function relu pada hidden layer dan softmax pada output layer mendapatkan akurasi tertinggi dalam skenario sederhana dalam mencari fungsi aktivasi terbaik
- Fungsi aktivasi softmax memberikan akurasi yang baik karena softmax cocok untuk klasifikasi dengan label multiclass
- Relu dapat mempercepat konvergensi stochastic gradient descent dimana gradien yang dimiliki maksimum satu dan nilai dari relu berkisar dari nol sampai tak hingga

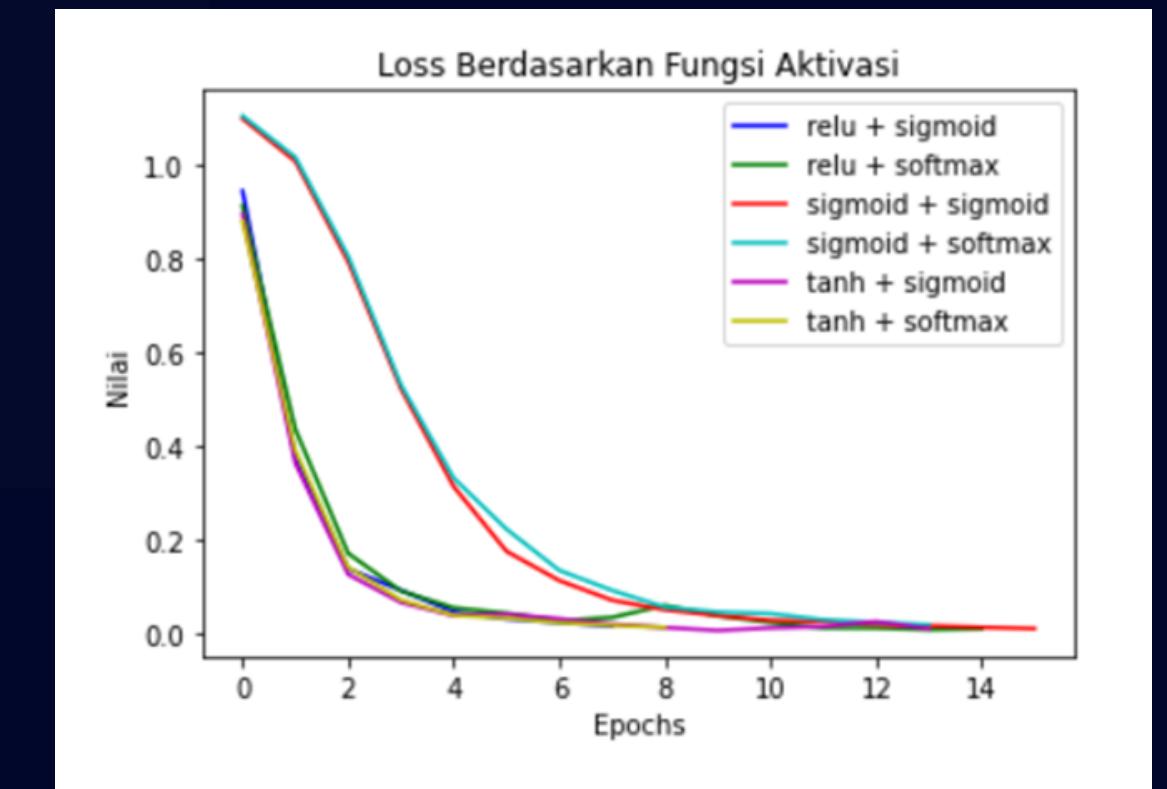
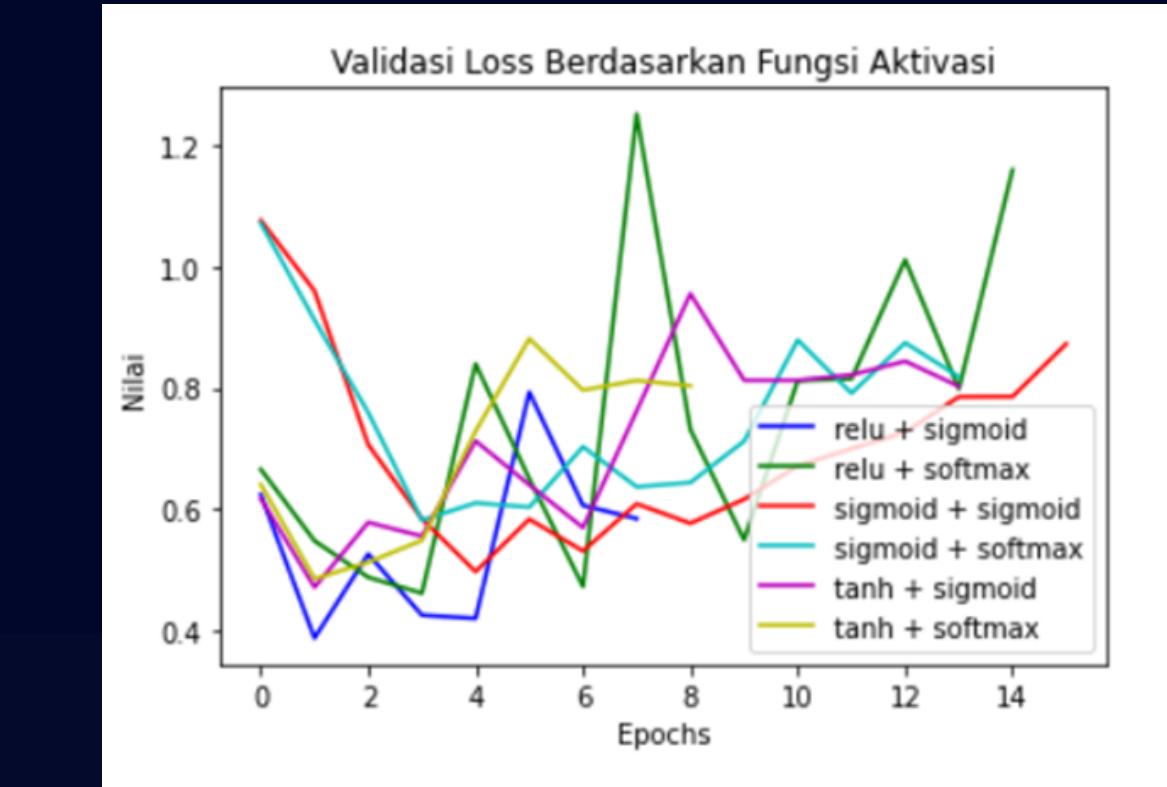
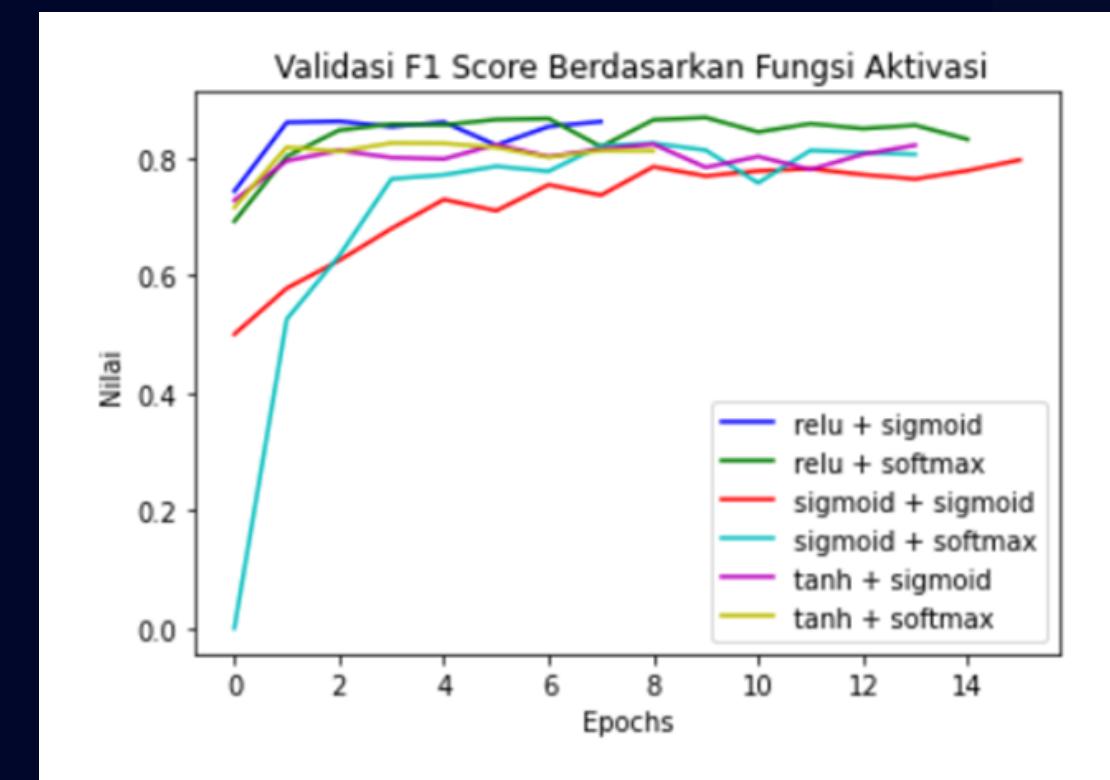
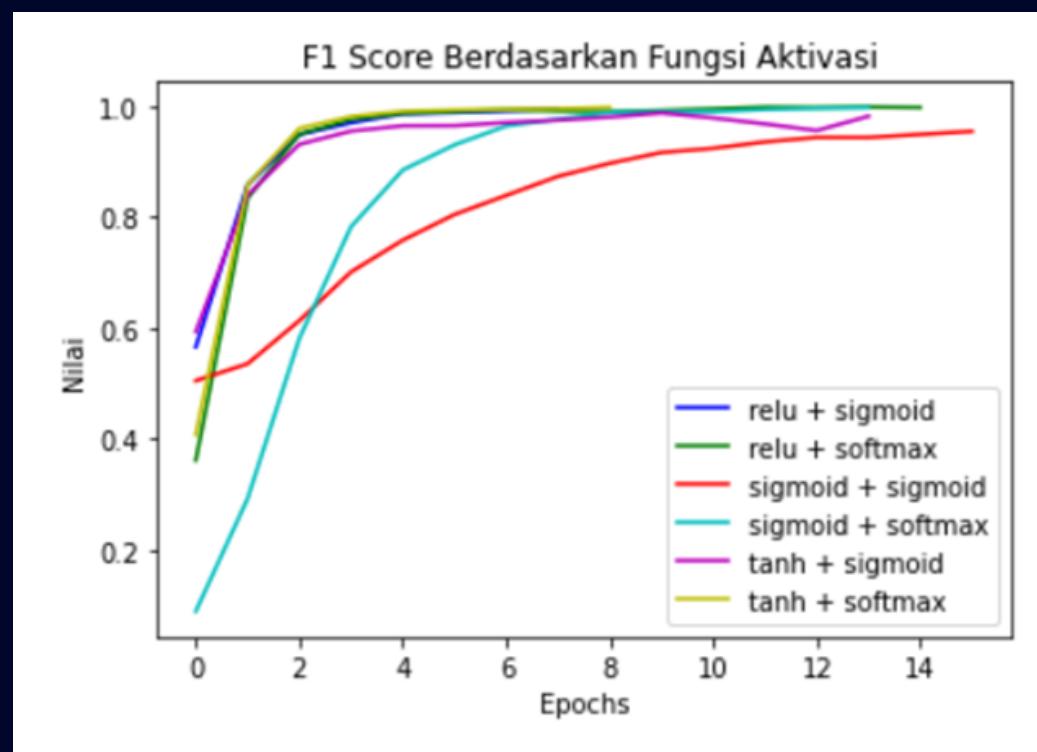


| No | Kombinasi Aktifasi | Optimizer | Fungsi loss | Early stopping | Loss | Accuracy | F1 Score |
|----|--|-----------|-------------|----------------|--------|----------|----------|
| 1 | <ul style="list-style-type: none"> • Relu (hidden) • Sigmoid (output) | Adam | Categorical | 8 | 0.5686 | 0.8330 | 0.8228 |
| 2 | <ul style="list-style-type: none"> • Relu (hidden) • Softmax (output) | Adam | Categorical | 11 | 0.6973 | 0.8522 | 0.8591 |
| 3 | <ul style="list-style-type: none"> • Sigmoid (hidden) • Sigmoid (output) | Adam | Categorical | 16 | 0.7553 | 0.8311 | 0.7964 |
| 4 | <ul style="list-style-type: none"> • Sigmoid (hidden) • Softmax (output) | Adam | Categorical | 12 | 0.5828 | 0.8369 | 0.8394 |
| 5 | <ul style="list-style-type: none"> • Tanh (hidden) • Sigmoid (output) | Adam | Categorical | 8 | 0.5911 | 0.8349 | 0.8233 |
| 6 | <ul style="list-style-type: none"> • Tanh (hidden) • Softmax (output) | Adam | Categorical | 9 | 0.7797 | 0.8138 | 0.8134 |

Tabel 4. 3 Hasil evaluasi training perbandingan fungsi aktifasi menggunakan layer

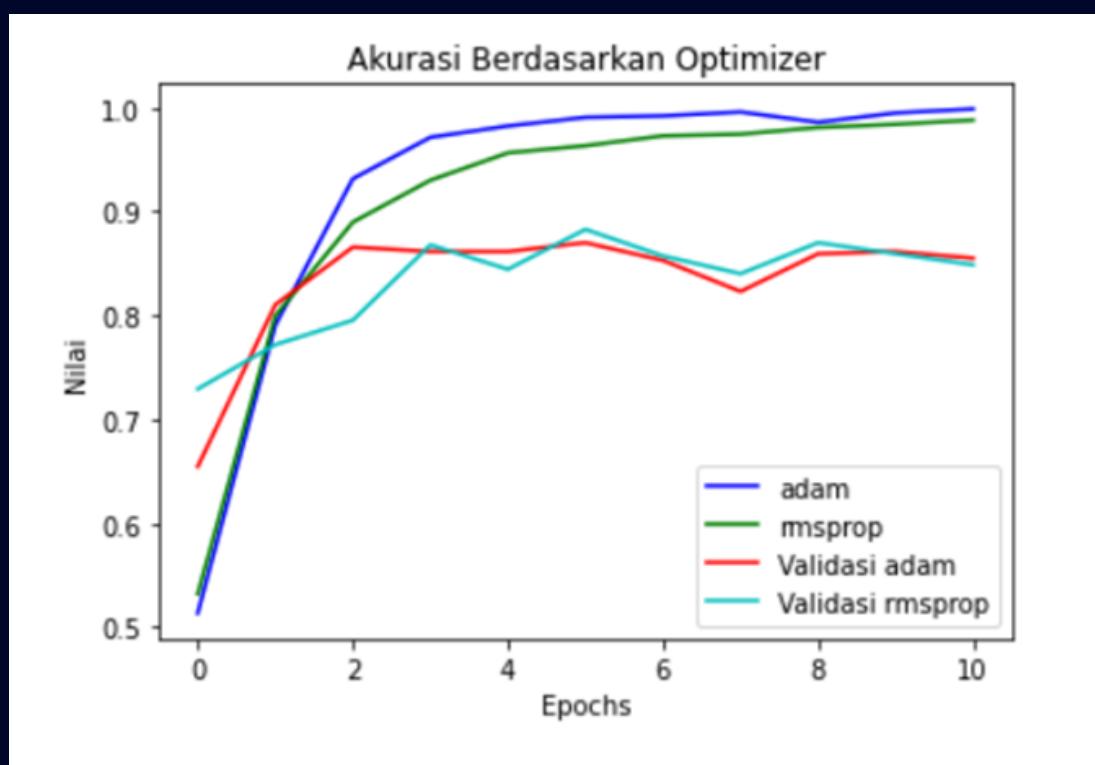
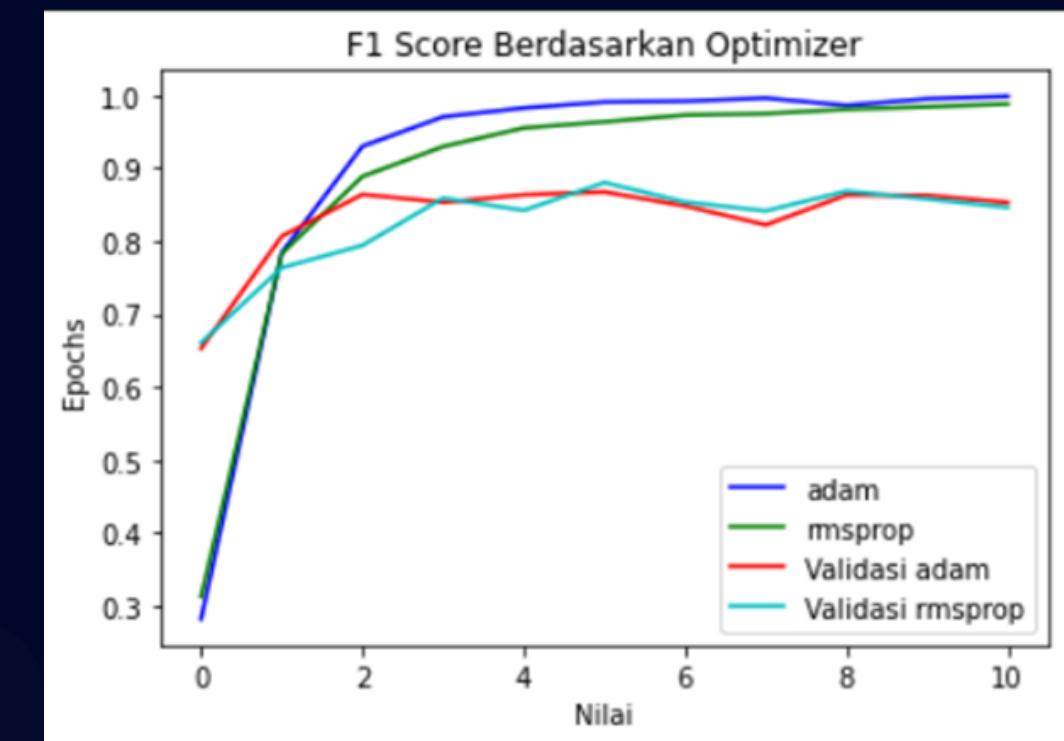
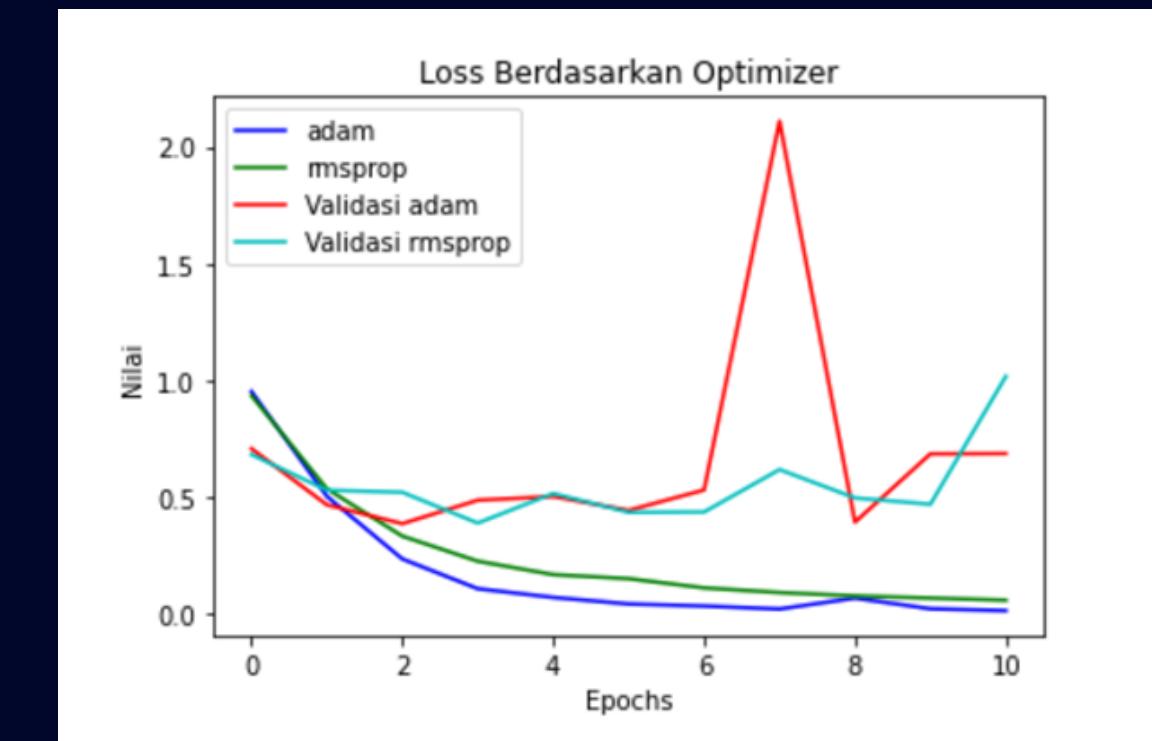
SKENARIO SEDERHANA: FUNGSI AKTIVASI LAYER

- Activation function relu pada hidden layer dan softmax pada output layer mendapatkan akurasi tertinggi dalam skenario sederhana dalam mencari fungsi aktivasi terbaik
- Fungsi aktivasi softmax memberikan akurasi yang baik karena softmax cocok untuk klasifikasi dengan label multiclass
- Relu dapat mempercepat konvergensi stochastic gradient descent dimana gradien yang dimiliki maksimum satu dan nilai dari relu berkisar dari nol sampai tak hingga



SKENARIO SEDERHANA: OPTIMIZER

- Optimizer RMSProp mampu menghasilkan hasil yang lebih baik dibandingkan dengan optimizer adam
- Model mengalami overfitting
- RMSProp lebih unggul dalam arsitektur model sederhana dan dataset yang cenderung sedikit

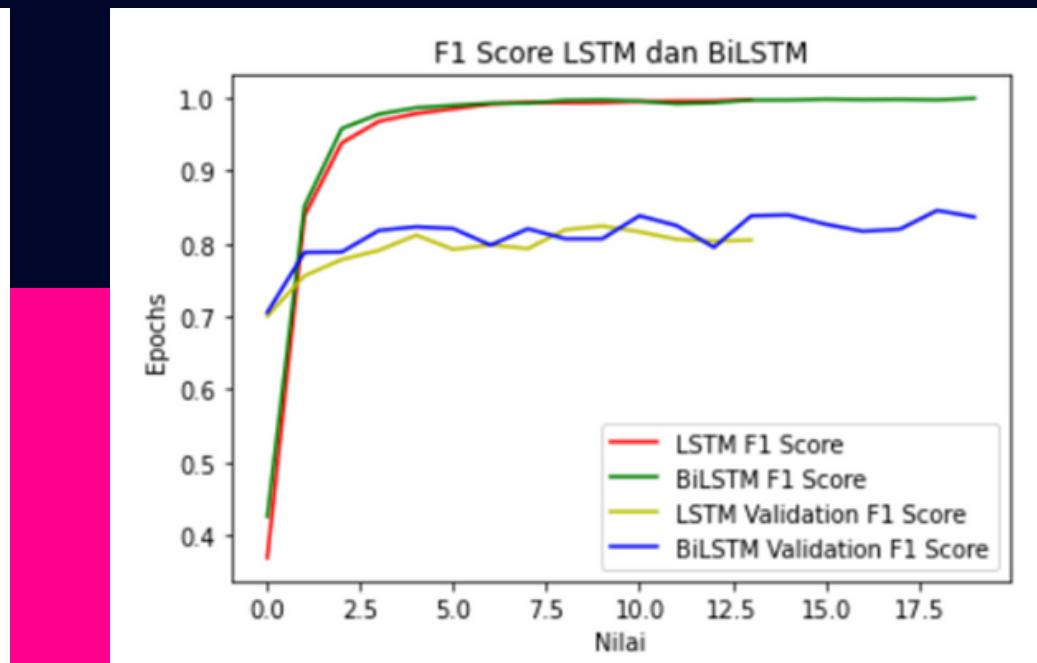
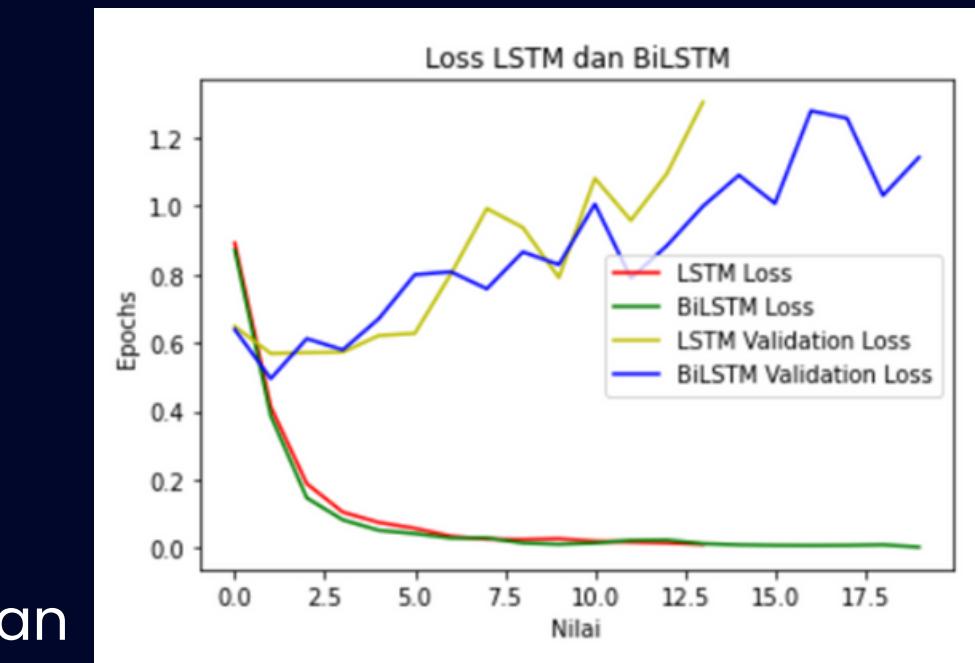
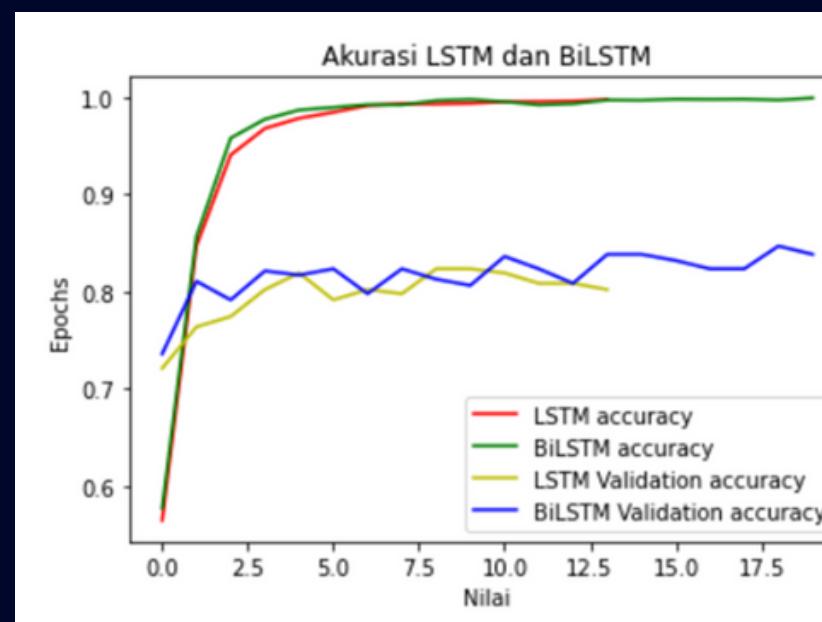


| No | Model | Optimizer | Fungsi loss | Early stopping | Loss | Accuracy | F1 Score |
|----|--|-----------|-------------|----------------|--------|----------|----------|
| 1 | <ul style="list-style-type: none"> • Embedding (16) • BiLSTM(64/relu) • Flatten • Dense(3/softmax) | Adam | Categorical | 10 | 0.5555 | 0.8599 | 0.8658 |
| 2 | <ul style="list-style-type: none"> • Embedding (16) • BiLSTM(64/relu) • Flatten • Dense(3/softmax) | RMSPRO | Categorical | 11 | 0.5011 | 0.8752 | 0.8720 |

Tabel 4. 4 Hasil evaluasi training model terhadap optimizer

SKENARIO SEDERHANA: ARSITEKTUR MODEL

- Model BiLSTM lebih unggul dibandingkan dengan model LSTM dari segi akurasi, loss dan f1 score
- Hal ini dikarenakan BiLSTM dapat bekerja secara dua arah untuk mengetahui konteks kalimat secara lebih luar
- Model mengalami overfitting, walaupun lebih baik daripada skenario sederhana

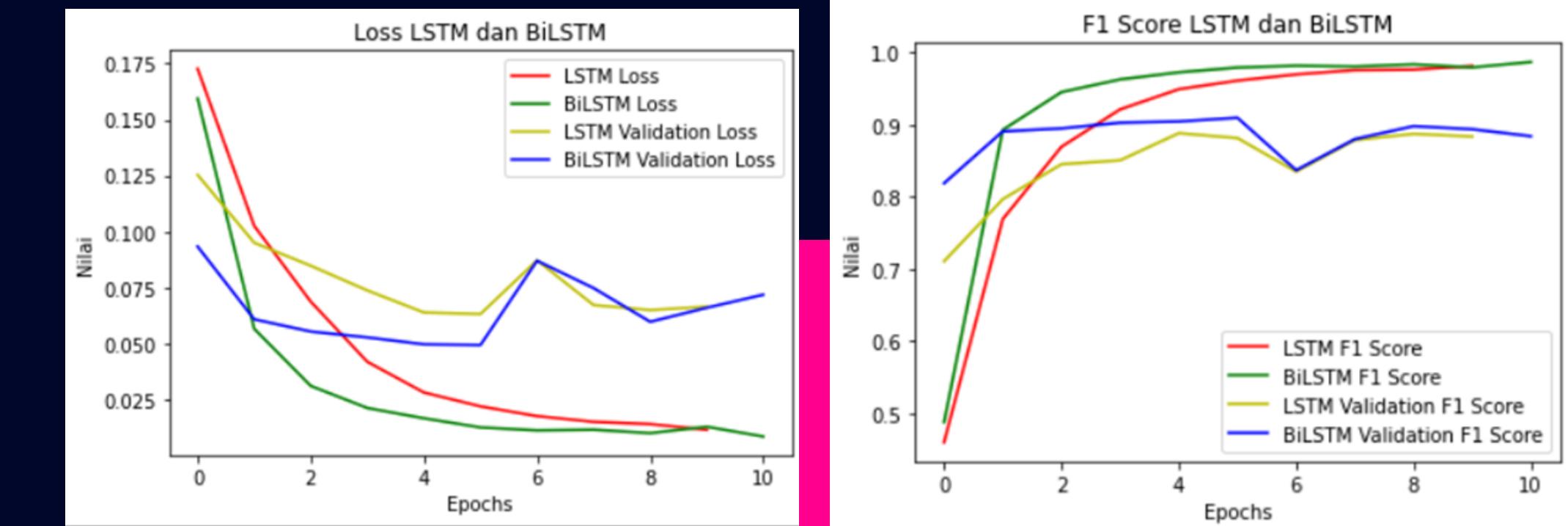
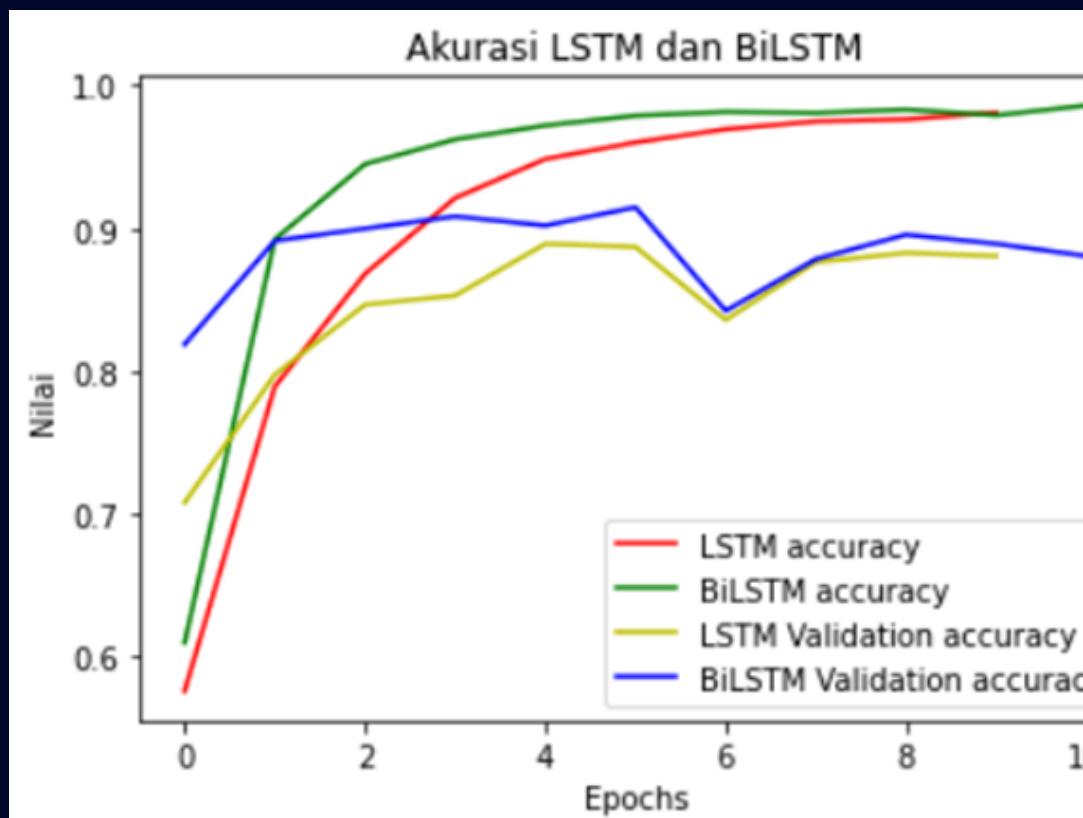


| No | Layer/Aktifasi | Optimizer | Fungsi loss | Early stopping | Loss | Accuracy | F1 Score |
|----|--|-----------|-------------|----------------|--------|----------|----------|
| 1 | <ul style="list-style-type: none"> • Embedding (16) • LSTM (64/tanh) • Flatten • Dense (3/softmax) | Adam | Categorical | 9 | 0.5386 | 0.8157 | 0.8007 |
| 2 | <ul style="list-style-type: none"> • Embedding (16) • BiLSTM(64/tanh) • Flatten • Dense(3/softmax) | Adam | Categorical | 8 | 0.4712 | 0.8522 | 0.8427 |

Tabel 4. 2 Hasil evaluasi training perbandingan arsitektur *layer* model.

SKENARIO KOMPLEKS: OPTIMIZER MODEL

- Optimizer Adam mampu menghasilkan hasil yang lebih baik dibandingkan dengan optimizer RMSProp
- Model mengalami overfitting
- Adam lebih unggul dalam arsitektur model kompleks



| No | Model | Optimizer | Early stopping | Loss | Accuracy | F1 Score |
|----|---|-----------|----------------|--------|----------|----------|
| 1 | <ul style="list-style-type: none"> Embedding (128) <u>LSTM(128/relu/Dropout 0.2)</u>, <u>LSTM(128/relu/Dropout 0.2)</u>, <u>LSTM(128/relu/Dropout 0.2)</u>, Flatten Dense(512/relu) <u>Dropout (0.5)</u> <u>Dense(3/softmax)</u> | RMSPRO P | 11 | 0.4310 | 88.87% | 89.39% |
| 2 | <ul style="list-style-type: none"> Embedding (128) <u>BiLSTM(128/relu/Dropout 0.2)</u>, <u>BiLSTM(128/relu/Dropout 0.2)</u>, <u>BiLSTM(128/relu/Dropout 0.2)</u>, Flatten Dense(512/relu) <u>Dropout(0.5)</u> <u>Dense(3/softmax)</u> | RMSPRO P | 10 | 0.4187 | 90.79 | 91.07% |

Tabel 4. 6 Hasil training model terhadap model kompleks

KESIMPULAN

- Model deep learning dengan arsitektur LSTM dan BiLSTM dapat melakukan sentimen analysis ujaran kebencian Twitter
- Model dengan performa terbaik ada pada model deep learning arsitektur kompleks, dengan fungsi aktivasi relu dan softmax, optimizer Adam serta loss function categorical crossentropy dengan akurasi 94,05%
- Hasil dari Hyper parameter tuning:
 - Arsitektur BiLSTM
 - Fungsi aktivasi relu dan softmax
 - optimizer Adam jika arsitektur kompleks, RMSProp jika arsitektur sederhana
- Callback Early stopping sangat membantu dalam melakukan training model dalam mengefisiensikan waktu dan komputasi



SARAN

- Dataset yang digunakan lebih banyak lagi jumlahnya agar model dibangun jauh lebih baik lagi karena mengenali lebih banyak variasi data yang ada.
- Mencari dataset yang lebih variatif. Jika menggunakan data Twitter, diusahakan menghindari data tweet yang merupakan retweet.
- Dilakukan penelitian lain yang membahas topik dan data yang sama, namun dengan metode yang berbeda.
- Menggunakan feature extraction yang berbeda seperti TF-IDF agar mendapatkan model yang lebih optimal.



THANK YOU

Sentiment Analysis Ujaran
Kebencian Twitter Menggunakan
LSTM dan Bidirectional LSTM

Adrianus Charlie H.A/195314174

