

BAB IV HASIL DAN PEMBAHASAN

4.1. Pengujian Data

Pengujian data pada penelitian ini dibagi menjadi 2 yaitu melalui Long Short Term Memory (LSTM) dan analisis *Bidirectional Encoder Representations dari Transformers*, yaitu teknologi open source berbasis jaringan neural untuk pre-training *Natural Language Processing (NLP)*

4.1.1. Pengujian Data Dengan Long Short Term Memory

4.1.1.1. Tahap Pengumpulan Data

Data penelitian yang diambil berdasarkan teknik web *crawling* dari media sosial Twitter API dari bulan Januari sampai Juli 2023. 5000 data tweet dengan kata kunci pencarian ujaran kebencian pemerintah. Tweet yang dikumpulkan adalah tweet berbahasa Indonesia yang berasal dari region Indonesia.

Crawling diawali dengan membuat koneksi twitter ke. Untuk membuat koneksi ini harus memiliki akun Twitter dan login ke Twitternya untuk memperoleh Acces Token. Setelah tahapannya dilalui dann koneksi sudah terbentuk, kemudian dihubungkan operator Search Twitter. Pada pojok kanan atas panel parameter search twitter dilakukan pengaturan untuk memasukkan kata kunci yang akan dicari, jumlah twitter yang harapan, waktu yang ditentukan, lokasi, dan sebagainya. Penerapannya pada data twitter, *crawling* menggunakan *automation* program dan juga *Application Programming Interface (API)* sebagai jalur komunikasi dalam mendapatkan data.

Dengan API data dapat dikumpulkan lebih spesifik sesuai dengan kata kunci diinputkan tanpa harus mengetahui element HTML pada sebuah website.

Menggunakan model pada gambar 4.1, diperoleh data sebanyak 5000 data tweet yang kemudian disimpan dalam format .CSV,

4.1.1.2. Pre-Processing Data

Pra pemrosesan data dilakukan terhadap data teks (corpus dan data set).

1. Case Folding

Case folding merupakan bentuk *text preprocessing* yang paling sederhana dan efektif dengan tujuan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil dengan memanfaatkan modul yang tersedia di python. Tahap *case folding* yang utama adalah mengubah teks menjadi lowercase, seperti contoh berikut :

```
kalimat = "Berikut ini adalah 5 negara dengan pendidikan terbaik di dunia adalah Korea Selatan, Jepang, Singapura, Hong Kong, dan Finlandia."
lower_case = kalimat.lower()
print(lower_case)
# berikut ini adalah 5 negara dengan pendidikan terbaik di dunia adalah korea selatan, jepang, singapura, hong kong, dan finlandia.
```

Tabel 4. 1 Case Folding

Sebelum	Sesudah
Aksi statmen Rocky Gerung (RG) yg terindikasi dengan memaki-maki dengan kata Anjing atau babi penguasa yang sudah dapat dikategorikan <i>hate speech</i> terhadap Presiden Jokowi sontak menuai reaksi keras bahkan ancaman dari masyarakat pendukung Presiden Jokowi yang kini mencapai sekitar 90 %. Kondisi ini sangat rawan dan terus bereskalasi yg dapat mengganggu jalannya pesta demokrasi 2024. Kondisi ini memang dilematis, diabaikan karena Presiden Jokowi tdk	Aksi statmen Rocky Gerung (RG) yg terindikasi <i>hate speech</i> terhadap Presiden Jokowi sontak menuai reaksi keras bahkan ancaman dari masyarakat pendukung Presiden Jokowi yang kini mencapai sekitar 90 %. Kondisi ini sangat rawan dan terus bereskalasi yg dapat mengganggu jalannya pesta demokrasi 2024. Kondisi ini memang dilematis, disebabkan oleh Presiden Jokowi tdk ingin melaporkan RG; polisi menindak RG karena tekanan masyarakat, juga mengundang pro kontra. Lepas dari

Sebelum	Sesudah
ingin melaporkan RG; polisi menindak RG dengan kata bangsatnya karena tekanan masyarakat, juga mengundang pro kontra. Lepas dari pro kontra itu ternyata masyarakat sangat sensitif tentang rasa bahasa membedakan antar kritik dan ujaran kebencian. Yg pasti sebagai Kepala Negara, Presiden Jokowi tdk ingin rakyatnya terjebak dalam ketidakpastian hukum dengan tindakan yang merobek sendi demokrasi Bahkan Jokowi tdk ingin rakyatnya terpenjara karena kritis. Aksi RG bisa saja memicu Presiden Jokowi mengeluarkan Perpu bila dipandang mengarah pada ketidak stabilan politik...	pro kontra itu ternyata masyarakat sangat sensitif tentang rasa bahasa membedakan antar kritik dan ujaran kebencian. Yg pasti sebagai Kepala Negara, Presiden Jokowi tdk ingin rakyatnya terjebak dalam ketidakpastian hukum. Bahkan Jokowi tdk ingin rakyatnya terpenjara karena kritis. Aksi RG bisa saja memicu Presiden Jokowi mengeluarkan Perpu bila yang mengarah pada ketidak stabilan politik...

2. Tokenisasi

Tokenizing adalah proses pemisahan teks menjadi potongan-potongan yang disebut sebagai token untuk kemudian di analisa. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Didalam NLP, token diartikan sebagai "kata" meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat. Fungsi `split()` pada python dapat digunakan untuk memisahkan teks. Perhatikan contoh dibawah ini :

Tabel 4. 2 Tokenizing

Sebelum	Sesudah
Aksi statmen Rocky Gerung (RG) yg terindikasi <i>hate speech</i> terhadap Presiden Jokowi sontak menuai reaksi keras bahkan ancaman dari masyarakat pendukung Presiden Jokowi yang kini mencapai sekitar 90 %. Kondisi ini sangat rawan dan terus bereskalasi yg dapat mengganggu	<ol style="list-style-type: none"> 1. Aksi statemen Rocky Gerung (RG) yg terindikasi <i>hate speech</i> terhadap Presiden Jokowi sontak menuai reaksi keras bahkan ancaman dari masyarakat pendukung Presiden Jokowi yang kini mencapai sekitar 90 %. 2. Kondisi ini sangat rawan dan terus

Sebelum	Sesudah
<p>jalannya pesta demokrasi 2024. Kondisi ini memang dilematis, diabaikan karena Presiden Jokowi tdk ingin melaporkan RG; polisi menindak RG karena tekanan masyarakat, juga mengundang pro kontra. Lepas dari pro kontra itu ternyata masyarakat sangat sensitif tentang rasa bahasa membedakan antar kritik dan ujaran kebencian. Yg pasti sebagai Kepala Negara, Presiden Jokowi tdk ingin rakyatnya terjebak dalam ketidakpastian hukum. Bahkan Jokowi tdk ingin rakyatnya terpenjara karena kritis. Aksi RG bisa saja memicu Presiden Jokowi mengeluarkan Perpu bila dipandang mengarah pada ketidak stabilan politik...</p>	<p>bereskalasi yg dapat mengganggu jalannya pesta demokrasi 2024. Kondisi ini memang dilematis, disebabkan oleh Presiden Jokowi tdk ingin melaporkan RG; polisi menindak RG karena tekanan masyarakat, juga mengundang pro kontra. Lepas dari pro kontra itu ternyata masyarakat sangat sensitif tentang rasa bahasa membedakan antar kritik dan ujaran kebencian.</p> <p>3. Yg pasti sebagai Kepala Negara, Presiden Jokowi tdk ingin rakyatnya terjebak dalam ketidakpastian hukum. Bahkan Jokowi tdk ingin rakyatnya terpenjara karena kritis. Aksi RG bisa saja memicu Presiden Jokowi mengeluarkan Perpu bila yang mengarah pada ketidak stabilan politik...</p>

kalimat = "rumah idaman adalah rumah yang bersih"

pisah = kalimat.split()

print(**pisah**)# output

['rumah', 'idaman', 'adalah', 'rumah', 'yang', 'bersih']

3. Pembersihan Kata (stop word)

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia adalah "yang", "dan", "di", "dari", dll.

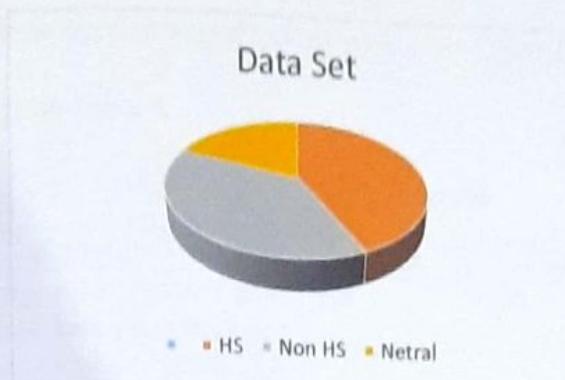
4. Stemming

Stemming adalah proses menghilangkan infleksi kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (*root word*). Misalnya kata "mendengarkan", "dengarkan", "didengarkan" akan ditransformasi menjadi kata "dengar". Idenya adalah ketika anda mencari dokumen "cara membuka lemari", anda juga ingin melihat dokumen yang menyebutkan "cara terbuka lemari" atau "cara dibuka lemari" meskipun terdengar tidak enak. Salah satu algoritma yang tertua dan paling populer adalah algoritma Porter. Algoritma ini tersedia dalam modul NLTK melalui kelas `PorterStemmer()`.

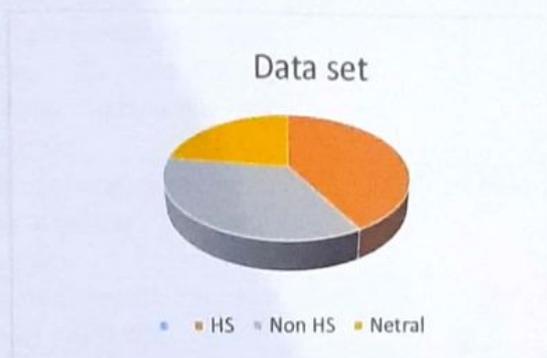
```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
kata = ["program", "programs", "programer", "programing", "programers"]
for k in kata:
    print(k, " : ", ps.stem(k))# ouput
# program : program
# programs : program
# programer : program
# programing : program
# programers : program
```

4.1.1.3. Tahap Analisis Sentimen

Pada tahap ini dilakukan pelabelan terhadap data yang telah dibersihkan. Data dibagi menjadi data latih data uji. Pelabelan data latih dilakukan secara manual dengan memberikan label positif, negative, atau netral pada teks tweet pengguna twitter. Peneliti memberikan label kepada sebanyak 500 data latih yang akan digunakan untuk melabeli data tweet sebagaimana gambar 4.1 dan 4.2. berikut.



Gambar 4. 1 Kalimat HS dan Non HS pada Train 1



Gambar 4. 2 Kalimat HS dan Non HS pada Train 2

Tabel 4. 3 Data Uji

No	Kalimat	Jenis
1	Siapa yang mau geser... pak, jelaskan kok kamu sumber hoax urus tuh mobil ESEMKA katanya dulu ada 1000 orang yang inden. PEMBOHONG !	HS
2	Mundur dari Jabatan dong... Gak malu apa kampanye pakai duit negara... Itu duit rakyat, jangan dibuar kampane pak #SaveUrutMalu	HS
3	Beajar saling menghargai karna kita semua manusia. Termasuk engkau dan aku	Non_HS
4	Pilih yang sudah nyata kerjanya , Pak Jokowi pilihanku	Non_HS
5	Tolong pak turun kan harga pesawat dan jangan naikkan harga minyak goreng	Non_HS
6	Beajar saling menghargai karna kita semua manusia. Termasuk engkau dan aku	Netral
7	Pembelajaran yang baik bagi generasi penerus bangsa	Netral
8	Tarif pajak yang diterapkan telah disesuaikan dengan kenaikan Upah Minimum Regional di setiap provinsi di Indonesia	Netral

Dari pengujian data tersebut didapat hasil yang beberapa contohnya diberikan pada Tabel 4.

Tabel 4. 4 Pensaringan Data Uji

No	Kalimat	Jenis	Hasil Prediksi
1	Siapa yang mau geser... pak, jelaskan kok kamu sumber hoax urus tuh mobil ESEMKA katanya dulu ada 1000 orang yang inden. PEMBOHONG !	HS	HS
2	Mundur dari Jabatan dong... Gak malu apa kampanye pakai duit negara... Itu duit rakyat, jangan dibuar kampanye pak #SaveUrutMalu	HS	HS
3	Beajar saling menghargai karna kita semua manusia. Termasuk engkau dan aku	Non_HS	Non_HS
4	Pilih yang sudah nyata kerjanya , Pak Jokowi pilihanku	Non_HS	Non_HS
5	Tolong pak turun kan harga pesawat dan jangan naikkan harga minyak goreng	Non_HS	Non_HS
6	Beajar saling menghargai karna kita semua manusia. Termasuk engkau dan aku	Netral	Netral
7	Pembelajaran yang baik bagi generasi penerus bangsa	Netral	Netral
8	Tarif pajak yang diterapkan telah disesuaikan dengan kenaikan Upah Minimum Regional di setiap provinsi di Indonesia	Netral	Netral

4.1.2. Pengujian Data dengan BERT

Beberapa langkah utama dalam pengujian BERT

1. Pre-processing Dataset

Pre processing dilakukan untuk mengubah dataset yang tidak terstruktur menjadi terstruktur sehingga mempermudah data untuk diproses dengan melakukan

beberapa tahapan yaitu case folding, data cleaning, tokenisasi, dan normalisasi kata tidak baku.

1. Case Folding

Tahapan ini digunakan untuk mengubah semua karakter menjadi huruf kecil. Tahapan ini dilakukan karena data yang diperoleh tidak selalu terstruktur dan konsisten dalam penggunaan huruf kapital, maka case folding dilakukan untuk menyamaratakan penggunaan huruf kapital. *Case folding* dilakukan dengan menggunakan fungsi `lower()` yang telah tersedia pada *library* Python.

Tabel 4. 5 Hasil Case Folding

No	Review_text	category
0	error terus aplikasi nya ga bs di buka	negative
1	app ga jelas, saya mau melakukan refund tapi saya upload ss bukti transaksi ga bisa? :) beli pulsa 40rb tpi ga masuk , yaudah ikhlasin aja deh rp.40050	negative
2	maaf belom bisa transfer k bank jabar banten :)) y ?	neutral
3	apakah harus ad atm untuk bisa transfer uang melalui flip?	neutral
4	memuaskan bangetttt	positive
5	membantu tapi terlalu lelet	neutral
6	sangat membantu sekali, krn saya tf :) ke bank lain selalu kena admin, terimakasih flip..	positive
7	respon sangat lama, mencerminkan tanggung jawab yang kurang pada :) aplikasi	negative
8	mantap tanpa biaya admin terbaik	positive
9	bagus sekali aplikasinya untuk masuk aja tidak bisa bagaimana untuk transaksi? bakal bikin kecewa pengguna, :) diperbaiki bug aplikasinya!!	negative

2. Data Cleaning

Tahapan ini digunakan untuk menghilangkan angka, beberapa simbol, url, *username* (@username), *hashtag* (#), spasi berlebih, tanda baca, emoji, dan pengulangan karakter yang ada pada kalimat. Tahapan ini menggunakan regular expression untuk menemukan karakter yang akan dihapus.

Tabel 4. 6 Hasil Data Cleaning

No	Review_text	category
0	error terus aplikasinya ga bs di buka	negative
1	app ga jelas, saya mau melakukan refund tapi saya upload ss bukti transaksi ga bisa? beli pulsa 40rb tpi ga masuk, yaudah ikhlasin aja deh rp.40050	negative
2	maaf belum bisa transfer k bank jabar banten y?	neutral
3	apakah harus ad atm untuk bisa transfer uang melalui flip?	neutral
4	memuaskan bangetttt	positive
5	membantu tapi terlalu lelet	neutral
6	sangat membantu sekali, krn saya tf ke bank lain selalu kena admin, terimakasih flip..	positive
7	respon sangat lama, mencerminkan tanggung jawab yang kurang pada aplikasi	negative
8	mantap tanpa biaya admin terbaik	positive
9	bagus sekali aplikasinya untuk masuk aja tidak bisa bagaimana untuk transaksi? bakal bikin kecewa pengguna, diperbaiki bug aplikasinya!!	negative

3. Tokenisasi

Tahapan ini menggunakan *regular expression* untuk menemukan karakter yang akan dihapus. Tahapan ini digunakan untuk memecah kalimat menjadi list kata. Proses ini menggunakan fungsi `word_tokenize` yang disediakan oleh library NLTK

Tabel 4. 7 Hasil Tokenisasi

No	Review_text	category
0	[error, terus, aplikasi, nya, ga, bs, di, buka]	negative
1	[app, ga, jelas, saya, mau, melakukan, refund, tapi, saya, upload, ss, bukti, transaksi, ga, bisa, ?, beli, pulsa, rb, tpi, ga, masuk, , yaudah, ikhlasin, aja, deh, rp..]	negative
2	[maaf, belum, bisa, transfer, k, bank, jabar, banten, y. ?]	neutral
3	[apakah, harus, ad, atm, untuk, bisa, transfer, uang, melalui, flip,?]	neutral
4	[memuaskan, banget]	positive
5	[membantu, tapi, terlalu, lelet]	neutral
6	[sangat, membantu, sekali, krn, saya, tf, ke, bank, lain, selalu, kena, admin, terimakasih, flip..]	positive
7	[respon, sangat, lama, mencerminkan, tanggung, jawab, yang, kurang, pada, aplikasi]	negative
8	[mantap, tanpa, biaya, admin, terbaik]	positive
9	[bagus, sekali, aplikasinya, untuk, masuk, aja, tidak, bisa, bagaimana, untuk, transaksi, ?, bakal, bikin, kecewa, pengguna, diperbaiki, bug, aplikasinya, !]	negative

4. Normalisasi Kata Tidak Baku

Tahap normalisasi adalah tahap di mana dataset yang memiliki kata-kata tidak baku diubah menjadi kata yang baku atau sesuai dengan ejaan. hal ini dilakukan karena cukup banyak kalimat yang menggunakan kata gaul seperti: tdk, dmn, cpt, ga, enggak, ngga, gak. Jika kata tersebut tidak melalui tahap normalisasi, maka sistem akan menganggap kata ga, enggak, ngga, dan gak adalah kata yang berbeda, padahal kata tersebut memiliki makna yang seharusnya sama yaitu enggak.

Tabel 4. 8 Hasil Normalisasi kata Baku

No	Review_text	category
0	error terus aplikasi nya enggak bisa dibuka	negative
1	app enggak jelas, saya mau melakukan refund tapi saya upload ss bukti transaksi enggak bisa? beli pulsa 40rb tapi enggak masuk , ya sudah ikhlasin saja deh rp.40050	negative
2	maaf belum bisa transfer kak bank jabar banten y ?	neutral
3	apakah harus ad atm untuk bisa transfer uang melalui flip?	neutral
4	puas banget	positive
5	membantu tapi terlalu lelet	neutral
6	sangat membantu sekali, karena saya transfer ke bank lain selalu kena admin, terimakasih flip..	positive
7	respon cs sangat lama, mencerminkan tanggung jawab yang kurang pada aplikasi	negative
8	mantap tanpa biaya admin terbaik	positive
9	bagus sekali aplikasinya untuk masuk saja tidak bisa bagaimana untuk transaksi? bakal bikin kecewa pengguna, diperbaiki bug aplikasinya!!	negative

4.1.3. Dataset Splitting

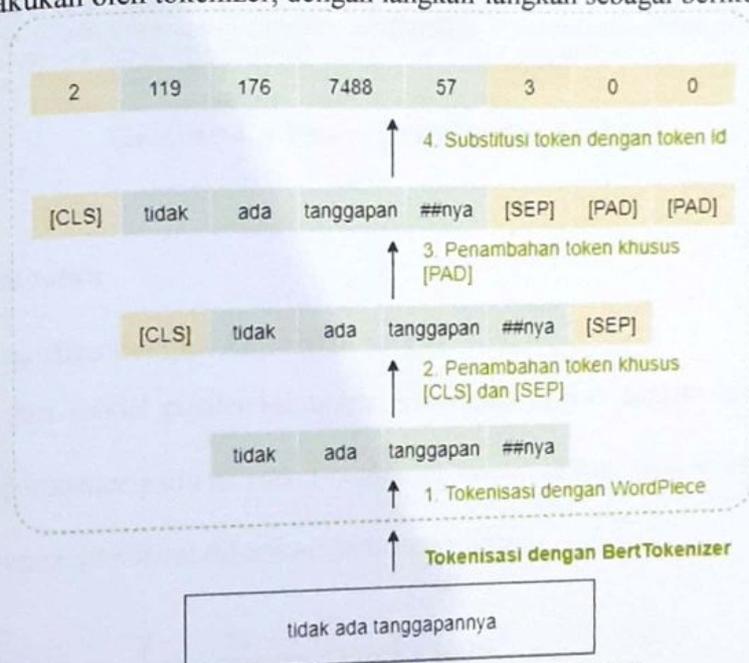
Dataset splitting adalah suatu teknik yang digunakan untuk melihat kinerja model dengan melakukan pembagian terhadap data yang akan kita olah menjadi beberapa bagian dalam hal ini *training*, *validation* dan *testing*. Dataset *training* digunakan untuk melatih model, dataset validasi digunakan untuk meminimalisir *overfitting* yang sering terjadi pada jaringan syaraf tiruan, sedangkan dataset *testing* sendiri digunakan sebagai test akhir untuk melihat keakuratan jaringan yang sudah dilatih dengan dataset *training*. Proporsi split dataset pada penelitian ini adalah 70% *train set*, 20% *validation set*, 10% *test set*. Hal ini bermakna bahwa dari semua data set yang telah dipisah-pisah, 70% data tersebut untuk model latihan, 20% adalah validasi untuk mencegah *overfitting* dan 10% sisanya untuk diujikan.

4.1.4. Implementasi BERT

Pada penelitian ini, penulis menggunakan teknik *fine-tuning* dengan model IndoBERT-base-p1, salah satu model yang menggunakan arsitektur BERT-base. Teknik ini menggunakan model yang telah dilatih sebelumnya dan hanya belajar sedikit lagi untuk mencapai titik optimal pada task yang baru. Model ini telah dilatih menggunakan 4 miliar kata dengan sekitar 250 juta kalimat formal dan sehari-hari dalam Bahasa Indonesia (Wilie *et al.*, 2020). Pada penelitian ini penulis menggunakan *library* Transformers yang disediakan oleh HuggingFace. Library ini menyediakan ribuan *pre-trained model* yang dapat digunakan untuk melakukan tugas-tugas klasifikasi, ekstraksi informasi, tanya jawab, *summarization*, translasi, text generation dan lain-lain dalam 100 bahasa. Transformers didukung oleh dua *library deep learning* yang terkemuka yaitu PyTorch dan TensorFlow.

4.1.5. Data Preparation

Sebelum dilakukan training pada BERT, dataset harus disesuaikan dengan input yang dapat diterima oleh BERT. Oleh karena itu dibutuhkan BertTokenizer, sebuah tokenizer yang bertujuan untuk melakukan tokenisasi pada kalimat-kalimat dan menghasilkan input yang sesuai. Hal ini dilakukan karena BERT menggunakan vocabulary yang spesifik yang mana tergantung dengan model apa yang dipakai. Proses mempersiapkan kalimat menjadi representasi input pada BERT dilakukan oleh tokenizer, dengan langkah-langkah sebagai berikut



Gambar 4. 3 Representasi Input BERT

Hasil dari proses *training* dan evaluasi dapat dilihat pada gambar berikut.

```

0%] | 0/62 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader
cpuset_checked))
(Epoch 1) TRAIN LOSS:0.8911 LR:0.00000300: 100%|██████████| 62/62 [00:50<00:00, 1.22it/s]
(Epoch 1) VALID LOSS:0.6478 ACC:0.8911 F1:0.79 REC:0.79 PRE:0.79 LR:0.00000300
(Epoch 2) TRAIN LOSS:0.5099 LR:0.00000300: 100%|██████████| 62/62 [00:51<00:00, 1.20it/s]
(Epoch 2) VALID LOSS:0.4463 ACC:0.84 F1:0.84 REC:0.84 PRE:0.84 LR:0.00000300
(Epoch 3) TRAIN LOSS:0.3815 LR:0.00000300: 100%|██████████| 62/62 [00:49<00:00, 1.26it/s]
(Epoch 3) VALID LOSS:0.4301 ACC:0.85 F1:0.85 REC:0.85 PRE:0.85 LR:0.00000300
(Epoch 4) TRAIN LOSS:0.3368 LR:0.00000300: 100%|██████████| 62/62 [00:50<00:00, 1.23it/s]
(Epoch 4) VALID LOSS:0.4254 ACC:0.85 F1:0.85 REC:0.84 PRE:0.85 LR:0.00000300
(Epoch 5) TRAIN LOSS:0.2833 LR:0.00000300: 100%|██████████| 62/62 [00:49<00:00, 1.24it/s]
(Epoch 5) VALID LOSS:0.4448 ACC:0.85 F1:0.84 REC:0.84 PRE:0.84 LR:0.00000300
(Epoch 5) VALID LOSS:0.4448 ACC:0.85 F1:0.84 REC:0.84 PRE:0.84 LR:0.00000300

```

Gambar 4. 4 Proses Training dan Evaluasi

4.2. Pembahasan

4.2.1. Pengujian dengan Algoritma LSTM

Performa dari model pendeteksi ujaran kebencian diukur dengan menggunakan beberapa parameter yaitu akurasi (*accuracy*), *recall*, dan *precision*. Formula dari ketiga parameter tersebut diberikan pada Persamaan

$$\text{Akurasi} = \left(\frac{TP + TN}{TP + TN + FP + FN} \right) \times 100\% \dots\dots\dots(1).$$

$$\text{Precision} = \frac{TP}{TP + FP} \dots\dots\dots(2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \dots\dots\dots(3)$$

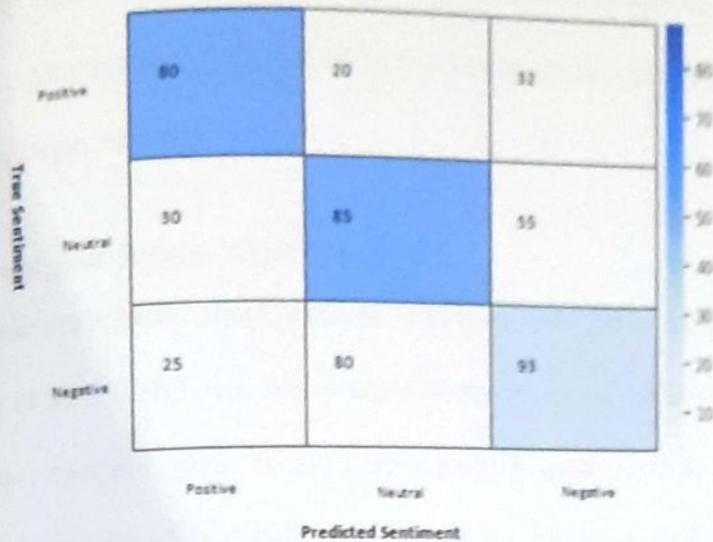
Dengan TP (*True Positive*) memiliki makna banyaknya data uji yang dikategorikan sebagai kelas positif, dalam hal ini *Hate Speech* (HS) dan sebenarnya memang HS. *True Negative* (TN) menyatakan banyaknya data uji yang dikategorikan sebagai kelas negatif, dalam hal ini *Non Hate Speech* (NHS)

dan sebenarnya memang NHS. Selanjutnya *False Positif* (FP) menyatakan banyaknya data uji yang dideteksi sebagai positif HS dan sebenarnya masuk dalam kategori NHS, sedangkan *False Negative* (FN) menyatakan banyaknya data uji yang dideteksi sebagai data negatif atau bukan ujaran kebencian (NHS) dimana kenyataannya merupakan ujaran kebencian (HS). Untuk mendeskripsikan secara detail performa model diberikan *confusion matrix* pada Gambar 4.5. dan 4.6. untuk pengujian pertama dan kedua.



Gambar 4.5 Confusion Matrix Pengujian Pertama

Dapat dilihat pada Gambar 4.5. total terdapat total 500 data uji dimana 80 data uji dianggap NHS dan kenyataannya juga NHS, banyaknya data yang salah dideteksi sebagai ujaran kebencian namun sebenarnya NHS adalah 85, banyaknya data yang dideteksi salah sebagai NHS adalah 90, dan banyaknya data yang dideteksi benar sebagai ujaran kebencian adalah 245. Atau dengan kata lain, nilai $TN = 80$, $FP = 85$, $FN = 90$, dan $TP = 245$. Sehingga didapat akurasinya adalah 55.88%, nilai *recall*-nya adalah 0.6889, dan nilai *precision*-nya adalah 0.5688.



Gambar 4. 6 Confusion Matrix Pengujian Kedua

Dapat dilihat pada Gambar 4.6., total terdapat total 5000 data uji dimana 80 data uji dianggap NHS dan kenyataannya juga NHS, banyaknya data yang salah dideteksi sebagai ujaran kebencian namun sebenarnya NHS adalah 85, banyaknya data yang dideteksi salah sebagai NHS adalah 93, dan banyaknya data yang dideteksi benar sebagai ujaran kebencian adalah 242. Atau dengan kata lain, nilai $TN = 80$, $FP = 85$, $FN = 93$, dan $TP = 242$.

Tabel 4. 9 Data Perhitungan dengan LSTM

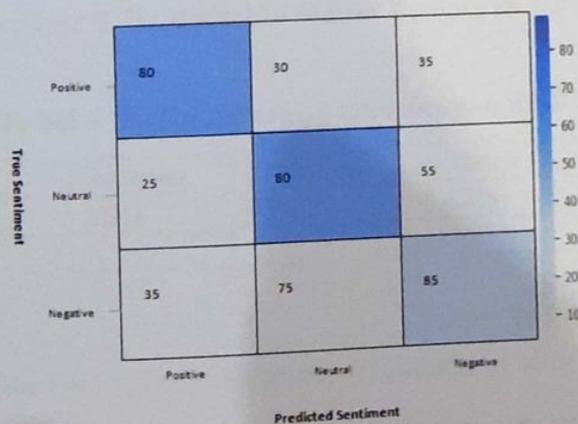
Data Uji Set	Akurasi	Precision	Recall
Hate Speech	85,42%	70,21%	56.41%

Sehingga didapat akurasinya adalah 85.42%, nilai *recall*-nya adalah 0.7021, dan nilai *precision*-nya adalah 0.5641. Hasil uji terbaik dengan parameter akurasi adalah pengujian kedua dengan akurasi 56.41%, untuk parameter *recall* juga terdapat peningkatan pada pengujian kedua yaitu menjadi 0.7021, namun terdapat penurunan *precision* dari 0.5641 pada pengujian pertama menjadi 0.5641. Makna dari *recall* adalah proporsi dari data HS yang dideteksi sebagai HS dengan

keseluruhan data HS yang ada. Sehingga dari keseluruhan data uji pada pengujian kedua yang termasuk HS, 70.21% dideteksi benar sebagai HS, dan 29.79% salah dideteksi sebagai Non HS.

4.2.2. Pengujian dengan Algoritma BERT

Dengan menambahkan pengklasifikasi jaringan saraf satu-lapisan-tersembunyi sederhana di atas BERT dan menyempurnakan BERT, didapat mencapai kinerja yang hampir canggih, yaitu 10 poin lebih baik daripada metode dasar meskipun kami hanya memiliki 3.400 data poin. Selain itu, meskipun BERT sangat besar, rumit, dan memiliki jutaan parameter, kita hanya perlu menyempurnakannya dalam 2-4 epoch saja. Hasil itu bisa dicapai karena BERT dilatih dalam jumlah besar dan sudah menyandikan banyak informasi tentang bahasa kita. Performa mengesankan yang dicapai dalam waktu singkat, dengan sejumlah kecil data telah menunjukkan mengapa BERT adalah salah satu model NLP terkuat yang tersedia saat ini.



Gambar 4. 7 Confusion Matrix Pengujian Pertama

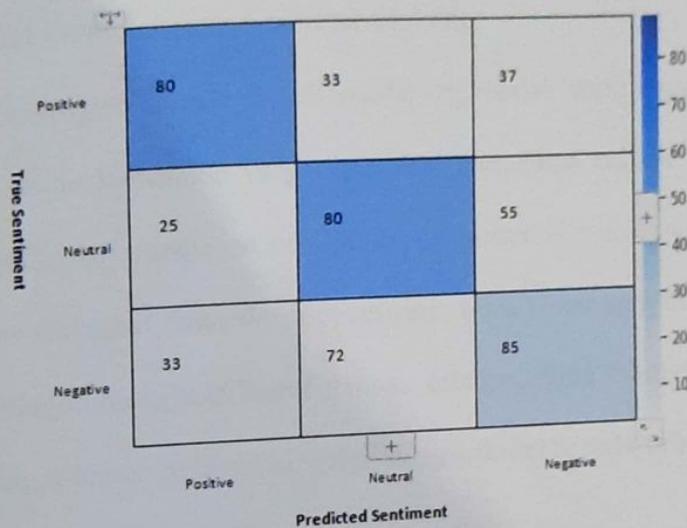
$$\text{Akurasi} = \frac{TP + TN + FN}{TP + TN + FP + FN + TN + FN} \times 100\% \dots\dots\dots (4).$$

$$\text{Precision}_{\text{Positive}} = \frac{TP}{TP + FP} \dots \dots \dots (5)$$

$$\text{Recall}_{\text{Positive}} = \frac{TP}{TP + FNT + FN} \dots \dots \dots (6)$$

$$\text{F1score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots \dots \dots (7)$$

Dari *confusion matrix* tersebut dapat dilihat bahwa model memiliki kesulitan dalam memprediksi data dengan sentimen netral, namun untuk sentimen positif dan negatif sudah cukup baik.



Gambar 4. 8 Confusion Matrix Pengujian Kedua

Tabel 4. 10 Data Perhitungan dengan BERT

Data Uji Set	Akurasi	Precision	Recall
Hate Speech	86%	83%	87%

Dari hasil *classification report* tersebut model mendapat skor cukup tinggi pada sentimen negatif, dan skor relatif lebih rendah pada sentimen positif dan netral.

4.2.3. Perbandingan Antara Pengujian Algoritma LSTM dan BERT

Berdasarkan hasil uji coba dapat dianalisis bahwa analisis sentimen berhasil dilakukan menggunakan teknik *fine-tuning* yang menghasilkan akurasi sebesar 86% dengan pemilihan *hyperparameter* yaitu *batch size* 32, *learning rate* $3e-6$, dan *epoch* 5. Sedangkan, Berdasarkan hasil penelitian yang dilakukan dengan menggunakan *data testing* 5000 kalimat dari *dataset*, dengan sampel sebesar 10% yakni 500 ujaran dengan Algoritma *Long Short Term Memory* sudah cukup baik dalam mendeteksi kalimat ujaran kebencian dengan nilai parameter *recall* mencapai 0.7021 dan nilai akurasi sebesar 85,42%.

Nilai parameter lainnya masih cenderung rendah yang dapat disebabkan oleh kalimat ujaran kebencian yang digunakan mencakup bahasa informal, dan diambil langsung dari media sosial dimana penulisan dari kata-kata seringkali berubah karena disingkat, banyaknya ejaan yang tidak benar atau tidak konsisten, juga kecenderungan mengganti huruf tertentu sebagai angka sehingga tidak dapat terdeteksi sebagai ujaran kebencian sesuai dengan kriteria pada model yang telah dibangun