BAB IV
PEMBAHASAN

Pada bab sebelumnya, telah diuraikan tentang kerangka teoritis dan metodologi penelitian yang digunakan dalam studi ini. Pada Bab 4 ini, akan di jelaskan detail mengenai implementasi sistem yang digunakan dalam rangka mencapai tujuan penelitian. Bagian ini mencakup spesifikasi perangkat keras dan perangkat lunak yang digunakan, serta langkah-langkah yang diambil untuk melakukan pengambilan data dari Twitter (Twitter crawling) dan proses preprocessing dataset sebelumnya. Selain itu, penuilis juga akan menyajikan implementasi dari model bahasa INDOBERT yang menjadi fokus utama penelitian ini. Akhirnya, Bab 4 ini akan ditutup dengan hasil evaluasi dari model yang telah diimplementasikan.

4.1 Implementasi Sistem

Dalam poin ini akan menjelaskan implementasi sistem yang digunakan dalam penelitian ini, termasuk spesifikasi perangkat keras dan perangkat lunak yang digunakan.

4.1.1 Spesifikasi Perangkat Keras

Penuilis menggunakan perangkat keras berikut untuk menjalankan penelitian ini. perangkat keras yang digunakan telah dipilih dengan cermat untuk memastikan kinerja yang optimal selama proses eksperimen dan analisis data.

1. Processor AMD Ryzen 3600x 6 Core 3,8GHz
2. Ram 16 GB DDR4
3. SSD NVME 512GB
4. VGA AMD Radeon RX570

4.1.2 Spesifikasi Perangkat Lunak

Berikut adalah spesifikasi perangkat lunak yang digunakan dalam penelitian ini.

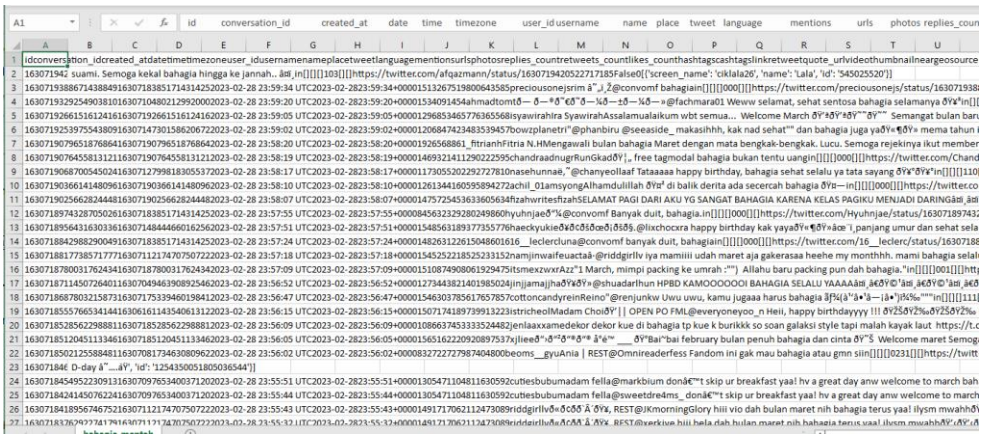
1. Windows 10 Pro 64-bit
2. Google Colab Pro
3. Python 3
4. Node JS 18.07
5. Pandas 2.0.3
6. *Python Library : twint, transformers, torch, nltk, numpy, re, matplotlib, seaborn, sklearn, Sastrawi, keras.*
7. Node.JS Library : *Tweet-Harvest*
8. Docker

4.2 Implementasi Twitter Crawling

Untuk mendapatkan data twitter yang di butuhkan penulis melakukan metode crawling dengan menggunakan library python. Crawling dilakukan dengan filter kata kunci kekerasan verbal berdasarkan pada tabel 3.1. Penulis melakukan beberapa kali percobaan crawling data twitter, beberapa metode dapat berjalan namun menemui beberapa keterbatasan, diantaranya :

1. Menggunakan metode *twint*, library ini berjalan pada docker machine. Eksekusi di jalankan dengan menggunakan filter Pada range waktu 03 April 2023 – 01 Mei 2023, metode ini mengacu pada artikel blog [19]. Metode ini bekerja namun dengan beberapa kekurangan penyimpanan dengan menggunakan file .csv data yang tersimpan terdapat banyak noise seperti karakter non-ascii seperti emoticon dll sulit untuk di cleaning

sedangkan jika di simpan dengan format file .txt hasilnya data minim noise dan rapi. Dokumentasi hasil crawling menggunakan twint yang di simpan dengan format file csv pada gambar 4.1 dan twint yang di simpan dengan format file txt pada gambar 4.2.



Gambar 4.1 Hasil crawling twint format file .csv

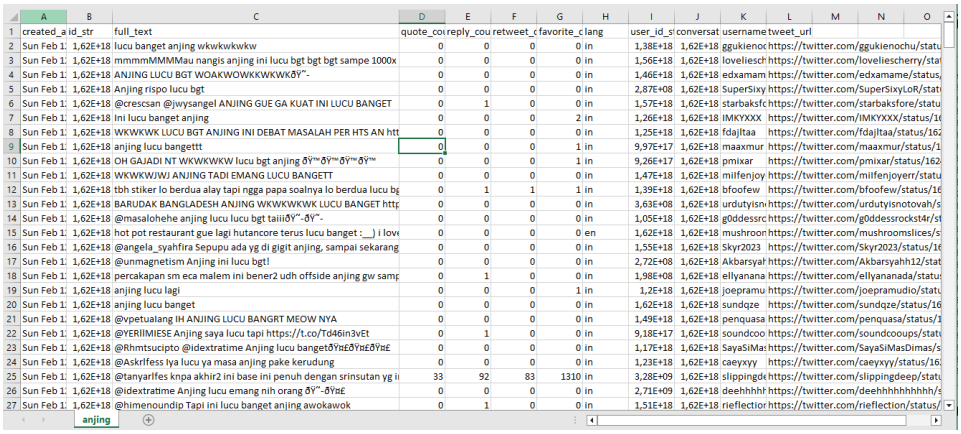
Gambar 4.2 Hasil crawling twint format file .txt

Kekurangan lainnya dalam metode ini yaitu jika limit yang di tentukan terlalu besar, atau saat ekseskusi ada gangguan seperti koneksi internet atau komputer terjadi not responding maka data yang berhasil tersimpan tidak akan mencapai limit. Dengan metode twint penulis berhasil crawling beberapa keyword seperti anjing, bodoh, dan cacat mental, data tersebut di gunakan untuk penelitian ini. Namun saat memasuki bulan Mei 2023,

metode ini tidak lagi dapat di gunakan, karena twitter menutup akses kolom search untuk publik atau pengguna twitter yang belum login kedalam akun. Oleh karena itu penulis berganti menggunakan metode Node.JS yang menggunakan Twitter-Harvest yang akan dijabarkan pada point selanjutnya.

2. Menggunakan metode Node.JS dengan library tweet-harvest, metode ini membutuhkan auth-token pengguna twitter asli yang merupakan syarat wajib agar dapat melakukan search pada twitter. Metode ini dapat di jalankan dengan menggunakan jupyter notebook ataupun *google colab*, yang tentu jika menggunakan *google colab* ini merupakan kelebihan karena tidak akan mengurangi resiko jika menggunakan perangkat keras seperti komputer, runtime tidak akan terganggu dengan koneksi internet dan keadaan kelistrikan kita. Dokumentasi penggunaan tweet-harvest bisa di lihat pada gambar 4.3 dan hasil dari crawling pada gambar 4.4.

Gambar 4.3 Proses crawling tweet-harvest



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	created_a	id_str	full_text	quote_co	reply_cou	retweet_c	favorite_c	lang	user_id_s	conversat	username	tweet_url			
2	Sun Feb 1	1,62E+18	lucu banget anjing wkwkwkwkw	0	0	0	0	in	1,38E+18	1,62E+18	ggukienoc	https://twitter.com/ggukienochu/statu			
3	Sun Feb 1	1,62E+18	mmmmMMMMau nangis anjing ini lucu bgt bgt bgt sampe 1000x	0	0	0	0	in	1,56E+18	1,62E+18	loveliesch	https://twitter.com/loveliescherry/stat			
4	Sun Feb 1	1,62E+18	ANJING LUCU BGT WOAKWOWKKWKWKðŸ˜	0	0	0	0	in	1,46E+18	1,62E+18	edxamam	https://twitter.com/edxamame/status/			
5	Sun Feb 1	1,62E+18	Anjing rispo lucu bgt	0	0	0	0	in	2,87E+08	1,62E+18	SuperSixy	https://twitter.com/SuperSixyLoR/stat			
6	Sun Feb 1	1,62E+18	@crescsan @jwysangel ANJING GUE GA KUAT INI LUCU BANGET	0	1	0	0	in	1,57E+18	1,62E+18	starbaksfc	https://twitter.com/starbaksfore/statu			
7	Sun Feb 1	1,62E+18	Ini lucu banget anjing	0	0	0	2	in	1,26E+18	1,62E+18	IMKYXXX	https://twitter.com/IMKYXXX/status/16			
8	Sun Feb 1	1,62E+18	WKWKWK LUCU BGT ANJING INI DEBAT MASALAH PER HTS AN htt	0	0	0	0	in	1,25E+18	1,62E+18	fdajltaa	https://twitter.com/fdajltaa/status/162			
9	Sun Feb 1	1,62E+18	anjing lucu bangettt	0	0	0	1	in	9,97E+17	1,62E+18	maaxmur	https://twitter.com/maaxmur/status/1			
10	Sun Feb 1	1,62E+18	OH GAJADI NT WKWKWKW lucu bgt anjing ðŸ™ðŸ™ðŸ™	0	0	0	1	in	9,26E+17	1,62E+18	pmixar	https://twitter.com/pmixar/status/162			
11	Sun Feb 1	1,62E+18	WKWKWJWJ ANJING TADI EMANG LUCU BANGETT	0	0	0	0	in	1,47E+18	1,62E+18	miifenjoy	https://twitter.com/miifenjoyerr/statu			
12	Sun Feb 1	1,62E+18	tbh stiker lo berdua alay tapi ngga papa soalnya lo berdua lucu bg	0	1	1	1	in	1,39E+18	1,62E+18	bfoofew	https://twitter.com/bfoofew/status/16			
13	Sun Feb 1	1,62E+18	BARUDAK BANGLADESH ANJING WKWKWKWK LUCU BANGET http	0	0	0	0	in	3,63E+08	1,62E+18	urdutyisn	https://twitter.com/urdutyisnotovah/s			
14	Sun Feb 1	1,62E+18	@masalohehe anjing lucu lucu bgt taiiiðŸ˜ðŸ˜	0	0	0	0	in	1,05E+18	1,62E+18	g0ddessrc	https://twitter.com/g0ddessrockst4r/st			
15	Sun Feb 1	1,62E+18	hot pot restaurant gue lagi hutancore terus lucu banget :__) i love	0	0	0	0	en	1,62E+18	1,62E+18	mushroon	https://twitter.com/mushroomslices/s			
16	Sun Feb 1	1,62E+18	@angela_syahfira Sepupu ada yg di gigit anjing, sampai sekarang	0	0	0	0	in	1,55E+18	1,62E+18	Skyr2023	https://twitter.com/Skyr2023/status/16			
17	Sun Feb 1	1,62E+18	@unmagnetism Anjing ini lucu bgt!	0	0	0	0	in	2,72E+08	1,62E+18	Akbarsyah	https://twitter.com/Akbarsyahh12/stat			
18	Sun Feb 1	1,62E+18	percakapan sm eca malem ini bener2 udh offside anjing gw samp	0	1	0	0	in	1,98E+08	1,62E+18	ellyanana	https://twitter.com/ellyananada/status			
19	Sun Feb 1	1,62E+18	anjing lucu lagi	0	0	0	1	in	1,2E+18	1,62E+18	joepramu	https://twitter.com/joepramudio/statu			
20	Sun Feb 1	1,62E+18	anjing lucu banget	0	0	0	0	in	1,62E+18	1,62E+18	sundqze	https://twitter.com/sundqze/status/16			
21	Sun Feb 1	1,62E+18	@vpetualang IH ANJING LUCU BANGRT MEOW NYA	0	0	0	0	in	1,49E+18	1,62E+18	penquasa	https://twitter.com/penquasa/status/1			
22	Sun Feb 1	1,62E+18	@YERlIMIESE Anjing saya lucu tapi https://t.co/Td46in3vEt	0	1	0	0	in	9,18E+17	1,62E+18	soundcoo	https://twitter.com/soundcooups/statu			
23	Sun Feb 1	1,62E+18	@Rhmtsucipto @idextratime Anjing lucu bangetðŸ¤£ðŸ¤£ðŸ¤£	0	0	0	0	in	1,17E+18	1,62E+18	SayaSiMas	https://twitter.com/SayaSiMasDimas/s			
24	Sun Feb 1	1,62E+18	@Askrlfess Iya lucu ya masa anjing pake kerudung	0	0	0	0	in	1,23E+18	1,62E+18	caeyxyy	https://twitter.com/caeyxyy/status/162			
25	Sun Feb 1	1,62E+18	@tanyarlfes knpa akhir2 ini base ini penuh dengan srinsutan yg i	33	92	83	1310	in	3,28E+09	1,62E+18	slippingde	https://twitter.com/slippingdeep/statu			
26	Sun Feb 1	1,62E+18	@idextratime Anjing lucu emang nih orang ðŸ˜ðŸ¤£	0	0	0	0	in	2,71E+09	1,62E+18	deehhhhh	https://twitter.com/deehhhhhhhhhh/s			
27	Sun Feb 1	1,62E+18	@himenoundip Tapi ini lucu banget anjing awokawok	0	1	0	0	in	1,51E+18	1,62E+18	rieflectior	https://twitter.com/rieflection/status/			

anjing

Gambar 4.4 Hasil crawling tweet-harvest

Keseluruhan dataset yang di peroleh dalam crawling adalah :

- anjing.csv
- cacat_mental.csv
- dilindungi_hukum.csv
- lonte.csv
- sampah.csv
- tidak_punya_otak.csv
- goblok.csv
- babi.csv
- belajar_lagi.csv
- dajjal.csv
- murahan.csv
- penjilat.csv
- tolol.csv

4.3 Preparing Data

Pada tahap ini data di proses agar siap untuk dilabeling secara manual dan dilabeling secara otomatis secara supervised learning, beberapa proses yang dilakukan adalah, shaping dataset, cleaning data, prepare data untuk labeling manual, dan menggabungkan seluruh dataset.

4.3.1 Shaping Dataset

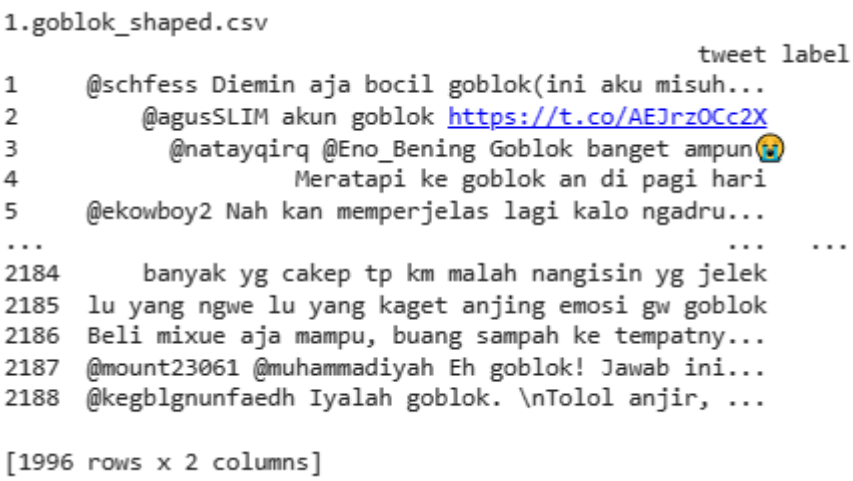
Shaping dataset dilakukan untuk memastikan bahwa seluruh dataset memiliki struktur yang seragam dan hanya memuat data Twitter yang relevan. Karena kedua metode pengumpulan data menggunakan crawling, menghasilkan struktur dataset yang berbeda, yakni beberapa dataset memiliki 12 kolom, sementara yang lain memiliki 15 kolom. Oleh karena itu, proses shaping ini bertujuan untuk menyamakan struktur dataset agar memudahkan dalam mengelompokkan data untuk melakukan labeling secara manual, mempermudah program untuk melakukan labeling menggunakan supervised learning, serta memudahkan proses pelatihan model INDOBERT. Dengan dataset yang telah terstruktur dengan baik, peneliti dapat lebih fokus pada proses analisis sentimen dan meningkatkan efisiensi dalam penggunaan data secara keseluruhan.

Shaping dataset dijalankan dengan menggunakan fungsi *shape_dataframe*, *Code* tersebut berfungsi untuk membentuk dan menyimpan DataFrame baru dari data yang diberikan dalam format file CSV menggunakan fungsi *shape_dataframe*. Pada setiap iterasi, dataset akan diolah satu per satu dan DataFrame baru akan disimpan dalam file CSV dengan nama yang sesuai. Untuk mengakomodasi proses banyak dataset, dilakukan perulangan berdasarkan

jumlah dataset yang akan diolah dengan menamai variabel DataFrame baru sesuai dengan nama dataset. Selanjutnya, fungsi shape_dataframe membaca data CSV, mempertahankan hanya data dengan bahasa "Indonesia", dan mengambil kolom kedua sebagai kolom "tweet". DataFrame baru kemudian dibuat dengan kolom "tweet" dari data asli dan kolom "label" dengan data kosong. DataFrame baru akan disimpan dalam file CSV sesuai dengan nama dataset. Proses ini memungkinkan untuk membentuk dan menyimpan dataset yang siap digunakan dalam analisis sentimen.. Dokumentasi struktur asli dataset ada pada gambar 4.5, dan hasil dari shaping dataset ada pada gambar 4.6.

```
-                    -
                    created_at           id       id_str  \
1  Sat Apr 29 23:51:54 +0000 2023  1,65246E+18  1,65246E+18
2  Sat Apr 29 23:50:35 +0000 2023  1,65246E+18  1,65246E+18
3  Sat Apr 29 23:55:18 +0000 2023  1,65246E+18  1,65246E+18

                                    full_text quote_count  reply_count  \
1  @schfess Diemin aja bocil goblok(ini aku misuh...           0          0.0
2      @agusSLIM akun goblok https://t.co/AEJrzOCc2X           0          0.0
3        @natayqirq @Eno_Bening Goblok banget ampun😱           0          0.0

   retweet_count  favorite_count  geo lang user_id_str conversation_id  \
1            0.0             0.0  NaN   in  1,2439E+18     1,65221E+18
2            0.0             0.0  NaN   in    83425259     1,59757E+18
3            0.0             0.0  NaN   in   122219531     1,65222E+18

  conversation_id_str                                media_url_https  \
1        1,65221E+18                                            NaN
2        1,59757E+18  https://pbs.twimg.com/media/Fu63-k0aQAIuOfY.jpg
3        1,65222E+18                                            NaN

  media_type      username Unnamed: 16
1        NaN  QueenOfRivia         NaN
2      photo  ecko_zidanez         NaN
3        NaN    skinnyfolk         NaN
```

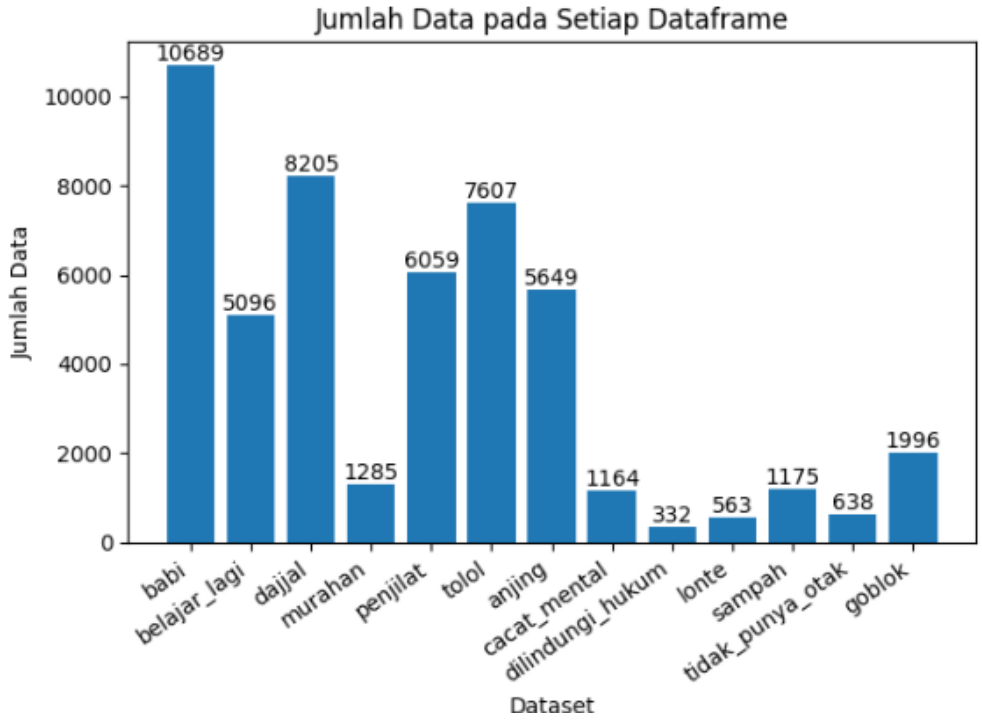
Gambar 4.5 Struktur asli dataset



Gambar 4.6 Struktur dataset setelah di shape

Data yang telah di shape mempunyai struktur kolom tweet dan label, dimana pada kolom label nantinya akan digunakan untuk menyimpan data sentiment yang di berikan secara manual maupun otomatis. Keseluruhan data yang berhasil di shaped selanjutnya di buatkan barplot pada gambar 4.7 untuk menghetahui jumlah pada setiap dataset.

```
Total Keseluruhan data: 50458
```



Gambar 4.7 *Summary* data setelah tahap *shaping* dataset

4.3.2 Cleaning Dataset

Proses cleaning dataset dilakukan untuk memperbaiki dan mengoptimalkan isi dari setiap tweet dalam dataset. Fungsi *clean_data* berperan dalam membersihkan *tweet* dari karakter khusus, mention, URL, serta simbol-simbol yang tidak relevan. Selain itu, tindakan cleaning ini juga mencakup penghapusan kata-kata yang bukan merupakan kata sejati, penghilangan tanda baca ganda, dan penghapusan baris dengan kurang dari 2 kata. Tahapan cleaning ini dilakukan pada setiap dataset dalam loop untuk memastikan keseragaman dan kebersihan data sebelum dilakukan proses labeling secara manual dan training model

menggunakan supervised learning. Selanjutnya, hasil data yang telah dibersihkan akan disimpan dalam file CSV baru dengan nama yang sesuai dengan dataset. Proses cleaning ini bertujuan untuk mempersiapkan data dengan kualitas terbaik sebelum dilakukan analisis sentimen dan pelatihan model INDOBERT.

Langkah-langkah yang ada pada cleaning dataset biasanya di lakukan pada tahap preprocessing data, namun dalam penelitian ini, penulis perlu melakukan lakukan tahap ini sebelum data di labeling secara manual, untuk mempermudah kerja annotator dalam menilai sentiment pada data twitter, dan tentunya akan memudahkan program dalam melakukan labeling secara semi supervised learning nantinya. Berikut adalah gambar 4.8 dataset sebelum cleaning data, gambar 4.9 dataset setelah di cleaning data dan summary data setelah di lakukan cleaning data pada gambar 4.10.

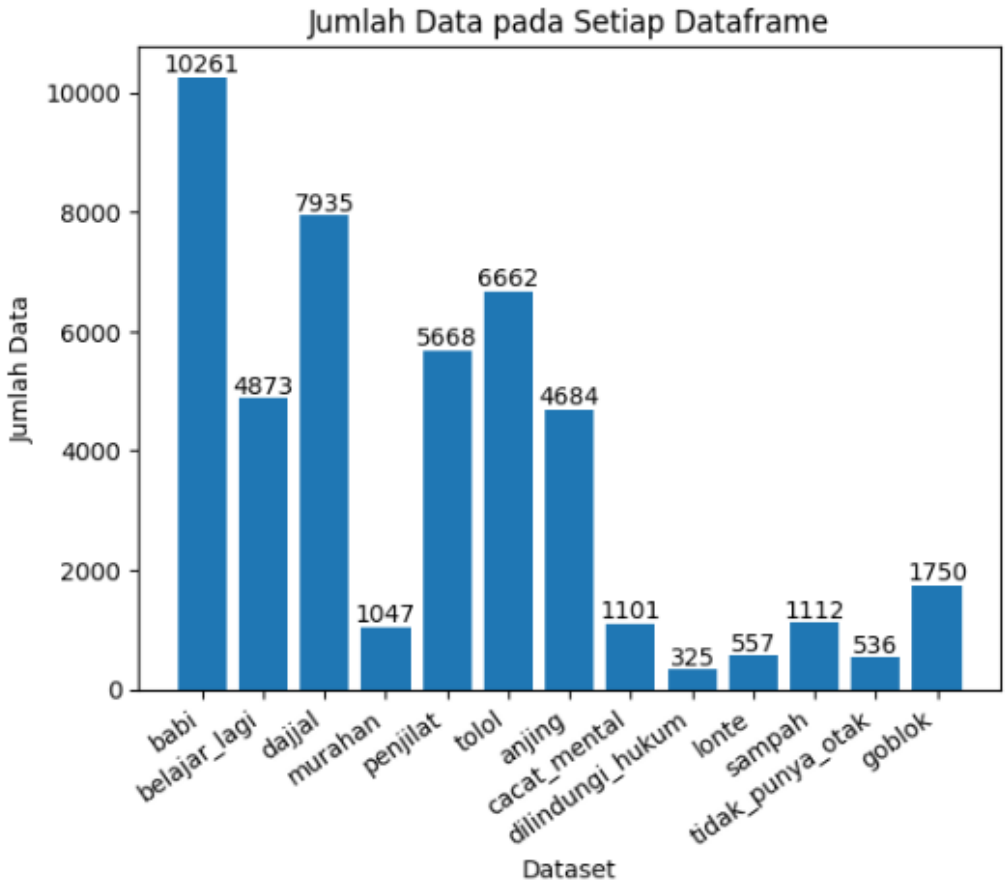
```
Before
                                                  tweet label
0     @Midjan_La_2 Pria pria jahanam Dajjal bedebah ...
2     @zoelfick Dulu sempet mikir, gimana caranya Da...
5                 wkwkwkw nnti dajjal turun dri lampung
6     jnt di ig komennya dibatasin anjing. trs kompl...
7                     @Mr_OmarMalik Nawaz sharif dajjal
...                                                 ...   ...
8568  @Miduk17 Eh longgor kafir dajjal.\nLawan malay...
8569  HUAAAAAA GOODLUCK BUAT KITA SEMUA YANG WAR TIK...
8570                      *Dajjal ney dusri ankh lgwali*
8571                             Oh Dajjal orang German
8572  @412Uin @RachelMarshal12 @Polyglot7777 @CNNInd...

[8205 rows x 2 columns]
```

Gambar 4.8 Hasil sebelum *Cleaning* data

```
After
                                                  tweet label
0                Pria pria jahanam Dajjal bedebah laknat
2     Dulu sempet mikir, gimana caranya Dajjal menye...
5                  wkwkwkw nnti dajjal turun dri lampung
6     jnt di ig komennya dibatasin anjing. trs kompl...
7                                  Nawaz sharif dajjal
...                                                 ...   ...
8568  Eh longgor kafir dajjal.Lawan malaysia aja jar...
8569  HUAA GOODLUCK BUAT KITA SEMUA YANG WAR TIKET Y...
8570                    *Dajjal ney dusri ankh lgwali*
8571                           Oh Dajjal orang German
8572  Masih lebih pro bani bipang. Penjajahan ratusa...
```

Gambar 4.9 Hasil sebelum *Cleaning* data

```
Total Data Yang di Hilangkan: 3947
Total Keseluruhan data: 46511
```



Gambar 4.10 *Summary* data setelah *cleaning* data

4.3.3 *Data Sampling* and *Preparation*

Pada tahap ini, setelah dataset di-*shape* dan dibersihkan (*cleaned*), penulis menentukan untuk mengambil secara acak 10% data dari keseluruhan dataset untuk dilabeli secara manual. Data sebanyak 10% tersebut akan menjadi bagian yang akan digunakan dalam proses Semi-Supervised Learning dengan metode Self-Learning. Sementara itu, data yang tidak terpilih pada tahap pengambilan 10% akan dijadikan sebagai data unlabeled yang nantinya akan diberi label oleh model pada proses Semi-Supervised Learning. Dengan pendekatan ini, diharapkan model dapat memanfaatkan data yang berlabel dan tidak berlabel secara efisien untuk meningkatkan kinerja analisis sentimen.

Beberapa hal perlu diperhatikan dalam langkah data sampling untuk memastikan hasilnya membuat model dalam Semi-Supervised Learning dapat belajar secara merata pada setiap dataset. Langkah pertama adalah melakukan perhitungan untuk mengambil 10% dari keseluruhan data yang telah dibersihkan. Perhitungan ditunjukkan pada gambar 4.11.

```
# Menghitung total data cleaned dari setiap dataset
total_data_cleaned = 0
for dataset in List_Data:
    df_name = 'df' + dataset
    total_data_cleaned += eval(f"{df_name}.shape[0]")

# Menghitung 10% dari total_data_cleaned
percentage = 0.1
num_data_to_label = int(percentage * total_data_cleaned)

print("Jumlah data yang harus dilabeli secara manual:", num_data_to_label)
```

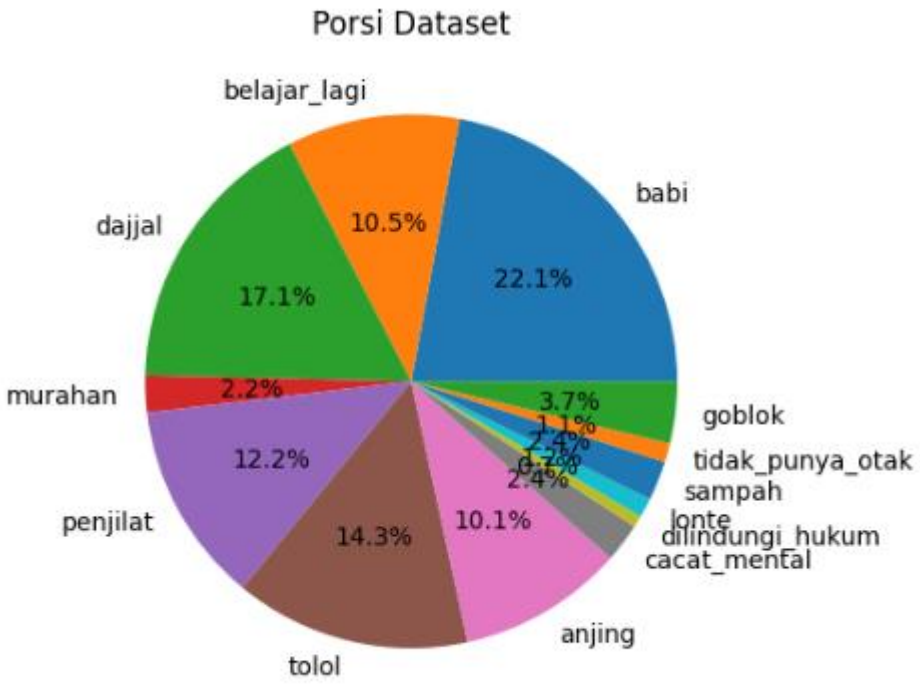
```
Jumlah data yang harus dilabeli secara manual: 5626
```

Gambar 4.11 Perhitungan 10% data untuk labeling manual

Pengambilan proporsi data sebesar 10% ini bertujuan untuk memperoleh sejumlah data yang cukup untuk dilabeli secara manual tanpa mengorbankan representasi yang merata dari setiap dataset. Dengan proporsi yang telah ditentukan, proses labeling manual akan dijalankan dengan lebih efisien dan memungkinkan model belajar dari beragam data yang mewakili seluruh dataset secara seimbang. Hasil perhitungan porsi pada setiap dataset ditunjukkan pada gambar 4.12 dan chartnya pada gambar 4.13

```
Porsi setiap dataset:
babi: 1026 data (22.1%)
belajar_lagi: 487 data (10.5%)
dajjal: 793 data (17.1%)
murahan: 104 data (2.2%)
penjilat: 566 data (12.2%)
tolol: 666 data (14.3%)
anjing: 468 data (10.1%)
cacat_mental: 110 data (2.4%)
dilindungi_hukum: 32 data (0.7%)
lonte: 55 data (1.2%)
sampah: 111 data (2.4%)
tidak_punya_otak: 53 data (1.1%)
goblok: 174 data (3.7%)
```

Gambar 4.12 Hasil perhitungan porsi setiap database



Gambar 4.13 Chart porsi setiap database

Hasil dari data sampling ini akan menjadi bagian integral dalam proses Semi-Supervised Learning yang berfokus pada penggabungan data berlabel dan unlabeled untuk meningkatkan performa model analisis sentimen secara keseluruhan.

Setelah diketahui porsi untuk setiap dataset, data akan diambil secara acak dari masing-masing dataset dan digabungkan menjadi satu dataset baru. Proses pengambilan data dilakukan dengan menggunakan frac=1 dan random_state=42, sehingga data terpilih akan diacak dan menghindari bias dalam urutan data. Selain itu, perlu dilakukan pengaturan ulang index untuk memastikan data yang tergabung dalam dataset baru memiliki index yang berurutan. Hasil dari proses ini ditunjukkan pada Gambar 4.14, yang menggambarkan dataset baru yang telah

terbentuk melalui penggabungan data dari setiap dataset yang telah diambil secara acak.

```python
import pandas as pd
import random


# Inisialisasi DataFrame kosong
combined_df = pd.DataFrame(columns=['tweet', 'label'])

# Ambil data secara acak dari setiap dataset sesuai dengan porsi yang ditentukan
for dataset, porsi in porsi_dataset.items():
    df = pd.read_csv('2.'+ dataset + '_cleaned.csv')  # Ganti dengan nama file dataset
    sampled_df = df.sample(n=porsi, random_state=42)
    #sampled_df['label'] = dataset  # Kolom label diisi dengan nama dataset
    combined_df = pd.concat([combined_df, sampled_df], ignore_index=True)

# Acak seluruh data di dalam DataFrame
combined_df = combined_df.sample(frac=1, random_state=42).reset_index(drop=True)


combined_df.to_csv('combined_data.csv', index=False)

# Tampilkan hasil
print(combined_df)
```

```
                                                  tweet label
0     Dari pemerintah maupun federasi sepakbola kami...   NaN
1           Katanya dajjal itu munculnya dari one piece   NaN
2     Gak onok sg nyalhno Bonek, mek golonganmu tok ...   NaN
3      Aku bacanya mlah babi tp emang babi kan bacanya.   NaN
4                         Kok mati lampu sih pln anjing   NaN
...                                                 ...   ...
4640  gampang,  kalo mereka punya otak buat dipake. ...   NaN
4641                           Iya ih parah bangett sihh   NaN
4642                                tolol banget aowkak   NaN
4643                                  sedih bat anjing    NaN
4644  Sbnrny aq mnusia yg suka pErDaMaIaN, tpi ancri...   NaN

[4645 rows x 2 columns]
```

Gambar 4.14 Data sampling

Selanjutnya, data yang tidak terpilih pada combined_data akan disimpan sebagai unlabeled_data. Data ini akan di-label secara otomatis menggunakan metode Supervised Learning. Proses labeling otomatis ini akan memanfaatkan model yang telah dilatih sebelumnya untuk melakukan prediksi label pada data yang tidak berlabel. Selanjutnya, data yang telah dilabeli akan digunakan sebagai tambahan data berlabel dalam proses Semi-Supervised Learning. Berikut adalah perintah dan hasil dari langkah di atas pada gambar 4.15.

```python
# Buat dataframe kosong all_data
all_data = pd.DataFrame(columns=['tweet', 'label'])

# Menggabungkan dataframe dari setiap dataset
for dataset in List_Data:
    df = pd.read_csv('2.' + dataset + '_cleaned.csv')  # Ganti dengan nama t

    # Hapus data yang sudah ada di combined_df
    df = df[~df['tweet'].isin(combined_df['tweet'])]

    # Gabungkan dataframe ke all_data
    all_data = pd.concat([all_data, df], ignore_index=True)

# Tampilkan hasil
print(all_data)

all_data.to_csv('unlabeled_data.csv')
```

```
                                                  tweet label
0                       Manajemen keuangan cokk babi babi   NaN
1                 Gua selamin deh, lu nyri yang mana meng?   NaN
2                                        lu babi ngepet   NaN
3                                  Tipikal tipikal babi   NaN
4                              Raja Babi ngak terima cuy.   NaN
...                                                  ...   ...
41825      banyak yg cakep tp km malah nangisin yg jelek   NaN
41826  lu yang ngwe lu yang kaget anjing emosi gw goblok   NaN
41827  Beli mixue aja mampu, buang sampah ke tempatny...   NaN
41828  Eh goblok! Jawab ini duluGa usah sok jauh mera...   NaN
41829  Iyalah goblok. Tolol anjir, bisa bisanya lu As...   NaN

[41830 rows x 2 columns]
```

Gambar 4.15 Sisa data dari Langkah Data Sampling

4.4 Labeling Data

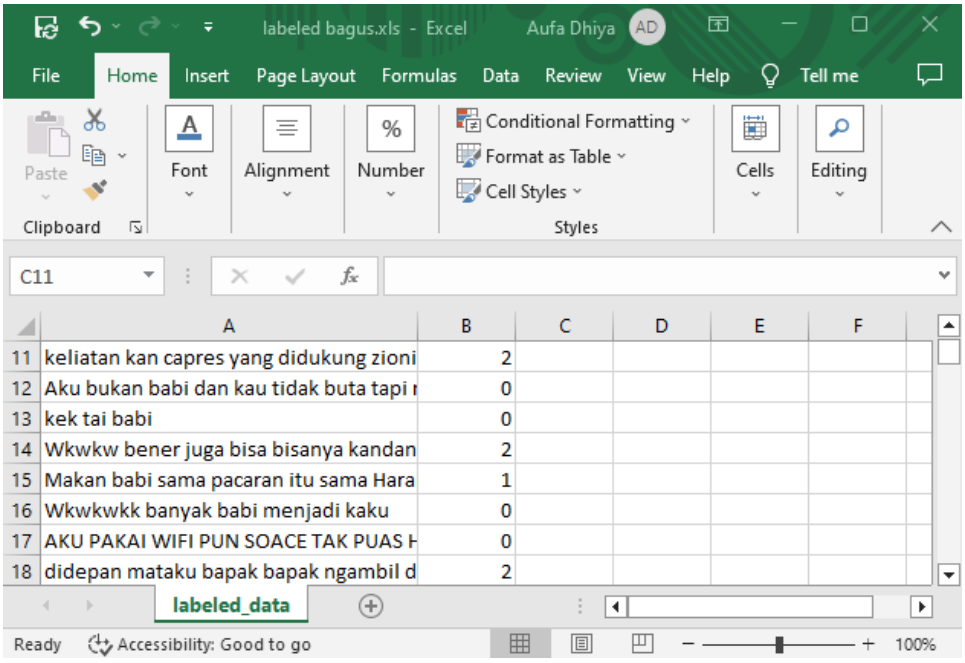
Proses ini melibatkan dua pendekatan, yakni labeling secara manual oleh manusia dan labeling otomatis menggunakan model yang telah dilatih sebelumnya. Dengan kombinasi dari kedua pendekatan ini, penuilis bertujuan untuk memanfaatkan data berlabel dan unlabeled secara optimal, meningkatkan performa model, dan memberikan analisis sentimen yang lebih akurat dan efisien pada data berbahasa Indonesia. Selanjutnya, akan dijelaskan secara rinci tentang langkah-langkah dalam proses labeling data ini.

4.4.1 Labeling Data Manual

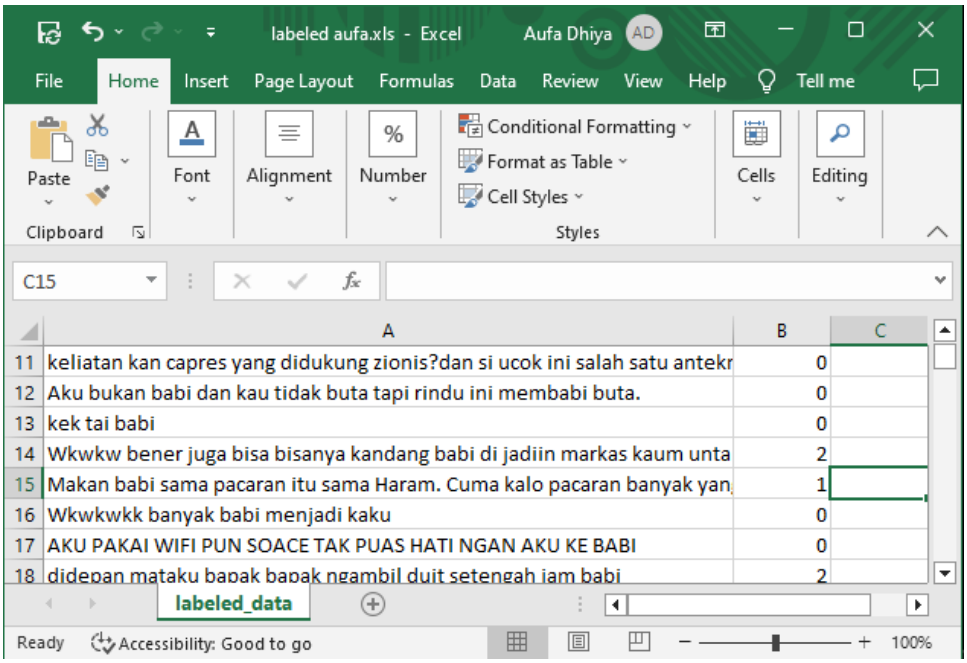
Proses Labeling data secara manual dilakukan pada data yang telah di-sampling dari keseluruhan dataset. Data tersebut kemudian akan dilabeli oleh tiga anotator berbeda dengan ketentuan sebagai berikut:

1. Data yang memiliki sentimen negatif akan diberi label angka "2".
2. Data yang memiliki sentimen positif akan diberi label angka "1".
3. Data yang memiliki sentimen netral akan diberi label angka "0".

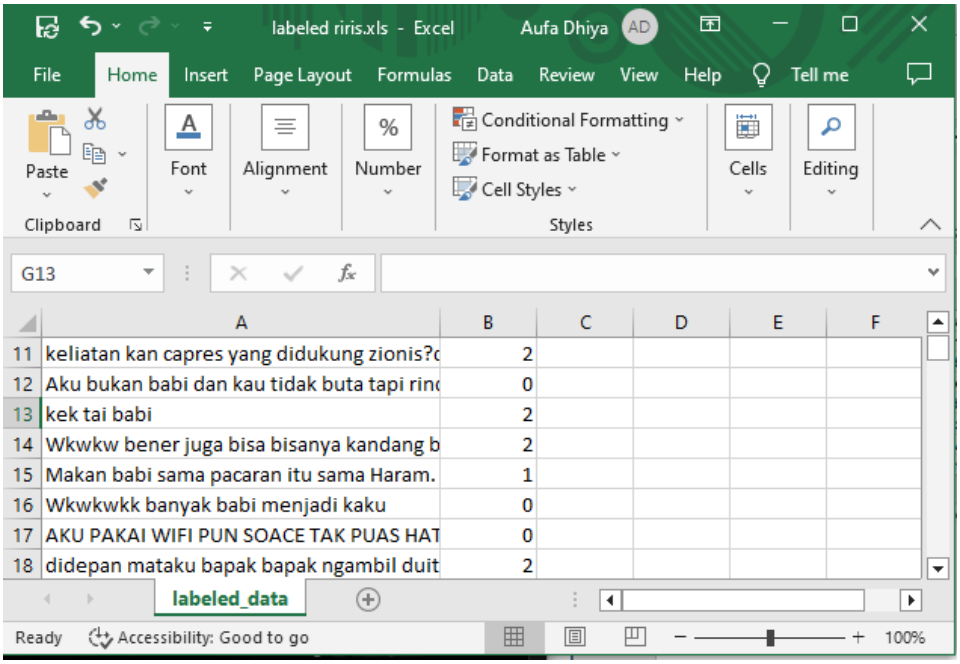
Penggunaan label angka 0, 1, dan 2 dipilih untuk mempermudah dan mempercepat proses annotator dalam melabeli data. Dengan sistem ini, annotator dapat lebih cepat mengidentifikasi sentimen pada setiap data dan memberikan label yang sesuai dengan sentimen yang terkandung dalam tweet. Proses ini menjadi langkah krusial dalam persiapan data berlabel yang akan menjadi dasar bagi model dalam proses Semi-Supervised Learning. Dengan ketentuan label yang jelas dan efisiensi proses labeling, diharapkan analisis sentimen pada data berbahasa Indonesia dapat dilakukan secara lebih efektif dan akurat. Hasil dari labeling manual ditunjukkan pada gambar 4.16, 4.17, dan 4,18.



Gambar 4.16 Data terlabel annotator 1



Gambar 4.17 Data terlabel annotator 2



Gambar 4.18 Data terlabel annotator 3

Setelah seluruh data terkumpul, dilakukan komparasi dan menggabungkan data dari ketiga anotator, serta menentukan label final dengan memilih mayoritas atau label dari anotator pertama(penulis) jika mayoritas tidak ada. Code dalam melakukan melakukan komparasi ada pada gambar 4.19.

```python
import pandas as pd

# Baca data dari ketiga anotator dari file Excel
anotator1_df = pd.read_excel('labeled aufa.xlsx')
anotator2_df = pd.read_excel('labeled riris.xlsx')
anotator3_df = pd.read_excel('labeled bagus.xlsx')

# Gabungkan ketiga DataFrame dengan menggunakan concat
combined_df = pd.concat([anotator1_df, anotator2_df['label_anotator2'], anotator3_df['label_anotator3']], axis=1)

# Hitung mayoritas label untuk setiap data
combined_df['label_final'] = combined_df[['label', 'label_anotator2', 'label_anotator3']].mode(axis=1)[0]

# Jika tidak ada mayoritas, pilih label dari anotator pertama
combined_df['label_final'] = combined_df['label_final'].fillna(combined_df['label'])

# Buat DataFrame baru yang berisi tweet dan label final
final_df = combined_df[['tweet', 'label_final']]

# Simpan DataFrame final menjadi file CSV
final_df.to_csv('final_data.csv', index=False)
```

Gambar 4.19 Komparasi Data Labeled

Hasil dari komparasi yang ada pada final_data.csv inilah yang akan menjadi acuan model supervised learning untuk melabeli keseluruhan data yang lain.

4.4.2 Labeling Data Otomatis

Pada bagian ini, akan membahas tentang proses Labeling Data Otomatis yang menggunakan salah satu pendekatan dalam proses Semi-Supervised Learning untuk analisis sentimen. Dalam metode ini, data yang tidak memiliki label (unlabeled data) akan diberi label secara otomatis menggunakan model yang telah dilatih sebelumnya. Proses ini memanfaatkan hasil prediksi sentimen yang telah dipelajari oleh model dari data berlabel (labeled data). Dengan mengimplementasikan teknik ini, penulis bertujuan untuk memanfaatkan data secara efisien dan meningkatkan jumlah data berlabel untuk melatih model analisis sentimen. Selanjutnya, akan dijelaskan secara rinci tentang langkah-langkah dan pendekatan yang digunakan dalam proses Labeling Data Otomatis ini untuk meningkatkan performa model secara keseluruhan.

4.4.2.1 Preprocessing Data

Sebelum data digunakan untuk melatih model Supervised Learning, penting untuk melakukan langkah preprocessing data guna memastikan kualitas dan keseragaman data. Dalam tahap ini, dilakukan stopword removal dan replace slangword menggunakan library nlp untuk menghilangkan kata-kata yang tidak relevan dan menggantikan kata-kata slang dengan bentuk baku yang sesuai. Selain itu, dalam proses stemming, digunakan stemmer dari sastrawi untuk mengubah kata-kata dalam dataset menjadi bentuk dasar atau kata dasar, mengatasi variasi kata yang serupa, sehingga kata-kata yang memiliki akar kata

yang sama akan dianggap sebagai satu entitas. Untuk melihat implementasi kode dapat ditemukan pada Gambar 4.20. Dengan langkah-langkah ini, diharapkan data yang sudah melalui proses preprocessing akan lebih siap digunakan oleh model Supervised Learning labeling.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from tqdm import tqdm  # Tambahkan import tqdm
from indoNLP.preprocessing import remove_stopwords, replace_slang

# Fungsi untuk melakukan proses stemming menggunakan PySastrawi
def stemming(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(text)

# Fungsi untuk membersihkan data teks
def clean_text(text):
    if pd.notna(text):  # Periksa apakah nilai teks adalah string valid (bukan NaN)
        # Case folding (ubah menjadi huruf kecil)
        text = text.lower()

        # Menghapus karakter berulang (lebih dari 2 karakter sama berulang)
        text = re.sub(r'(.)\1{2,}', r'\1\1', text)

        # Menghapus simbol-simbol
        text = re.sub(r'[^\w\s]', '', text)

        # Menghapus stopwords
        text = remove_stopwords(text)

        # Mengganti slang words
        text = replace_slang(text)

        return text
    else:
        return text  # Jika nilai teks adalah NaN, kembalikan nilai tersebut tanpa melakukan proses lebih lanjut

# Baca data dari file CSV
final_data = labeled_df
unlabeled_data = pd.read_csv('unlabeled_data.csv')

# Proses data pada kolom 'tweet' menggunakan stemming dan preprocessing
# Tambahkan progress bar menggunakan tqdm
tqdm.pandas(desc="Processing final_data")
final_data['tweet'] = final_data['tweet'].progress_apply(clean_text)
final_data['tweet'] = final_data['tweet'].progress_apply(stemming)

tqdm.pandas(desc="Processing unlabeled_data")
unlabeled_data['tweet'] = unlabeled_data['tweet'].progress_apply(clean_text)
unlabeled_data['tweet'] = unlabeled_data['tweet'].progress_apply(stemming)

# Simpan hasil ke dalam file CSV baru
final_data.to_csv('final_data_processed.csv', index=False)
unlabeled_data.to_csv('unlabeled_data_processed.csv', index=False)
```

```
Processing unlabeled_data: 100%|██████████| 47846/47846 [02:19<00:00, 341.96it/s]
```

Gambar 4.20 Proses Preprocessing data

Setelah di lakukan preprocessing data, hasil dari preprocessing akan di disajikan pada gambar 4.21, selanjutnya data dicek distribusi label sentimentnya,

yang ada pada dataset final_data_preprocessed.csv hasil perhitungan distribusi label sentimen ditunjukkan pada gambar 4.22.

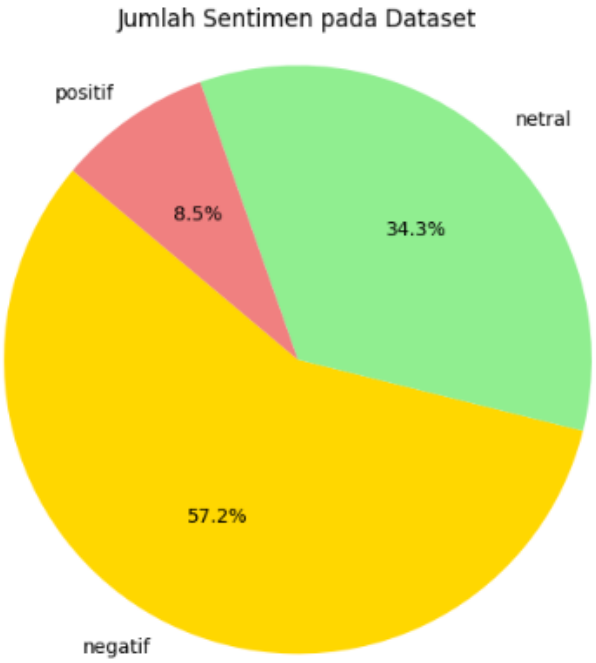
```
[ ]  print(labeled_data)
```

```
                                                  tweet  label
0            super intensif go anjg banget asli ah babi      2
1                                              lu babi      2
2                 hahaha sumpah nih igau pasal tuh babi      0
3                        that lu etik kagak suara babi      0
4                                              wes babi      0
...                                                 ...    ...
4642                                       tolol goblok      2
4643  blokk goblokk pantesan solo setan nih gampang ...      2
4644  jaman pele dek offside gua bilang teknik blok ...      2
4645  tolol kau bukti ban argumen noh orang bela pol...      2
4646  najis mark nyamperin gua tuh kayak ansgyayabsy...      2

[4647 rows x 2 columns]
```

```
[ ]  unlabeled_data = unlabeled_data.drop('label', axis=1)
     print(unlabeled_data.head(5))
```

```
                          tweet
0  manajemen uang cokk babi babi
1     gua lamin deh lu nyri meng
2                 lu babi ngepet
3           tipikal tipikal babi
4    raja babi enggak terima cuy
```

Gambar 4.21 Hasil Preprocessing data



Gambar 4.22 Distribusi data berdasarkan Sentimen

Tersaji pada diagram bahwa dataset final_data mempunyai data sentimen 8,5% positif, 34,3% netral dan 57,2% negative. Selanjutnya data telah siap untuk digunakan melatih model supervised-learning.

4.4.2.1 Implementasi Labeling dengan Supervised Learning

Langkah labeling dilakukan pada unlabeled_data.csv dengan melatih model bi-lstm menggunakan data yang telah terlabeli secara manual, data yang telah dilabeli secara manual di bagi menjadi 2 data yaitu data latih dan data validasi, dengan masing masing porsi yaitu 80/20 hal ini bertujuan memberikan data latih yang cukup untuk klasifikasi multi kelas [20].

Pada tahap ini penulis menentukan beberapa parameter :

1. Beberapa variabel dan list seperti elapsed_times, accuracies, losses, sparse_categorical_accuracies, output_list, dan time_list diinisialisasi untuk menyimpan informasi yang berkaitan dengan proses pelatihan.

2. Parameter vocab_size dan max_length ditentukan untuk tokenization data teks.
3. Model Sentiment Analysis diinisialisasi menggunakan Sequential dengan Embedding sebagai input layer, Bidirectional LSTM dengan dropout sebagai hidden layer, dan Dense sebagai output layer untuk klasifikasi sentimen.
4. Model Sentiment Analysis di-compile menggunakan Adam optimizer, sparse categorical crossentropy sebagai loss function, dan sparse categorical accuracy sebagai metrics untuk mengukur kinerja model pada proses pelatihan.

Labeling dilakukan secara bertahap, pertama-tama model dilatih hingga mencapai akurasi 95%, kemudian model akan melabeli data sebanyak 250 data, setelah data berhasil dilabeli, maka akan dilakukan validasi dengan menggunakan data validasi, jika hasilnya sudah sesuai, 250 data yang telah dilabeli akan ditambahkan ke data latih untuk melatih model yang selanjutnya akan melabeli 250 data lainnya, proses ini akan terus berulang sampai seluruh data berhasil dilabeli.

Implementasi pertama labeling menunjukkan hasil dari *sparse categorical accuracy* menunjukkan angka yang dapat melebihi 95% namun untuk hasil dari validasi classification report menunjukkan hasil yang berbeda, hasil percobaan pertama ditunjukkan pada gambar 4.23.

```
Batch 165 labeled. Total labeled data: 45896
8/8 [==============================] - 0s 7ms/step
233/233 [==============================] - 5s 21ms/step - loss: 0.0170 - sparse_categorical_accuracy: 0.9925
117/117 [==============================] - 1s 10ms/step
30/30 [==============================] - 0s 11ms/step
Batch 166 labeled. Total labeled data: 46146
8/8 [==============================] - 0s 10ms/step
233/233 [==============================] - 5s 21ms/step - loss: 0.0221 - sparse_categorical_accuracy: 0.9906
117/117 [==============================] - 1s 8ms/step
30/30 [==============================] - 0s 8ms/step
Batch 167 labeled. Total labeled data: 46396
4/4 [==============================] - 0s 9ms/step
233/233 [==============================] - 5s 21ms/step - loss: 0.0235 - sparse_categorical_accuracy: 0.9919
117/117 [==============================] - 1s 10ms/step
30/30 [==============================] - 0s 10ms/step
Batch 168 labeled. Total labeled data: 46502
Final Validation Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.37      0.39       330
           1       0.29      0.26      0.28        76
           2       0.67      0.73      0.70       524

    accuracy                           0.56       930
   macro avg       0.46      0.45      0.45       930
weighted avg       0.55      0.56      0.55       930

Final Validation Confusion Matrix:
[[121  41 168]
 [ 37  20  19]
 [136   8 380]]
```

Gambar 4.23 Percobaan Labeling pertama

Hasil dari percobaan pertama Rata-rata dari precision, recall, dan f1-score memiliki nilai sekitar 0.54, menunjukkan performa yang seimbang antara presisi dan recall meskipun rata-rata f1-score memiliki nilai yang baik, performa pada kelas 1 (kelas minoritas) sangat rendah, dengan nilai f1-score hanya sekitar 0.29. Matriks konfusi menunjukkan bahwa model memiliki masalah dalam mengidentifikasi sentimen 1(positif) dan sentimen 2 (negatif) dengan benar, serta banyaknya kesalahan dalam mengklasifikasikan data.

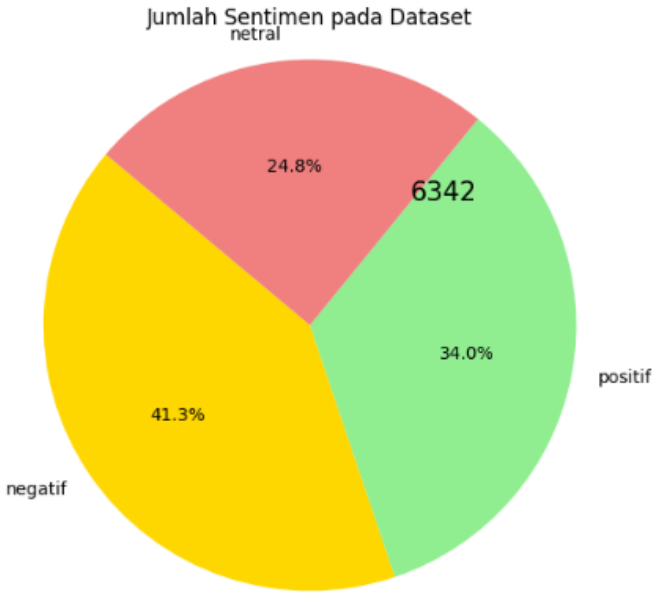
Percobaan kedua dilakukan dengan mengubah beberapa *hyper-parameter* seperti menaikkan learning-rate ke 0,003 dan menambahkan layer-dropout dan layer bi-lstm dengan menentukan dropout sebesar 0,002. Hasil dari percobaan ke 2 disajikan pada gambar 4.24.

```
117/117 [==============================] - 2s 21ms/step - loss: 0.0135 - sparse_categorical_accuracy: 0.9935
117/117 [==============================] - 1s 8ms/step
30/30 [==============================] - 0s 8ms/step
Batch 166 labeled. Total labeled data: 46144
8/8 [==============================] - 0s 10ms/step
117/117 [==============================] - 3s 27ms/step - loss: 0.0126 - sparse_categorical_accuracy: 0.9927
117/117 [==============================] - 1s 10ms/step
30/30 [==============================] - 0s 9ms/step
Batch 167 labeled. Total labeled data: 46394
4/4 [==============================] - 0s 10ms/step
117/117 [==============================] - 2s 20ms/step - loss: 0.0127 - sparse_categorical_accuracy: 0.9930
117/117 [==============================] - 1s 8ms/step
30/30 [==============================] - 0s 9ms/step

Final Validation Classification Report:
              precision    recall  f1-score   support

           0       0.48      0.42      0.45       589
           1       0.31      0.38      0.34       183
           2       0.64      0.68      0.66       801

    accuracy                           0.54      1573
   macro avg       0.48      0.49      0.48      1573
weighted avg       0.54      0.54      0.54      1573

Final Validation Confusion Matrix:
[[245 101 243]
 [ 56  69  58]
 [209  51 541]]
```

Gambar 4.24 Percobaan Labeling kedua

Hasil dari percobaan kedua Rata-rata dari precision, recall, dan f1-score memiliki nilai sekitar 0.54, menunjukkan performa yang seimbang antara presisi dan recall meskipun rata-rata f1-score memiliki nilai yang baik, performa pada sentiment 1 (positif) sedikit meningkat dengan nilai f1-score hanya sekitar 0.31. Matriks konfusi menunjukkan bahwa model masih memiliki masalah dalam mengidentifikasi sentimen 1(positif) dan sentimen 2 (negatif) dengan benar, serta banyaknya kesalahan dalam mengklasifikasikan data. Hal ini terjadi karena data yang tidak seimbang menyebabkan *overfitting*.

Percobaan ketiga, penulis menambahkan data positif dari sumber github [21], data kemudian di tambahkan ke data yang sudah terlabel secara manual, kemudian di hitung kembali distribusi sentiment yang ada pada dataset, distribusi label sentiment disajikan pada gambar 4.25. dan hasil percobaan ketiga pada gambar 4.26.



Gambar 4.25 Distribusi label sentimen

```
Final Validation Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.42      0.41       328
           1       0.89      0.85      0.87       397
           2       0.66      0.67      0.66       543

    accuracy                           0.66      1268
   macro avg       0.65      0.65      0.65      1268
weighted avg       0.67      0.66      0.66      1268

Final Validation Confusion Matrix:
[[137  26 165]
 [ 34 339  24]
 [165  15 363]]
```

Gambar 4.26 Hasil akurasi percobaan labeling ketiga

Penulis memutuskan data hasil percobaan ketiga yang di pakai untuk Melatih model INDOBERT. Meskipun akurasi data sentimen yang akan digunakan untuk melatih model INDOBERT saat ini sebesar 64%, hasil akhir dari model tidak

dapat dengan pasti diprediksi hanya berdasarkan akurasi data tersebut. Model INDOBERT memiliki kemampuan untuk memahami dan memproses bahasa Indonesia, dan kemampuan ini dapat membantu dalam memperbaiki performa model.

4.5 Implementasi INDOBERT

Pada langkah ini, karena melatih model INDOBERT membutuhkan *resource* yang tinggi, perlu dibuat beberapa optimalisasi seperti penggunaan *data loader* agar menghemat penggunaan memori saat pelatihan, sehingga seluruh dataset tidak dimuat secara bersamaan dalam memori. Untuk tujuan ini, digunakan juga *blocks* untuk membuat *data loader* yang akan menghasilkan komentar-komentar yang sudah di *tokenisasi*. Komentar dan sentimen memiliki panjang maksimal 130 kata. Untuk menganalisis sentimen, digunakan *layer* tambahan, termasuk *dropout layer* dengan probabilitas 0.1 [11]. Penulis juga melakukan *fine-tuning* dengan mengatur *hyperparameter* seperti berikut, mengambil beberapa rekomendasi untuk INDOBERT[22]:

1 *Epoch* : 10

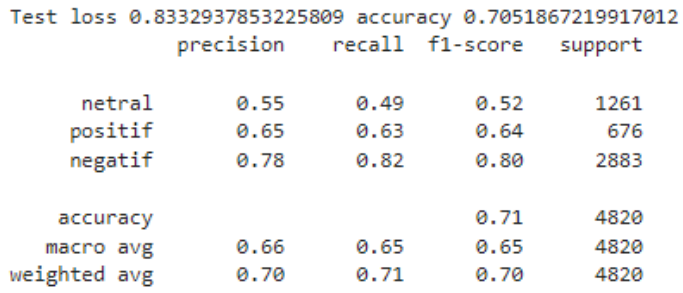
2 *Learning rate* : *1e-3*

3 *Batch size* : 32

Pemilihan *hyperparameter* dengan *Epoch* sebesar 10 menunjukkan bahwa seluruh dataset pelatihan akan dilewati oleh model. Ini berarti model akan memperbarui bobotnya sebanyak 10 kali iterasi. Selanjutnya, *learning rate* sebesar 1e-3 mengindikasikan seberapa besar langkah perubahan bobot yang diambil oleh model saat melakukan pembaruan berdasarkan *gradient* dari fungsi *loss*. Semakin kecil *learning rate*, pembaruan menjadi lebih halus dan stabil,

tetapi bisa memerlukan lebih banyak iterasi untuk mencapai konvergensi. Terakhir, batch size sebesar 32 menentukan berapa banyak sampel data yang digunakan dalam setiap iterasi saat melatih model. *Batch size* yang lebih besar dapat mempercepat proses pelatihan karena lebih banyak sampel yang dievaluasi sekaligus, tetapi juga dapat memakan lebih banyak memori.

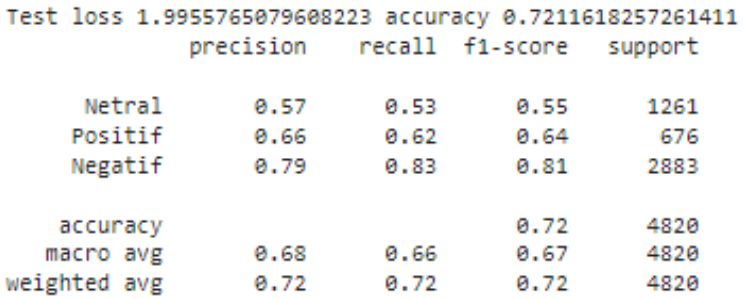
Pada percobaan pertama didapatkan beberapa data seperti Untuk Epoch optimal dilakukan sebanyak 10 kali. Gambar 4.27 menunjukkan hasil akurasi 5 Epoch, gambar 4.28 menunjukkan hasil akurasi 10 Epoch dan gambar 4.29 menunjukkan hasil akurasi 15 Epoch.



Test loss 0.8332937853225809 accuracy 0.7051867219917012

	precision	recall	f1-score	support
netral	0.55	0.49	0.52	1261
positif	0.65	0.63	0.64	676
negatif	0.78	0.82	0.80	2883
accuracy			0.71	4820
macro avg	0.66	0.65	0.65	4820
weighted avg	0.70	0.71	0.70	4820

Gambar 4.27 Hasil akurasi 5 Epoch



Test loss 1.9955765079608223 accuracy 0.7211618257261411

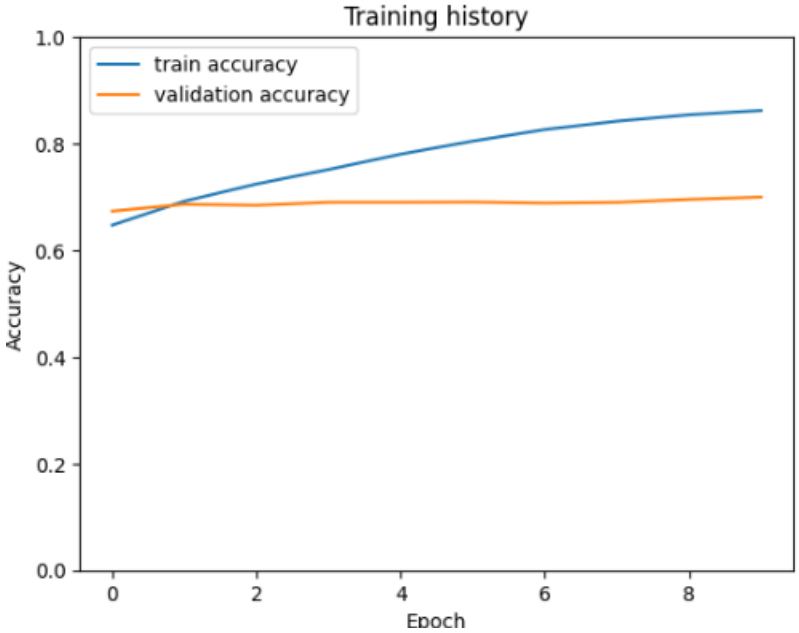
	precision	recall	f1-score	support
Netral	0.57	0.53	0.55	1261
Positif	0.66	0.62	0.64	676
Negatif	0.79	0.83	0.81	2883
accuracy			0.72	4820
macro avg	0.68	0.66	0.67	4820
weighted avg	0.72	0.72	0.72	4820

Gambar 4.28 Hasil akurasi 10 Epoch

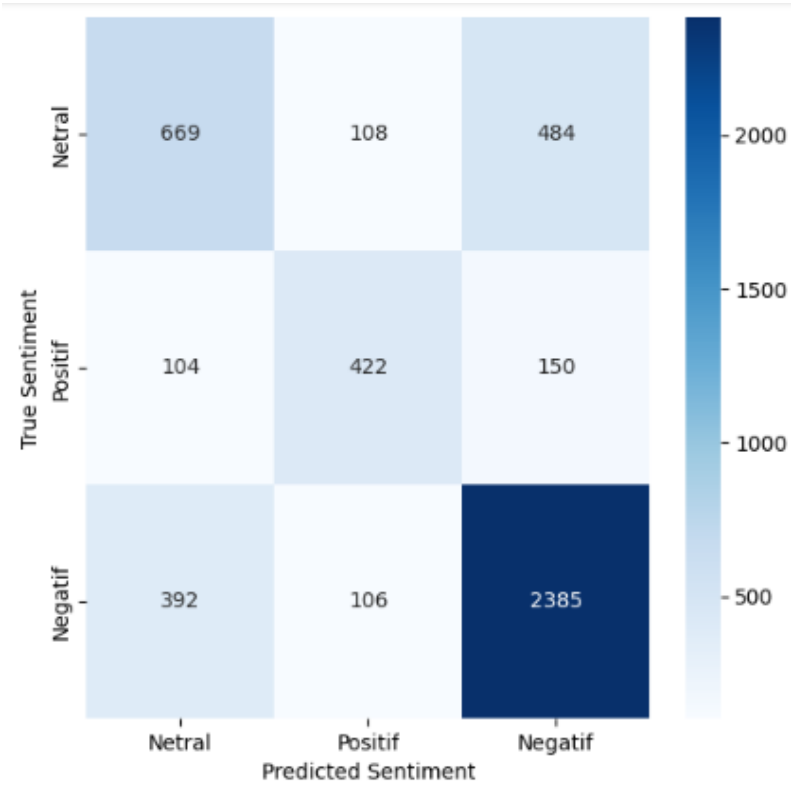
```
Test loss 0.8364062545117953 accuracy 0.7197095435684647
              precision    recall  f1-score   support

      netral       0.59      0.50      0.54      1261
     positif       0.64      0.64      0.64       676
     negatif       0.78      0.84      0.81      2883

    accuracy                           0.72      4820
   macro avg       0.67      0.66      0.66      4820
weighted avg       0.71      0.72      0.71      4820
```

Gambar 4.29 Hasil akurasi 15 Epoch

Hasil akurasi dari tiga percobaan tersebut mengindikasikan tingkat akurasi masing-masing sebesar 70%, 72%, dan 71%. Penurunan performa tersebut mungkin disebabkan oleh kecenderungan overfitting akibat jumlah epoch yang tinggi. Overfitting terjadi ketika model terlalu berfokus pada data pelatihan dan kehilangan kemampuan untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Fenomena ini mengakibatkan kemampuan prediksi model menjadi kurang akurat dan cenderung tidak mampu menghasilkan hasil yang konsisten. Lebih lanjut, pada hasil eksperimen dengan 10 epoch, Gambar 4.30 memperlihatkan kurva performa pelatihan,sementara Gambar 4.31 menggambarkan *confussion matrix* hasil prediksi.

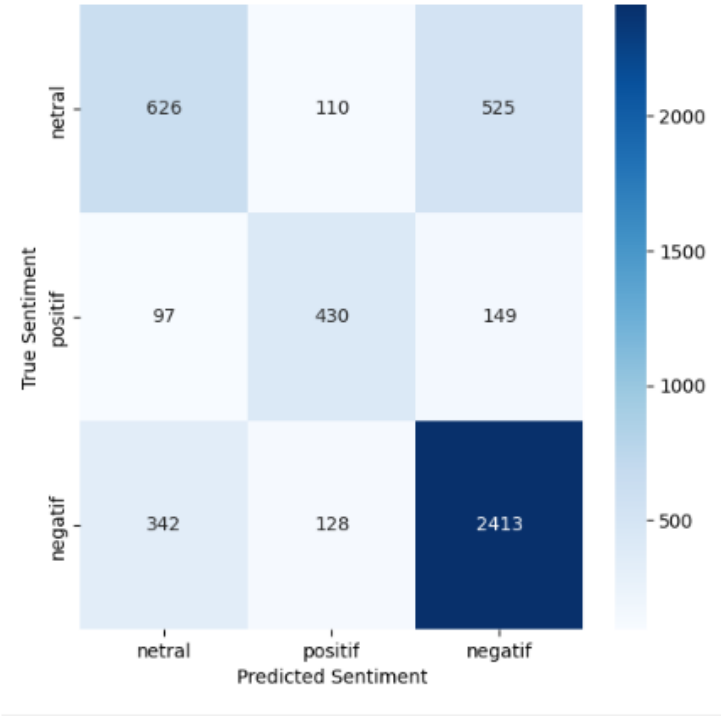


Gambar 4.30 Kurva performa training model

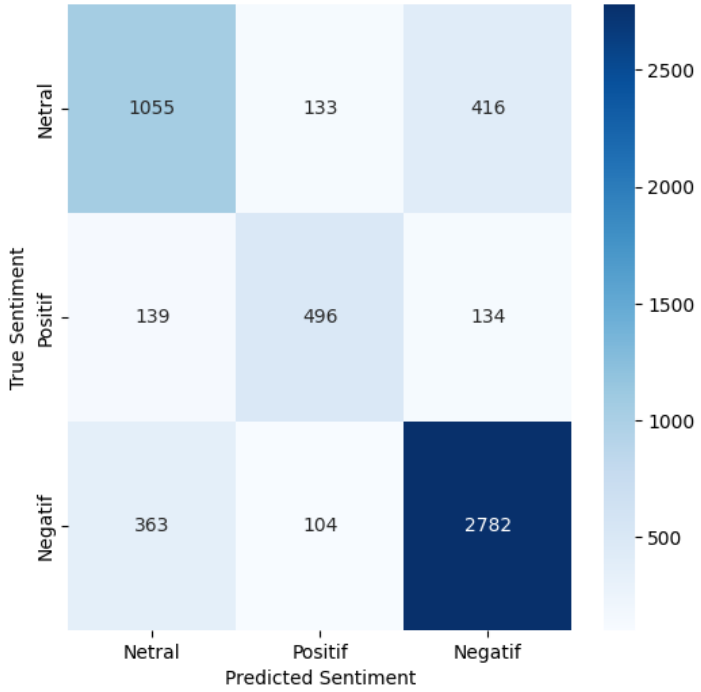


Gambar 4.31 Diagram confussion matrix

Dilanjutkan dengan percobaan keempat, dilakukan penyesuaian terhadap hyperparameter dan parameter lainnya, seperti learning rate (2e-5), dropout (0.003), dan batch size (16). Hasil dari percobaan keempat ditampilkan dalam Gambar 4.32. Selanjutnya, percobaan kelima melibatkan perubahan pada porsi data latih menjadi 50%, sementara data uji dan validasi masing-masing menjadi 25%. Hasil eksperimen ini diperlihatkan dalam Gambar 4.33.



Gambar 4. 32 Diagram confussion matrix Percobaan ke 4



Gambar 4. 33 Diagram confussion matrix Percobaan ke 5

Dalam eksperimen keempat, terjadi peningkatan hasil yang signifikan karena nilai pada diagonal dari matriks hasil prediksi sentimen mencapai puncak tertinggi, yaitu dengan 2413 sentimen negatif, 430 sentimen positif, dan 626 sentimen netral. Sementara itu, pada eksperimen kelima, hasil keseluruhan dari prediksi sentimen juga mengalami peningkatan, namun peningkatan ini dipengaruhi oleh penambahan data validasi dari 10% menjadi 25%. Oleh karena itu, pada eksperimen kelima, meskipun terjadi peningkatan akurasi dalam model prediksi sentimen, namun peningkatannya tidak begitu signifikan.

4.6 Evaluasi

4.6.1 Labeling Supervised Learning

Pada hasil labeling dataset menggunakan model Supervised-Learning, karena porsi dataset sangat tidak berimbang, dan kategori dataset terlalu banyak, penulis memutuskan untuk menambahkan data dengan sentimen positif ke dataset yang mana data dengan sentiment positif memang memiliki jumlah data yang lebih sedikit. Selain itu, juga dilakukan fine-tuning pada hyperparameter model untuk memastikan bahwa model memiliki performa yang optimal dalam melakukan analisis sentimen pada dataset yang sangat beragam ini.

Selain itu, penting untuk dicatat bahwa evaluasi hasil labeling menggunakan metode Supervised-Learning juga mencakup analisis lebih mendalam terhadap kualitas label yang diberikan oleh model. Penulis melakukan validasi silang (cross-validation) untuk mengukur stabilitas dan konsistensi hasil prediksi model terhadap berbagai subset data. Hal ini memungkinkan penuilis untuk mengidentifikasi potensi perbedaan kinerja model terhadap data yang berbeda, serta menilai apakah model mampu menggeneralisasi sentimen dengan baik pada data yang belum pernah dilihat sebelumnya. Proses validasi silang juga membantu mengatasi masalah overfitting dan menghasilkan estimasi akurasi yang lebih realistis pada situasi pengujian di dunia nyata.

Dalam evaluasi hasil labeling menggunakan metode Supervised-Learning, penuilis juga melakukan analisis lebih lanjut terhadap distribusi kesalahan (error analysis) yang dilakukan oleh model pada setiap kategori sentimen. Ini memberikan wawasan yang berharga tentang area di mana model memiliki performa yang baik dan di mana ia masih perlu ditingkatkan. Dengan memahami

jenis kesalahan yang umum terjadi, penuilis dapat mengambil tindakan yang lebih spesifik dalam merancang strategi perbaikan model, seperti peningkatan jumlah data pada kategori yang paling rentan terhadap kesalahan atau melakukan penyesuaian pada preprosesing data yang lebih tepat. Seluruh analisis ini menjadi dasar yang kuat dalam menilai kemampuan model dalam menghadapi data yang tidak terstruktur dan beragam, serta membantu penuilis mengambil langkah-langkah yang tepat untuk meningkatkan performa model dalam tugas analisis sentimen lebih lanjut.

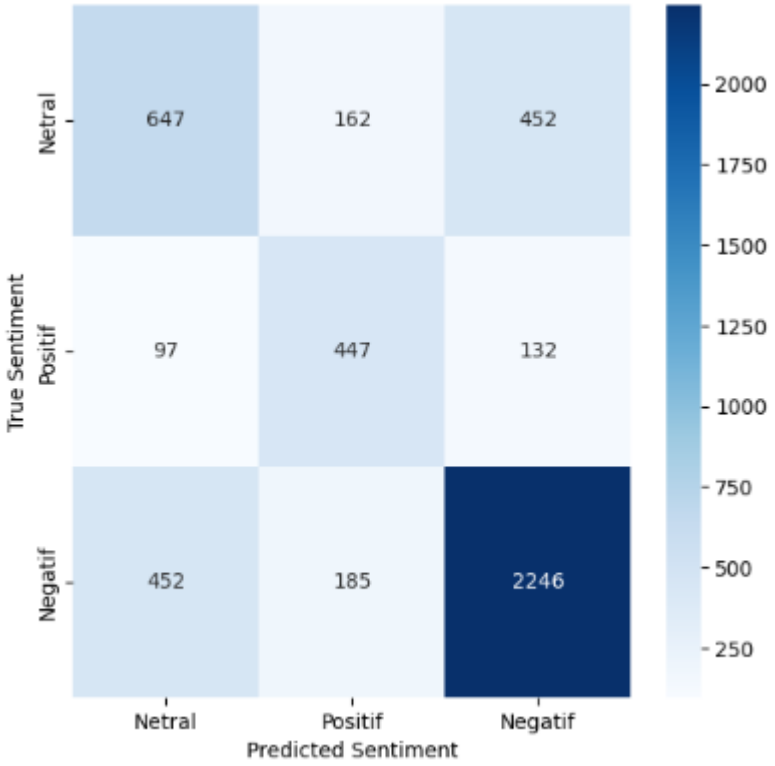
4.6.2 Implementasi INDOBERT

Pada bagian terakhir dari Bab 4, penuilis menyajikan hasil evaluasi dari model yang diimplementasikan. Sebagai perbandingan untuk hasil evaluasi yang didapatkan dari model INDOBERT. Pada penelitian ini penulist melakukan perbaingan dengan model BERT *Uncased*, model ini adalah base model dari INDOBERT dan mempunyai kinerja yang sangat baik dalam menganalisa sentiment dan klasifikasi. Sebagai perbandingan, telah dilakukan percobaan dengan model BERT dengan menggunakan data yang sama, dan parameter yang sama. Gambar 4.34 menunjukkan hasil yang diperoleh BERT yaitu 69% dibandingkan hasil yang di dapat dari model INDOBERT yang ditunjukkan pada gambar 4.8 mendapat hasil akurasi 72%.

```
Test loss 1.4093390743414693 accuracy 0.6929460580912863
              precision    recall  f1-score   support

      Netral       0.54      0.51      0.53      1261
     Positif       0.56      0.66      0.61       676
     Negatif       0.79      0.78      0.79      2883

    accuracy                           0.69      4820
   macro avg       0.63      0.65      0.64      4820
weighted avg       0.70      0.69      0.69      4820
```

Gambar 4.34 Hasil akurasi model BERT

Hal ini menunjukkan bahwa INDOBERT bekerja lebih baik untuk menganalisa sentimen Bahasa Indonesia. Dan hasil lainnya pada confussion matrix, pada gambar 4.35 Model BERT berhasil memprediksi sentimen dengan benar sebanyak 2246 sentimen negatif, 447 sentimen positif dan 647 sentimen netral. Berbanding dengan INDOBERT yang mendapat 2413 sentimen negatif, 430 sentimen positif, dan 626 sentimen netral pada gambar 4.32.



Gambar 4. 35 Diagram confussion matrix BERT

Dalam evaluasi model INDOBERT analisis sentimen berbahasa Indonesia, penuilis menemukan bahwa jumlah Epoch memiliki dampak terhadap akurasi prediksi. Terlalu banyak Epoch dapat mengakibatkan overfitting, di mana model "menghafal" data pelatihan dan gagal melakukan generalisasi pada data uji. Kemudian untuk penambahan porsi data uji, meskipun kenaikan akurasi terlihat, penambahan data validasi tidak selalu berdampak signifikan pada hasil akhir.

Pemahaman ini memberikan panduan penting untuk mengoptimalkan kinerja model dan menjaga keseimbangan antara akurasi dan generalisasi dalam analisis sentimen berbahasa Indonesia. Melalui evaluasi ini, penuilis mendapatkan pemahaman yang lebih mendalam tentang kinerja model dalam analisis sentimen berbahasa Indonesia, serta faktor-faktor yang dapat memengaruhi hasil prediksi dan akurasi model secara keseluruhan.