

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Hasil penelitian analisis sentimen dengan kata kunci “KAI Access” menggunakan metode NBC dan SVM untuk menganalisis data dari *review* berbahasa Indonesia baik positif, netral maupun negatif. Data Google Play Store yang diambil selama periode 23 Januari 2023 hingga 25 Juni 2023. Data yang digunakan selama training sebanyak 900 data *review* dengan data berlabel positif 300 data berlabel negatif 300 dan 300 data berlabel netral. Pada tahap testing menggunakan data sebanyak 450 data dengan 150 data berlabel positif, 150 data berlabel negatif dan 150 data berlabel netral. Analisis Sentimen *Review* Aplikasi KAI Access dari Google Play Store Menggunakan Perbandingan NBC dan SVM, dibangun menggunakan Bahasa Pemrograman Python dengan metode NBC dan SVM untuk mendapatkan hasil perhitungan sentimen menggunakan framework Flask. Penelitian ini akan memberikan informasi tentang *review* analisis sentiment pengguna KAI Access di Google Play Store.

4.2 PENGAMBILAN DATA GOOGLE PLAY STORE

Pengambilan data ini dilakukan di media Google Play Store untuk mendapatkan data *review* yang berkaitan dengan kata kunci “KAI Access”. Data *review* yang diambil mulai dari tanggal 23 Januari 2023 hingga 25 Juni 2023 dengan 5.000 data *review* pengguna. Proses pengambilan data *review* dilakukan menggunakan Google Colab. Data *review* yang diambil berformat CSV sehingga dapat dibuka di Microsoft Excel.

4.3 PENGUMPULAN DATA GOOGLE PLAY STORE

Pengumpulan data dilakukan dengan Bahasa pemrograman Python. Python yang digunakan untuk proses pengumpulan data *review* kemudian disimpan sebagai file *comma separated values* (CSV). Dalam pengambilan

data *review* yang dihasilkan memiliki variasi dalam berbagai bentuk termasuk angka, username. Sehingga diperlukan proses *preprocessing* untuk mendapatkan data *review* yang diinginkan. Contoh data *review* yang sudah diambil ditunjukkan pada table 4.1

Tabel 4.1 Data *Review*

No	Review
1.	Kenapa tidak bisa melakukam pembayaran melalui LinkAja ?? Padahal samgat memudahkan. Pake Qris masih bermasalah. Please respon
2.	Aplikasi BUMN kok nyusain? 403 for Biden apa kali. Daftar berhasil. Masuk berhasil. Tapi isi data went wrong for Biden. Udah 10 kali bikin email sama beli nomor hp. Akhirat bakal disusain lu nanti!
3.	Sangat mengecewakan. Jelek jelek jelek jelek
4.	APLIKASI RUSAK, SAYA UDAH BUY TIKET, UDAH DI BOARDING MASIH GABISA MASUK, RUWET BANYAK ATURAN
5.	Makin parah nih aplikasi banyak banget gangguan nya, gimana mau ningkatin pelayanan kalo masalah gini aja terus ke ulang2
6.	APLIKASI MODEL OPO IKI GAK JELASS BLASSS..
7.	Ngotak lah tod, di KAI udah ada data terkait Jumlah vaksin yang diterima oleh calon penumpang, tetapi buat yang vaksin 2 masih dikasih akses buat beli Tiket, tapi ujung2nya ditolak
8.	Aplikasi gak guna

Tabel 4.1 menunjukkan hasil pengambilan data *review* yang belum terstruktur dan memiliki variasi bentuk, sehingga diperlukan beberapa *preprocessing* karena masih banyak variasi bentuk yang perlu dihilangkan karena tidak digunakan dalam pembuatan analisis sentimen agar menjadi data yang terstruktur.

4.4 CLEANING DATA

Tujuan dari tahap *cleaning* data adalah untuk menghilangkan data yang tidak diperlukan seperti tanda baca, angka dan simbol dari kalimat. Contoh data *Cleaning* data yang sudah diambil ditunjukkan pada table 4.2.

Tabel 4.2 Data *Cleaning*

No	Review
1	kenapa tidak bisa melakukam pembayaran melalui linkaja padahal sangat memudahkan pake qris masih bermasalah please respon
2	aplikasi bumn kok nyusain kali daftar berhasil masuk berhasil tapi isi biden udah kali bikin email sama beli nomor hp akhirat bakal disusain lu nanti
3	sangat mengecewakan jelek jelek jelek jelek
4	aplikasi model opo iki gak jelaass blasss
5	aplikasi gak guna
6	Buruk
7	aplikasi smpah di buat registrasi malah ngebug
8	ga guna apk nya menyusahkan orang dan ribet amat

4.5 TOKENIZATION DATA

Data *Tokenization* bertujuan untuk memisahkan kata dari kalimat sehingga kata tersebut dapat berdiri sendiri. Contoh data *tokenization* yang sudah diambil ditunjukkan pada table 4.3.

Tabel 4.3 Data *Tokenization*

Review
1,"['kenapa', 'tidak', 'bisa', 'melakukan', 'pembayaran', 'melalui', 'linkaja', 'padahal', 'sangat', 'memudahkan', 'pake', 'qris', 'masih', 'bermasalah', 'please', 'respon']"
2,"['aplikasi', 'bumn', 'kok', 'nyusain', 'apa', 'kali', 'daftar', 'berhasil', 'masuk', 'berhasil', 'tapi', 'isi', 'data', 'kali', 'bikin', 'email', 'sama', 'beli', 'nomor', 'hp', 'akhirat', 'bakal', 'disusain', 'lu', 'nanti']"
3,"['sangat', 'mengecewakan', 'jelek', 'jelek', 'jelek', 'jelek']"
4,"['aplikasi', 'model', 'opo', 'iki', 'gak', 'jelaass', 'blasss']"

5,"['aplikasi', 'gak', 'guna']"
6,['buruk']
7,"['aplikasi', 'smpah', 'di', 'buat', 'registrasi', 'malah', 'ngebug']"
8,"['ga', 'guna', 'apk', 'nya', 'menyusahkan', 'orang', 'dan', 'ribet', 'amatttt']"

4.6 STOPWORD REMOVAL

Data *Stopword Removal* digunakan untuk menghapus kata-kata yang sering dipakai namun kurang bermakna banyak. Contoh *stopword removal* yang sudah diambil ditunjukkan pada table 4.4.

Tabel 4.4 *Stopword Removal*

Review
1,"['tidak', 'melakukam', 'pembayaran', 'melalui', 'linkaja', 'padahal', 'sangat', 'memudahkan', 'pake', 'qris', 'bermasalah', 'please', 'respon']"
2,"['aplikasi', 'bumn', 'kok', 'nyusain', 'apa', 'kali', 'daftar', 'berhasil', 'masuk', 'berhasil', 'isi', 'data', 'udah', 'kali', 'bikin', 'email', 'sama', 'beli', 'nomor', 'hp', 'akhirat', 'bakal',
3,"['sangat', 'mengecewakan', 'jelek', 'jelek', 'jelek', 'jelek']"
4,"['aplikasi', 'model', 'opo', 'iki', 'gak', 'jelaass', 'blasss']"
5,"['aplikasi', 'gak']"
6,['buruk']
7,"['aplikasi', 'smpah', 'buat', 'registrasi', 'malah', 'ngebug']"
8,"['ga', 'apk', 'nya', 'menyusahkan', 'orang', 'ribet', 'amatttt']"

4.7 STEMMING DATA

Stemming Data bertujuan untuk mengubah kata menjadi bentuk kata dasarnya. Contoh *stemming data* yang sudah diambil ditunjukkan pada table 4.5

Tabel 4.5 *Stemming Data*

1,['tidak melakukam bayar lalu linkaja padahal sangat mudah pake qris masalah please respon']
2,['aplikasi bumn kok nyusain kali daftar hasil masuk hasil isi data kali bikin email sama beli nomor hp akhirat bakal disusain lu']

3,['sangat kecewa jelek jelek jelek jelek']
4,['aplikasi model opo iki gak jelaass blasss']
5,['aplikasi gak']
6,['buruk']
7,['aplikasi smpah buat registrasi malah ngebug']
8,['ga apk nya susah orang ribet amat']

4.8 HASIL PREPROCESSING DATA

Hasil *Preprocessing* data dilakukan untuk proses pengolahan data yang berfungsi memperbaiki data *review* yang belum terstruktur dengan melakukan tahapan-tahapan agar menjadi data yang terstruktur. Contoh hasil data *review* yang sudah dilakukan proses *preprocessing* agar dapat digunakan untuk melakukan perhitungan di tahap selanjutnya ditunjukkan pada table 4.6.

Tabel 4.6 Data Hasil *Preprocessing*

No	Review
1.	kenapa tidak bisa melakukam pembayaran melalui linkaja padahal samgat memudahkan pake qris masih bermasalah please respon
2.	aplikasi bumnn kok nyusain kali daftar berhasil masuk berhasil tapi isi data kali bikin email sama beli nomor hp akhirat bakal disusain lu nanti
3.	sangat mengecewakan jelek jelek jelek jelek
4.	aplikasi model opo iki gak jelaass blasss
5.	aplikasi gak guna
6.	Buruk
7.	aplikasi smpah di buat registrasi malah ngebug
8.	ga guna apk nya menyusahkan orang dan ribet amat

4.8.1 Pelabelan Manual Naïve Bayes Classifier

Pelabelan manual merupakan proses memberikan laba positif, negative, dan netral terhadap data *review* yang sudah dilakukan proses *preprocessing*. Proses pelabelan manual dilakukan di Microsoft excel secara

manual agar data *review* dapat di analisis. Data *review* yang sudah diberi label di dapatkan 900 data *review* yang digunakan untuk training dengan data berlabel positif 300 data *review*, berlabel netral 300 data *review*, dan berlabel negatif 300 dan pada tahap *testing* didapatkan 450 data *review* 150 berlabel positif, 150 berlabel negatif dan 150 berlabel netral data *review* dari metode NBC dan. Contoh data *review* yang sudah dilakukan pelabelan manual ditunjukkan pada table 4.7.

Tabel 4.3 Pelabelan Manual NBC

No.	Review	Kelas	Label
1.	kenapa tidak bisa melakukam pembayaran melalui linkaja padahal sangat memudahkan pake qris masih bermasalah please respon	Negatif	-1
2.	aplikasi bumnn kok nyusain kali daftar berhasil masuk berhasil tapi isi data udah kali bikin email sama beli nomor hp akhirat bakal disusain lu nanti	Negatif	-1
3.	sangat mengecewakan jelek jelek jelek jelek	Negatif	-1
4.	aplikasi model opo iki gak jelaass blasss	Negatif	-1
5.	aplikasi gak guna	Negatif	-1
6.	Buruk	Negatif	-1
7.	aplikasi sampah di buat registrasi malah ngebug	Negatif	-1
8.	ga guuna apk nya menyusahkan orang dan ribet amat	Negatif	-1

Tabel 4.7 Menunjukkan hasil pelabelean manual bahwa data *review* dengan kelas positif diberi label 1, kelas negative diberi label -1, dan kelas netral diberi label 0. Data pelabelan ini digunakan untuk memberikan nilai sentiment yang dihitung akurasinya.

4.8.2 Pelabelan Manual Support Vector Machine

Tabel 4.3 Menunjukkan hasil pelabelan manual bahwa data review dengan kelas positif diberi label 1, kelas negative diberi label -1, dan kelas netral diberi label 0. Data pelabelan ini digunakan untuk memberikan nilai sentiment yang dihitung akurasi.

Pelabelan manual merupakan proses memberikan label positif, negatif, dan netral terhadap data *review* yang sudah dilakukan proses *preprocessing*. Proses pelabelan manual dilakukan di Microsoft excel secara manual agar data *review* dapat di analisis. Data *review* yang sudah diberi label di dapatkan 900 data *review* yang digunakan untuk training dengan data berlabel positif 300 data review, berlabel netral 300 data review, dan berlabel negatif 300 data *review* dan pada tahap *testing* didapatkan 450 data *review* 150 berlabel positif, 150 berlabel negatif dan 150 berlabel netral dari metode SVM. Contoh data review yang sudah dilakukan pelabelan manual ditunjukkan pada table 4.8.

Tabel 4.4 Pelabelan Manual SVM

No.	Review	Kelas	Label
1.	kenapa tidak bisa melakukam pembayaran melalui linkaja padahal sangat memudahkan pake qris masih bermasalah please respon	Negatif	-1
2.	aplikasi bumh kok nyusain kali daftar berhasil masuk berhasil tapi isi data udah kali bikin email sama beli nomor hp akhirat bakal disusain lu nanti	Negatif	-1
3.	sangat mengecewakan jelek jelek jelek jelek	Negatif	-1
4.	aplikasi model opo iki gak jelaass blasss	Negatif	-1
5.	aplikasi gak guna	Negatif	-1
6.	buruk	Negatif	-1
7.	aplikasi sampah di buat registrasi malah ngebug	Negatif	-1

8.	ga guuna apk nya menyusahkan orang dan ribet amat	Negatif	-1
----	---	---------	----

4.8.3 Training

Training data merupakan proses untuk menghasilkan model klasifikasi secara otomatis dapat digunakan untuk proses klasifikasi, proses *training* menggunakan metode NBC dan SVM dengan fungsi ekstraksi TF-IDF pada data teks, dengan jumlah data sebanyak 900 data *review*.

Perhitungan TF-IDF, dilakukan secara manual menggunakan 8 data *review* d1, d2, d3, d4, d5, d6, d7, d8 dan terdapat beberapa komponen seperti T yaitu kata, D yaitu kalimat atau data *review* untuk mengetahui berapa banyak data dimana suatu kata muncul. Perhitungan TF-IDF ditunjukkan pada table 4.9.

Tabel 4.9 Perhitungan TF-IDF

term/kata	d1	d2	d3	d4	d5	d6	d7	d8	df	idf
bisa	1								1	0,90308999
link	1								1	0,90308999
qris	1								1	0,90308999
please	1								1	0,90308999
respon	1								1	0,90308999
aplikasi		1		1	1	1			4	0,30103
bumn		1							1	0,90308999
daftar		1							1	0,90308999
berhasil		2							2	0,60205999
data		1							1	0,90308999
email		1							1	0,90308999
jelek			4						4	0,30103
model				1					1	0,90308999
jelas				1					1	0,90308999
guna					1			1	2	0,60205999
buruk						1			1	0,90308999

sampah							1		1	0,90308999
registrasi							1		1	0,90308999
ngebug							1		1	0,90308999
ribet								1	1	0,90308999

Tabel 4.10 Perhitungan TF-IDF

TF*IDF									
term/kata	d1	d2	d3	d4	d5	d6	d7	d8	
Bisa	0,90308999								
Link	0,90308999								
Qris	0,90308999								
Please	0,90308999								
Respon	0,90308999								
Aplikasi		0,30103		0,30103	0,301029996	0,301029996			
Bumn		0,90308999							
Daftar		0,90308999							
Berhasil		1,2041998							
Data		0,90308999							
Email		0,90308999							
Jelek			1,2041998						
Model				0,90308999					

Jelas				0,90 308 999				
Guna					0,602 0599 91			0,602 0599 91
Buruk						0,903 0899 9		
Sampah							0,903 0899 87	
Registrasi							0,903 0899 87	
Ngebug							0,903 0899 87	
Ribet								0,903 0899 87

4.8.4 Testing

Testing merupakan proses yang digunakan untuk memprediksi kelas dan label berdasarkan data *training* yang sudah dibangun untuk mengetahui tingkat keakuratan pemodelan dengan jumlah *review* sebanyak 450 data *review*.

4.8.5 Hasil Klasifikasi Naïve Bayes Classifier

Melakukan data klasifikasi data *training* menggunakan perhitungan *confusion matrix* untuk menentukan nilai aktual dan nilai prediksi berdasarkan akurasi yang diberikan oleh aplikasi. Penelitian ini memiliki 3 kelas sentimen yaitu positif, negatif, dan netral. Data *training* yang akan dihitung adalah 900 data *review* yang diberi label secara manual sebagai positif, negatif, dan netral. *Multiclass confusion matrix* memiliki istilah yang mewakili hasil dari proses klasifikasi diantaranya true positif (Aktual Post), true negatif (Aktual Neg), true netral (Aktual Net), false positif (Aktual

Post), dan valse negatif (Aktual Neg), valse netral (Aktual Net). Hasil perhitungan *multiclass confusion* ditunjukkan pada table 4.11.

Tabel 4.5 Hasil Perhitungan *Confusion Matrix* NBC

Kelas Aktual	Kelas Prediksi		
	Positive	Negatif	Netral
Positif	198	15	210
Negatif	35	87	18
Netral	96	55	180

4.8.6 Hasil Klasifikasi Support Vector Machine

Melakukan data klasifikasi data *training* menggunakan perhitungan *confusion matrix* untuk menentukan nilai actual dan nilai prediksi berdasarkan akurasi yang diberikan oleh aplikasi. Penelitian ini memiliki 3 kelas sentimen yaitu positif, negatif, dan netral. Data *training* yang akan dihitung adalah 450 data *review* yang diberi label secara manual sebagai positif, negatif, dan netral. *Multiclass confusion matrix* memiliki istilah yang mewakili hasil dari proses klasifikasi diantaranya true positif (Aktual Post), true negatif (Aktual Neg), true netral (Aktual Net), valse positif (Aktual Post), dan valse negatif (Aktual Neg), valse netral (Aktual Net). Hasil perhitungan *multiclass confusion* ditunjukkan pada table.4.12.

Tabel 4.6 Hasil Perhitungan *Confusion Matrix* SVM

Kelas Aktual	Kelas Prediksi		
	Positive	Negatif	Netral
Positif	98	3	32
Negatif	10	64	19
Netral	41	12	121

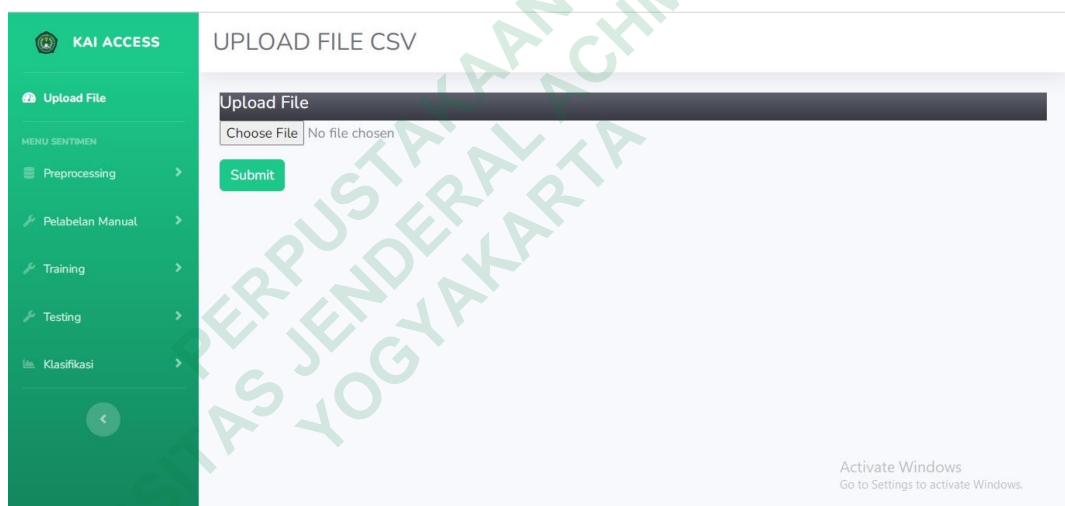
4.9 IMPLEMENTASI DESAIN INTERFACE

Tampilan antarmuka yang digunakan pengguna untuk berinteraksi secara langsung disebut dengan interface. Dalam penelitian Analisis Sentimen Google Play Store *review* aplikasi KAI Access dengan membandingkan metode NBC dan

SVM. Bahasa Pemrograman yang digunakan adalah Python dengan menggunakan framework Flask SQLAlchemy dan pengolah database menggunakan MySQL. Berikut ini beberapa kode dan hasil implementasi halaman yang terdapat analisis sentimen.

4.9.1 Implementasi Halaman Upload File

Implementasi halaman upload file data KAI Access digunakan untuk melakukan upload file dan menampilkan file CSV yang sudah terupload ke dalam *directory* agar memudahkan proses *preprocessing*, *training*, *testing* dan klasifikasi. Implementasi halaman upload file KAI Access ditunjukkan pada Gambar 4.1.



Gambar 4.1 Implementasi halaman upload file

Berikut ini potongan kode yang berfungsi untuk melakukan upload file.

```
# UPLOAD DATA MENTAH
@app.route('/', methods=["POST", "GET"])
def uploadFile():

    if request.method == 'POST':

        # Mengunggah file menggunakan Flask
        upload_file = request.files['file']

        # Mendapatkan nama file data yang diunggah
        data_filename = secure_filename(upload_file.filename)

        # Mengunggah file menggunakan Flask ke direktori yang
        ditentukan (UPLOAD_FOLDER dalam konfigurasi app)
```

```
upload_file.save(os.path.join(app.config['UPLOAD_FOLDER'],d
ata_filename))
```

4.9.2 Implementasi Halaman *Preprocessing* (Data *Cleaning*)

Implementasi halaman *preprocessing* data *cleaning* digunakan untuk menghapus atau menghilangkan kata yang tidak diperlukan seperti tanda baca, angka dan simbol pada suatu kalimat.



Gambar 4.2 Implementasi Halaman *Preprocessing* Data *Cleaning*

Berikut ini potongan kode yang berfungsi untuk melakukan preprocessing data data cleaning

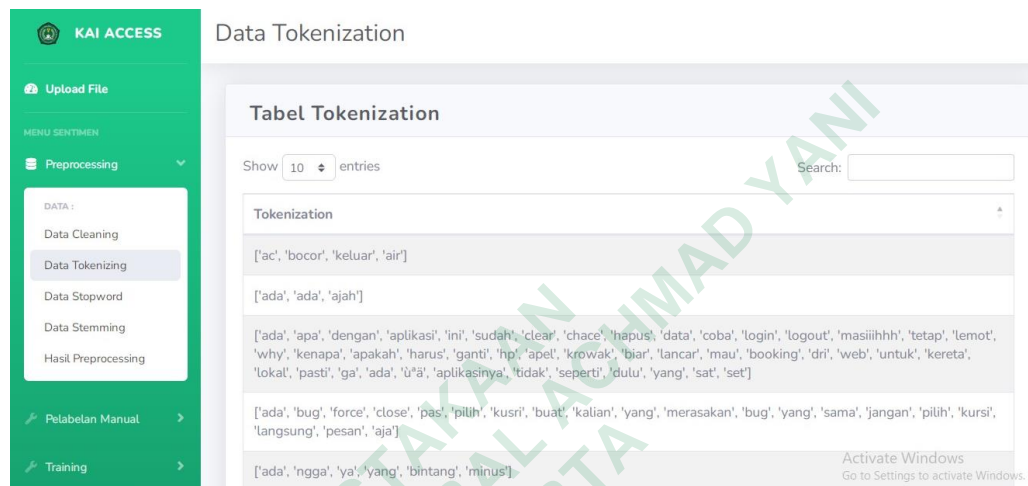
```
# fungsi untuk menjalankan cleaning data, dengan function
cleaning text
def cleaning_text(data):
# Mengganti kata yang dimulai dengan b'@ dan mengikuti karakter-
karakter lainnya dengan spasi
data = re.sub(r"b'\@[\\w]*", ' ', str(data))

# Mengganti kata yang dimulai dengan b' dan mengikuti karakter-
karakter lainnya dengan spasi
data = re.sub(r"b'[\\w]*", ' ', data)

# Mengganti URL yang dimulai dengan https: dengan spasi
data = re.sub(r'https\:.*$', " ", data)
```

4.9.3 Implementasi Halaman Preprocessing (Data Tokenization)

Implementasi halaman *preprocessing* data *tokenization* bertujuan untuk memisahkan kata dari kalimat sehingga kata tersebut dapat berdiri sendiri. Implementasi halaman *preprocessing* data *tokenization* ditunjukkan pada Gambar 4.3.



Gambar 4.3 Halaman *Preprocessing* Data *Tokenization*

Berikut ini potongan kode yang berfungsi untuk melakukan *preprocessing* data *tokenization*

```
# Proses Tokenisasi
tokenz =
pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"data_clean.csv"))
text = tokenz["content"]
data_tokenz = []
for i in text:
    data_tokenz.append(tokenizing(i))

# convert data Tokenisasi ke csv
text_tokenz = pd.DataFrame()
text_tokenz['content'] = data_tokenz

text_tokenz.to_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"data_tokenz.csv"))
```

4.9.4 Implementasi Halaman Preprocessing (Stopword Removal)

Implementasi halaman *preprocessing* data *stopword removal* digunakan untuk menghapus kata yang sering dipakai namun kurang

bermakna banyak. Implementasi halaman *preprocessing* data *stopword removal* ditunjukkan pada Gambar 4.4.



Gambar 4.4 Halaman *Preprocessing Stopword Removal*

Berikut ini potongan kode yang berfungsi untuk melakukan *preprocessing* data *stopword removal*

```
# Proses Penghilangan Stopword
Stopword=pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"data_clean.csv"))
    text = stopword["content"]
    data_stopword = []
    for i in text:
        data_stopword.append(filltering(i))
# convert data Stopword ke csv
text_stopword = pd.DataFrame()
text_stopword['content'] = data_stopword

text_stopword.to_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"stopword.csv"))
```

4.9.5 Implementasi Halaman *Preprocessing (Stemming Data)*

Implementasi halaman *preprocessing* data *stemming* bertujuan untuk mengubah kata menjadi bentuk kata dasarnya. Implementasi halaman *preprocessing* data *stemming* ditunjukkan pada Gambar 4.5.



Gambar 4.5 Halaman *Preprocessing Stemming* Data

Berikut ini potongan kode yang berfungsi untuk melakukan *preprocessing data stemming*

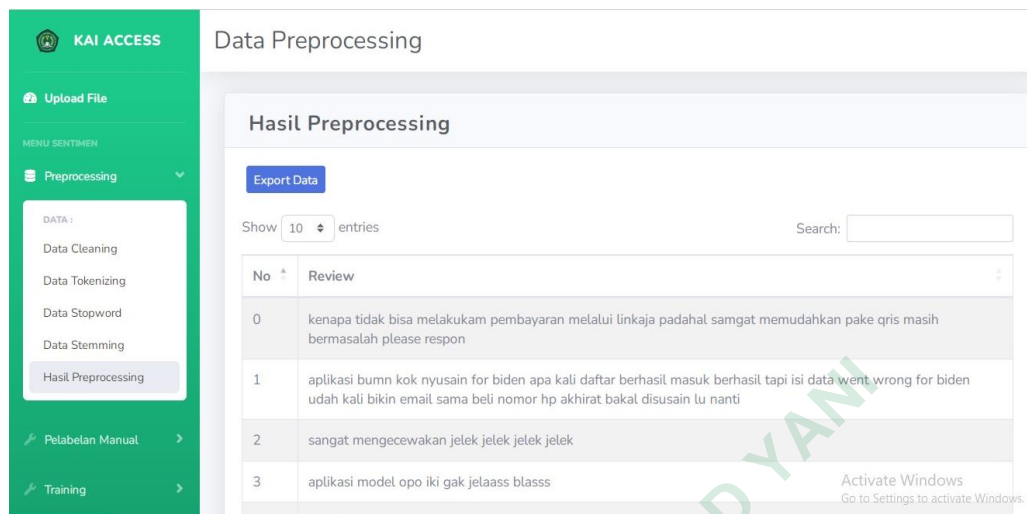
```
# Proses Stemming
stemmed = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"stopword.csv"))
text = stemmed["content"].head(50)
data_stemming = []
for i in text:
    data_stemming.append(stemming(i))

# convert data Stemming ke csv
txt_stemming = pd.DataFrame()
txt_stemming['text_stemming'] = data_stemming

txt_stemming.to_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"stemming.csv"))
```

4.9.6 Implementasi Halaman Hasil Preprocessing

Implementasi halaman hasil *preprocessing* merupakan proses pengolahan data *review* yang ada dengan melakukan langkah-langkah untuk menyempurnakan data *review* yang masih memiliki komponen sehingga menjadi data yang diinginkan. Implementasi halaman hasil *preprocessing* ditunjukkan pada Gambar 4.6.



Gambar 4.6 Halaman Hasil *Preprocessing*

Berikut ini potongan kode yang berfungsi untuk melakukan hasil *preprocessing*

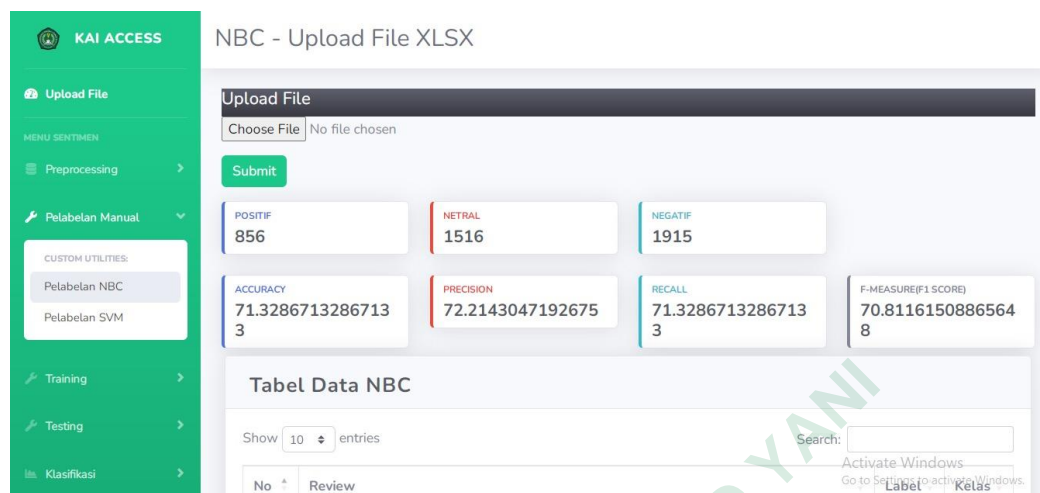
```
# MENAMPILKAN DATA DI HASIL PREPROCESSING
@app.route('/dataset')
def show_dataset():
    # Membaca data yang telah diproses dari file CSV

    uploaded_df=pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
    'data_preprocessing.csv'))
    uploaded_df_html = uploaded_df.values

    results = []
    for i in uploaded_df_html[0:8]:
        results.append(i[1]) # Menyimpan isi kolom 'content' ke dalam list
    results
```

4.9.7 Implementasi Halaman Preprocessing Pelabelan Manual NBC

Implementasi halaman pelabelan manual NBC digunakan untuk menampilkan hasil *Accuracy*, *Precision*, *Recall* dan *F1-score* berdasarkan data *review* yang sudah diberi kelas, label terhadap kalimat yang ada pada data *review* secara manual. Proses pelabelan manual dilakukan menggunakan Microsoft Excel. Implementasi halaman pelabelan manual NBC ditunjukkan pada Gambar 4.7.



Gambar 4.7 Halaman Pelabelan Manual NBC

Berikut ini potongan kode yang berfungsi untuk melakukan pelabelan manual naïve bayes classifier

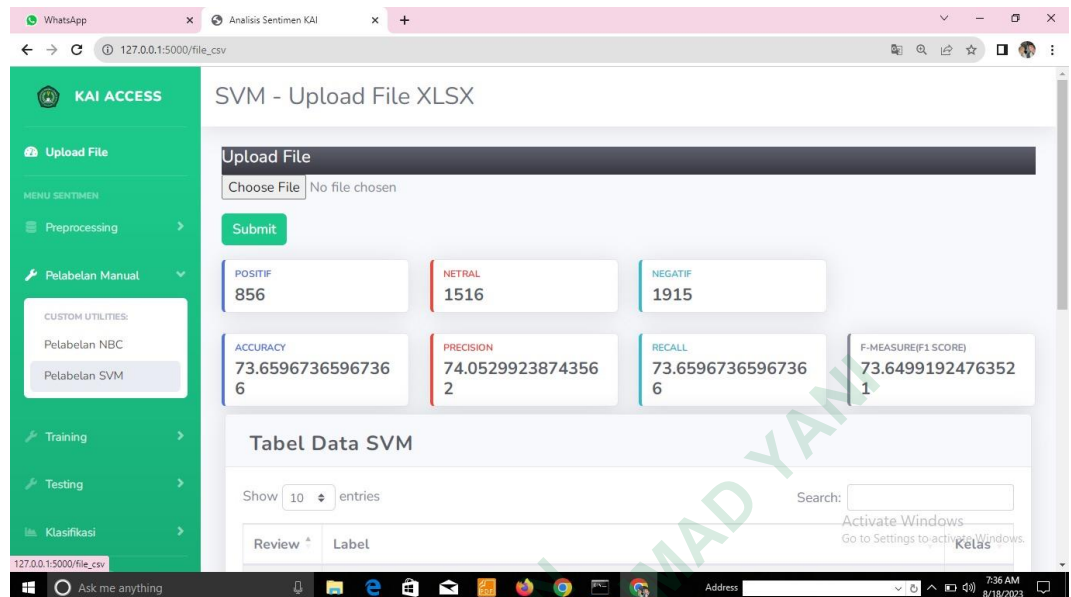
```
# PELABELAN MANUAL NBC
@app.route('/file_excel', methods=["POST", "GET"])
def uploadExcel():
    if request.method == 'POST':
        # Mengunggah file excel
        upload_file = request.files['file']
        data_filename = secure_filename(upload_file.filename)

        upload_file.save(os.path.join(app.config['UPLOAD_FOLDER'],
        "pelabelan_manual_nbc.xlsx"))

        # Membaca file excel yang diunggah dan menyimpan dalam format CSV
        uploaded_df=pd.read_excel(os.path.join(app.config['UPLOAD_FOLDER']
        ], "pelabelan_manual_nbc.xlsx"))
```

4.9.8 Implementasi Halaman Pelabelan Manual SVM

Implementasi Halaman pelabelan manual SVM digunakan untuk menampilkan hasil *Accuracy*, *Precision*, *Recall* dan *F-Measure* berdasarkan data *review* yang sudah diberi kelas, label terhadap kalimat yang ada pada data *review* secara manual. Proses pelabelan manual dilakukan menggunakan Microsoft Excel. Implementasi halaman pelabelan Manual ditunjukkan pada Gambar 4.8.



Gambar 4.8 Halaman Pelabelan Manual SVM

Berikut potongan kode yang berfungsi untuk melakukan pelabelan manual svm

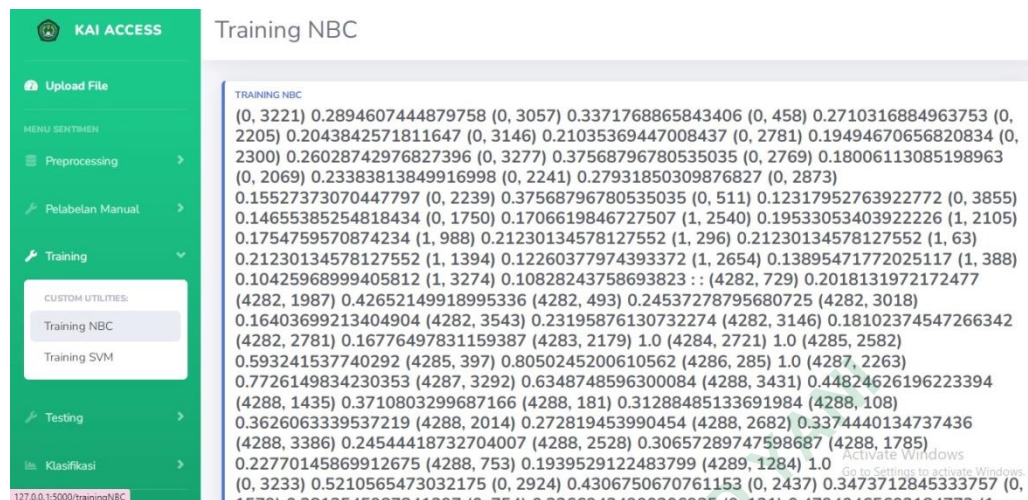
```
# PELABELAN MANUAL SVM
@app.route('/file_csv', methods=["POST", "GET"])
def uploadCSV():
    if request.method == 'POST':
        # Mengunggah file CSV yang diunggah oleh pengguna
        upload_file = request.files['file']
        data_filename = secure_filename(upload_file.filename)

        upload_file.save(os.path.join(app.config['UPLOAD_FOLDER'],
        "pelabelan_manual_svm.xlsx"))

        # Membaca data yang diunggah dari file Excel
        uploaded_df=pd.read_excel(os.path.join(app.config['UPLOAD_FOLDER']
        ], "pelabelan_manual_svm.xlsx"))
```

4.9.9 Implementasi Halaman Training NBC

Implementasi halaman *training* NBC digunakan untuk membuat model klasifikasi yang digunakan untuk mengklasifikasi sentiment secara otomatis. Langkah-langkah dalam training data menggunakan metode Naïve Bayes Classifier dan *library multinominal* NBC dengan fitur ekstrasi menggunakan (TF-IDF) pada data teks. Implementasi halaman training data NBC ditunjukkan pada Gambar 4.9.



Gambar 4.9 Halaman Training Data NBC

Berikut potongan kode yang berfungsi untuk melakukan *training* data naïve bayes classifier

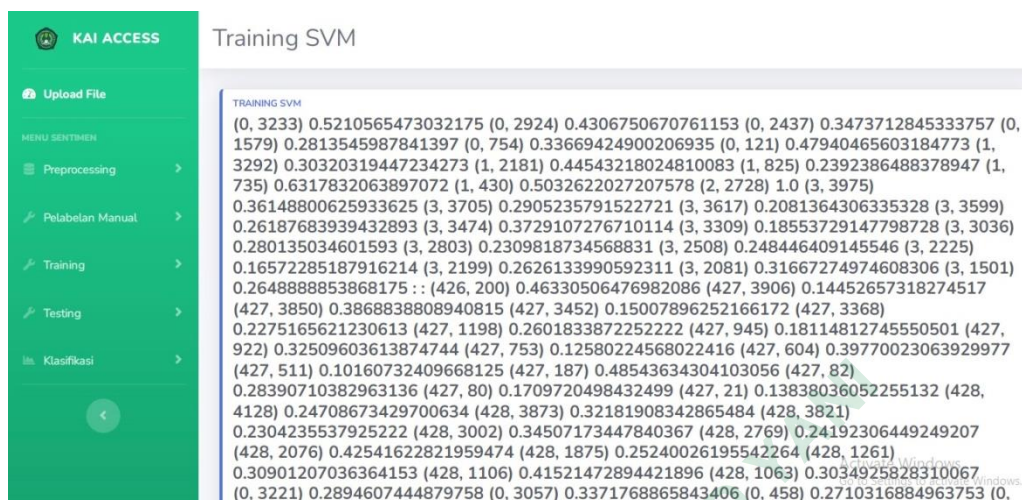
```
# TRAINING NBC
@app.route('/trainingNBC', methods=['GET', 'POST'])
def training_TFIDF():

# Membaca data dari file CSV yang telah diproses
uploaded_df=
pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
"pelabelan_manual_nbc.csv"),encoding='latin-1',
on_bad_lines='skip')
    uploaded_df.dropna()

# Memisahkan data menjadi fitur (X) dan label (Y)
X = uploaded_df['content']
Y = uploaded_df['label']
```

4.9.10 Implementasi Halaman Training SVM

Implementasi halaman *training* SVM digunakan untuk membuat model klasifikasi yang digunakan untuk mengklasifikasi sentiment secara otomatis. Langkah-langkah dalam *training* data dengan metode SVM dan *library multinominal* Support Vector Machine dengan fitur ekstraksi menggunakan (TF-IDF) pada data teks. Implementasi halaman *training* data SVM ditunjukkan pada Gambar 4.10.



Gambar 4.10 Halaman *Training SVM*

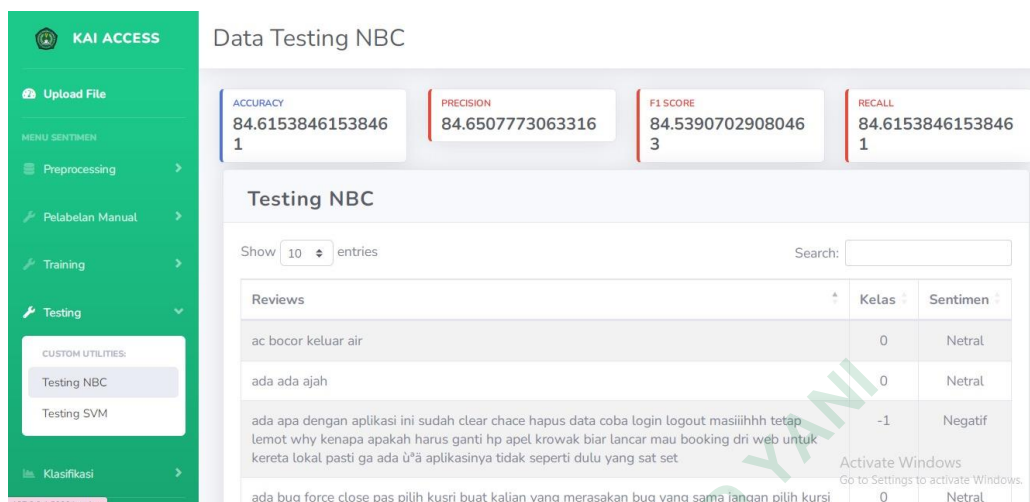
Berikut ini potongan kode yang berfungsi untuk melakukan *training* data support vector machine

```
# TRAINING SVM
@app.route('/trainingSVM', methods=['GET', 'POST'])
def trainingSVM():
    # Membaca data dari file CSV yang telah diunggah
    uploaded_df=pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
    "pelabelan_manual_svm.csv"),encoding='latin-1',
    on_bad_lines='skip')
    uploaded_df.dropna()

    # Memisahkan data menjadi fitur (X) dan label (Y)
    X = uploaded_df['content']
    Y = uploaded_df['label']
```

4.9.11 Implementasi Halaman *Testing Data NBC*

Implementasi halamann *testing* data NBC digunakan untuk menampilkan hasil *Accuracy*, *Precision*, *Recall* dan *F-Measure* berdasarkan data *review* yang sudah diprediksi kelas dan labelnya secara otomatis, untuk menentukan seberapa akurat model yang dibangun pada *training* data. Implementasi halaman *testing* data NBC ditunjukkan pada Gambar 4.11.



Gambar 4.11 Halaman *Testing* Data NBC

Berikut ini potongan kode yang berfungsi untuk melakukan *testing* data NBC

```
# TESTING NBC
@app.route('/testing', methods=["GET"])
def data_testing():
    # Membaca data dari file CSV yang telah diproses
    all_data = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
    "data_preprocessing.csv"), header=0)

    # Membuka model NBC yang telah dilatih sebelumnya
    model = open('data.pickle', 'rb')
    nbc_classifier = pickle.load(model)
```

4.9.12 Implementasi Halaman *Testing* SVM

Implementasi halamann *testing* data SVM digunakan untuk menampilkan hasil *Accuracy*, *Precision*, *Recall* dan *F-Measure* berdasarkan data *review* yang sudah diprediksi kelas dan labelnya secara otomatis, untuk menentukan seberapa akurat model yang dibangun pada *training* data. Implementasi halaman *testing* data SVM ditunjukkan pada Gambar 4.12.

ACCURACY
89.74358974358975

PRECISION
89.78965497697257

F1 SCORE
89.75620617534872

RECALL
89.74358974358975

Testing SVM

Show 10 entries Search:

Reviews	Kelas	Sentimen
ac bocor keluar air	0	Netral
ada ada ajah	0	Netral
ada apa dengan aplikasi ini sudah clear chace hapus data coba login logout masiihhh tetap lemot why kenapa apakah harus ganti hp apel krowak biar lancar mau booking dri web untuk kereta lokal pasti ga ada ùa aplikasinya tidak seperti dulu yang sat set	-1	Negatif
ada bug force close pas pilih kusri buat kalian yang merasakan bug yang sama jangan pilih kursi	0	Netral

Gambar 4.12 Halaman *Testing* Data SVM

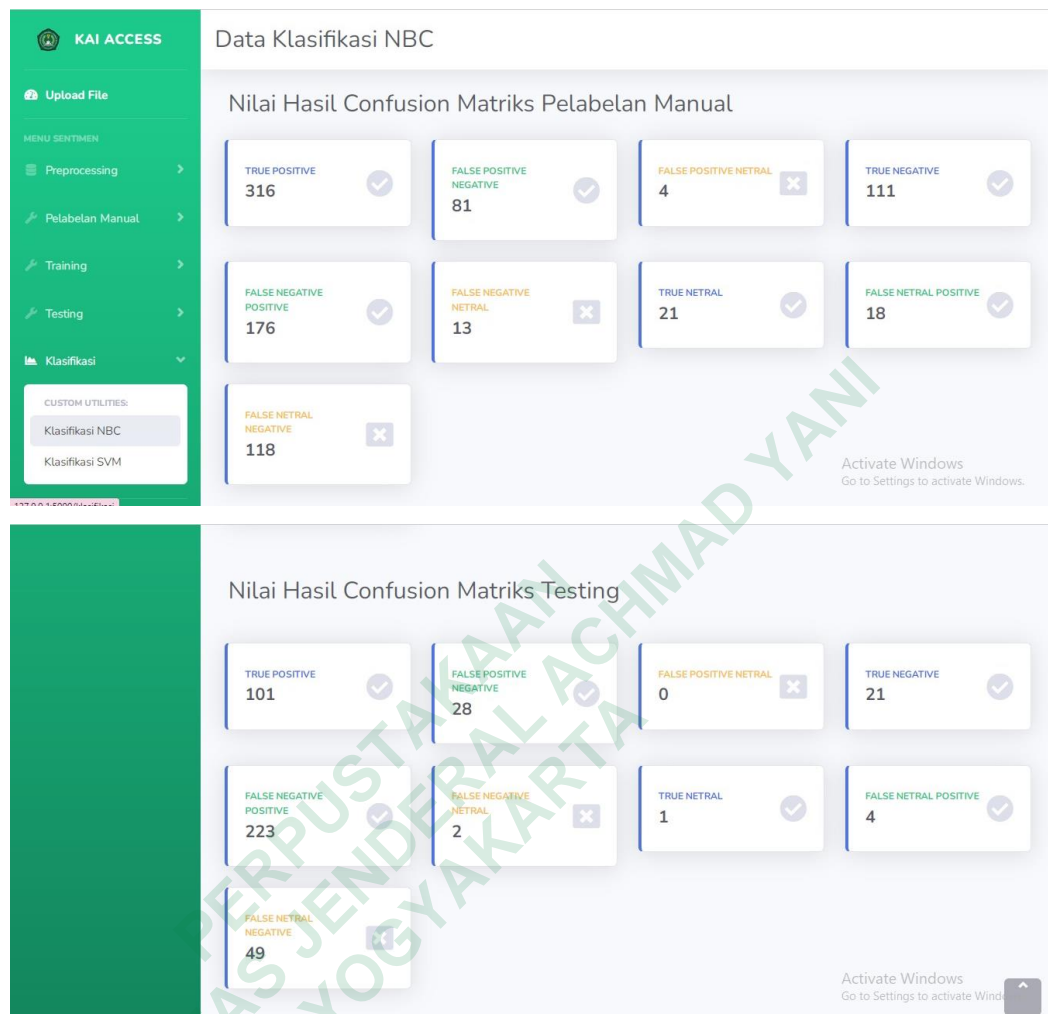
Berikut ini potongan kode yang berfungsi untuk melakukan *testing* data SVM

```
# TESTING SVM
@app.route('/testing_svm', methods=["GET"])
def testing_svm():
    # Membaca data dari file CSV "data_preprocessing.csv"
    all_data = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
    "data_preprocessing.csv"), header=0)

    # Memuat model klasifikasi yang telah dilatih sebelumnya (NBC) dari
    file 'data.pickle'
    model = open('data.pickle', 'rb')
    nbc_classifier = pickle.load(model)
```

4.9.13 Implementasi Halaman Klasifikasi NBC

Implementasi halaman klasifikasi Naïve Bayes Classifier digunakan untuk menampilkan data *review* yang sudah diprediksi kelas dan labelnya secara otomatis berdasarkan model yang sudah dibangun pada tahap *training*. Implementasi halaman klasifikasi Naïve Bayes Classifier ditunjukkan pada Gambar 4.13.



Gambar 4.13 Halaman Klasifikasi Data NBC

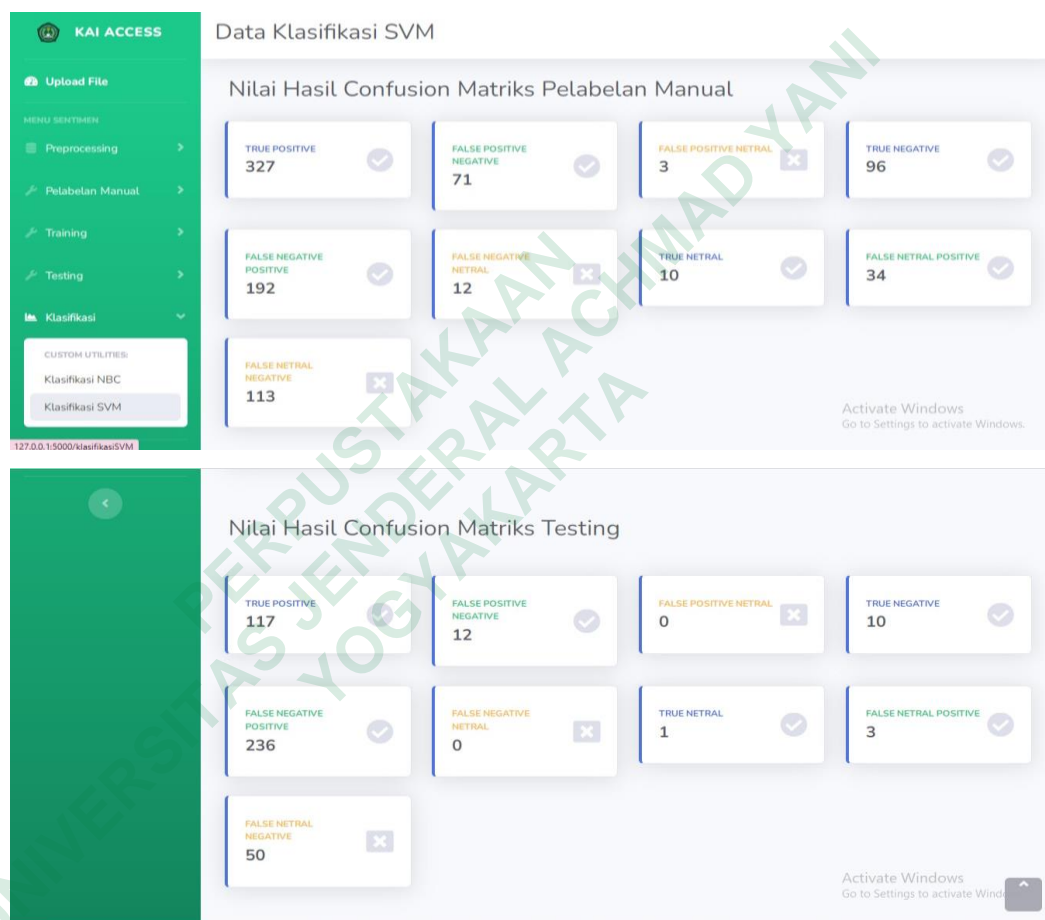
Berikut ini potongan kode yang berfungsi untuk melakukan klasifikasi data NBC

```
# KLASIFIKASI NBC
@app.route('/klasifikasi', methods=["GET"])
def sentimen():
    # Membaca data dari file CSV yang telah diproses
    data=pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'],
    "pelabelan_manual_nbc.csv"))
    data = pd.DataFrame(data)
    data = data.dropna()

    # Memisahkan data menjadi fitur (X) dan label (Y)
    X = data['content']
    Y = data['kelas']
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    test_size=0.2, random_state=42)
```


4.9.14 Implementasi Halaman Klasifikasi SVM

Implementasi Halaman klasifikasi Support Vector Machine digunakan untuk menampilkan data review yang sudah diprediksi kelas dan labelnya secara otomatis berdasarkan model yang sudah dibangun pada tahap training. Implementasi halaman klasifikasi Support Vector Machine ditunjukkan pada Gambar 4.14.



Gambar 4.14 Halaman Klasifikasi Data SVM Testing

Berikut ini potongan kode yang berfungsi untuk melakukan klasifikasi data NBC

```
# KLASIFIKASI SVM
# Memisahkan data menjadi fitur (X) dan label (Y)
X = data['content']
Y = data['kelas']
# Memisahkan data menjadi data latih dan data uji
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)
```

4.10 PEMBAHASAN

Analisis sentimen berdasarkan kata kunci “KAI Access” menggunakan data *review* sebanyak data untuk dilakukan analisis. Data *review* sebanyak 5.000 data yang diambil dari Google Colab, selanjutnya diberi kelas dan label secara manual dengan rata-rata akurasi yang cukup bagus sebesar 84% NBC dan 89% SVM. Data yang sudah diprediksi secara otomatis pada tahap *training*. Pada pengujian data *review* untuk dilakukan prediksi yang sudah dibangun pada tahap *training* sebanyak 900 data, hasil prediksi didapatkan data *review* berlabel positif sebanyak 300 data, data *review* berlabel netral sebanyak 300 data dan data *review* berlabel negatif sebanyak 300 data *review*.

Analisis Sentimen Review Aplikasi KAI ACCESS dari Google Play Store Menggunakan Perbandingan Metode Naïve Bayes Classifier dan Support Vector Machine dibangun menggunakan Bahasa pemograman Python dengan framework Flask, pada sistem analisis sentimen memiliki beberapa menu yaitu, menu upload data, menu *preprocessing* yang didalamnya terdapat menu data *cleaning*, menu *tokenizing*, menu data *stopword removal*, menu *stemming* data, menu hasil *preprocessing*, menu pelabelan manual, menu *training* data, menu *testing* data dan menu klasifikasi. Pada menu upload data berfungsi untuk mengupload file data mentah, menu *preprocessing* digunakan untuk melakukan proses pengolahan data teks dengan memerlukan beberapa tahapan-tahapan, menu pelabelan manual berfungsi untuk menampilkan hasil akurasi dari data *review* yang sudah diberi kelas atau label berdasarkan kalimat secara manual, menu *training* data digunakan untuk membangun model klasifikasi sentimen secara otomatis dengan fungsi ekstraksi menggunakan (TF-IDF) pada data teks, menu *testing* data berfungsi untuk menampilkan hasil akurasi berdasarkan data *review* yang diprediksi secara otomatis untuk menentukan seberapa akurat model yang dibangun pada *training*, menu klasifikasi NBC dan klasifikasi SVM berfungsi untuk menampilkan data *review* yang sudah diprediksi secara otomatis.

Hasil klasifikasi yang digunakan dalam sistem analisis sentimen ini adalah hasil tabel. Tabel tersebut secara otomatis menunjukkan jumlah data *review* yang diprediksi berdasarkan kelas positif, negatif dan netral. Sistem analisis sentimen ini tentunya masih terdapat banyak kekurangan seperti dalam pemanggilan file karena belum bisa dilakukan pemanggilan secara otomatis dan hasil klasifikasinya belum terdapat grafik.

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA