

## **BAB IV**

### **PENGEMBANGAN PERANGKAT LUNAK**

#### **4.1 Pendahuluan**

Pada bab ini membahas mengenai proses pengembangan perangkat lunak yang menjadi hal utama dalam penelitian ini. Proses pengembangan perangkat lunak dengan menggunakan metode *waterfall* terdapat lima fase yaitu *requirement, system design, coding and testing, implementasi, dan maintenance*.

#### **4.2 Metode Pengembangan Perangkat Lunak**

Pada penelitian ini, metode penelitian perangkat lunak yang digunakan adalah metode *waterfall* yang memiliki 5 tahapan, yaitu *requirement, system design, coding and testing, implementasi dan maintenance*.

##### **4.2.1 Requirement**

Pada tahap ini, hal yang perlu dilakukan adalah untuk menganalisis hal apa saja yang diperlukan untuk mengatasi masalah-masalah yang berkaitan dengan kebutuhan sistem. Tahap ini dilakukan dengan melakukan analisis terhadap perangkat keras, perangkat lunak, dan data.

##### **4.2.1.1 Analisis Perangkat Keras**

Dalam melakukan pengembangan perangkat lunak, diperlukan sebuah perangkat keras yang digunakan sebagai alat dalam membuat sistem. Perangkat keras yang digunakan dalam mengembangkan perangkat lunak adalah sebagai berikut:

Processor : AMD Ryzen 5 4500U with Radeon Graphics

RAM : 20 GB

SSD : 512 GB

#### 4.2.1.2 Analisis Perangkat Lunak

Dalam pengembangan perangkat lunak ini juga memerlukan perangkat lunak lain. Dalam penelitian ini perangkat lunak yang digunakan adalah sebagai berikut:

Sistem Operasi : Windows 11 64-bit

Bahasa Pemrograman : *Python*

Code Editor : *Visual Studio Code*

*Library* : *Pandas, sklearn, streamlit, re, nltk, sastrawi, string, os, pickle, io*

#### 4.2.1.3 Analisis Data

Pada penelitian ini, data yang digunakan merupakan data sekunder. Data yang digunakan dalam ini diambil dari <https://www.kaggle.com/datasets/jovangt13/kendaraan-listrik>. Untuk rincian data, dapat dilihat dalam tabel V-1 berikut.

Tabel IV- 1 Rincian Dataset

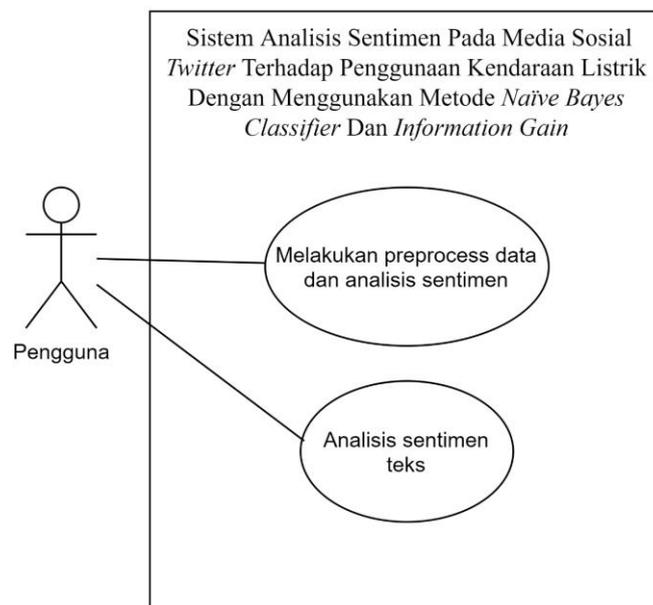
<u>Klasifikasi</u>	<u>Jumlah</u>
<u>Positif</u>	<u>1903</u>
<u>Negatif</u>	<u>1603</u>
<u>Netral</u>	<u>668</u>
<u>Total</u>	<u>4174</u>

#### 4.2.2 System Design

Pada tahap ini hal yang dilakukan yaitu membuat *usecase*, *scenario usecase*, *Activity Diagram*, *sequence Diagram*, dan *Class Diagram*.

##### 4.2.2.1 Use case Diagram

*Use case Diagram* menggambarkan *user* sebagai aktor terhadap sistem sehingga tercipta sebuah interaksi. *Diagram* usecase untuk penelitian ini digambarkan seperti dibawah ini.



Gambar IV- 1 *Use case Diagram*

##### 4.2.2.2 Use case Skenario

*Use case* skenario merupakan penggambaran tindakan spesifik terhadap actor dan sistem yang dibuat pada Gambar IV-1 Berikut ini skenario penerapan dalam format tabel.

Tabel IV- 2 Skenario *preprocess data*

Identifikasi	
Nomor	1
Nama	Melakukan <i>Preprocess</i> Data dan analisis sentimen
Actor	Pengguna
Tujuan	Pengguna memasukkan data yang akan dilakukan <i>preprocessing</i> . Setelah itu, jika data sudah selesai diproses akan muncul <i>radio button</i> untuk memilih analisis dengan <i>Naive bayes</i> atau <i>Information gain dan Naive bayes</i>
Deskripsi	pada <i>use case</i> ini pengguna akan memasukkan data yang akan digunakan dalam preprocess
Kondisi awal	Belum ada data yang diinput
Skenario utama	
Pengguna	Sistem
	1. Menampilkan <i>interface</i> awal untuk menginput file csv dan tombol ' <i>browse file</i> '
2. Menekan tombol ' <i>browse files</i> '	
	3. Menampilkan <i>dialog box</i> untuk mencari <i>file</i>
4. Memilih <i>file</i> berekstensi CSV	

5. Menekan tombol ' <i>open</i> ' pada <i>dialog box</i>	
	6. Membaca <i>file</i>
	7. Menampilkan isi <i>file</i> CSV
8. Menekan tombol ' <i>Preprocess Data</i> '	
	9. Menampilkan data hasil <i>preprocessing text</i>
	10. Menampilkan <i>radio button</i> untuk memilih metode analisis sentimen.
11. Memilih metode analisis	
12. Menekan tombol ' <i>Analisis</i> '	
	13. Menampilkan hasil evaluasi berdasarkan metode yang dipilih
Kondisi Akhir	Menampilkan hasil evaluasi berdasarkan metode yang dipilih
Skenario alternatif	
Pengguna	Sistem
	1. Menampilkan interface awal dan dan button ' <i>browse file</i> '

2. Menekan tombol ' <i>browse files</i> '	
	3. Menampilkan <i>dialog box</i> untuk mencari <i>file</i>
4. Memilih <i>file</i> yang tidak berekstensi CSV atau file CSV yang dimasukkan tidak memiliki kolom yang dibutuhkan	
	5. Menampilkan pesan error
Kondisi Akhir	Sistem menampilkan pesan kesalahan input

Tabel IV- 3 *Skenario Analisis sentiment teks*

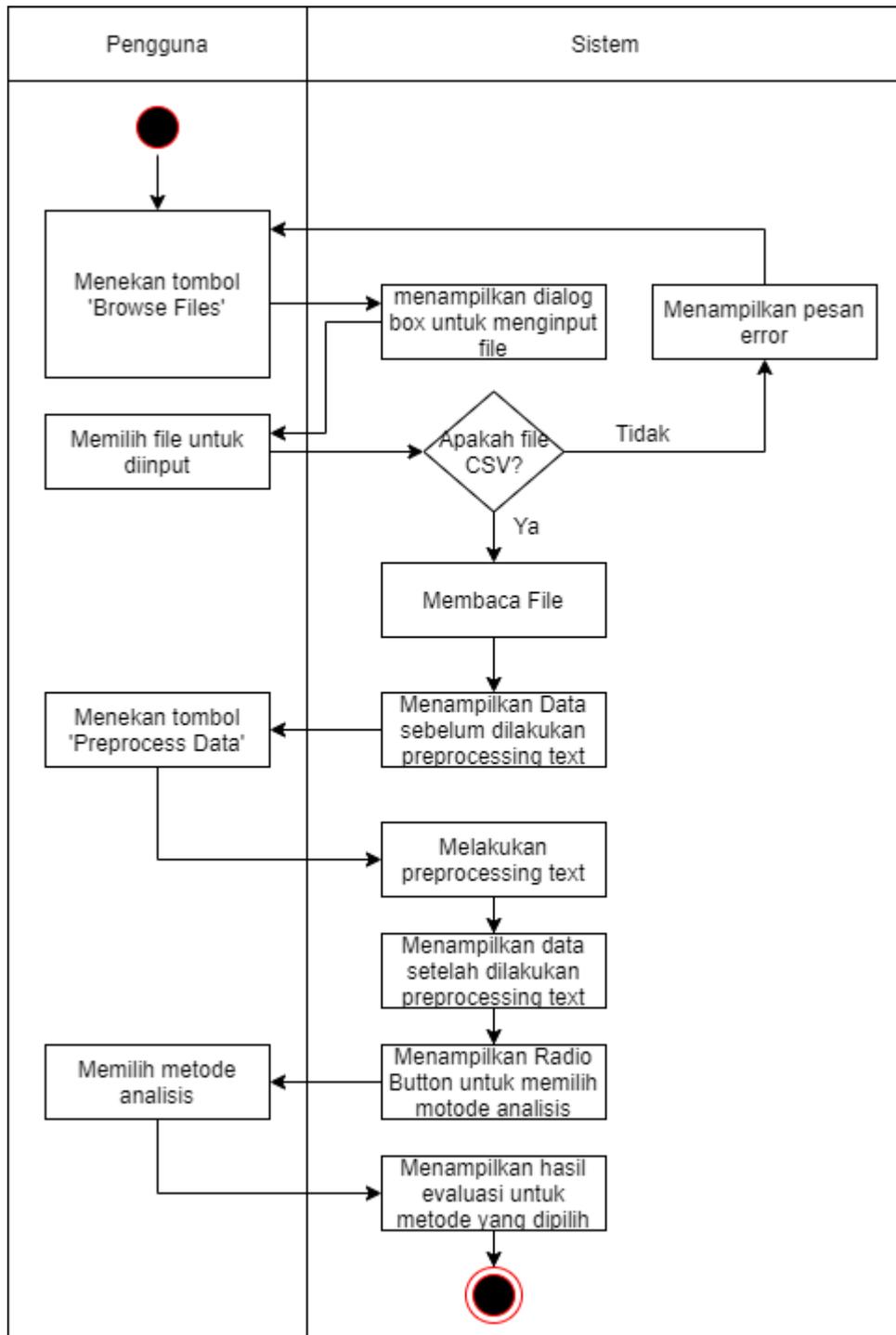
Identifikasi	
Nomor	2
Nama	Analisis sentiment teks
Actor	Pengguna
Tujuan	Pengguna memasukkan teks yang akan diklasifikasi
Deskripsi	pada use case ini pengguna akan memasukkan teks yang akan digunakan dalam proses perhitungan
Kondisi awal	Belum ada text yang diinput

Skenario utama	
Pengguna	Sistem
	1. Menampilkan interface awal dan kolom untuk input text
2. Memasukkan text	
3. Menekan tombol 'Analisis'	
	4. Melakukan preprocessing text
	5. Melakukan klasifikasi terhadap text berdasarkan model yang telah dilatih
	6. Menampilkan hasil preprocessing text dan klasifikasi text
Kondisi Akhir	interface menampilkan hasil preprocessing text dan klasifikasi text
Skenario alternatif	
Pengguna	Sistem
	1. Menampilkan interface awal dan kolom untuk input text

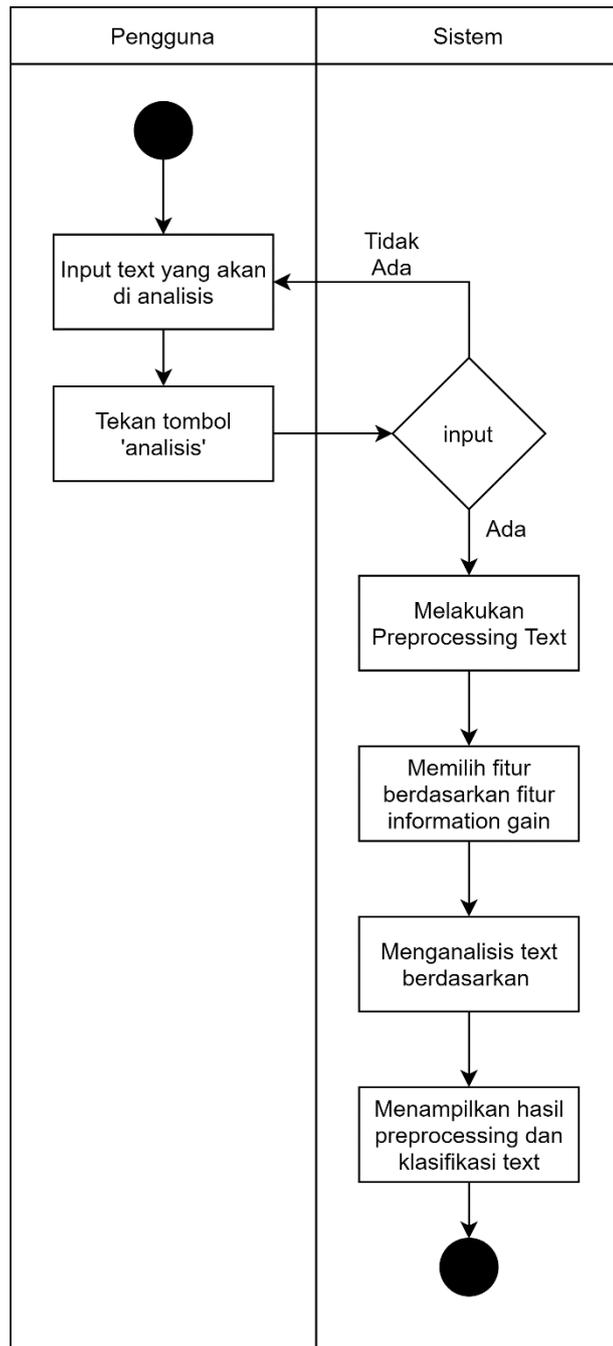
2. Menekan tombol 'analisis' tanpa memasukkan teks	
	3. Menampilkan pesan error 'silahkan masukkan text'
Kondisi akhir	Interface menampilkan pesan error 'silahkan masukkan text'

#### 4.2.2.3 Activity Diagram

Activity *Diagram* adalah *Diagram* yang menggambarkan alur program yang perankan oleh actor dan dengan sistem. Alur program digambarkan dengan *Diagram* dibawah.



Gambar IV- 2 Activity diagram Preprocess Data dan Sentiment Analisis

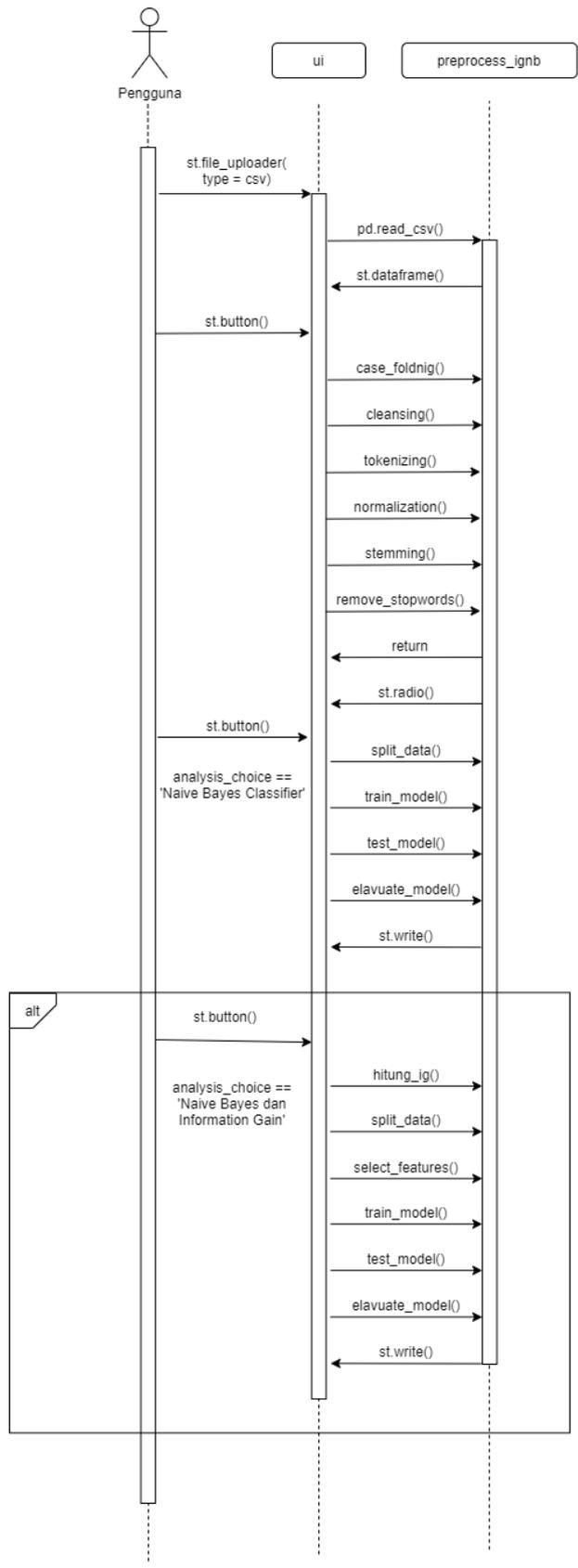


Gambar IV- 3 Activity Diagram Analisis sentimen teks

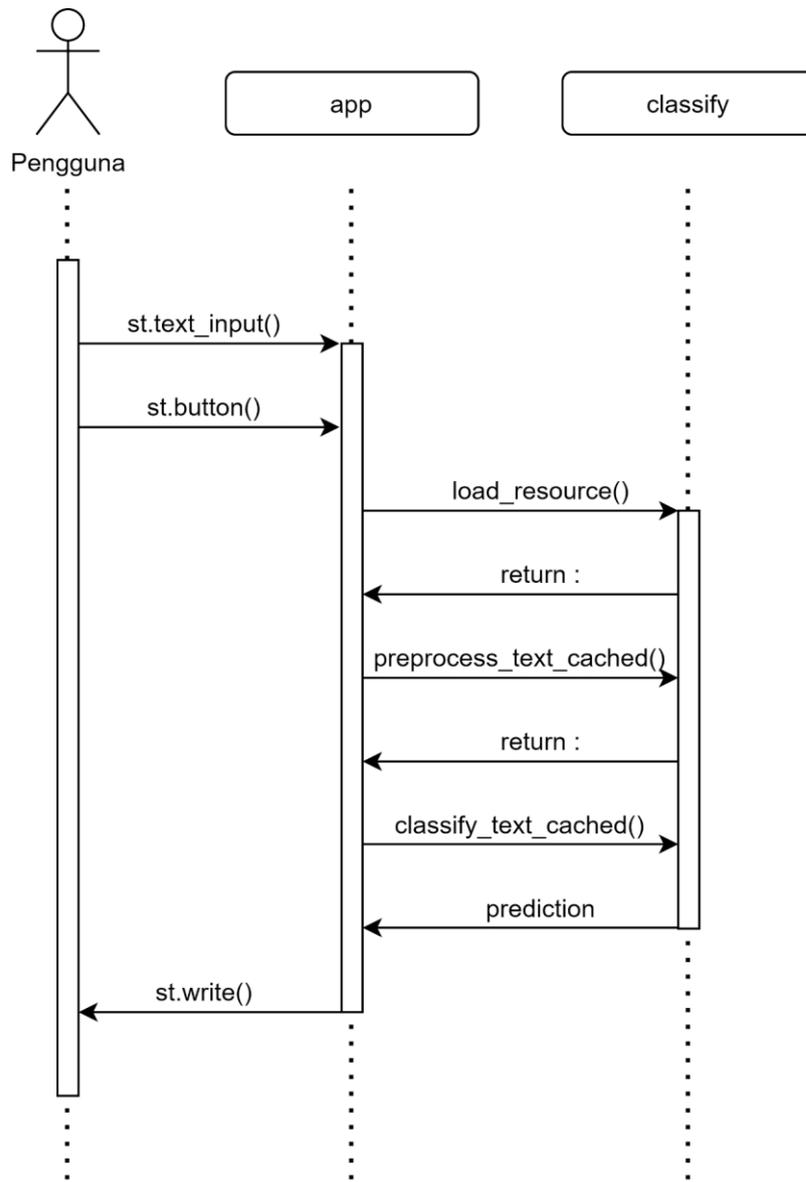
#### 4.2.2.4 Sequence Diagram

*Sequence Diagram* bertujuan untuk mengetahui bagaimana suatu sistem berinteraksi satu dengan yang lain dengan urutan tertentu. Berikut adalah gambaran

dari *sequence Diagram* yang sedang dikembangkan peneliti.



Gambar IV- 4 Analisis Sentimen Naïve Bayes



Gambar IV- 5 Sequence Diagram sentimen analisis teks

#### 4.2.2.5 Class Diagram

*Class diagram* merupakan suatu *diagram* dalam UML yang diperlukan untuk menggambarkan struktur dari suatu sistem. *Diagram* ini menggambarkan kelas yang dimiliki sistem serta hubungan di antaranya. Berikut ini adalah *diagram* kelas untuk perangkat lunak yang sedang dikembangkan.



Gambar IV- 6 Class Diagram

### 4.2.3 Coding and Testing

Pada tahap ini dibagi menjadi dua tahap yaitu tahap *coding* dan tahap *testing*..

#### 4.2.3.1 Coding

Pada tahap *coding*, hasil analisis dan desain sistem yang sudah dibuat

dalam tahap sebelumnya diterjemahkan kedalam bahasa pemrograman.

#### 4.2.3.1.1 Preprcess\_ignb.py

Tabel IV- 4 *function* dari *preprocess\_ignb.py*

No.	Function	Keterangan
1	<code>case_folding(text)</code>	Mengubah semua karakter dalam teks menjadi huruf kecil.
2	<code>cleaning(text)</code>	Menghapus URL, symbol, tanda baca, spasi berlebih, mention, dan angka.
3	<code>tokenizing(text)</code>	Memecahkan teks menjadi kata Tunggal
4	<code>normalisasi(tokens, normal_csv)</code>	Mengubah kata slang atau informal dalam token, dengan padanan formal menggunakan file CSV yang dimasukkan
5	<code>stopword_remove(tokens, stopword_csv)</code>	Menghapus kata yang tidak berhubungan dengan topik dari daftar token berdasarkan CSV ynag dimasukkan
6	<code>stemming(tokens)</code>	Mengubah kata-kata menjadi kata dasarnya dengan stemmer Sastrawi
7	<code>preprocess_text(text)</code>	Menjalankan semua function prapemrosesan pada teks
8	<code>split_data(data)</code>	Membagi data menjadi data latih dan data uji

9	<code>evaluate_model(y_test, y_pred)</code>	Menghitung dan mengembalikan metrik evaluasi seperti akurasi, <i>confusion matrix</i> , presisi, recall, dan skor F1 untuk prediksi model.
10	<code>train_model_nb(x_train, y_train)</code>	Melatih classifier <i>Naive bayes</i> pada data pelatihan dan mengembalikan model yang dilatih serta vektorizer yang digunakan.
11	<code>test_model_nb(model, vectorizer, x_text)</code>	Menggunakan model <i>Naive bayes</i> yang dilatih dan vektorizer untuk memprediksi sentimen pada data pengujian.
12	<code>sentiment_analyze_nb(data)</code>	Melakukan analisis sentimen menggunakan classifier <i>Naive bayes</i> pada dataset, mengembalikan metrik evaluasi.
13	<code>hitung_ig(texts, labels)</code>	Menghitung <i>Information gain</i> untuk setiap fitur (kata) dalam teks dan mengembalikan DataFrame dengan nama fitur dan skor <i>Information gain</i> yang sesuai.
14	<code>select_features(x_train, gain_df,</code>	Memilih fitur dari data pelatihan berdasarkan nilai ambang untuk

	threshold)	<i>Information gain</i> , mengembalikan fitur yang dipilih, mask, vektorizer, dan jumlah fitur yang dipilih.
15	train_model_ignb(x_train_selected, y_train)	Melatih classifier <i>Naive bayes</i> pada fitur yang dipilih dari data pelatihan.
16	test_model_ignb(model, mask, vectorizer, x_text)	Menggunakan model <i>Naive bayes</i> berbasis <i>Information gain</i> yang dilatih dan vektorizer untuk memprediksi sentimen pada data pengujian.
17	sentimentt_analyze_ignb(data, gain_df, threshold)	Melakukan analisis sentimen menggunakan classifier <i>Naive bayes</i> berbasis <i>Information gain</i> pada dataset, menerapkan pemilihan fitur, dan mengembalikan metrik evaluasi.
18	app()	Fungsi utama untuk aplikasi Streamlit, menangani unggahan file, prapemrosesan, dan memanggil fungsi analisis sentimen berdasarkan pilihan pengguna.

#### 4.2.3.1.2 Classify.py

Tabel IV- 5 *function* dari *classify.py*

No.	Function	Keterangan
1	case_folding(text)	Mengubah semua karakter dalam teks menjadi huruf kecil.
2	cleaning(text)	Menghapus URL, symbol, tanda baca, spasi berlebih, mention, dan angka.
3	tokenizing(text)	Memecahkan teks menjadi kata Tunggal
4	Normalisasi ( tokens, normal_csv)	Mengubah kata slang atau informal dalam token, dengan padanan formal menggunakan file CSV yang dimasukkan
5	stopword_remove (tokens, stopwords_csv)	Menghapus kata yang tidak berhubungan dengan topik dari daftar token berdasarkan CSV yang dimasukkan
6	Stemming (tokens)	Mengubah kata-kata menjadi kata dasarnya dengan stemmer Sastrawi
7	preprocess_text(text)	Menjalankan semua function prapemrosesan pada teks
8	load_recource (normal_csv, stopwords_csv)	Memuat sumber daya yang diperlukan untuk normalisasi dan penghapusan stopwords dari file CSV

		yang diberikan. Mengembalikan kamus untuk normalisasi kata slang ke kata formal dan himpunan kata stopwords yang akan dihapus.
9	<code>classify(processed_text)</code>	Mengklasifikasikan teks yang telah diproses menggunakan model yang telah dilatih. Mengubah teks menjadi vektor fitur, memilih fitur yang relevan, dan mengembalikan hasil prediksi dari model.

#### 4.2.3.1.3 App.py

Tabel IV- 6 *function* dari *app.py*

No.	Function	Keterangan
1	<code>preprocess_text_cached</code> ( <code>text</code> , <code>normal_csv</code> , <code>stopword_csv</code> )	Memuat sumber daya untuk normalisasi dan penghapusan stopwords menggunakan fungsi <code>load_resources</code> dari modul <code>classify</code> , kemudian menjalankan prapemrosesan teks dengan fungsi <code>preprocess_text</code> .
2	<code>classify_text_cached</code> ( <code>processed_text</code> )	mengklasifikasikan teks yang sudah diproses menggunakan fungsi <code>classify</code> dari modul <code>classify</code>

3	main()	Menampilkan antarmuka pengguna untuk analisis sentimen, menerima input teks dari pengguna
---	--------	-------------------------------------------------------------------------------------------

#### 4.2.3.2 Testing

Pada tahap *testing*, hasil dari tahap *coding* dilakukan *testing* dengan metode *blackbox*. Testing ini dilakukan untuk memastikan bahwa sistem berjalan dengan baik dan sesuai dengan sebagaimana seharusnya.

#### 4.2.4.2.1. Preprocess data dan Analisis sentimen

Tabel IV- 7 Rencana Pengujian *Preprocess Data dan Analisis Sentimen*

Kode	Skenario Test
A1	Menginput <i>file</i> CSV dengan menekan tombol ' <i>browse file</i> ' untuk data dan menekan "Preprocess Data"
A2	Menginput <i>file</i> yang bukan CSV
A3	Memilih salah satu metode analisis dan menekan tombol "Analyze"

#### 4.2.4.2.2. Klasifikasi teks

Tabel IV- 8 Rencana Pengujian Klasifikasi teks

Kode	Skenario Test
D1	Menginput teks yang akan dianalisis, dan menekan tombol 'analisis'

D2	Tidak ada teks yang diinput
----	-----------------------------

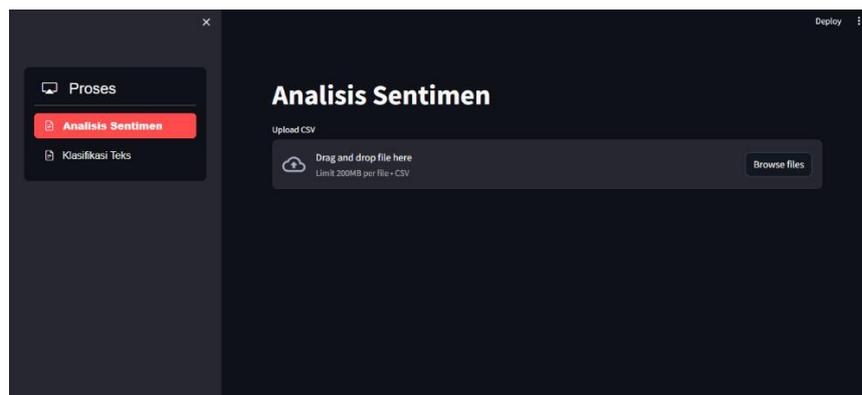
#### 4.2.4 Implementasi

Tahap implementasi di sini mencakup serangkaian aktivitas yang dilakukan untuk menerapkan dan menguji sistem. Implementasi ini terdiri dari dua langkah, yaitu uji coba implementasi, dan pelatihan pengguna.

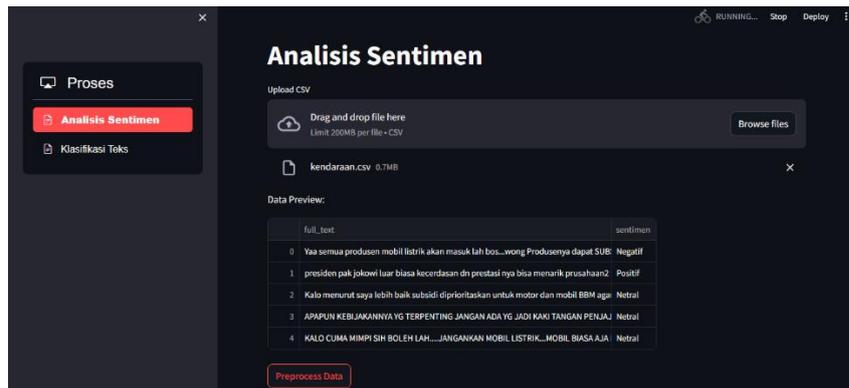
##### 4.2.4.1. Implementasi Interface

Untuk interface sistem dapat dilihat pada gambar-gambar berikut.

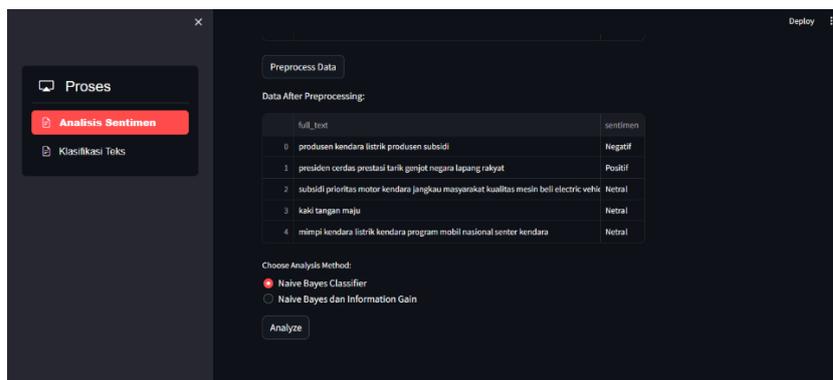
##### 4.1.4.1.1. *Preprocess Data dan analisis sentimen*



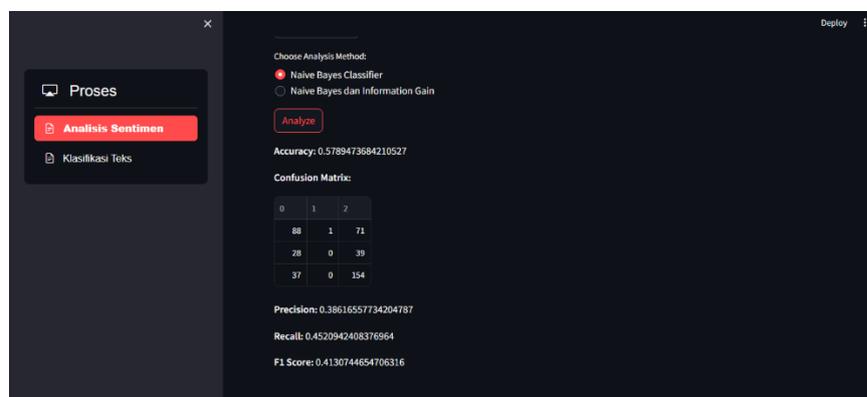
Gambar IV- 7 *Interface* halaman awal



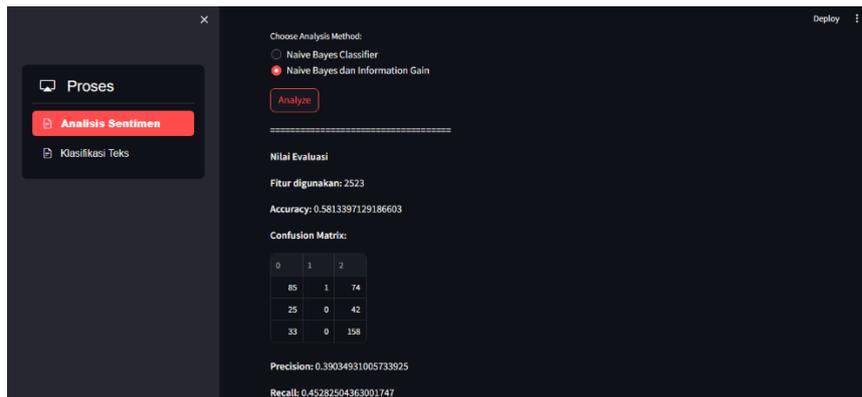
Gambar IV- 8 Interface setelah input file



Gambar IV- 9 Interface setelah tombol 'Preprocess Data' ditekan

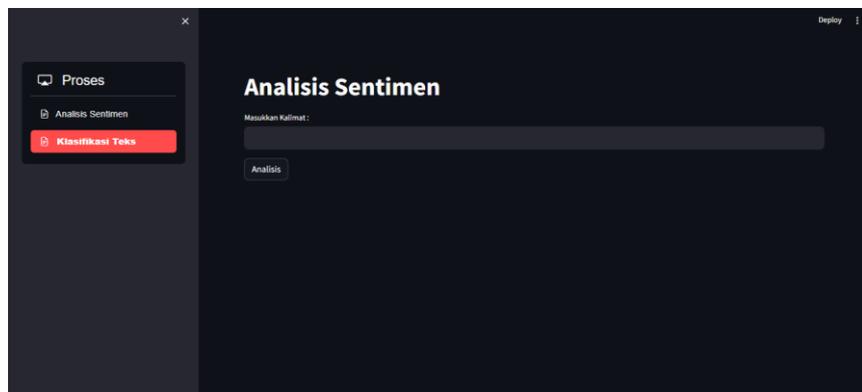


Gambar IV- 10 Hasil Sentimen Analisis Naive bayes

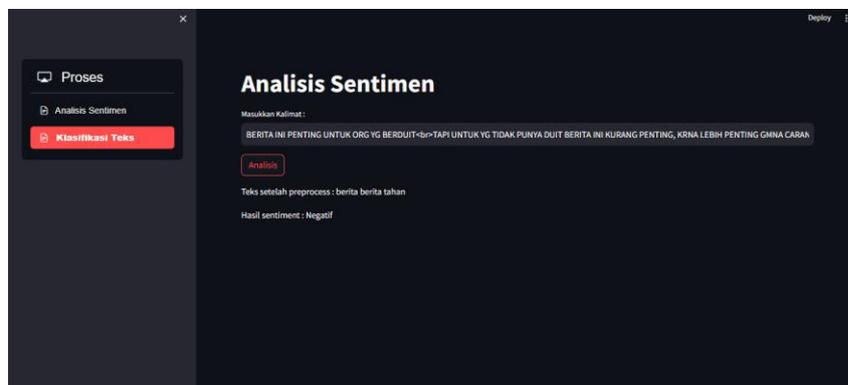


Gambar IV- 11 Sentimen Analisis Information gain dan Naive bayes

#### 4.1.4.1.2. Klasifikasi Teks



Gambar IV- 12 Interface awal klasifikasi teks



Gambar IV- 13 Interface setelah input teks dan menekan 'analisis'

#### 4.2.4.2. Implementasi Pengujian

Langkah pertama dalam tahap ini adalah melakukan uji coba terhadap sistem yang telah dikembangkan. Pengujian dilakukan dengan metode *blackbox* dengan teknik *Equivalence Partitioning*. Dalam pengujiannya menggunakan metode ini meliputi, yaitu skenario test, hasil yang diharapkan, hasil pengujian dan Kesimpulan. Uji coba ini bertujuan untuk memastikan bahwa sistem dapat berjalan sesuai dengan yang diharapkan.

#### 4.2.4.2.1. Preprocess data dan analisis sentimen

Tabel IV- 9 Pengujian *Preprocess Data dan analisis sentimen*

Kode	Skenario Test	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A1	Menginput <i>file</i> CSV dengan menekan tombol ' <i>browse file</i> ' dan menekan "Preprocess Data"	Menampilkan data awal yang belum dilakukan <i>preprocessing text</i> , lalu menekana tombol " <i>Preprocess Data</i> " untuk melakukan preproses teks, setelah selesai,	Sistem menampilkan data awal yang belum dilakukan <i>preprocessing text</i> , lalu menekana tombol " <i>Preprocess Data</i> " untuk melakukan preproses teks,	Berhasil

		muncul 2 radio <i>button</i> untuk memilih metode anlaisis	setelah selesai, muncul 2 radio <i>button</i> untuk memilih metode anlaisis	
A2	Menginput <i>file</i> yang bukan CSV	Menampilkan pesan error	Sistem menampilkan pesan error	Berhasil
A3	Memilih salah satu metode analisis dan menekan tombol “Analyze”	Menampilkan hasil evaluasi berupa <i>accuracy</i> , <i>onfusion</i> <i>matrix</i> , <i>precision</i> , <i>recall</i> , dan <i>f1-</i> <i>score</i> untuk metode yang dipilih	Sistem menampilkan hasil evaluasi berupa <i>accuracy</i> , <i>onfusion</i> <i>matrix</i> , <i>precision</i> , <i>recall</i> , dan <i>f1-</i> <i>score</i> untuk metode yang dipilih	Berhasil

#### 4.2.4.2.2. Klasifikasi teks

Tabel IV- 10 Pengujian Klasifikasi teks

Kode	Skenario Test	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
D1	Menginput teks, lalu menekan tombol 'analisis'	Melakukan preprocess text dan menampilkan teks hasil preprocessing dan sentimen hasil analisis	Sistem melakukan preprocess text dan menampilkan teks hasil preprocessing dan sentimen hasil analisis	Berhasil
D2	Tidak ada teks yang diinput	Menampilkan pesan error	Sistem menampilkan pesan error	Berhasil

#### 4.2.4.3. Pelatihan pengguna

Tahap ini adalah memberikan pelatihan kepada pengguna. Pelatihan ini bertujuan untuk memastikan bahwa pengguna dapat menggunakan sistem dengan efektif. pelatihan mencakup pengenalan antarmuka sistem, serta cara penggunaan fitur-fitur utama.

#### 4.2.5 *Maintenance*

Tahap ini adalah langkah penting dalam memastikan keberlanjutan sistem analisis sentimen yang telah dikembangkan. Pada tahap ini, beberapa hal yang

dilakukan untuk menjaga agar sistem tetap berfungsi dengan baik terhadap perubahan atau masalah yang mungkin timbul selama penggunaan. Hal yang dapat dilakukan adalah sebagai berikut

#### **4.2.5.1.        *Monitoring***

Monitoring adalah langkah awal dalam proses maintenance, di mana sistem diawasi secara terus-menerus untuk mendeteksi setiap anomali atau masalah yang mungkin terjadi. Aktivitas monitoring meliputi pemantauan kinerja sistem dan pemantauan penggunaan sistem.

#### **4.2.5.2.        *Evaluasi sistem***

Evaluasi sistem dilakukan secara berkala untuk menilai apakah sistem masih memenuhi kebutuhan pengguna dan standar performa yang diharapkan. Evaluasi ini mencakup pengujian terhadap komponen perangkat keras dan perangkat lunak, serta analisis umpan balik dari pengguna.

#### **4.2.5.3.        *Pembaharuan sistem***

Seiring dengan perkembangan teknologi dan perubahan kebutuhan pengguna, sistem mungkin memerlukan pembaruan atau peningkatan. Proses pembaruan sistem meliputi pembaharuan perangkat lunak, dan pembaharuan data.

#### **4.2.5.4.        *Perbaikan dan pemulihan***

Jika terdeteksi masalah atau kerusakan pada sistem, langkah-langkah perbaikan dan pemulihan harus segera dilakukan untuk meminimalkan dampak negatif pada pengguna. Aktivitas perbaikan dan pemulihan meliputi identifikasi

masalah, dan *fixing bug*.

#### **4.2.5.5. Dokumentasi**

Semua aktivitas *maintenance* harus didokumentasikan dengan baik untuk tujuan audit dan referensi di masa yang akan datang. Dokumentasi ini mencakup laporan monitoring, catatan pembaruan sistem, dan log perbaikan.

### **4.3 Kesimpulan**

Dalam penelitian ini, metode *waterfall* digunakan untuk merancang dan membangun perangkat lunak. Metode ini terdiri dari 5 tahap, yaitu *requirement*, *system design*, *coding and testing*, implementasi, dan *maintenance*. Penelitian ini menunjukkan metode *waterfall* dapat dipakai untuk mengembangkan perangkat lunak untuk melakukan analisis sentimen.