

BAB IV

PERANCANGAN

Bab ini membahas tahapan perancangan aplikasi yang dalam penelitian, pembahasan analisis sistem terdahulu, perancangan antarmuka, proses bisnis aplikasi, dan spesifikasi kebutuhan sistem yang akan dibuat.

4.1 Analisis Sistem Terdahulu

Analisis sistem terdahulu memiliki tujuan untuk menganalisis sistem yang sudah pernah dibuat sebelumnya oleh peneliti lain. Hal yang dianalisis ialah cara kerja sistem, kelebihan, dan kekurangannya. Hasil analisis sistem terdahulu ini akan melihat perbedaan hasil output dari sistem yang sudah lama dan melengkapi kekurangan sistem yang sudah ada.

- Penelitian yang dilakukan oleh Ivan Dwi Nugraha dan Yufis Azhar (2022), telah membuat sebuah model yang mampu mendeteksi depresi dari *tweet* yang dibuat oleh pengguna *twitter* dengan menggunakan LSTM-RNN. Model yang dibuat oleh Ivan Dwi dan Yufis Azhar memiliki akurasi 86%. Penelitian tersebut sangat baik, namun penelitian tersebut tidak membuat sebuah website untuk ditampilkan atau digunakan oleh publik.

4.2 Spesifikasi Kebutuhan Sistem Baru

Penentuan spesifikasi sistem yang akan dibuat pada penelitian ini bertujuan agar hasil dari sistem yang dibuat optimal untuk digunakan. Sistem yang dibuat akan berbasis website dengan menggunakan *streamlit framework*. Kebutuhan spesifikasi sistem akan dijelaskan sebagai berikut.

4.3.1 Spesifikasi Kebutuhan Sistem Perangkat Lunak

Penelitian ini akan menghasilkan sebuah sistem berbasis website yang artinya penggunaan perangkat lunak sangat diperlukan dalam pembuatan sistem. Penggunaan perangkat lunak yang digunakan pada penelitian ini dapat dilihat pada Tabel 4.1

Tabel 4.1 Spesifikasi Perangkat Lunak Sistem

No	Perangkat Lunak
1	Python 3.12
2	Visual Studio Code
3	Jupyter Notebook
4	Google Colab
5	Web Browser
6	Koneksi Internet
7	Sistem Operasi Windows 10

Tabel 4.1 adalah spesifikasi yang digunakan dalam pembuatan sistem berbasis website dan pembuatan model LSTM. Penelitian ini menggunakan *Python* versi 3.12. *Python* versi 3.12 merupakan 1 versi dibelakang versi paling terbaru yaitu 3.13. Penggunaan versi 3.12 memiliki tujuan agar proses pembuatan model berjalan secara lancar tanpa hambatan keperluan versi *Python* yang diperlukan oleh *library Python* nantinya. Penelitian ini menggunakan 3 kode editor berbeda, yaitu *Visual Studio Code*, *Jupyter Notebook*, dan *Google Colab*. Masing-masing kode editor digunakan dengan tujuan berbeda. *Visual Studio Code* akan digunakan untuk pembuatan antar muka website dengan menggunakan *library streamlit*. *Jupyter Notebook* akan digunakan untuk melatih model LSTM. *Google Colab* akan digunakan untuk proses pengumpulan data melalui *YouTube API v3* dan proses praproses. *Web Browser* yang digunakan pada penelitian ini adalah *Microsoft Edge* dan sistem operasi yang digunakan adalah *Windows 10*.

4.3.2 Spesifikasi Kebutuhan Sistem Perangkat Keras

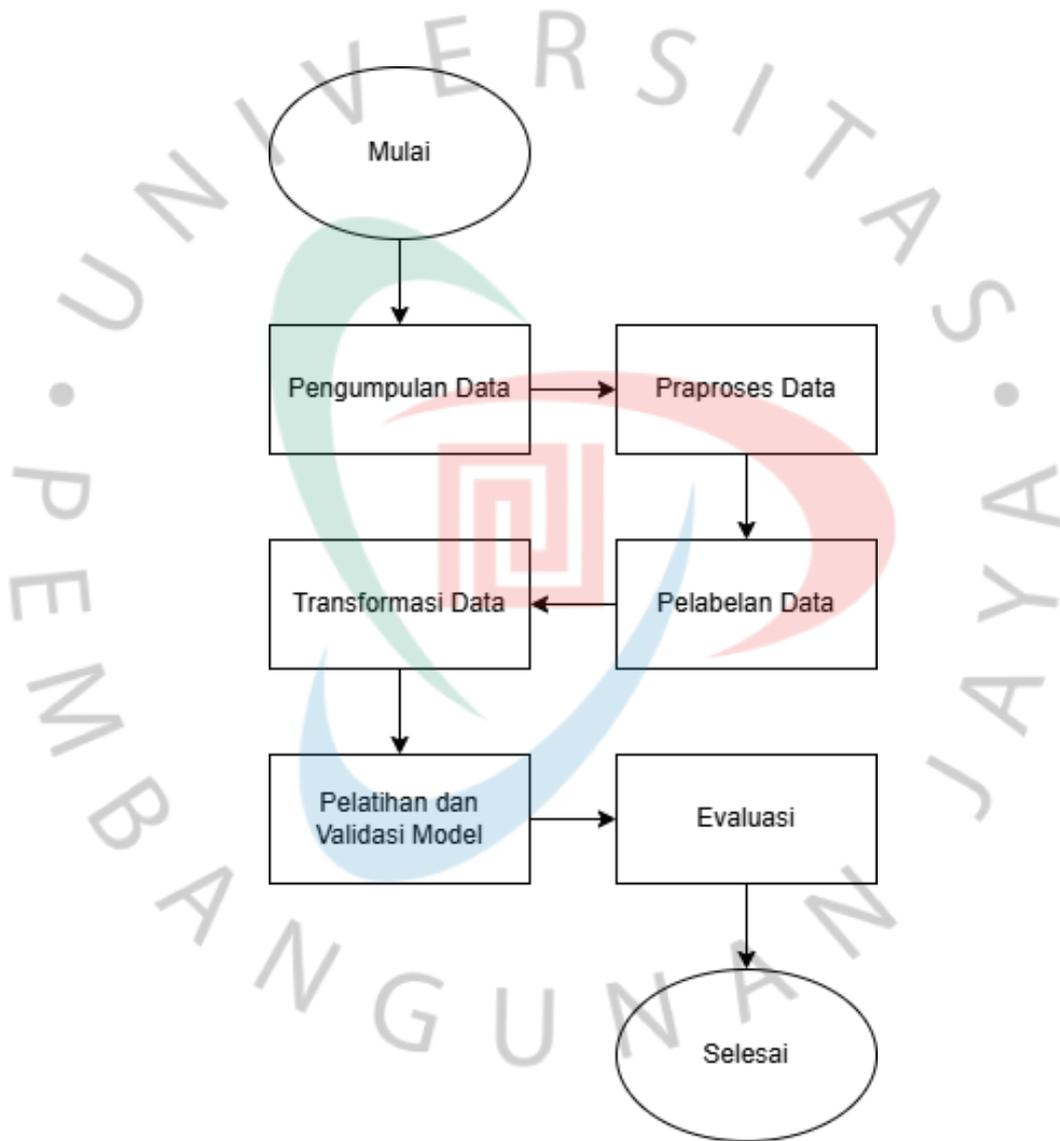
Perangkat keras digunakan sebagai sarana untuk menjalankan perangkat lunak. Maka dari itu menentukan perangkat keras penting agar perangkat lunak yang digunakan dapat dibuat secara optimal. Perangkat keras yang digunakan dalam pembuatan sistem dapat dilihat pada Tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak Sistem

Perangkat Keras	Perangkat Keras
CPU	Intel i7 7 th Gen
GPU	Nvidia GTX 1050
RAM	4GB + 8GB DDR4
Penyimpanan	SSD 512GB NVME + 1TB HDD

Tabel 4.2 adalah spesifikasi perangkat keras yang digunakan untuk membuat sistem. CPU yang digunakan dalam penelitian ini adalah Intel i7 7th Gen dan GPU Nvidia GTX 1050. CPU dan GPU akan digunakan dalam pembuatan model *machine learning* LSTM. RAM akan digunakan untuk menjalankan *web server streamlit* dan proses pemuatan dataset pada *Jupyter Notebook*.

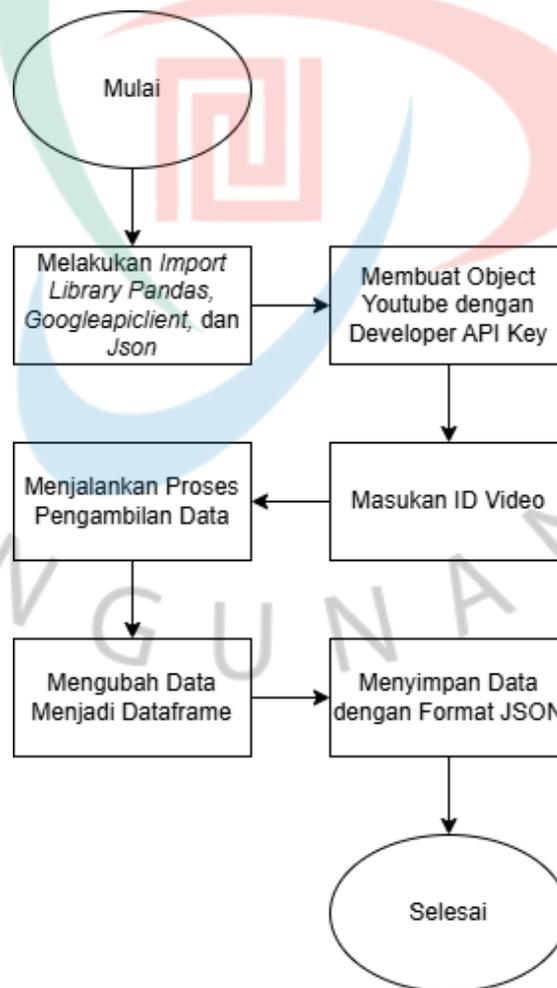
4.3.3 Perancangan Model



Gambar 4.1 Flowchart Pembuatan Model

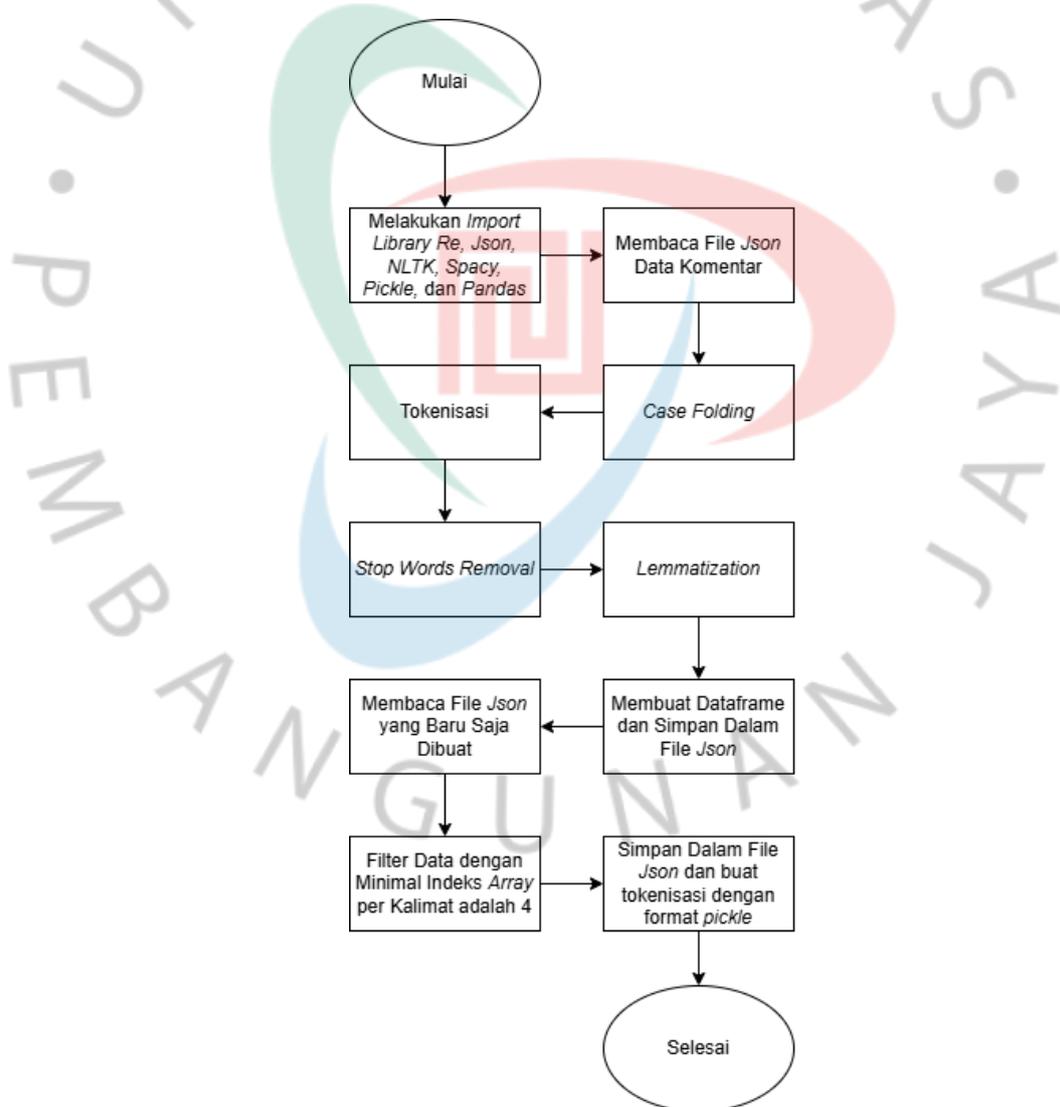
Gambar 4.1 merupakan *flowchart* pembuatan model. Proses ini dimulai dengan pengumpulan data hingga proses evaluasi model. Proses pengumpulan data dilakukan dengan menggunakan *YouTube API v3* yang disediakan oleh Google

Cloud Console secara gratis. Setelah data terkumpul, data tersebut akan disimpan dengan format file json yang selanjutnya akan dilakukan ke tahap praproses data. Praproses data akan membersihkan data yang sudah terkumpul menjadi bentuk yang mudah untuk dianalisis. Setelah data melewati proses praproses data, data tersebut akan diberikan label. Data yang sudah memiliki label akan diproses ketahap selanjutnya, yaitu transformasi data. Data transformation merupakan rangkaian proses yang digunakan untuk mengubah data menjadi bentuk yang dapat diolah (Azrul, Purnamasari, & Ali, 2024). Transformasi data ini akan mengubah data teks menjadi numerik. Data telah diubah ke dalam bentuk numerik ini akan digunakan oleh model untuk melakukan proses latihan dan validasi. Setelah model melakukan latihan dan validasi, tahap terakhir adalah evaluasi. Tahap evaluasi akan meninjau kinerja model yang telah dibuat, proses ini akan menentukan apakah model sudah sesuai dengan tujuan penelitian.



Gambar 4.2 *Flowchart* Pengumpulan Data

Gambar 4.2 adalah *flowchart* pengumpulan data untuk penelitian. Proses ini dimulai dengan melakukan *import library* yang diperlukan seperti *pandas*, *googleapiclient*, dan *json*. Selanjutnya, pembuatan sebuah fungsi *Python* yang akan digunakan untuk menarik data melalui *api*. Proses ini dilakukan dengan membuat sebuah obyek yang diberi nama *youtube* yang memiliki nilai sebuah fungsi *build* dengan parameter versi *api youtube* dan *developer key* yang diberikan oleh *Google Cloud Console*. Setelah memasukan *developer key*, langkah selanjutnya adalah memasukan id video yang ingin diambil komentarnya. Komentar akan diambil dari id video yang telah ditentukan, data komentar tersebut akan diubah ke dalam bentuk *Dataframe* dari *library pandas* dan akan disimpan dengan file berformat *json*.



Gambar 4.3 *Flowchart* Praproses Data

Gambar 4.3 adalah *flowchart* praproses data untuk penelitian. Praproses dimulai dengan meng-*import library* yang diperlukan seperti re, NLTK, Json, SpaCy, Pickle, dan Pandas. Praproses dimulai dengan membaca file *json* yang telah disimpan dari hasil pengumpulan data. Setelah file *json* dibaca, data dalam file tersebut akan memasuki tahapan praproses yang akan dijelaskan sebagai berikut.

1. Case Folding

Case folding bertujuan untuk mengubah teks menjadi huruf kecil dan membersihkan data teks dari *html tag*, *link url*, emoji, angka, spasi ganda, dan huruf yang telah ditentukan.

2. Tokenisasi

Tokenisasi bertujuan untuk penguraian kalimat menjadi sebuah potongan kata-kata yang dilakukan dengan memotong kata pada white space atau spasi.

3. *Stop Words Removal*

Stop Words Removal bertujuan untuk menghilangkan kata yang umum digunakan dan memiliki sedikit bobot dalam maknanya. Kata-kata tersebut sering dihilangkan dalam proses analisis teks.

4. Lematisasi

Lematisasi bertujuan untuk mengekstrakan kata dasar dari sebuah kata dengan pendekatan menggunakan kosakata/kamus data. Lematisasi bertujuan untuk mendapatkan kata-kata yang valid dari kata dasar.

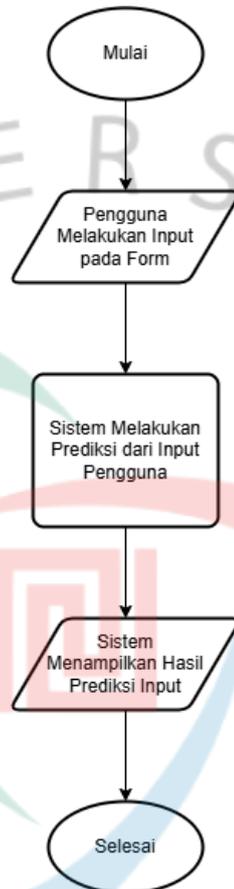
Setelah data melewati proses praproses, data akan disimpan dalam file *json* setelah itu dilakukan proses penyaringan data. Penyaringan data dilakukan dengan kondisi indeks setiap kalimat harus memiliki minimal 4 indeks *array* kata, jika tidak memiliki 4 indeks *array* kata, maka data tersebut akan dihapus. Tujuan dari penyaringan data ini adalah untuk menyaring data yang terlalu pendek yang berpotensi hanya sebagai spam atau indeks kosong saja.

4.3 Perancangan Sistem

Perancangan sistem dalam penelitian ini menggunakan diagram UML. Diagram UML sering digunakan sebagai bahasa yang menjelaskan proses bisnis

suatu sistem sebelum sistem tersebut dibuat. Diagram UML yang digunakan adalah *flowchart*, *use case*, dan *activity diagram*.

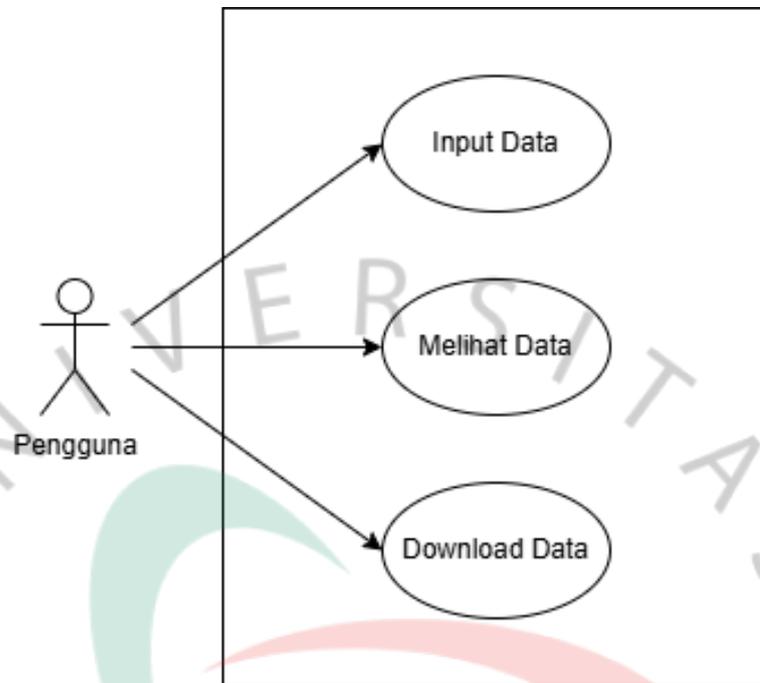
4.3.1 *Flowchart* Bisnis Proses Aplikasi



Gambar 4.4 *Flowchart* Aplikasi

Gambar 4.4 adalah *flowchart* dari aplikasi yang akan dibuat dalam penelitian ini. Proses bisnis dari aplikasi ini cukup sederhana dan cepat. Hal yang pertama dilakukan pengguna adalah mengakses aplikasi yang nanti dibuat dan memasukkan input melalui ke aplikasi, setelah itu aplikasi akan memproses data input tersebut dengan model LSTM yang sudah dibuat. Hasil dari prediksi akan ditampilkan pada aplikasi dengan label dan nilai prediksi dari model LSTM.

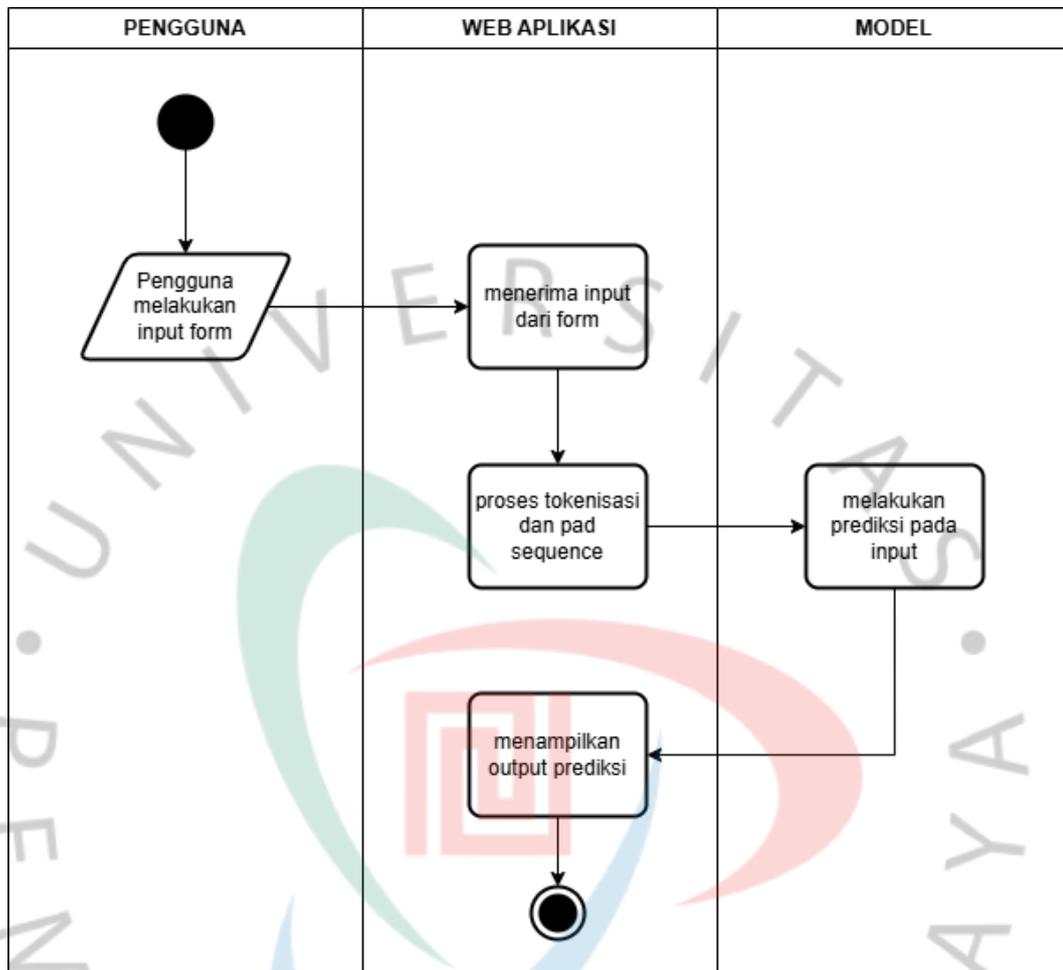
4.3.2 Use Case Aplikasi



Gambar 4.5 Use Case Aplikasi

Gambar 4.5 merupakan *use case* aplikasi yang akan dibuat. *Use case* ini menggambarkan akses yang dapat dilakukan oleh pengguna ketika aplikasi telah dibuat. Akses yang pertama adalah input data, pengguna dapat memasukkan data sebuah teks dalam *form* yang sudah disediakan. *Form* ini akan mengirim data yang telah diinput ke model yang telah dibuat. Kedua, pengguna dapat melihat data. Pengguna memiliki akses untuk melihat dataset penelitian, data test yang digunakan serta hasil identifikasi dari model, dan hasil dari prediksi dari komentar yang diinput oleh user melalui *form*. Terakhir, pengguna dapat mengunduh dataset dan datatest yang ada dalam aplikasi yang akan dibuat nanti.

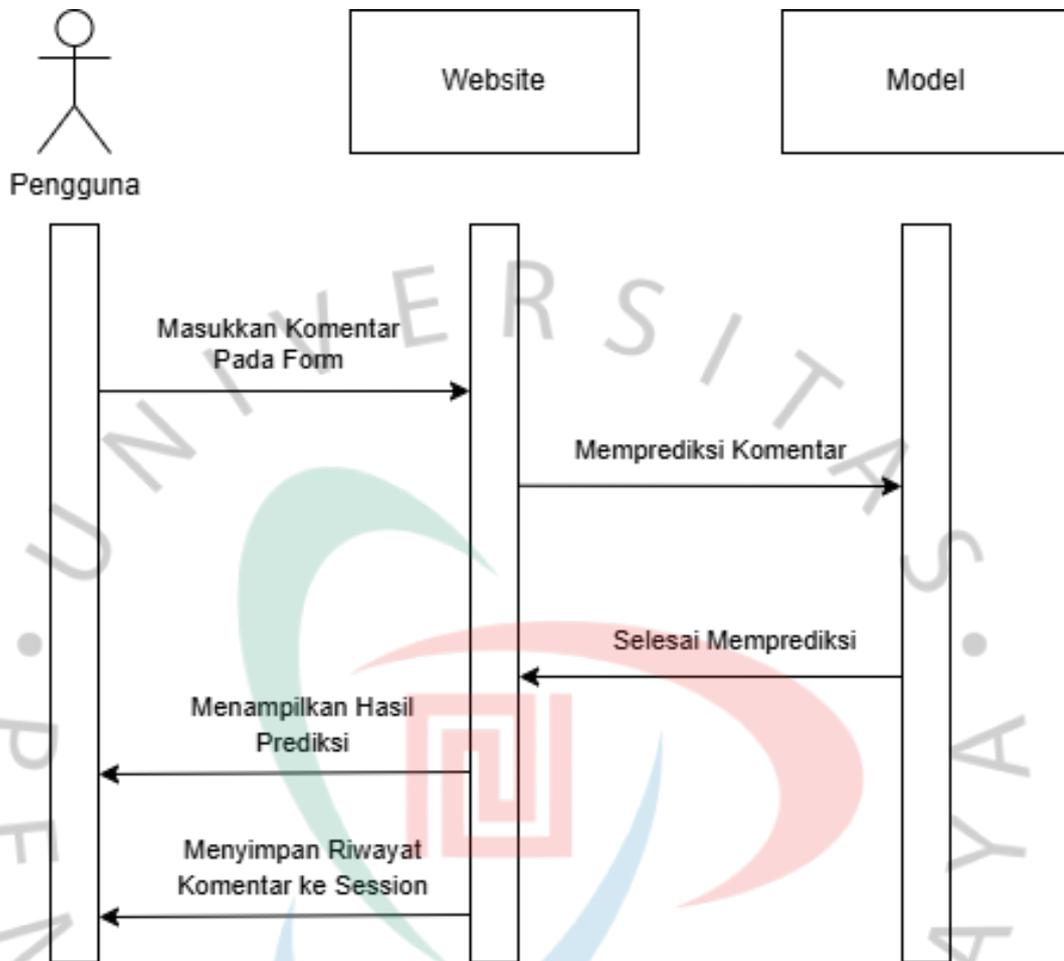
4.3.3 Activity Diagram Aplikasi



Gambar 4.6 Activity Diagram Aplikasi

Gambar 4.6 merupakan *activity diagram* dari aplikasi yang akan dibuat. Diagram ini memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari sistem aplikasi digambarkan secara vertikal. Tujuan dari aktivitas ini untuk menjelaskan dalam suatu proses aplikasi yang akan dibuat. Proses ini menggambarkan bagaimana pengguna melakukan input komentar melalui *form* dan sistem aplikasi memproses data tersebut kepada model dan menampilkan hasil output prediksi ke sistem aplikasi.

4.3.4 Sequence Diagram Aplikasi



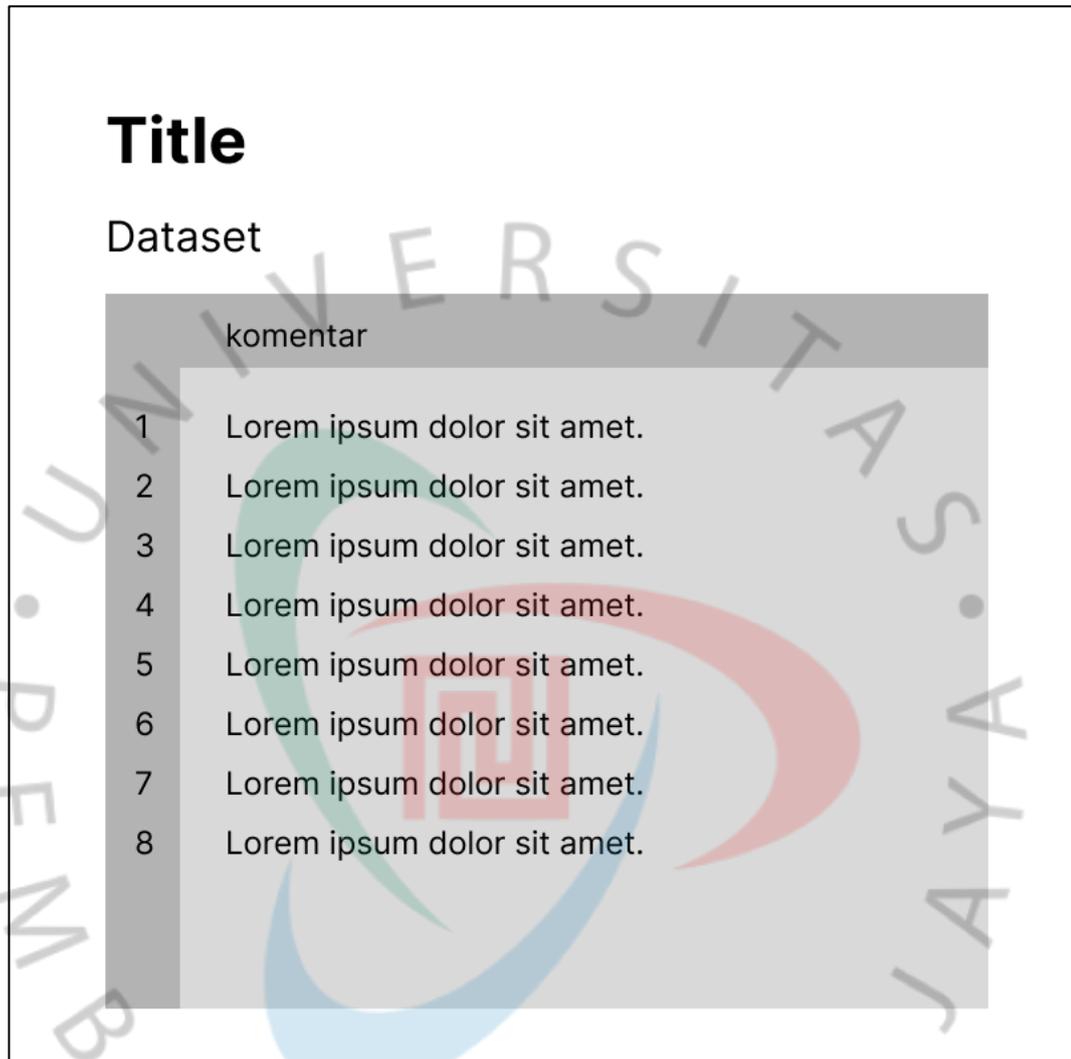
Gambar 4.7 Sequence Diagram Aplikasi

Gambar 4.7 merupakan *sequence diagram* dari aplikasi yang akan dibuat. Proses *sequence diagram* menggambarkan serangkaian langkah yang dilakukan sebagai respon dari proses aktivitas yang menghasilkan output, dalam *sequence diagram* ini terdapat satu aktor, yaitu pengguna. Pengguna akan memasukan komentar pada aplikasi, lalu komentar tersebut akan diproses model untuk diprediksi angka persentasenya. Hasil prediksi akan ditampilkan kepada pengguna, selain itu riwayat komentar juga akan disimpan ke dalam session aplikasi untuk pengguna melihat daftar komentar yang sudah pernah diprediksi sebelumnya.

4.4 Perancangan Antarmuka Aplikasi

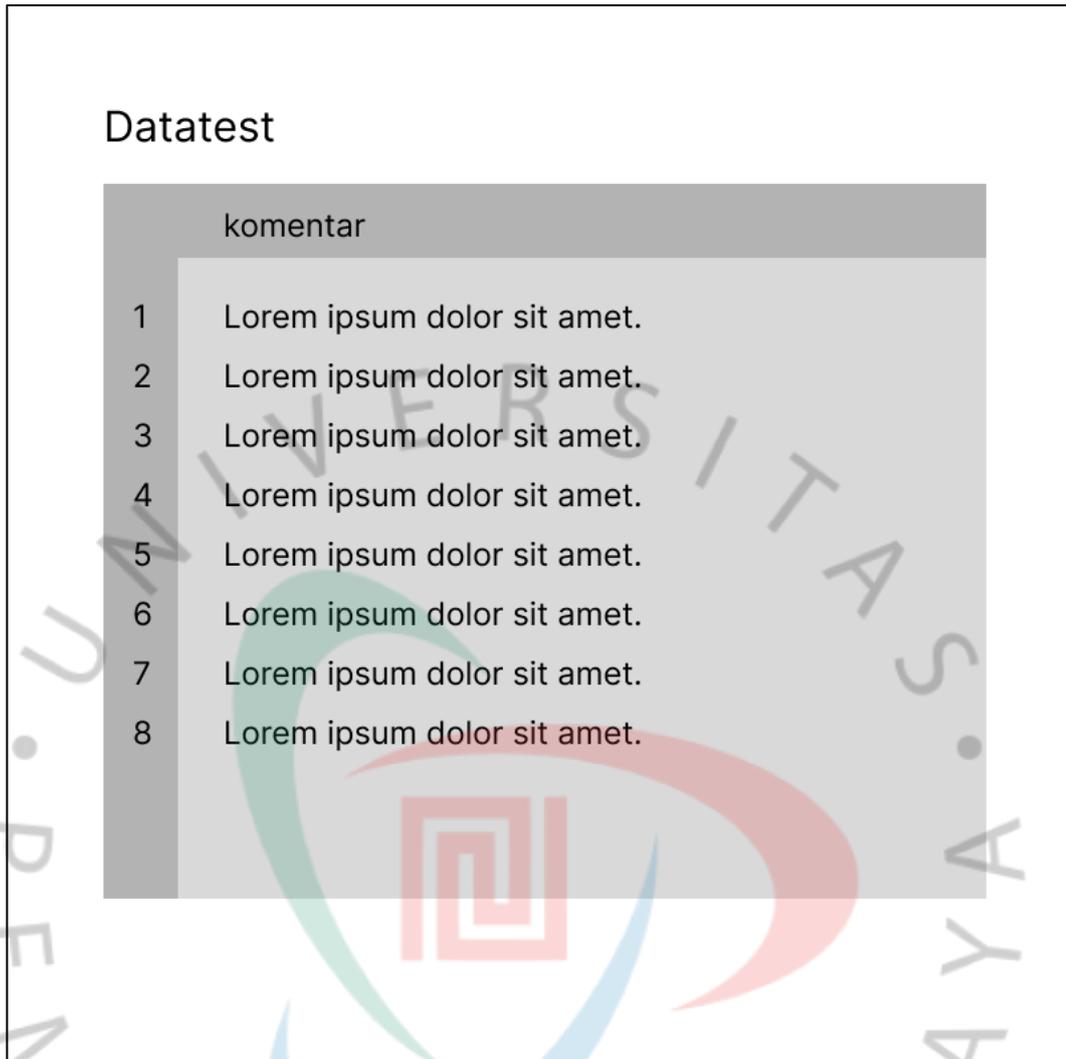
Perancangan antarmuka aplikasi dilakukan dengan tujuan untuk mengetahui gambaran prototipe aplikasi yang akan dibuat. Prototipe antarmuka ini akan

memberikan gambaran letak elemen desain, interaksi pengguna dengan aplikasi, dan penyajian data kepada pengguna.



Gambar 4.8 Prototipe Antarmuka Dataset Aplikasi

Gambar 4.8 adalah antarmuka pengguna untuk menampilkan dataset yang digunakan dalam penelitian ini. Pengguna dapat melihat dataset komentar yang digunakan dan dapat juga mengunduh dataset tersebut jika diperlukan.



Gambar 4.9 Prototipe Antarmuka Datatest Aplikasi

Gambar 4.9 adalah antarmuka pengguna untuk menampilkan datatest yang digunakan dalam penelitian ini sebagai contoh untuk memprediksi komentar dari model yang sudah dibuat. Pengguna dapat melihat datatest komentar yang digunakan dalam penelitian ini dan dapat juga mengunduh datatest tersebut jika diperlukan.

Keyword sering muncul

	keyword	frekuensi
1	Lorem ipsum dolor sit amet.	22
2	Lorem ipsum dolor sit amet.	18
3	Lorem ipsum dolor sit amet.	11
4	Lorem ipsum dolor sit amet.	10
5	Lorem ipsum dolor sit amet.	9
6	Lorem ipsum dolor sit amet.	5
7	Lorem ipsum dolor sit amet.	2

Gambar 4.10 Prototipe Antarmuka Kata Kunci Aplikasi

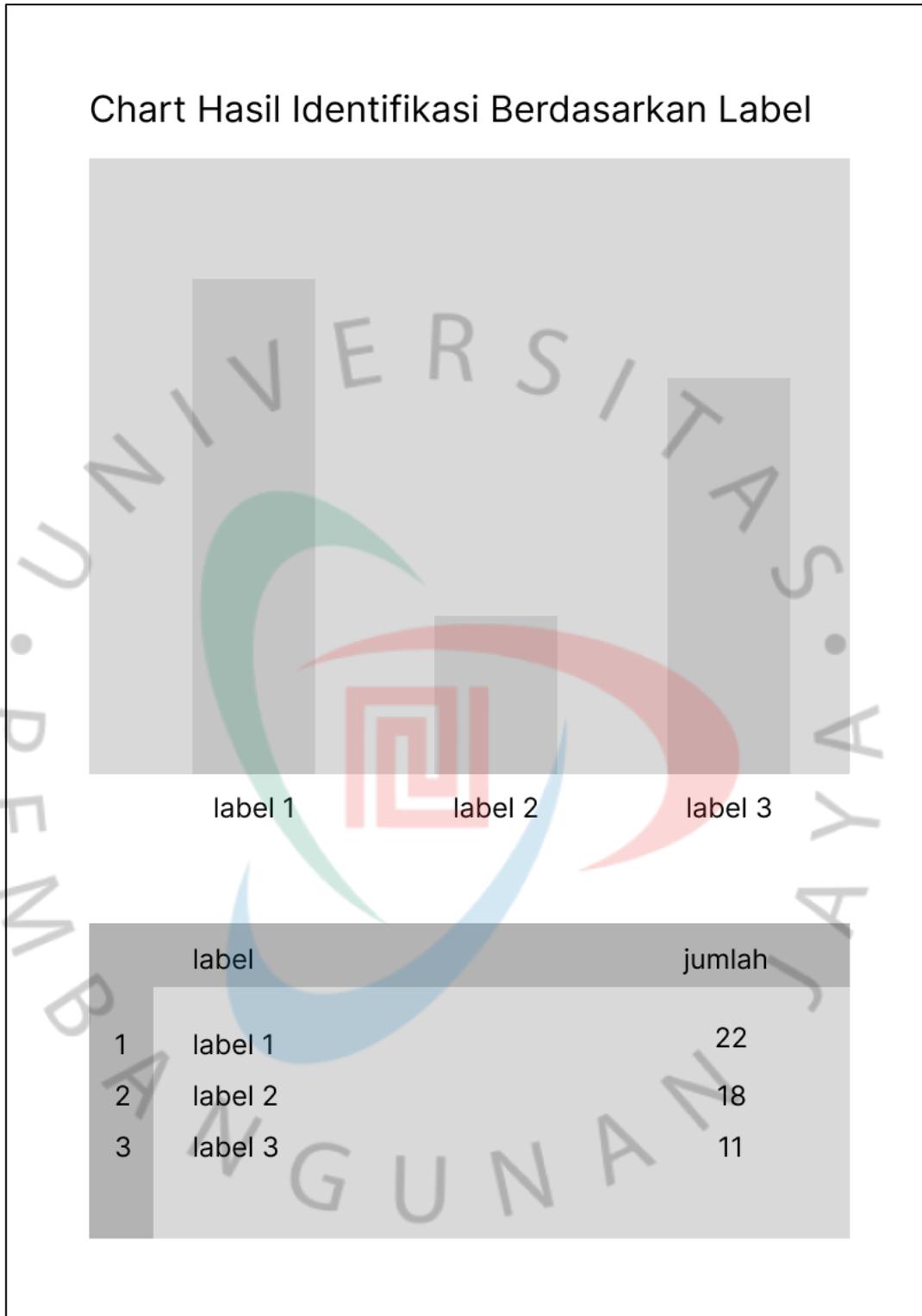
Gambar 4.10 adalah antarmuka pengguna untuk menampilkan kata kunci yang sering muncul dari kata kunci yang digunakan dalam penelitian. Terdapat 111 kata kunci yang telah ditetapkan dalam penelitian yang digunakan untuk pelabelan data pada hasil komentar. Daftar komentar ini hanya menjadi data untuk ditampilkan saja, kata kunci ini tidak akan mempengaruhi hasil prediksi. Kata kunci ini digunakan pada saat latihan model, namun pada aplikasi ini ditampilkan sebagai data daftar saja.

Hasil Identifikasi

	komentar	label	prediksi
1	Lorem ipsum	label	0.55
2	Lorem ipsum	label	1
3	Lorem ipsum	label	0.75
4	Lorem ipsum	label	0.98
5	Lorem ipsum	label	0.05

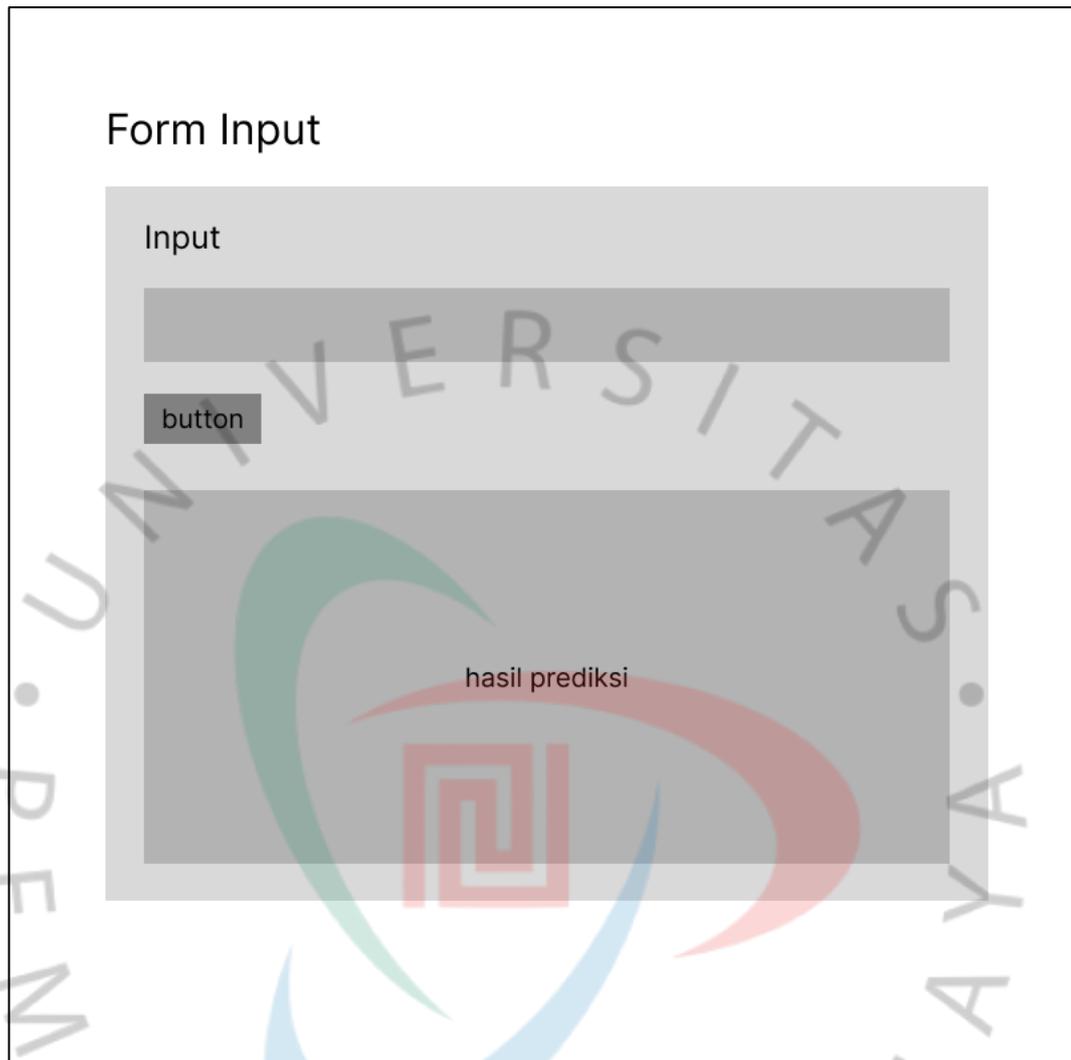
Gambar 4.11 Prototipe Antarmuka Hasil Identifikasi Aplikasi

Gambar 4.11 adalah antarmuka pengguna untuk menampilkan hasil prediksi model. Data yang akan ditampilkan pada pengguna adalah komentar, label, dan nilai prediksi model. Nilai prediksi model akan menentukan label komentar yang diprediksi. Pada bagian komentar akan menampilkan teks asli dari komentar yang dianalisis oleh model. Setiap komentar yang ditampilkan dalam bentuk teks yang dapat dibaca, sehingga pengguna dapat melihat konteks asli dari data yang dievaluasi. Pada bagian label akan menunjukkan kategori label berdasarkan hasil dari prediksi model. Pada bagian nilai prediksi model, bagian ini akan menampilkan skor atau probabilitas yang diberikan oleh model untuk setiap komentar. Nilai ini akan menampilkan dengan rentang 0 hingga 1. Nilai dari prediksi akan dikalikan 100 agar data ditampilkan secara persen. Hal ini dilakukan agar pembacaan data lebih mudah oleh pengguna.



Gambar 4.12 Prototipe Antarmuka Bagan Aplikasi

Gambar 4.12 adalah antarmuka pengguna untuk menampilkan bagan dari jumlah label pada datatest. Bagian ini akan memberikan ringkasan data dalam berbentuk bagan dan tabel dengan informasi label beserta banyaknya jumlah label.



Gambar 4.13 Prototipe Antarmuka *Form* Aplikasi

Gambar 4.13 adalah antarmuka pengguna untuk menampilkan *form* input pengguna. Pengguna dapat memasukkan data berupa teks yang nantinya akan diproses oleh model untuk diprediksi dan ditampilkan hasilnya.

4.5 Perancangan Pengujian Aplikasi

Peneliti melakukan perancangan pengujian aplikasi dengan menggunakan metode pengujian *black box* dan *white box*. Tujuan pembuatan rancangan pengujian adalah sebagai gambaran ketika aplikasi nanti akan diuji. Proses ini berguna untuk memberikan informasi input dan output aplikasi ketika akan diuji. Proses perancangan pengujian aplikasi akan dijelaskan sebagai berikut.

4.5.1 Black Box

Tabel 4.3 Rancangan Pengujia *Black Box*

Skenario	Hasil
Pengguna mengakses aplikasi pertama kali	Aplikasi akan menampilkan data yang sudah disiapkan seperti dataset, data test, kata kunci, hasil identifikasi dataset, dan bagan batang.
Pengguna memasukan input teks pada <i>form</i> yang disediakan.	Aplikasi menampilkan input teks yang dimasukan oleh pengguna.
Pengguna menekan tombol <i>submit</i> pada form untuk proses prediksi.	Aplikasi melakukan prediksi pada input teks dan menampilkannya pada pengguna setelah selesai prediksi.

Tabel 4.3 adalah rancangan skenario pengujian *black box*. Pengujian *black box* dilakukan untuk melihat proses aplikasi pada bagian tampilan dan menguji apakah ada error yang muncul ketika pengguna berinteraksi dengan aplikasi.

4.5.2 White Box

Tabel 4.4 Rancangan Pengujian *White Box*

Skenario	Hasil
Kode aplikasi untuk melakukan <i>case folding</i> .	Dataset komentar diubah menjadi huruf kecil dan dibersihkan dari <i>html tag</i> , <i>link url</i> , <i>multiple space</i> , angka, dan emoji.
Kode aplikasi untuk melakukan tokenisasi.	Dataset komentar diubah dalam bentuk <i>array</i> token.
Kode aplikasi untuk <i>stop words removal</i> .	Dataset komentar menghilangkan kata berdasarkan kamus dari <i>library</i> NLTK dan kamus manual.
Kode aplikasi untuk lematisasi	Dataset komentar akan mengubah kata menjadi kata dasar berdasarkan dari <i>library SpaCy</i> .
Kode aplikasi untuk filter data dengan indeks kurang dari 4	Dataset komentar akan menghapus data komentar yang panjang data tokennya kurang dari 4 indeks.
Kode aplikasi untuk menyimpan hasil praproses.	Dataset komentar yang sudah dilakukan tahap praproses akan disimpan dalam bentuk json.
Kode aplikasi untuk menyimpan tokenisasi.	Membuat file yang menyimpan proses tokenisasi dengan format <i>pickle</i> dari <i>library pickle</i>

Kode aplikasi untuk memuat model dan tokenisasi yang telah dibuat	Aplikasi memuat model dan tokenisasi yang telah dibuat
Kode aplikasi untuk pengguna memasukan input teks pada <i>form</i>	Aplikasi memproses input pengguna dan menjalankan prediksi dengan model

Tabel 4.4 adalah rancangan skenario pengujian *white box*. Pengujian *white box* dilakukan untuk melihat proses kode yang telah dibuat apakah berjalan sesuai dengan yang ditentukan. Proses pengujian dimulai dari tahap praproses, pemuatan model dan tokenisasi, dan proses prediksi input yang dilakukan oleh pengguna.

