BAB IV
PERANCANGAN

Pada bab ini menjelaskan secara detail mengenai analisis sistem terdahulu, spesifikasi sistem kebutuhan baru, perancangan sistem, perancangan tatap muka dan langkah-langkah perancangan sistem.

4.1 Analisis Sistem Terdahulu

Analisis sistem terdahulu bertujuan untuk mengevaluasi cara kerja, kelebihan, kekurangan, dan kebutuhan dari sistem sebelumnya. Pada tahap ini, peneliti dapat mengidentifikasi kebutuhan sistem dengan melakukan pengamatan. Analisis sistem yang telah ada juga membantu memastikan perubahan yang akan dilakukan pada sistem yang akan dikembangkan dan mengatasi kekurangan yang terdapat pada sistem sebelumnya.

Pada penelitian yang dilakukan oleh Adi Yahyadi dan Fitri Latifah (2022). Topik yang dibahas pada sistem terdahulu adalah sentimen terhadap kebijakan PPKM di Tengah pandemi *covid-19* . Metode pengolahan data atau pra-proses data yang digunakan hanya mencakup pembersihan data , case folding , dan tokenisasi.

4.2 Spesifikasi Kebutuhan Sistem Baru

Tahap yang dilakukan setelah analisis sistem sebelumnya adalah menetapkan spesifikasi kebutuhan untuk sistem yang akan dikembangkan. Spesifikasi sistem ini penting untuk mencapai tujuan penelitian, yaitu pengembangan aplikasi analisis sentimen menggunakan metode LSTM. Kebutuhan sistem meliputi beberapa komponen yang dijelaskan sebagai berikut.

1. Spesifikasi Perangkat Lunak (Software)

Perangkat lunak adalah elemen krusial dalam proses pengembangan sistem. Dalam penelitian ini, spesifikasi perangkat lunak ditetapkan oleh peneliti untuk mendukung pengembangan sistem. Di bawah ini adalah tabel spesifikasi perangkat lunak yang digunakan.

Tabel 4. 1 Kebutuhan Perangkat Lunak

No	Kebutuhan Perangkat Lunak
1	Web Browser
2	Visual Studio Code
3	Python 3.9
4	Sistem Operasi Windows
5	Koneksi Internet

Tabel 4.1 berisi spesifikasi perangkat lunak yang digunakan oleh peneliti dalam pengembangan aplikasi dan model LSTM. Web browser digunakan untuk menampilkan dan menjalankan aplikasi. Interpreter Python 3.9 dipilih karena stabil dan telah lama dirilis. Sistem operasi yang digunakan adalah Windows, yang terhubung ke internet.

2. Spesifikasi Perangkat Keras (*Hardware*)

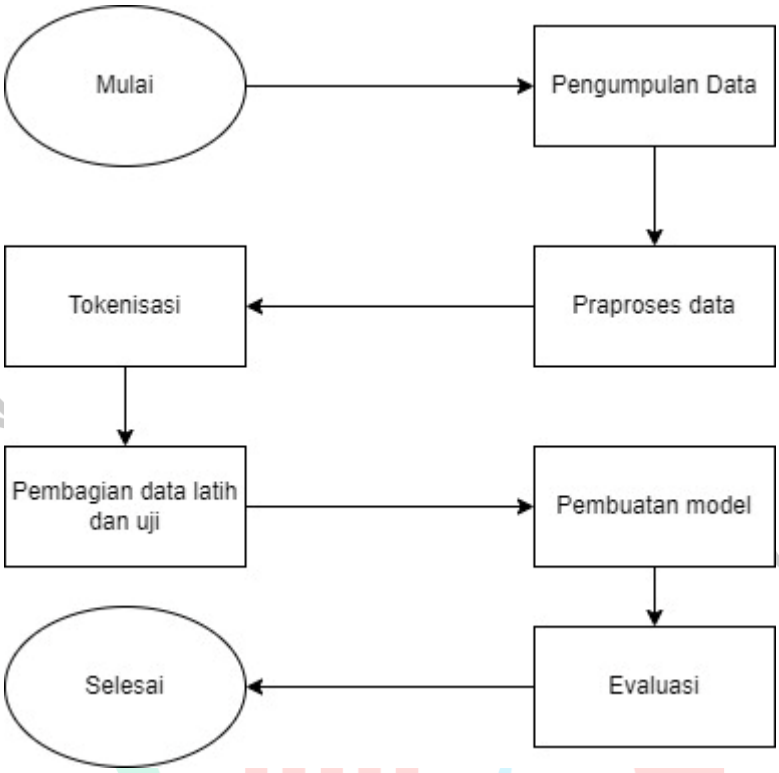
Perangkat keras diperlukan untuk mendukung pengoperasian perangkat lunak dan perangkat keras lainnya. Berikut ini adalah perangkat keras yang digunakan peneliti dalam pengembangan aplikasi.

Tabel 4. 2 Kebutuhan Perangkat keras

No	Perangkat Keras	Spesifikasi
1	*Processor*	Intel i5
2	RAM	8GB

Tabel 4.2 mencantumkan perangkat keras yang digunakan untuk menjalankan seluruh perangkat lunak. Peneliti menggunakan RAM 8GB untuk menghindari lag saat mengolah dataset. Prosesor yang digunakan adalah Intel Core i5 untuk melakukan pelatihan model LSTM. Semua data yang terkait dengan aplikasi dan model disimpan pada HDD.

3. Pembuatan Model

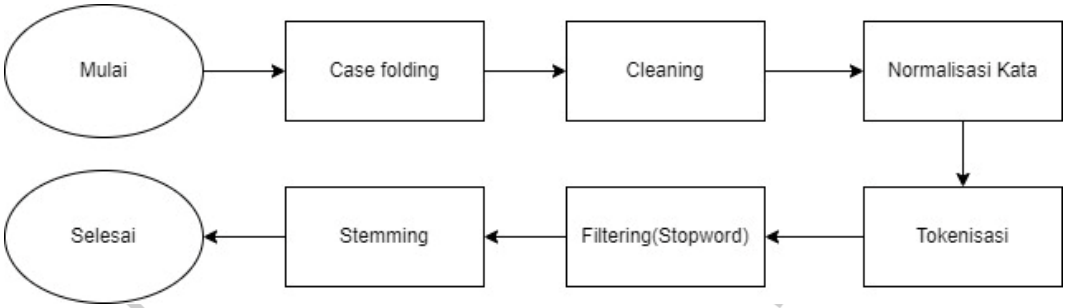


Gambar 4. 1 Flowchart Pembuatan Model

Gambar 4.1 adalah flowchart yang digunakan untuk membuat dan menguji model NLP. Proses pembuatan model merupakan bagian utama dalam penelitian ini. Tahapan yang dilakukan meliputi pengumpulan data, praproses data, pelabelan data, pembagian data latih dan uji, tokenisasi kata, pembuatan model, melakukan prediksi data menggunakan berbagai skenario pengujian, serta evaluasi model di berbagai skenario pengujian.

Pada tahap pengumpulan data, penulis mencari dan menyimpan informasi yang dibutuhkan dalam pembangunan sistem. Pengumpulan data dilakukan dengan Teknik scrapping. Pengambilan data menggunakan scraping dengan API adalah metode yang efisien dan terstruktur, di mana aplikasi mengirim permintaan ke endpoint API untuk mengakses dan mengunduh data yang diperlukan dari server, memungkinkan peneliti untuk mendapatkan informasi yang relevan dengan cara yang lebih cepat dan terorganisir dibandingkan dengan scraping tradisional dari

halaman web. Kata kunci yang digunakan adalah subsidi kendaraan listik dan kendaraan Listrik kemudian disimpan dalam format xlsx.



Gambar 4. 2 Flowchart Pra-proses Data

Praproses data (*data preprocessing*) adalah serangkaian langkah yang dilakukan untuk mempersiapkan dan membersihkan data mentah sebelum digunakan dalam analisis atau model pembelajaran mesin. Proses ini penting untuk meningkatkan kualitas data sehingga dapat menghasilkan hasil analisis yang lebih akurat dan andal pada tahap selanjutnya.

- *Case folding* adalah proses dalam pengolahan teks yang mengubah semua karakter huruf besar (*uppercase*) menjadi huruf kecil (*lowercase*). Tujuan dari casefolding adalah untuk memastikan konsistensi dalam pemrosesan teks, di mana kata-kata atau frasa yang sama tetapi ditulis dengan huruf besar atau kecil dianggap identik. Proses ini penting dalam analisis teks dan pengolahan bahasa alami untuk memastikan bahwa perbandingan string atau pencarian teks tidak dipengaruhi oleh perbedaan kapitalisasi.
- Cleaning atau pembersihan dalam konteks pengolahan data adalah proses menghapus atau memodifikasi data yang tidak diperlukan, tidak valid, atau tidak relevan dari dataset. Tujuan utama dari proses cleaning adalah untuk meningkatkan kualitas data dengan menghilangkan noise atau gangguan, sehingga data yang tersedia menjadi lebih konsisten dan dapat diandalkan untuk analisis lebih lanjut. Beberapa langkah yang umum dilakukan dalam proses cleaning data meliputi penghapusan data yang hilang atau tidak lengkap, penanganan nilai yang hilang, deduplikasi data, dan normalisasi

format data. Proses ini penting karena data yang bersih dan terstruktur akan memungkinkan pengguna untuk membuat keputusan yang lebih baik dan membangun model yang lebih akurat dalam berbagai aplikasi analisis data dan pembelajaran mesin.

- Normalisasi kata adalah proses dalam pengolahan teks yang bertujuan untuk mengubah variasi kata ke bentuk standar atau dasar yang sama. Tujuan utama dari normalisasi kata adalah untuk mengurangi kompleksitas dan meningkatkan konsistensi dalam analisis teks atau pemodelan bahasa.
- Tokenisasi adalah proses dalam pengolahan teks yang mengubah sebuah teks atau dokumen menjadi serangkaian token. Token adalah unit terkecil dari teks yang memiliki makna, seperti kata atau tanda baca. Proses tokenisasi membagi teks menjadi token-token ini berdasarkan aturan tertentu, seperti spasi antar kata atau tanda baca sebagai pemisah. Contoh sederhana dari tokenisasi adalah mengubah kalimat "Saya suka makan nasi goreng" menjadi token individu seperti ["Saya", "suka", "makan", "nasi", "goreng"]. Tokenisasi adalah langkah penting dalam pemrosesan bahasa alami dan analisis teks karena memungkinkan komputer untuk memahami dan mengolah teks dalam bentuk yang lebih terstruktur dan dapat dimengerti.
- Stopword removal adalah proses dalam pengolahan teks yang bertujuan untuk menghapus kata-kata umum yang tidak memberikan nilai tambah dalam analisis teks atau pemodelan bahasa. Kata-kata ini disebut sebagai "stopwords" karena mereka sering muncul dalam teks namun tidak membawa makna atau informasi yang signifikan untuk pemahaman konten atau tujuan analisis tertentu. Tujuan dari stopword removal adalah untuk meningkatkan kualitas analisis teks dengan mengurangi noise atau kata-kata yang tidak relevan, sehingga memfokuskan perhatian pada kata-kata kunci atau fitur penting dalam teks. Hal ini dapat meningkatkan kinerja dalam tugas-tugas seperti klasifikasi teks, analisis sentimen, atau pencarian informasi.
- Stemming adalah proses dalam pengolahan teks yang menghilangkan infleksi atau afiks dari kata-kata untuk menghasilkan bentuk dasar atau kata

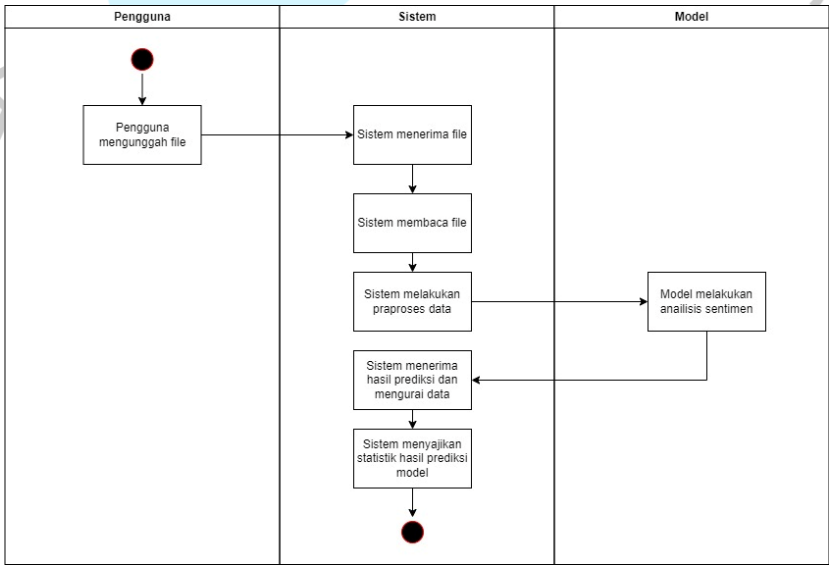
dasar yang disebut "stem". Tujuan utama dari stemming adalah untuk mengurangi kata-kata yang berbeda bentuknya menjadi bentuk yang sama sehingga kata-kata tersebut dapat dihitung sebagai satu entitas yang sama dalam analisis teks. Contoh sederhana dari stemming adalah mengubah kata-kata seperti "berjalan", "berjalanlah", dan "berjalan-jalan" menjadi kata dasar "jalan".

- Pelabelan data adalah proses memberikan tag atau label pada data untuk mengidentifikasi atau mengkategorikan informasi tertentu yang terkandung dalam data tersebut. Tujuan dari pelabelan data adalah untuk memungkinkan sistem komputer atau model pembelajaran mesin untuk memahami dan memproses informasi dengan lebih baik berdasarkan pada klasifikasi atau atribut yang telah ditentukan sebelumnya. Polaritas yang digunakan pada penelitian ini yaitu positif dan negatif.

4.3 Perancangan Sistem

Pada bagian ini, peneliti menjelaskan proses perancangan sistem untuk aplikasi analisis sentimen menggunakan Unified Modeling Language (UML). Model UML mencakup diagram Use Case, dan diagram Activity yang digunakan untuk mendesain sistem yang akan dikembangkan.

4.3.1 Activity Diagram

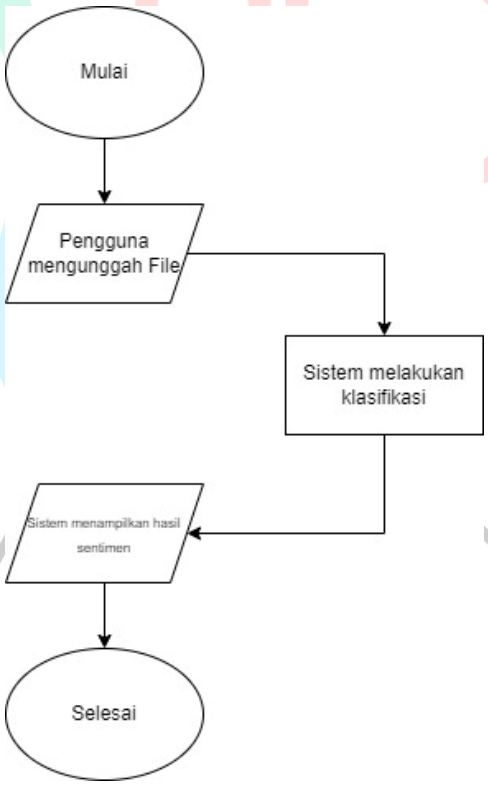


Gambar 4. 3 Activity Diagram

Gambar 4.3 menunjukkan diagram aktivitas yang menggambarkan alur kerja sistem. Diagram ini melibatkan tiga entitas utama yang saling terhubung: pengguna, sistem, dan model NLP. Proses dimulai ketika pengguna mengakses aplikasi dan mengunggah file dalam format xlsx.

Sistem kemudian menerima file masukan dan melakukan validasi terhadap formatnya. Setelah validasi selesai, sistem membaca konten file dan melakukan praproses data, mirip dengan langkah-langkah pengumpulan dataset pada model NLP. Data yang telah dibersihkan kemudian dianalisis oleh model NLP untuk menghasilkan hasil analisis. Hasil analisis ini disajikan oleh sistem kepada pengguna dalam bentuk statistik, seperti total sentimen yang ditemukan dan sentimen yang paling umum muncul dalam data tersebut.

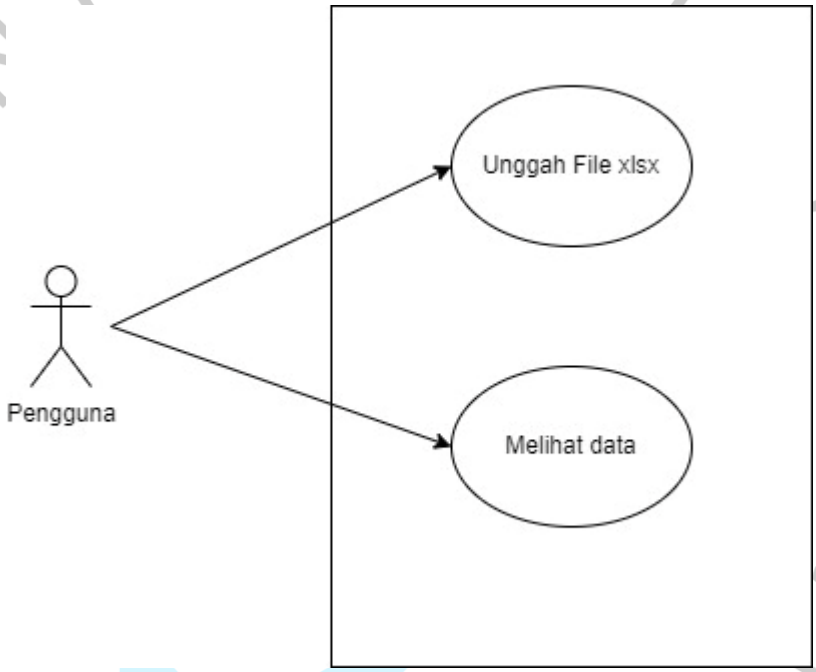
4.3.2 Flowchart



Gambar 4. 4 Flowchart Penggunaan

Gambar 4.4 menampilkan flowchart dari sistem analisis sentimen yang dibuat. Tahap pertama dari sistem ini adalah pengguna mengakses laman dan mengunggah file dalam format xlsx. File yang diunggah kemudian dibaca dan diverifikasi oleh sistem. Jika file sesuai dengan format yang ditentukan, sistem akan melakukan klasifikasi sentimen berdasarkan isi file tersebut. Hasil klasifikasi sentimen diolah oleh sistem untuk menampilkan statistik berupa total sentiment.

4.3.3 Use Case Diagram



Gambar 4. 5 Usecase Diagram Aplikasi

Gambar 4.4 menunjukkan bahwa pengguna aplikasi analisis sentimen memiliki akses ke berbagai fitur. Pengguna dapat mengakses aplikasi analisis sentimen, mengunggah file dalam format yang telah ditentukan. Selain itu, pengguna juga dapat melihat hasil pemrosesan data yang ditampilkan oleh sistem.

4.3.4 Skenario *Use Case*

Tabel 4. 3 Seknario Use case Unggah File

Nama U*se Case*	Unggah File
Aktor	*User*

Deskripsi	*User* melakukan unggah file ke aplikasi
Tindakan	*User* mengunggah file ke aplikasi sesuai format

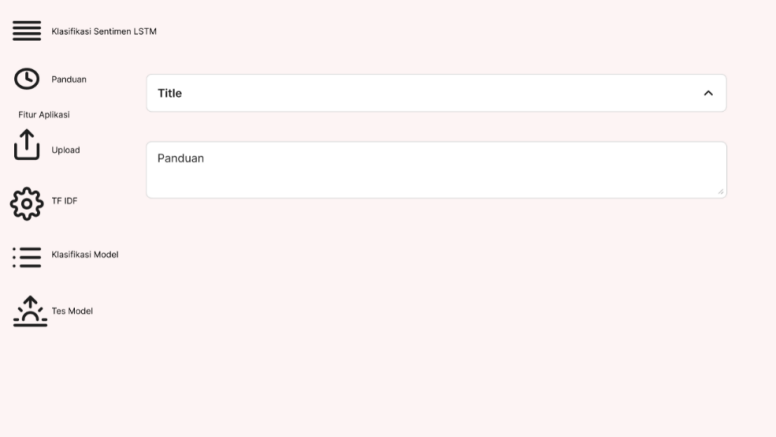
Tabel 4.1 adalah tabel use case untuk unggah file. Aktor yang dapat menggunakan fitur ini adalah semua pengguna yang mengakses aplikasi. Pengunggahan file oleh pengguna harus memenuhi persyaratan yang telah ditentukan.

Tabel 4. 4 Skenario Use Case Melihat Data

Nama U*se Case*	Melihat Data
Aktor	*User*
Deskripsi	*User* mampu melihat data yang disajikan sistem
Tindakan	*User* melihat data yang disajikan

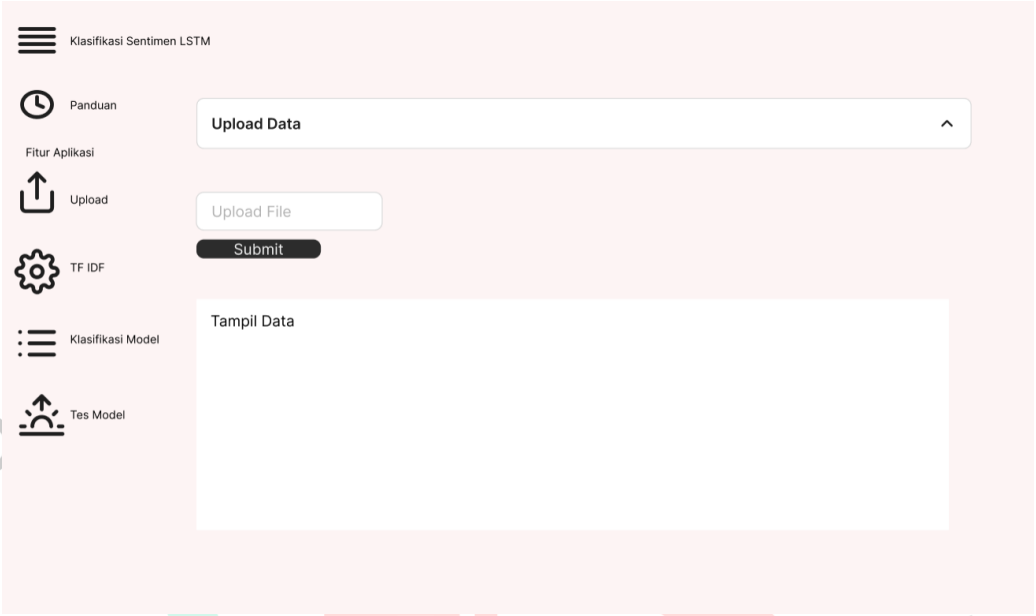
4.4 Perancangan Antar Muka

Perancangan antarmuka adalah proses merancang dan mengembangkan antarmuka pengguna yang mudah digunakan dan efektif untuk sistem yang sedang dibangun. Tujuannya adalah memastikan pengguna dapat berinteraksi dengan sistem dengan mudah dan memahami informasi yang disajikan dengan jelas. Berikut adalah perancangan antarmuka untuk pengguna.



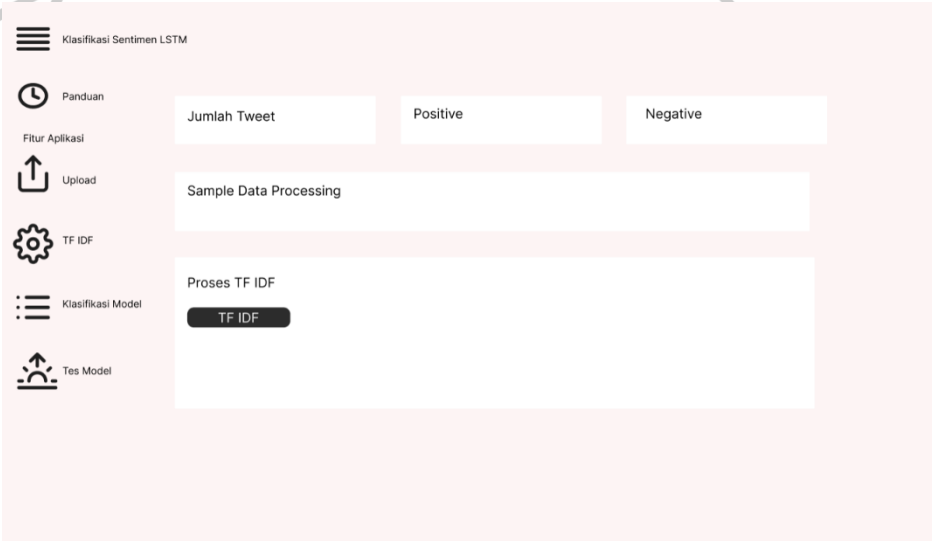
Gambar 4. 6 Rancangan Tampilan awal aplikasi

Gambar 4.5 merupakan tampilan awal aplikasi, tampilan awal aplikasi menampilkan judul dan panduan untuk menjalankan aplikasi.



Gambar 4. 7 Rancangan Tampilan Menu Upload

Gambar 4.6 merupakan tampilan dari menu *upload* data, Menu unggah data memungkinkan pengguna untuk mengunggah data mereka, yang kemudian akan ditampilkan dalam *dashboard* tampil data. Hasil dari tampilan data akan berupa kalimat yang sudah dilabeli sentiment positif dan negative.



Gambar 4. 8 Rancangan Tampilan Menu TF IDF

Gambar 4.7 merupakan rancangan menu TF-IDF pada website ini memungkinkan pengguna untuk menganalisis dan menilai kepentingan kata-kata dalam dokumen. Menu ini juga menampilkan total jumlah tweet keseluruhan, jumlah tweet positif, jumlah tweet negative serta menampilkan kalimat yang sudah dilabeli sentiment positif atau negatif.

4.1 Rancangan Pengujian

4.1.1 Rancangan Pengujian *Black Box*

Tabel 4.3 menggambarkan perencanaan pengujian black box yang mencakup beberapa skenario dan hasil yang diharapkan. Pengujian black box dirancang untuk mengevaluasi kinerja sistem dan mengurangi kemungkinan bug yang mungkin muncul pada antarmuka.

Tabel 4. 5 Skenario Pengujian Black Box

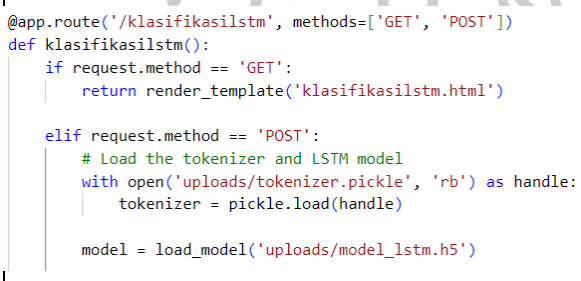
No	Skenario Pengujian	Hasil yang diharapkan
1	*User* memilih menu upload lalu memilih opsi *choose file* setelah itu menekan tombol *submit*.	*User* dapat memilih menu upload lalu sistem menampilkan halaman upload, Ketika user memilih opsi *choose file* maka sistem akan menampilkan jendela *file explorer* untuk memilih file lalu user dapat menekan tombol *submit* dan sistem menampilkan data.
2	*User* memilih menu TF IDF dan menekan tombol TF IDF	Sistem menampilkan halaman dan menampilkan hasil berupa jumlah data yang di unggah dan berapa

		jumlah sentiment positif dan negatif
3	*User* memilih menu Klasifikasi LSTM	Sistem memproses klasifikasi dan menampilkan hasil dari klasifikasi berupa *accuracy score*, *precision score*, *recall score* dan *f1 score*
4	*User* memilih menu Tes Model lalu menginput kalimat serta menekan tombol Tes model	Sistem memproses kalimat yang di masukan *user* berupa kalimat positif atau negatif lalu menampilkan hasil dari prediksi tersebut

4.5.2 Rancangan Pengujian *White Box*

Tabel 4.4 menggambarkan perencanaan pengujian white box yang terdiri dari beberapa skenario dan hasil yang diharapkan. Pengujian white box dirancang untuk mengevaluasi kesesuaian logika pemrograman berdasarkan input yang diberikan dan memverifikasi bahwa output yang dihasilkan sesuai dengan yang diharapkan.

Tabel 4. 6 Skenario Pengujian White Box

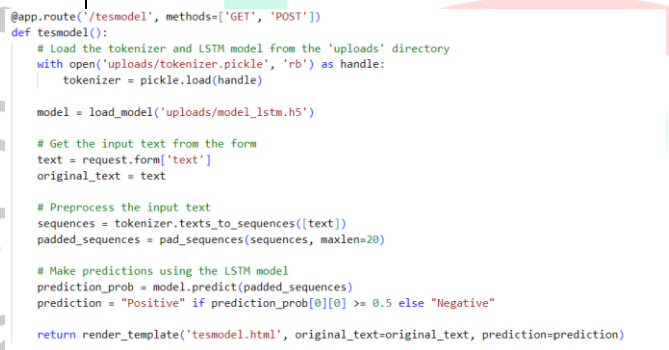
No	Skenario Pengujian	Hasil yang di harapkan
1		Aplikasi memuat model dan tokenizer

```
@app.route('/klasifikasilstm', methods=['GET', 'POST'])
def klasifikasilstm():
    if request.method == 'GET':
        return render_template('klasifikasilstm.html')

    elif request.method == 'POST':
        # Load the tokenizer and LSTM model
        with open('uploads/tokenizer.pickle', 'rb') as handle:
            tokenizer = pickle.load(handle)

        model = load_model('uploads/model_lstm.h5')
```

2		Menyamakan tipe huruf menjadi huruf kecil, menghapus username twitter dan *hyperlinks*
3		Menghilangkan emoji pada kalimat

4		Menghilangkan stopword pada kalimat
5		Melakukan prediksi yang kemudian dikonversi menjadi label "Positive" atau "Negative" berdasarkan nilai probabilitas.