

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab ini akan dibahas secara rinci mengenai proses pengembangan perangkat lunak untuk analisis layanan gojek berbasis aspek menggunakan algoritma Long-Short Term Memory (LSTM). Proses ini dilakukan dengan menggunakan metode pengembangan perangkat lunak RUP yang terdiri dari fase yaitu fase inepsi, fase elaborasi, fase konstruksi, dan fase transisi

4.2 Fase Inepsi

4.2.1 Pemodelan Bisnis

Dalam penelitian ini, kategori ulasan dibagi menjadi dua, yaitu positif dan negatif, dengan setiap kategori diberikan label berupa angka. Data ulasan yang digunakan dalam penelitian ini merupakan ulasan berbahasa Indonesia yang diperoleh dari *platform Play Store*. Metode yang digunakan dalam penelitian ini yaitu metode LSTM (*Long Short-Term Memory*).

Metode LSTM merupakan jenis jaringan saraf tiruan yang dirancang untuk mengatasi masalah ketergantungan jangka panjang (*long-term dependencies*) dengan menggunakan unit memori yang dikenal sebagai *cell state*. Unit ini dilengkapi dengan tiga gerbang utama, yaitu *input gate*, *forget gate*, dan *output gate*, yang memungkinkan LSTM untuk mempertahankan informasi penting sekaligus mengabaikan informasi yang sudah tidak relevan.

Dalam proses pelatihan LSTM, model disesuaikan dengan data pelatihan untuk mengenali pola dan hubungan dalam urutan data yang diberikan dengan tujuan mengoptimalkan parameter-parameter model seperti bobot, *input*,

output, dan lainnya. Proses ini melibatkan perhitungan operasi LSTM menggunakan sampel nilai input, pembobotan, nilai bobot, dan bias yang diinisialisasi secara acak. Berikut ini adalah perhitungan manual LSTM:

Diketahui:

$$x_t \quad \quad \quad : \quad 0.025$$

$$W_f, W_i, W_c, W_o \quad : \quad 0.45$$

$$b_f, b_i, b_c, b_o \quad : \quad 0.5$$

$$h_{t-1} \quad \quad \quad : \quad 0.1$$

$$c_{t-1} \quad \quad \quad : \quad 0.1$$

a. Menghitung *forget gate* (f_t)

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ &= \sigma(0.45 \cdot [0.1 + 0.025] + 0.5) \\ &= \sigma(0.045 + 0.025 + 0.5) \\ &= \sigma(0.57) \end{aligned}$$

Dengan fungsi *sigmoid* $\sigma(x)$:

$$\begin{aligned} \sigma(x) &= \frac{1}{1 + e^{-x}} \\ f_t &= \frac{1}{1 + e^{-0.57}} \\ &= 0.6389 \end{aligned}$$

b. Menghitung *input gate* (i_t)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$= \sigma(0.45 \cdot [0.1 + 0.025] + 0.5)$$

$$= \sigma(0.45 \cdot 0.125 + 0.5)$$

$$= \sigma(0.05625 + 0.5)$$

$$= \sigma(0.05625 + 0.5)$$

$$= \sigma(0.55625)$$

$$i_t = \frac{1}{1 + e^{-0.55625}}$$

$$= 0.6357$$

c. Menghitung *cell state* (c_t)

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c_t$$

$$= 0.6389 \cdot 0.1 + 0.6357 \cdot 0.2$$

$$= 0.6389 \cdot 0.1 + 0.12714$$

$$= 0.19103$$

d. Menghitung *ouput gate* (o_t)

$$o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o$$

$$= \sigma(0.45 \cdot [0.1 + 0.025] + 0.5)$$

$$= \sigma(0.45 \cdot 0.125 + 0.5)$$

$$= \sigma(0.05625 + 0.5)$$

$$= \sigma(0.55625)$$

$$o_t = \frac{1}{1 + e^{-0.55625}}$$

$$= 0.6357$$

e. Menghitung h_t

$$h_t = \tanh(c_t) \cdot o_t$$

$$\begin{aligned}
&= \tanh(0.19103) \cdot 0.6357 \\
&= (2 \cdot \sigma(2 \cdot 0.19103) - 1) \cdot 0.6357 \\
&= (2 \cdot \sigma(0.38206) - 1) \cdot 0.6357 \\
&= (2 \cdot \sigma(0.5945) - 1) \cdot 0.6357 \\
&= (1.1890 - 1) \cdot 0.6357 \\
&= 0.1890 \cdot 0.6357 \\
&= 0.1202
\end{aligned}$$

4.2.2 Kebutuhan Sistem

Dalam tahap ini diperlukan perancangan kebutuhan sistem yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional merupakan panduan utama dalam desain dan pengembangan sistem untuk memastikan bahwa sistem memenuhi kebutuhan pengguna dan tujuan bisnis. Sedangkan kebutuhan non fungsional memastikan bahwa sistem memenuhi standar kualitas, kinerja, dan keandalan yang diharapkan.

Tabel IV- 1. Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Perangkat lunak dapat melakukan <i>text preprocessing</i>
2.	Perangkat lunak dapat melakukan prediksi pada kalimat positif dan negatif yang diberikan oleh pengguna menggunakan metode LSTM (<i>Long Short-Term Memory</i>).
3.	Perangkat lunak dapat menerima masukan <i>dataset</i> berupa file dengan format CSV dari pengguna.

4.	Perangkat lunak dapat menampilkan hasil klasifikasi dalam bentuk <i>confusion matrix</i> dan <i>classification report</i> dari model yang sudah dibuat.
----	---

Tabel IV- 2.Kebutuhan Non Fungsional

No.	Kebutuhan Fungsional
1.	Perangkat lunak dapat menampilkan peringatan jika terjadi kesalahan

4.2.3 Analisis dan Desain

Tujuan pada tahap ini, yaitu menganalisis kebutuhan perangkat lunak dan merancang desain *use case diagram*.

4.2.3.1 Analisis Kebutuhan Perangkat Lunak

Penelitian ini bertujuan untuk mengembangkan sebuah perangkat lunak yang dapat menganalisis sentimen dari ulasan gojek melalui *platform Play Store*, yang diharapkan memiliki kemampuan yang mampu memprediksi kalimat positif dan negatif dalam sebuah kalimat dengan menggunakan algoritma LSTM.

1. Dapat menerima masukan dataset dari pengguna.
2. Dapat melakukan *text preprocessing*.
3. Dapat menentukan topik model LDA dengan aspek *user experience* dan *service*.
4. Mampu melakukan analisis sentimen berbasis aspek pada *dataset* yang diberikan oleh pengguna menggunakan metode LSTM (*Long Short-Term Memory*).

5. Mampu menampilkan hasil klasifikasi dalam bentuk *confusion matrix* dan hasil evaluasi model.

4.2.3.2 Analisis Data

Data yang digunakan pada penelitian ini merupakan ulasan dari *platform Play Store* yang diambil dari *website kaggle*⁴, dengan setiap ulasan sudah diberikan label positif dan negatif dalam berbentuk angka. Data disimpan dalam format *.csv* dengan total data yang digunakan 225.000. Setelah itu, data dibagi menjadi 80:20 dengan data latih 80% dengan total 161.371 data dan data uji 20% dengan total 54.171 data.

4.2.3.3 Analisis Text Preprocessing

Text preprocessing merupakan teknik yang dilakukan pada data teks mentah (*raw text*) untuk mempersiapkannya agar lebih mudah dianalisis dan diproses oleh model atau algoritma. Pada penelitian ini terdapat beberapa tahap *text preprocessing*, diantara yaitu *case folding*, *cleaning text*, normalisasi, tokenisasi, *stop word removal*, dan *stemming*. Contoh dataset sebelum *preprocessing* dapat dilihat pada Tabel IV-3.

Tabel IV- 3. Contoh Dataset Sebelum *Preprocessing*

No	Ulasan	Label
1.	Kebanyakan pembaruan	Negatif
2.	Payah jarang ada voucher semenjak bukan pak nadiem lagi	
3.	Rute gofoodnya dijauhin biar biaya antarnya nambah makin lama makin mahal dan ngga ada diskon error sering bgt customer service lama	

⁴ <https://www.kaggle.com/datasets/ucupsedaya/gojek-app-reviews-bahasa-indonesia>

4.	ONGKIRNYA MAHAL BANGET DAN TIDAK ADA PROMO ATAU VOUCHER PAKE GRAB AJA BANYAK PROMONYA	
5.	makin mahal harga gak stabil naik seenaknya gojek sendiri	
6.	Mantap	Positif
7.	Pelayanannya bagus	
8.	Bagus bgt sumpahh wajib download	
9.	Terus tingkatkan layanan gojek banyakin promonya banyakin cashback nya	
10.	Bagus banyak promo	

1. Case Folding

Case folding merupakan sebuah proses dalam pengolahan teks yang berfungsi untuk mengonversi seluruh karakter dalam suatu teks ke dalam format yang seragam, umumnya dengan mengubah semua huruf menjadi huruf kecil. Langkah ini bertujuan untuk menghilangkan perbedaan antara huruf kapital dan huruf kecil, sehingga analisis teks dapat dilakukan secara konsisten tanpa dipengaruhi oleh variasi bentuk huruf yang mungkin muncul pada data. Proses ini sangat penting untuk memastikan bahwa perbedaan penggunaan huruf besar dan kecil tidak memberikan dampak pada hasil analisis, terutama dalam konteks pengolahan data teks yang memerlukan standardisasi format. Dapat dilihat pada Tabel IV-4 hasil *case folding* pada teks.

Tabel IV- 4. Hasil *Case Folding*

No	Ulasan	Label
1.	kebanyakan pembaruan	Negatif
2.	payah jarang ada voucher semenjak bukan pak nadiem lagi	

3.	rute gofoodnya dijauhin biar biaya antarnya nambah makin lama makin mahal dan ngga ada diskon error sering bgt customer service lama	
4.	ongkirnya mahal banget dan tidak ada promo atau voucher pake grab aja banyak promonya	
5.	makin mahal harga gak stabil naik seenaknya gojek sendiri	
6.	mantap	
7.	pelayanannya bagus	
8.	bagus bgt sumpahh wajib download	Positif
9.	terus tingkatkan layanan gojek banyakin promonya banyakin cashback nya	
10.	bagus banyak promo	

2. *Cleaning*

Cleaning merupakan merupakan langkah-langkah penting dalam pemrosesan data teks yang bertujuan untuk menghapus elemen-elemen yang tidak diperlukan atau tidak relevan, sehingga teks yang dihasilkan menjadi lebih bersih dan siap untuk dianalisis lebih lanjut. Pada tahap ini, berbagai elemen yang tidak memberikan kontribusi signifikan terhadap pemahaman konten akan dihilangkan. Beberapa elemen yang umumnya dihapus selama proses *cleaning* termasuk karakter-karakter yang tidak penting seperti tanda baca misalnya koma, titik, tanda seru, serta simbol-simbol khusus seperti mata uang atau tanda matematis, angka yang tidak relevan, serta *whitespace* atau spasi berlebihan yang dapat mengganggu analisis data. Selain itu, *URL* dan alamat email yang sering ditemukan dalam teks, terutama yang diambil dari media sosial atau situs web, juga akan dihapus agar tidak mempengaruhi hasil analisis. Proses

pembersihan ini bertujuan untuk memastikan bahwa teks yang tersisa hanya berfokus pada informasi yang esensial dan relevan, memudahkan dalam proses analisis selanjutnya, seperti klasifikasi teks, ekstraksi informasi, atau analisis sentimen. Hasil dari proses *cleaning* ini dapat dilihat pada Tabel IV-5

Tabel IV- 5. Hasil *Cleaning*

No	Ulasan	Label
1.	kebanyakan pembaruan	Negatif
2.	payah jarang ada voucher semenjak bukan pak nadiem lagi	
3.	rute gofoodnya dijauhin biar biaya antarnya nambah makin lama makin mahal dan ngga ada diskon error sering bgt customer service lama	
4.	ongkirnya mahal banget dan tidak ada promo atau voucher pake grab aja banyak promonya	
5.	makin mahal harga gak stabil naik seenaknya gojek sendiri	
6.	mantap	Positif
7.	pelayanannya bagus	
8.	bagus bgt sumpahh wajib download	
9.	terus tingkatkan layanan gojek banyakin promonya banyakin cashback nya	
10.	bagus banyak promo	

3. Normalisasi

Normalisasi merupakan proses mengubah kata-kata yang tidak sesuai dengan kaidah baku menjadi bentuk yang lebih sesuai agar dapat diproses dengan lebih efisien. Hasil normalisasi dapat dilihat pada Tabel IV-6.

Tabel IV- 6. Hasil Normalisasi

No	Ulasan	Label
1.	kebanyakan pembaruan	Negatif
2.	payah jarang ada voucher semenjak bukan pak nadiem lagi	
3.	rute gofoodnya dijauhin biar biaya antarnya nambah makin lama makin mahal dan tidak ada diskon error sering banget customer service lama	
4.	ongkir mahal banget dan tidak ada promo atau voucher pakai grab aja banyak promonya	
5.	makin mahal harga tidak stabil naik seenaknya gojek sendiri	
6.	mantap	Positif
7.	pelayanannya bagus	
8.	bagus banget sumpah wajib download	
9.	terus tingkatkan layanan gojek banyakin promonya banyakin cashback nya	
10.	bagus banyak promo	

4. Tokenisasi

Tokenisasi merupakan proses pemecahan teks menjadi unit-unit yang lebih kecil dapat berupa kata, frasa, kalimat, atau simbol yang memiliki makna tertentu. Hasil tokenisasi dapat dilihat pada Tabel IV-7.

Tabel IV- 7. Hasil Tokenisasi

No	Ulasan	Label
1.	['kebanyakan', 'pembaruan']	Negatif
2.	['payah', 'jarang', 'voucher', 'semenjak', 'bukan', 'nadiem', 'lagi']	
3.	['rute', 'gofoodnya', 'dijauhin', 'biar', 'biaya', 'antarnya', 'nambah', 'makin', 'lama', 'makin', 'mahal', 'tidak', 'diskon',	

	'error', 'sering', 'banget', 'customer', 'service', 'lama']	Positif
4.	['ongkir', 'mahal', 'banget', 'tidak', 'promo', 'atau', 'voucher', 'pakai', 'grab', 'banyak', 'promonya']	
5.	['makin', 'mahal', 'harga', 'tidak', 'stabil', 'naik', 'seenaknya', 'gojek', 'sendiri']	
6.	['mantap']	
7.	['pelayanannya', 'bagus']	
8.	['bagus', 'banget', 'sumpah', 'wajib', 'download']	
9.	['terus', 'tingkatkan', 'layanan', 'gojek', 'banyakin', 'promonya', 'banyakin', 'cashback']	
10.	['bagus', 'banyak', 'promo']	

5. Stop Word Removal

Stop word removal merupakan proses penghilangan kata-kata yang dianggap tidak memberikan informasi penting dalam analisis teks atau pemrosesan bahasa alami. Kata-kata ini biasanya berupa kata penghubung, kata depan, kata ganti, atau kata-kata umum lainnya yang sering muncul dalam teks tetapi tidak berkontribusi signifikan terhadap makna atau konteks analisis. Hasil *stop word removal* dapat dilihat pada Tabel IV-8.

Tabel IV- 8. Hasil *Stop Word Removal*

No	Ulasan	Label
1.	['kebanyakan', 'pembaruan']	Negatif
2.	['payah', 'jarang', 'voucher', 'semenjak', 'nadiem']	
3.	['rute', 'gofoodnya', 'dijauhin', 'biar', 'biaya', 'antaranya', 'nambah', 'mahal', 'diskon', 'error', 'banget', 'customer', 'service']	

4.	['ongkir', 'mahal', 'banget', 'promo', 'voucher', 'pakai', 'grab', 'promonya']	Positif
5.	['mahal', 'harga', 'stabil', 'gojek']	
6.	['mantap']	
7.	['pelayanannya', 'bagus']	
8.	['bagus', 'banget', 'sumpah', 'wajib', 'download']	
9.	['tingkatkan', 'layanan', 'gojek', 'banyakin', 'promonya', 'banyakin', 'cashback']	
10.	['bagus', 'banyak', 'promo']	

6. *Stemming*

Stemming merupakan proses mengubah sebuah kata menjadi bentuk dasarnya dengan tujuan utama untuk mengelompokkan kata-kata yang memiliki akar yang sama. Langkah ini dilakukan untuk mengurangi variasi kata dalam dokumen, sehingga analisis teks dapat menjadi lebih sederhana dan terfokus pada makna inti dari kata-kata tersebut. Proses ini membantu mengidentifikasi hubungan antara kata-kata yang berbeda bentuk tetapi memiliki akar yang sama, sehingga data teks dapat dianalisis secara lebih efektif. Hasil *stemming* pada teks dapat dilihat pada Tabel IV-9.

Tabel IV- 9. Hasil *Stemming*

No	Ulasan	Label
1.	['kebanyakan', 'pembaruan']	Negatif
2.	['payah', 'jarang', 'voucher', 'semenjak', 'nadiem']	
3.	['rute', 'gofoodnya', 'dijauhin', 'biar', 'biaya', 'antarnya', 'nambah', 'mahal', 'diskon', 'error', 'banget', 'customer', 'service']	
4.	['ongkir', 'mahal', 'banget', 'promo', 'voucher', 'pakai', 'grab', 'promonya']	
5.	['mahal', 'harga', 'stabil', 'gojek']	

6.	['mantap']	Positif
7.	['pelayanannya', 'bagus']	
8.	['bagus', 'banget', 'sumpah', 'wajib', 'download']	
9.	['tingkatkan', 'layanan', 'gojek', 'banyakin', 'promonya', 'banyakin', 'cashback']	
10.	['bagus', 'banyak', 'promo']	

7. *Bag of Words*

Bag of Words (BoW) merupakan metode representasi teks dalam bentuk numerik yang sering digunakan dalam analisis teks dan pemrosesan bahasa alami karena kesederhanaannya serta kemampuannya untuk menangkap pola umum dalam teks. Pendekatan ini merepresentasikan teks tanpa memperhatikan aturan tata bahasa, urutan kata dalam kalimat, maupun konteks semantik antar kata, sehingga fokus utama dari metode ini adalah pada frekuensi kemunculan kata-kata individu dalam teks. Dalam metode BoW, kumpulan dokumen atau *corpus* diubah menjadi sebuah kosakata, yaitu daftar semua kata unik yang ditemukan dalam teks, tanpa memedulikan struktur atau hubungan antar kata.

Setelah kosakata terbentuk, setiap dokumen diubah menjadi vektor berdasarkan frekuensi kemunculan kata-kata dari kosakata tersebut, yang memungkinkan untuk perbandingan atau analisis numerik. Proses metode BoW dimulai dengan tokenisasi, yang memecah teks menjadi kata atau frasa untuk analisis. Selanjutnya, kosakata dibangun dengan mengumpulkan kata-kata unik dari seluruh dokumen dalam korpus. Terakhir, dibuat matriks frekuensi yang menggambarkan jumlah kemunculan kata dalam dokumen untuk analisis statistik. Contoh hasilnya dapat dilihat pada Tabel IV-10, IV-11, dan IV-12.

Tabel IV- 10. Contoh Data Hasil Token

<i>Text_ID</i>	<i>Text</i>	Hasil token
1.	Sangat bermanfaat sekali saat di butuhkan dalam keadaan urgent	'sangat', 'bermanfaat', 'sekali', 'saat', 'di', 'butuhkan', 'dalam', 'keadaan', 'urgent'
2.	Kebanyakan pembaruan	'kebanyakan', 'pembaruan'

Tabel IV- 11. Kata Unik

No	Kata unik
1.	bermanfaat
2.	butuhkan
3.	dalam
4.	di
5.	keadaan
6.	kebanyakan
7.	pembaruan
8.	saat
9.	sangat
10.	sekali
11.	urgent

Tabel IV- 12. Hasil Kata Unik Tiap Topik

<i>Text_ID</i>	Kata Unik										
	1	2	3	4	5	6	7	8	9	10	11
1.	1	1	1	1	1	0	0	1	1	1	1
2.	0	0	0	0	0	1	1	0	0	0	0

8. *Latent Dirichlet Allocation (LDA)*.

Setelah tahap *bag of words* selesai, langkah berikutnya dalam penelitian ini adalah melakukan pemodelan topik menggunakan *Latent Dirichlet Allocation (LDA)*. Model LDA memiliki peran signifikan dalam analisis sentimen karena mampu mengidentifikasi topik utama dalam ulasan pelanggan. Melalui pendekatan ini, distribusi kata pada setiap topik memberikan gambaran yang komprehensif tentang aspek-aspek yang sering menjadi perhatian pelanggan. Selain itu, LDA dievaluasi dengan menggunakan *coherence score*. Beberapa parameter utama yang diperlukan untuk model ini mencakup *Bag of Words*, jumlah topik, kamus atau *dictionary*, *chunksize*, *passes*, *random state*, dan *alpha*. Parameter "*passes*" menentukan jumlah iterasi atau siklus pelatihan yang dilakukan pada model. Pada parameter jumlah topik, LDA mengidentifikasi pola berdasarkan distribusi kata dalam dokumen dan menyesuaikan dengan jumlah topik yang sudah ditentukan. Selama evaluasi, *coherence score* digunakan untuk menilai sejauh mana topik yang dihasilkan dapat dipahami dengan jelas dan memiliki konsistensi yang tinggi.

Pada penelitian ini, dilakukan pelatihan beberapa model LDA dengan jumlah topik yang bervariasi, dimulai dari *num_topics* 2 untuk menghindari pemodelan dengan hanya 1 topik, yang tidak memberikan informasi berguna dalam banyak kasus analisis topik. Proses pelatihan kemudian diakhiri dengan *num_topics* 10, dikarenakan semakin banyak jumlah topik, semakin sulit untuk menginterpretasikan topik-topik tersebut. Selain itu, proses pelatihan dapat memakan waktu lebih lama dan berisiko menyebabkan *overfitting* akibat

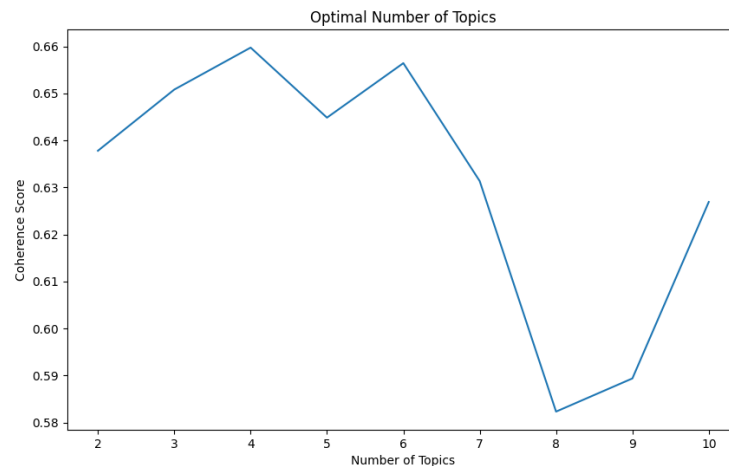
banyaknya topik kecil yang tidak bermakna. Nilai *coherence* digunakan untuk menilai kualitas topik yang dihasilkan oleh model, dan model dengan nilai *coherence* tertinggi dipilih sebagai model terbaik. Hasil nilai *coherence* dapat dilihat pada Tabel IV-13.

Tabel IV- 13. Hasil *Cohorence Score*

No	Num_Topics	Coherence_Score
1.	2	0.637802
2.	3	0.650813
3.	4	0.659738
4.	5	0.644843
5.	6	0.656431
6.	7	0.631361
7.	8	0.582332
8.	9	0.589385
9.	10	0.626915

Dapat dilihat pada Tabel IV-13, bahwa *Coherence Score* meningkat seiring dengan bertambahnya jumlah topik hingga mencapai puncaknya pada 4 topik, dengan skor tertinggi 0.659738. Hal ini menunjukkan bahwa model dengan 4 topik menghasilkan topik yang paling koheren dan relevan, sehingga lebih efektif dalam menangkap hubungan antar topik. Model dengan 6 topik masih menunjukkan *Coherence Score* yang cukup baik dengan nilai 0.656431, tetapi semakin rendah pada 8 topik dengan nilai 0.582332 dan 9 topik pada nilai 0.589385. Penurunan ini mengindikasikan bahwa semakin banyak topik yang dihasilkan, semakin sulit bagi model untuk mempertahankan relevansi antar

topik, yang dapat mengurangi kualitas informasi yang diperoleh. Dapat dilihat visualisasi *coherence score* pada Gambar IV-1.



Gambar IV- 1. Grafik Visualisasi Cohorence

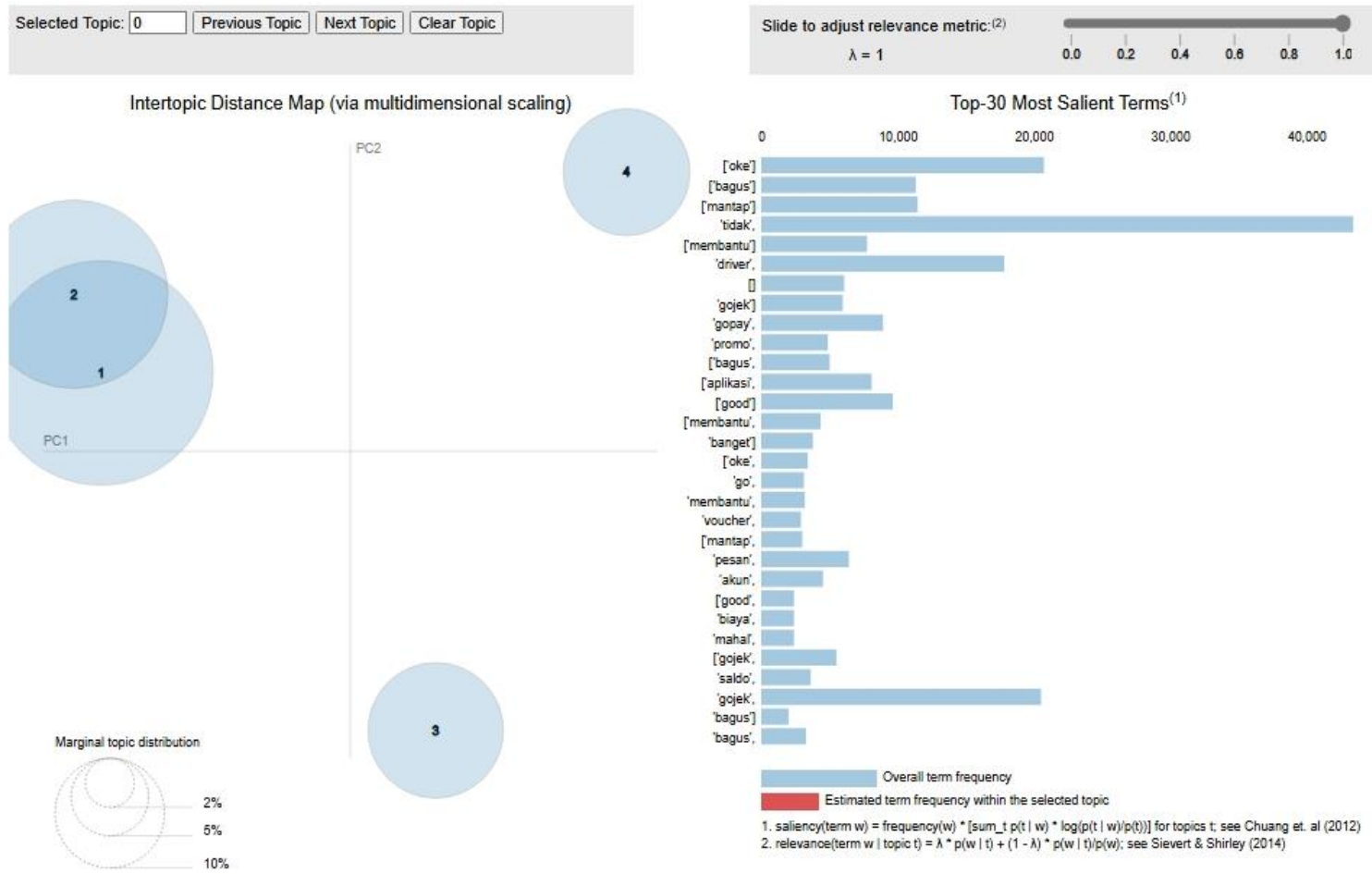
Setelah menentukan jumlah topik berdasarkan nilai *Coherence Score* tertinggi, model LDA akan dibangun dengan jumlah topik sebanyak 4, dengan setiap topik akan menampilkan 15 kata yang memiliki bobot masing-masing. Hasil dari pemodelan topik LDA dapat dilihat pada Tabel IV-14.

Tabel IV- 14. Hasil Pemodelan Topik LDA

Topik 1
'0.054*"tidak" + 0.02*"gopay" + 0.014"pakai" + 0.014*"gojek" + 0.013*"akun" + 0.010*"nya" + 0.010*"saldo" + 0.010*"tidak" + 0.009*"masuk" + 0.008*"aplikasi" + 0.006*"hp" + 0.006*"belum" + 0.006*"aja" + 0.005*"up"
Topik 2
'0.047*"tidak" + 0.035*"driver" + 0.024*"gojek" + 0.023*"nya" + 0.019*"aplikasi" + 0.013*"pesan"+ 0.012*"banget"+ 0.010*"aja" + 0.008*"gofood" + 0.008*"ya" + 0.008*"sulit"
Topik 3

'0.030*"promo" + 0.019*"go" + 0.018*"voucher" + 0.008*"aplikasi"'
Topik 4
'0.032*"gojek" + 0.010*"bagus" + 0.009*"kasih" + 0.008*"aplikasi"'

Pada langkah selanjutnya, keluaran dari model LDA yang telah dijalankan sebelumnya akan divisualisasikan menggunakan *pyLDAvis* yang merupakan sebuah alat visualisasi untuk menganalisis dan memahami struktur topik yang dihasilkan oleh model. Melalui visualisasi ini, topik-topik yang teridentifikasi akan dievaluasi lebih lanjut berdasarkan kata-kata yang telah diberi bobot dan dimodelkan dengan menggunakan metode LDA. Proses visualisasi topik ini akan menggunakan nilai *coherence* tertinggi yang diperoleh dari model LDA sebelumnya, yang dapat membantu dalam menentukan kualitas topik yang dihasilkan. Dalam hal ini, jumlah topik yang dipilih adalah 4, yang mengindikasikan bahwa, berdasarkan analisis terhadap kumpulan data ulasan yang telah diproses, terdapat 4 topik utama yang paling sering muncul dan menjadi pusat pembahasan dalam dataset tersebut. Pada panel kiri sisi kiri merupakan pemetaan jarak dari antar topik atau *intertopic distance map via multidimensional scaling* yang terdapat pada *cluster* topik. Sedangkan pada panel sisi kanan terdapat 30 buah terminologi yang paling relevan untuk topik tertentu. Visualisasi topik dapat dilihat pada Gambar IV-2.



Gambar IV- 2. Visualisasi Topik

Berdasarkan hasil visualisasi yang diperoleh dari interpretasi topik yang telah dianalisis, kata-kata yang sering muncul dalam setiap topik dapat dimanfaatkan untuk mengidentifikasi aspek-aspek yang relevan yang berkaitan dengan topik tersebut. Proses analisis ini berfungsi untuk memberikan pemahaman yang lebih mendalam mengenai fokus utama dari setiap topik yang telah diidentifikasi.

9. *Word2vec*

Langkah selanjutnya dalam penelitian ini adalah menerapkan teknik *word embedding* dengan menggunakan algoritma *Word2vec*. *Word embedding* adalah teknik yang digunakan untuk mengubah kata-kata dalam teks menjadi representasi vektor numerik, sehingga memungkinkan mesin untuk lebih memahami arti dan hubungan antar kata dalam konteksnya. Dalam analisis sentimen berbasis aspek pada ulasan aplikasi Gojek, penerapan *Word2Vec* akan memungkinkan kita untuk mengeksplorasi dan memahami bagaimana kata-kata yang berkaitan dengan penelitian ini saling berinteraksi, serta bagaimana hubungan antar kata-kata tersebut dapat memengaruhi sentimen keseluruhan dalam teks yang dianalisis. Dengan menghasilkan representasi vektor kata, kita dapat mencerminkan makna dan konteks kata-kata, yang memungkinkan kita untuk melakukan analisis yang lebih mendalam mengenai bagaimana kata-kata tersebut memengaruhi persepsi pengguna Gojek. Pada Tabel IV-15 dapat dilihat contoh dari *word vector* dari kata “mantap”.

Tabel IV- 15. Contoh *Word Vector* 100 Dimensi

<i>Word Vector</i>
[-5.3622725e-04 2.3643136e-04 5.1033497e-03 9.0092728e-03
-9.3029495e-03 -7.1168090e-03 6.4588725e-03 8.9729885e-03
-5.0154282e-03 -3.7633716e-03 7.3805046e-03 -1.5334714e-03
-4.5366134e-03 6.5540518e-03 -4.8601604e-03 -1.8160177e-03
2.8765798e-03 9.9187379e-04 -8.2852151e-03 -9.4488179e-03
7.3117660e-03 5.0702621e-03 6.7576934e-03 7.6286553e-04
6.3508903e-03 -3.4053659e-03 -9.4640139e-04 5.7685734e-03
-7.5216377e-03 -3.9361035e-03 -7.5115822e-03 -9.3004224e-04
9.5381187e-03 -7.3191668e-03 -2.3337686e-03 -1.9377411e-03
8.0774371e-03 -5.9308959e-03 4.5162440e-05 -4.7537340e-03
-9.6035507e-03 5.0072931e-03 -8.7595852e-03 -4.3918253e-03
-3.5099984e-05 -2.9618145e-04 -7.6612402e-03 9.6147433e-03
4.9820580e-03 9.2331432e-03 -8.1579173e-03 4.4957981e-03
-4.1370760e-03 8.2453608e-04 8.4986202e-03 -4.4621765e-03
4.5175003e-03 -6.7869602e-03 -3.5484887e-03 9.3985079e-03
-1.5776526e-03 3.2137157e-04 -4.1406299e-03 -7.6826881e-03
-1.5080082e-03 2.4697948e-03 -8.8802696e-04 5.5336617e-03
-2.7429771e-03 2.2600652e-03 5.4557943e-03 8.3459532e-03
-1.4537406e-03 -9.2081428e-03 4.3705525e-03 5.7178497e-04
7.4419081e-03 -8.1328274e-04 -2.6384138e-03 -8.7530091e-03
-8.5655687e-04 2.8265631e-03 5.4014288e-03 7.0526563e-03

-5.7031214e-03	1.8588197e-03	6.0888636e-03	-4.7980510e-03
-3.1072604e-03	6.7976294e-03	1.6314756e-03	1.8991709e-04
3.4736372e-03	2.1777749e-04	9.6188262e-03	5.0606038e-03
-8.9173904e-03	-7.0415605e-03	9.0145587e-04	6.3925339e-03]

4.2.3.4 Analisis Proses Klasifikasi

Langkah berikutnya dalam penelitian ini adalah melakukan penyetelan *hyperparameter* agar mengoptimalkan kinerja model dengan menggunakan *hyperparameter* tuning. Tujuan dari langkah ini adalah untuk menentukan kombinasi nilai *hyperparameter* yang paling efisien dalam membangun model. Rentang nilai *hyperparameter* yang digunakan dapat dilihat pada Tabel IV-16.

Tabel IV- 16. Rentang Nilai *Hyperparameter*

<i>Hyperparameter</i>	Rentang
LSTM Units	64, 128
<i>dropout</i>	0.2, 0.3
<i>recurrent dropout</i>	0.2, 0.3
<i>Epochs</i>	10, 20
<i>Batch_size</i>	128
<i>Learning Rate</i>	1e-3, 1e-4, 1e-5

Saat melakukan proses *hyperparameter tuning*, *Validation Accuracy* menjadi tolak ukur untuk menentukan model terbaik dari proses klasifikasi ini. Hasil dari *validation accuracy* tertinggi akan disimpan menjadi best *hyperparameter* model dengan format .h5. Kombinasi dengan *hyperparameter*

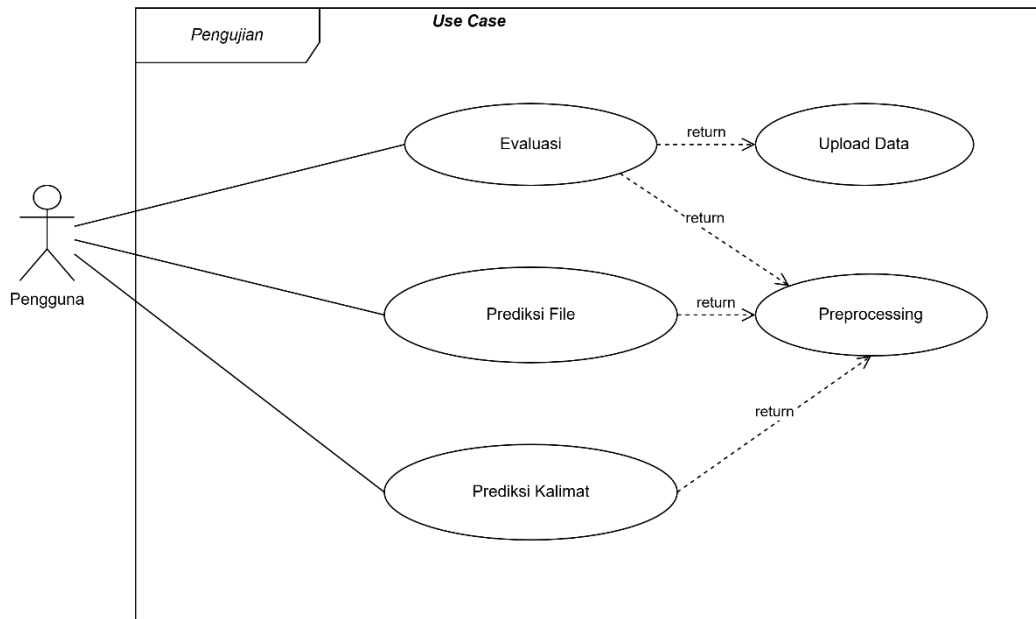
terbaik dari penelitian ini adalah menggunakan *LSTM units* 128, *Dropout* 0.2, *Reccurent dropout* 0.3, *learning rate* 0.0001 pada *epochs* 10 dan *batch size* 128.

4.2.4 Implementasi

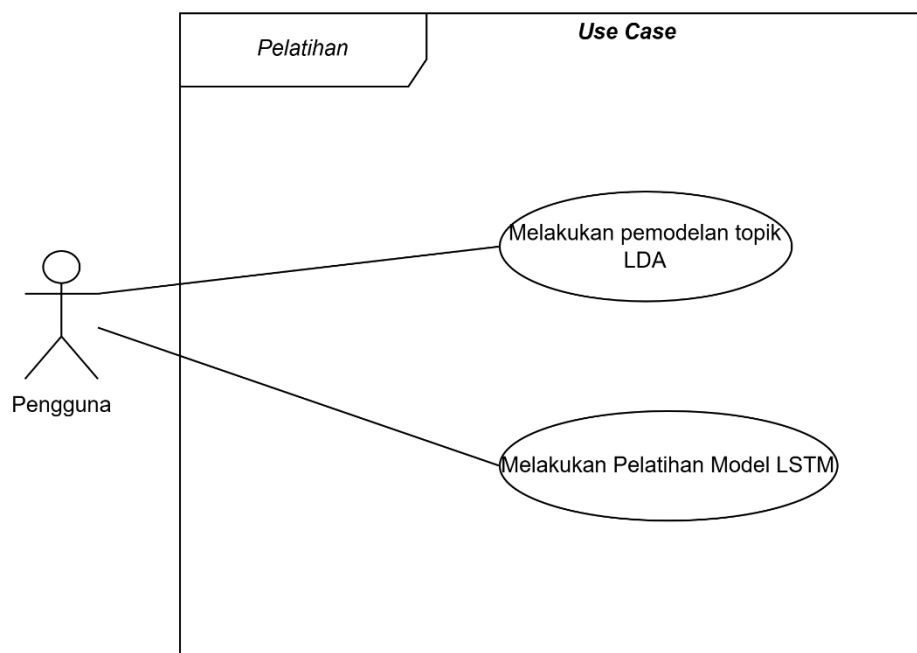
Perangkat lunak akan mempresentasikan penggambaran kasus penggunaan atau *use case* untuk menggambarkan secara rinci cara perangkat lunak yang akan berinteraksi dengan pengguna serta menjelaskan fungsionalitas utama yang akan disediakan.

4.2.4.1 Use Case

Diagram *use case* adalah salah satu jenis diagram yang digunakan dalam perancangan perangkat lunak untuk menggambarkan hubungan antara pengguna atau aktor dengan sistem yang sedang dikembangkan, serta skenario fungsi atau layanan yang disediakan oleh sistem kepada penggunanya. Dalam diagram ini, hubungan tersebut digambarkan secara jelas, sehingga memberikan gambaran yang komprehensif mengenai cara penggunaan sistem, berbagai fungsi utama yang dijalankan oleh sistem, dan aktor-aktor yang terlibat dalam setiap interaksi dengan sistem tersebut. Selain itu, diagram *use case* juga memungkinkan untuk memetakan alur atau urutan aktivitas yang dilakukan oleh aktor dalam berinteraksi dengan sistem. Visualisasi dari diagram ini sangat penting untuk pemahaman yang lebih baik tentang kebutuhan pengguna dan fungsionalitas yang harus dimiliki sistem. Diagram ini dapat dilihat lebih rinci pada Gambar IV-3 dan IV-4.



Gambar IV- 3. *Diagram Use Case Pengujian*



Gambar IV- 4. *Diagram Use Case Pelatihan*

4.2.4.2 Tabel Definisi Pengguna

Pengguna adalah pihak yang berperan sebagai aktor utama dalam kasus ini, dan informasi lengkap mengenai aktor tersebut dapat ditemukan dalam Tabel IV-17 dan IV-18.

Tabel IV- 17. Definisi Pengguna Pengujian

No	Aktor	Definisi
1.	Pengguna	Aktor orang atau kelompok yang menggunakan aplikasi atau sistem untuk menjalankan fungsi atau tugas tertentu yang disediakan oleh sistem tersebut. Pengguna dapat berperan sebagai aktor utama dalam use case diagram, yang menggambarkan interaksi mereka dengan sistem dalam menjalankan kasus penggunaan tertentu.

Tabel IV- 18. Definisi Pengguna Pelatihan

No	Aktor	Definisi
1.	Pengguna	Aktor orang atau kelompok yang dapat berinteraksi dengan perangkat lunak untuk melakukan proses pemodelan topik dan pelatihan model LSTM.

4.2.4.3 Tabel Definisi *Use Case*

Tabel dibawah ini berisi daftar definisi untuk setiap *use case* penggunaan yang terdapat dalam perangkat lunak. Definisi masing-masing kasus penggunaan dapat dilihat pada Tabel IV-19 dan Tabel IV-20.

Tabel IV- 19. Definisi *Use Case* Pengujian

No	<i>Use Case</i>	Penjelasan
1.	<i>Upload Data</i>	Prosedur untuk mengunggah data ke dalam sistem. Pengambilan data diambil dari sumber <i>Kaggle</i> .
2.	<i>Preprocessing</i>	Mengolah data mentah sebelum data dapat dilanjutkan dan dianalisis
3.	Evaluasi	Pada proses ini dilakukan pelatihan dan evaluasi pengujian data yang sudah dilakukan pra-pemrosesan
4.	Prediksi Data	Melakukan pengujian analisis dengan memanfaatkan model proses untuk menghasilkan label sentimen dan aspek dari data yang diunggah oleh pengguna.
5.	Prediksi Kalimat	Pada fase ini, teks yang sudah dimasukkan oleh pengguna akan diprediksi untuk menghasilkan keluaran berupa prediksi sentimen dan kategori aspek.

Tabel IV- 20. Definisi *Use Case* Pelatihan

No	<i>Use Case</i>	Penjelasan
1.	Melakukan Pemodelan Topik LDA	Sistem dapat melakukan pemodelan topik menggunakan dataset yang telah dilakukan pra-pemrosesan.
2.	Melakukan Pelatihan Model LSTM	Sistem dapat melakukan pelatihan model LSTM menggunakan dataset yang telah disiapkan untuk proses selanjutnya.

4.2.4.4 Tabel Skenario *Use Case*

Skenario *use case* yang telah dijelaskan dapat dilihat pada Tabel IV-21 , IV-22, IV-23, IV-24, dan IV-25.

Tabel IV- 21. Skenario *Use Case Upload Data*

Identifikasi	
Nomor	TS-001
Nama	<i>Upload Data</i>
Tujuan	Mengunggah file berupa teks dengan format CSV.
Deskripsi	Untuk melakukan analisis sentimen, pengguna harus mengunggah file yang berisi teks dan melakukan pra-pemrosesan data
Aktor	Pengguna
Kondisi Awal	Sistem belum menerima dataset
Skenario Utama	
Aktor	Sistem
1. Pengguna mengklik tombol “Pilih File”	
	2. Menampilkan jendela berkas pencarian
3. Pengguna memilih file	
	4. Sistem menampilkan nama dataset
5. Pengguna mengkonfirmasi unggahan dengan menekan tombol “Unggah”	
Kondisi Akhir	Sistem membaca dan menampilkan data.
Skenario Alternatif	
Aktor	Sistem
1. Pengguna mengklik tombol “Pilih File”	
	2. Menampilkan jendela berkas pencarian
3. Pengguna memilih file	
	4. Sistem menampilkan nama dataset
5. Pengguna mengkonfirmasi unggahan dengan menekan tombol “Unggah”	
	6. Sistem menampilkan pesan error format file csv tidak valid
Kondisi Akhir	Sistem tidak dapat menampilkan data

Tabel IV- 22. Skenario *Use Case Preprocessing*

Identifikasi	
Nomor	TS-002
Nama	<i>Preprocessing</i>
Tujuan	Mempersiapkan awal data sebelum analisis sentimen
Deskripsi	Mengolah data masukan dengan langka <i>preprocessing</i> .
Aktor	Pengguna
Kondisi Awal	Data telah diunggah ke sistem
Skenario Utama	
Aktor	Sistem
1. Pengguna menekan tombol “Proses dan Evaluasi Data”	
	2. Sistem melakukan <i>preprocessing</i> data
	3. Menampilkan data hasil <i>preprocessing</i>
Kondisi Akhir	Sistem berhasil melakukan pra-pemrosesan data

Tabel IV- 23. Skenario *Use Case Evaluasi*

Identifikasi	
Nomor	TS-003
Nama	Melakukan pengujian model LSTM yang sudah dilatih
Tujuan	Menggunakan model LSTM untuk klasifikasi sentimen dan identifikasi kategori aspek pada data yang telah dilakukan pra-pemrosesan
Deskripsi	Pada tahap ini, sistem akan melakukan proses pengujian data menggunakan model LSTM dengan data yang telah dibersihkan.
Aktor	Pengguna
Kondisi Awal	Pengguna menekan tombol Proses dan Evaluasi Data
Skenario Utama	
Aktor	Sistem

1. Pengguna mengklik tombol “Proses dan Evaluasi Data”	
	2. Melakukan proses pengujian model LSTM menggunakan data yang telah diproses.
	3. Sistem menampilkan hasil evaluasi berbentuk <i>confusion matrix</i> dan <i>classification report</i>
Kondisi Akhir	Proses pengujian LSTM selesai

Tabel IV- 24. Skenario *Use Case* Prediksi Data

Identifikasi	
Nomor	TS-004
Nama	Melakukan Prediksi
Tujuan	Memprediksi data yang diunggah oleh pengguna
Deskripsi	Model LSTM akan melakukan prediksi terhadap data prediksi yang diunggah oleh pengguna.
Aktor	Pengguna
Kondisi Awal	Sistem belum menerima dataset
Skenario Utama	
Aktor	Sistem
1. Pengguna membuka halaman “Prediksi”	
	2. Sistem menampilkan halaman prediksi
3. Pengguna mengklik tombol “Prediksi”	
	4. Sistem menampilkan jendela berkas pencarian
5. Pengguna memilih dataset yang akan diunggah	
	6. Sistem menampilkan nama dataset
7. Pengguna mengklik tombol “Evaluasi Prediksi”	
	8. Sistem menampilkan hasil prediksi sentimen dan aspek berupa tabel

	9. Sistem menampilkan hasil evaluasi berbentuk <i>confusion matrix</i> dan <i>classification report</i>
Kondisi Akhir	Sistem berhasil melakukan prediksi dataset

Tabel IV- 25. Skenario *Use Case* Prediksi Kalimat

Identifikasi	
Nomor	TS-005
Nama	Melakukan Prediksi Kalimat
Tujuan	Memprediksi kalimat dengan masukan dari pengguna
Deskripsi	Pada tahap ini model LSTM yang sudah tersimpan di dalam sistem akan melakukan prediksi
Aktor	Pengguna
Kondisi Awal	Teks yang dimasukkan oleh pengguna berbentuk ulasan yang berkaitan dengan kata kunci “Gojek”
Skenario Utama	
Aktor	Sistem
1. Pengguna memasukan teks dan mengklik tombol “Prediksi Kalimat”	
	2. Sistem menampilkan hasil prediksi sentimen dan aspek
Kondisi Akhir	Proses prediksi selesai.

Proses *use case* pelatihan terdapat skenario melakukan pemodelan topik dan melatih model LSTM. Skenario ini menggambarkan alur proses yang terjadi dalam melakukan pelatihan pada dataset berbahasa Indonesia yang telah dilakukan pra-pemrosesan.

Tabel IV- 26. Skenario *Use Case* Pemodelan Topik

Identifikasi	
Nomor	TR-001
Nama	Melakukan Pemodelan Topik
Tujuan	Menghasilkan model LDA untuk pengelompokkan topik dalam dataset
Deskripsi	Pada tahap ini dilakukan pemodelan topik LDA menggunakan dataset yang telah dilakukan pada tahap pra-pemrosesan
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan halaman <i>notebook</i> pelatihan
Skenario Utama	
Aktor	Sistem
1. Menyediakan data yang telah dibersihkan pada tahap pra-pemrosesan	
2. Menjalankan program	
	3. Memuat dataset
	4. Membuat dictionary
	5. Sistem mengubah setiap dokumen menjadi representasi BoW dengan format corpus
	6. Menyimpan dengan format dictionary.dict dan corpus.pkl
	7. Sistem membuat model LDA berdasarkan nilai <i>cohorence</i> tertinggi
	8. Sistem menyimpan model LDA dengan format lda_model.model
	9. Sistem memvisualisasikan model menggunakan pyLDAvis
	10. Visualisasi disimpan dalam format lda_visualization.html
Kondisi Akhir	Proses pemodelan topik selesai.

Tabel IV- 27 Skenario Use Case Pelatihan Model LSTM

Identifikasi	
Nomor	TR-002
Nama	Melakukan pelatihan model LSTM
Tujuan	Sistem menghasilkan model LSTM yang telah dilatih untuk proses pengujian
Deskripsi	Pada tahap ini dilakukan dengan pelatihan model LSTM menggunakan dataset yang telah dibersihkan pada pra-pemrosesan
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan halaman <i>notebook</i> pelatihan model LSTM
Skenario Utama	
Aktor	Sistem
1. Menyediakan data yang telah dibersihkan pada tahap pra-pemrosesan	
2. Menjalankan program	
	3. Memuat dataset yang telah diproses dan menampilkan beberapa baris untuk memverifikasi data yang dimuat.
	4. Mengubah teks menjadi token menggunakan metode tokenisasi menggunakan <i>word2vec</i>
	5. Memuat model <i>Word2Vec</i> yang telah dilatih sebelumnya untuk mendapatkan representasi vektor kata.
	6. Membagi data menjadi data latih dan data uji
7. Menyusun model LSTM dengan menggunakan nilai <i>hyperparameter</i> yang telah ditentukan	
	8. Melatih model LSTM dengan data latih
	9. Model disimpan ke dalam bentuk <i>file</i> dengan format <i>.h5</i>

Kondisi Akhir	Sistem menghasilkan model yang telah dilatih dengan dataset yang telah disiapkan
----------------------	--

4.3 Fase Elaborasi

Dalam tahap ini, bisnis direpresentasikan dalam model, data direncanakan, antarmuka dirancang, dan *activity diagram* serta *sequence diagram* digunakan.

4.3.1 Pemodelan Bisnis

Fokus utama dalam pemodelan bisnis ini yaitu merancang perangkat lunak berdasarkan hasil evaluasi tahap awal. Pengembangan perangkat lunak tersebut mencakup pengaturan data dan desain antarmuka sebagai bagian dari proses perancangan yang telah dibahas.

4.3.2 Perancangan Data

Perangkat lunak yang dikembangkan akan dirancang untuk menganalisis sentimen dari ulasan pengguna Gojek yang diambil melalui *Play Store* dengan memanfaatkan data masukan dari pengguna. Data yang digunakan berupa ulasan dalam bahasa Indonesia yang disimpan dalam format file .CSV.

4.3.3 Perancangan Antar Muka

Gambar IV-4, IV-5, IV-6 menampilkan desain perangkat lunak yang dirancang dengan teliti untuk memandu pembuatan antarmuka yang efektif, mudah dipahami, dan digunakan. Desain ini diperoleh melalui analisis kebutuhan dan preferensi pengguna agar menciptakan pengalaman yang intuitif dan memuaskan. Selain itu, desain ini bertujuan mendukung produktivitas dengan mempermudah navigasi, mempercepat akses fitur utama, dan meningkatkan interaksi pengguna dengan perangkat lunak, menciptakan lingkungan yang efisien dan menyenangkan.


SISTEM ANALISIS SENTIMEN GOJEK

Preprocessing & Evaluasi Prediksi

Upload CSV file

Drag and drop file here

Limit 200MB per file CSV



Dataset Preview

Proses & Evaluasi Data

Gambar IV- 5. Perancangan Antar Muka Sistem Proses dan Evaluasi Data

SISTEM ANALISIS SENTIMEN GOJEK


Preprocessing & Evaluasi Prediksi

Prediksi File Prediksi Kalimat

Upload File

Drag and drop file here

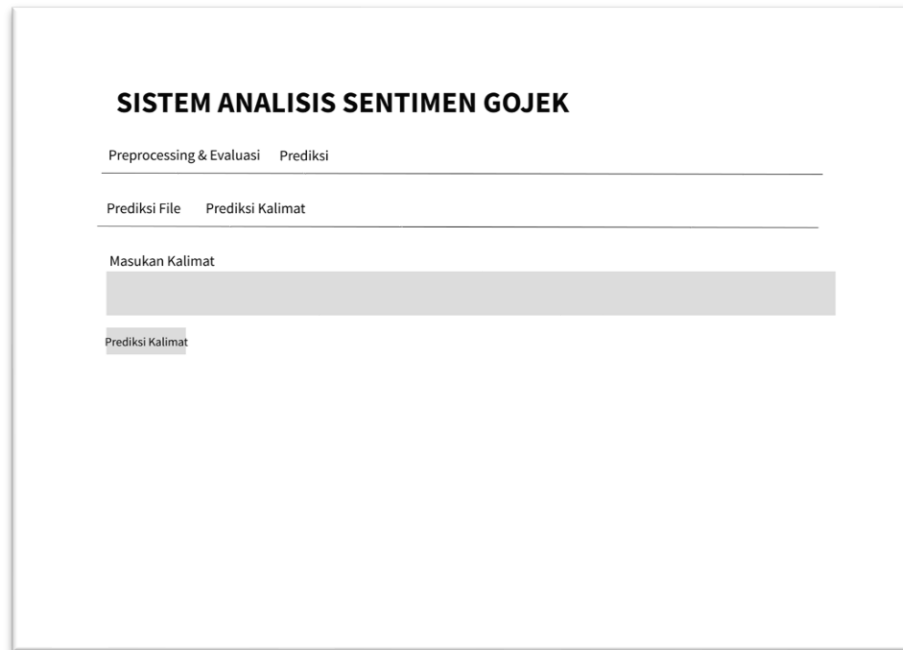
Limit 200MB per file CSV



Dataset Preview

Evaluasi Prediksi

Gambar IV- 6. Perancangan Antar Muka Sistem Prediksi Data



Gambar IV- 7. Perancangan Antar Muka Sistem Prediksi Kalimat

4.3.4 Kebutuhan Sistem

Kebutuhan sistem perangkat lunak yang dibangun memberikan perangkat keras atau *hardware*, perangkat lunak atau *software*, dan bahasa pemrograman. Bahasa pemrograman yang digunakan adalah bahasa Python. Dalam pengembangan sistem perangkat lunak, diperlukan komponen perangkat keras, perangkat lunak, dan pemilihan bahasa pemrograman yang sesuai. Berikut adalah rincian perangkat lunak yang digunakan dalam studi ini:

1. Perangkat Keras

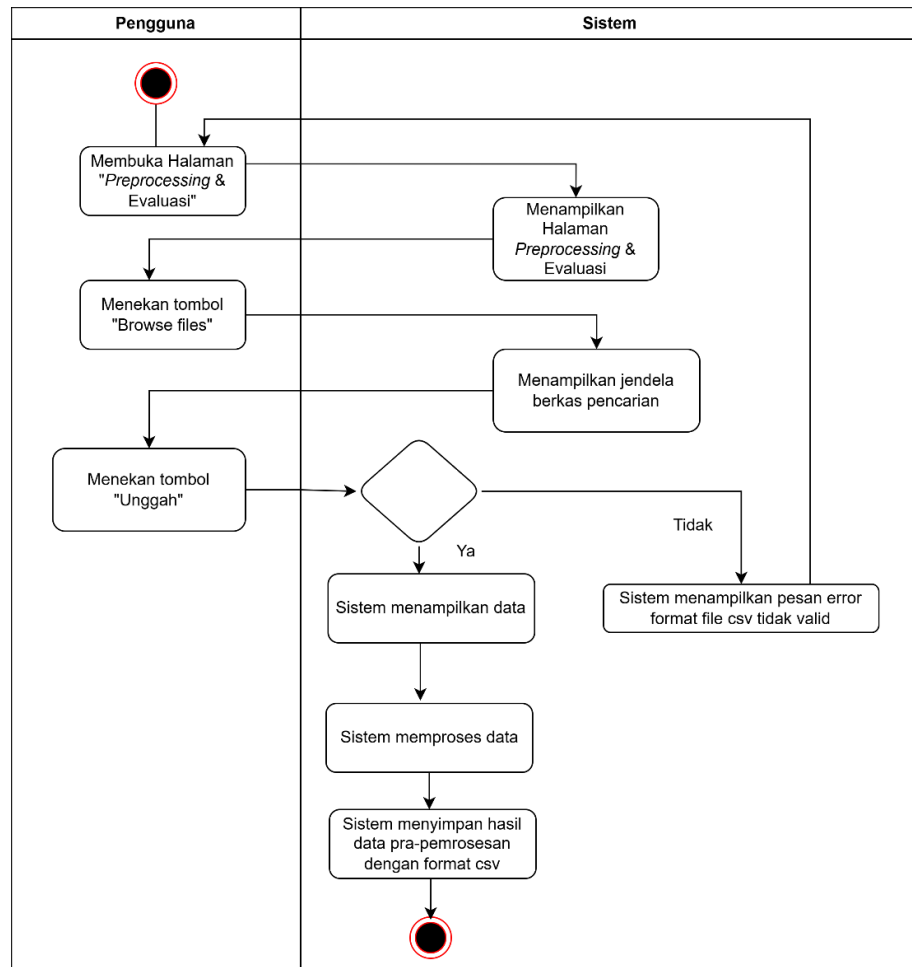
Processor	:	Intel Core i3
RAM	:	4GB
HDD	:	475GB

2. Perangkat Lunak

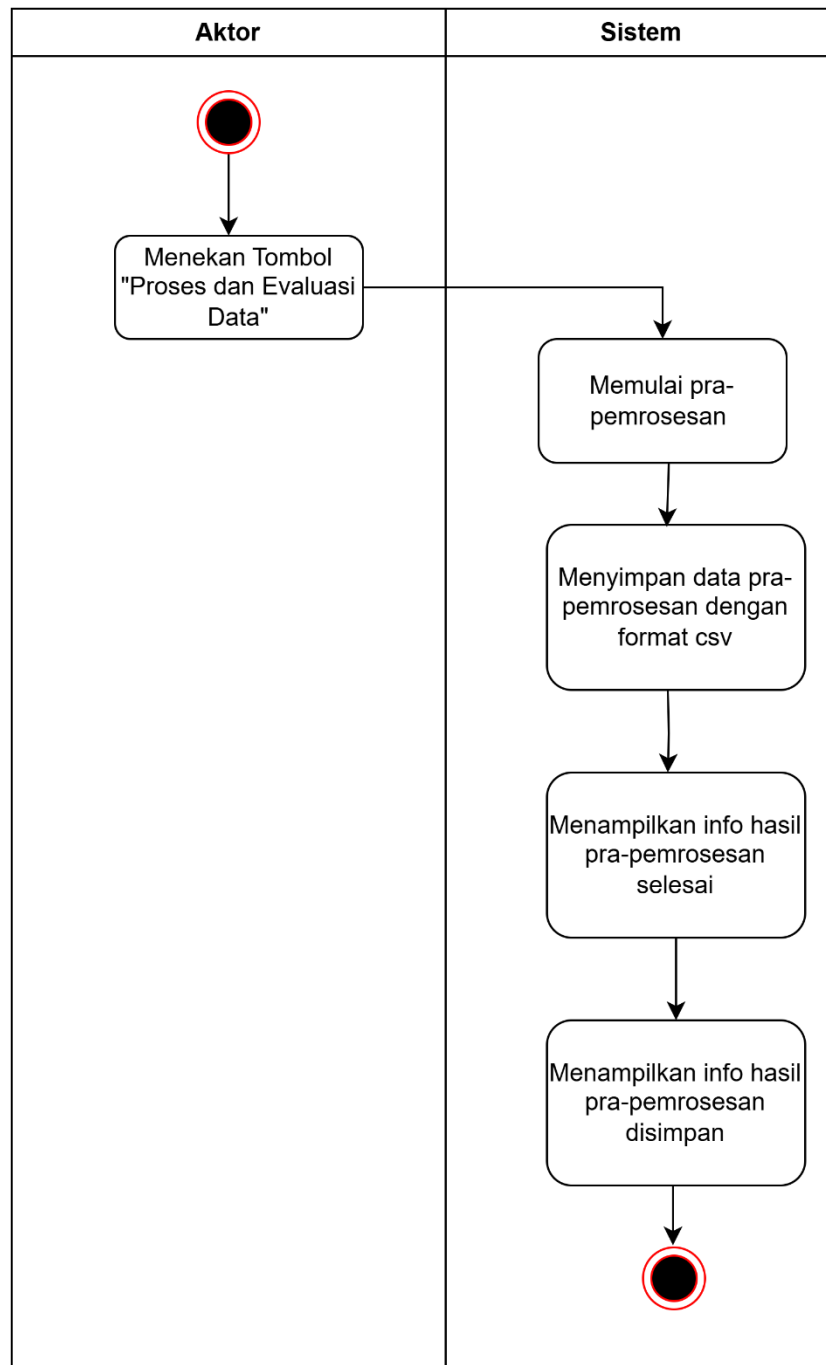
Sistem Operasi	: Windows 11 64-Bit
Bahasa Pemrograman	: <i>Python 3.11.5</i>
IDE	: Visual Studio Code
Library	: <i>Streamlit, Keras, Matplotlib,</i> <i>NLTK, Numpy, Pandas,</i> <i>Sastrawi, Sklearn, Tensorflow,</i> <i>Gensim, pyLDAvis</i>

4.3.5 Activity Diagram

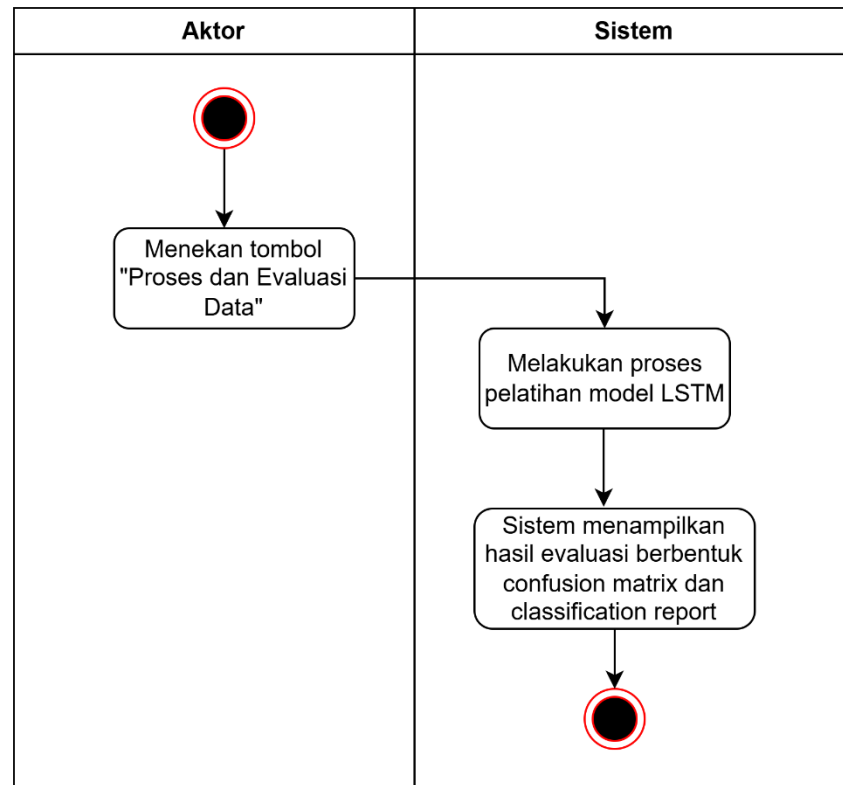
Diagram aktivitas menggambarkan urutan dan aliran proses dalam perangkat lunak, membantu pengembang memvisualisasikan interaksi antar komponen sistem. Diagram ini memetakan setiap langkah, termasuk kondisi dan keputusan yang mempengaruhi alur, untuk memudahkan perancangan sistem dan identifikasi kendala. Dalam *use case*, diagram aktivitas menunjukkan proses mulai dari interaksi pengguna hingga proses internal, dengan rincian tentang fungsi yang saling bergantung. Diagram-diagram tersebut terdapat pada Gambar IV-7, IV-8, IV-9, IV-10, dan IV-11 untuk menggambarkan organisasi dan hubungan proses.



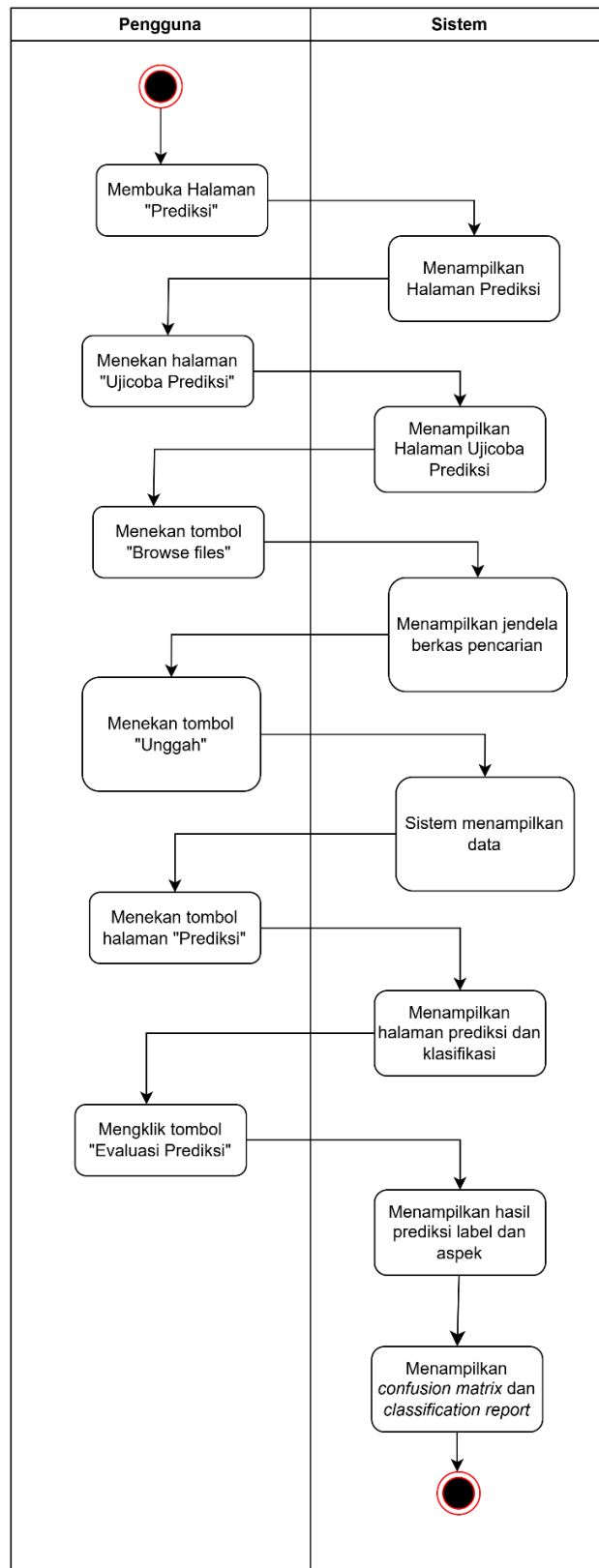
Gambar IV- 8. *Activity Diagram Upload Data*



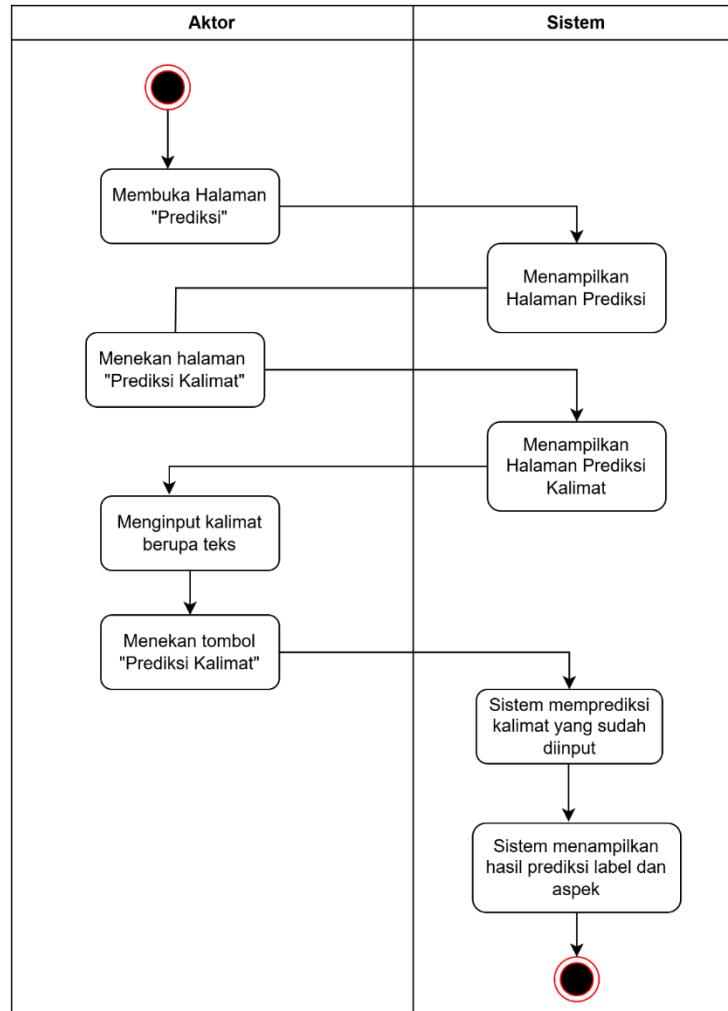
Gambar IV-9. *Activity Diagram Preprocessing*



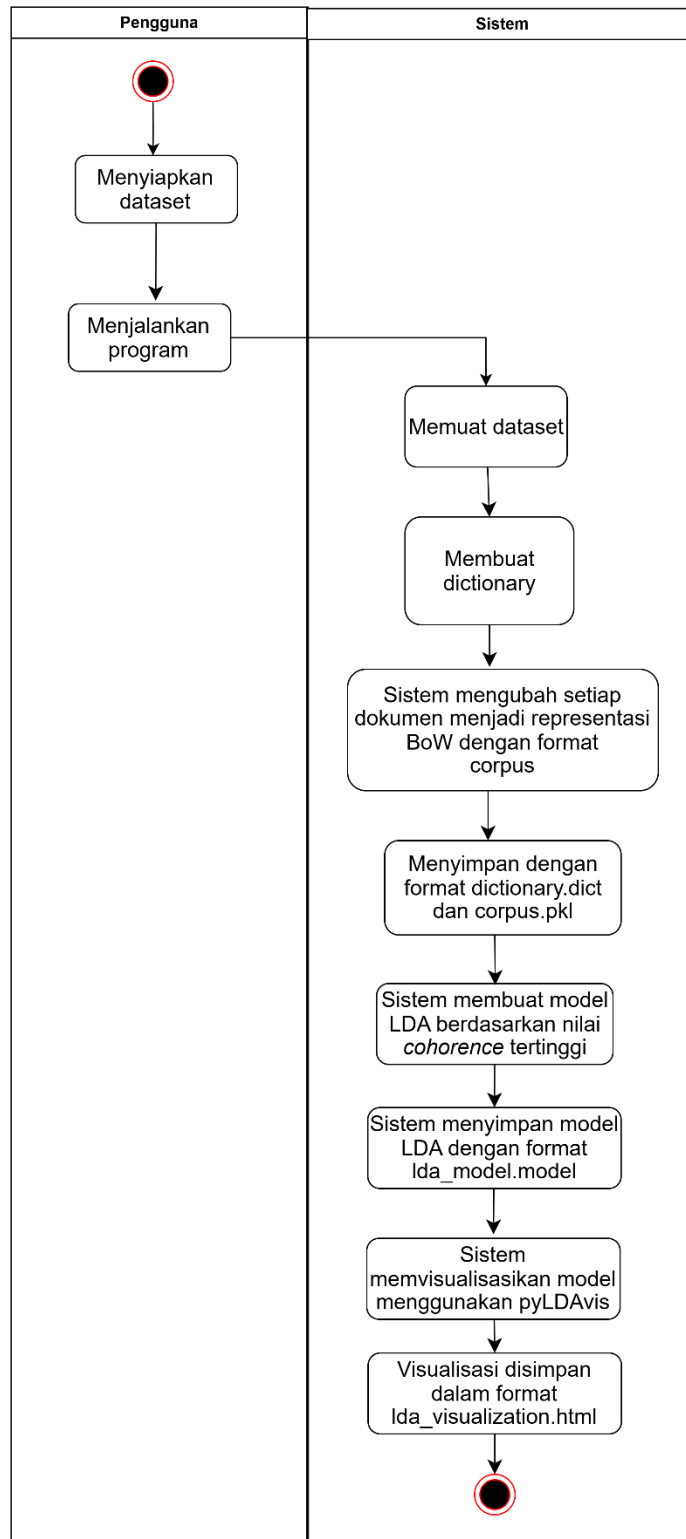
Gambar IV- 10. *Activity Diagram Evaluasi*



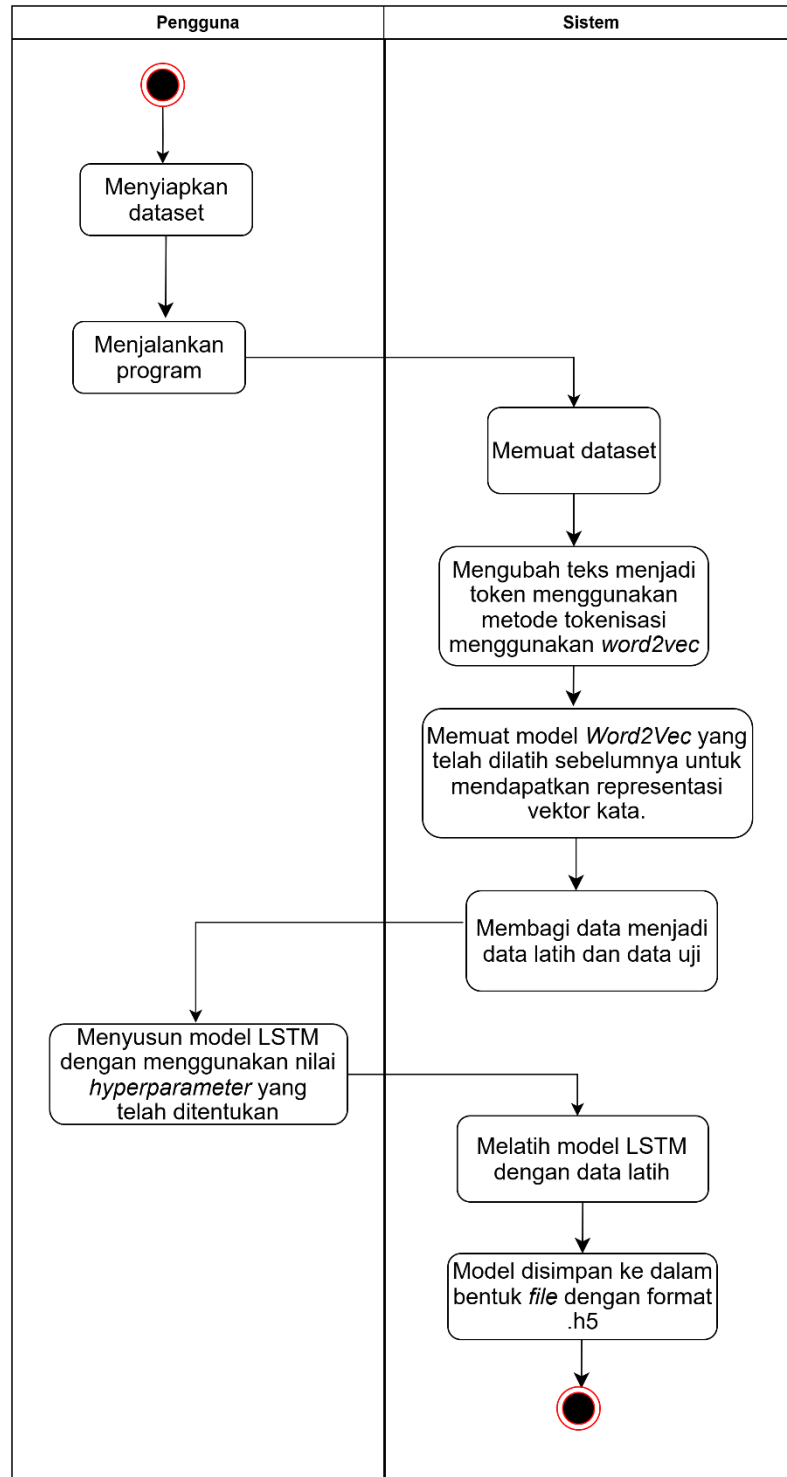
Gambar IV- 11. *Activity Diagram* Prediksi Data



Gambar IV- 12. *Activity Diagram* Prediksi Kalimat



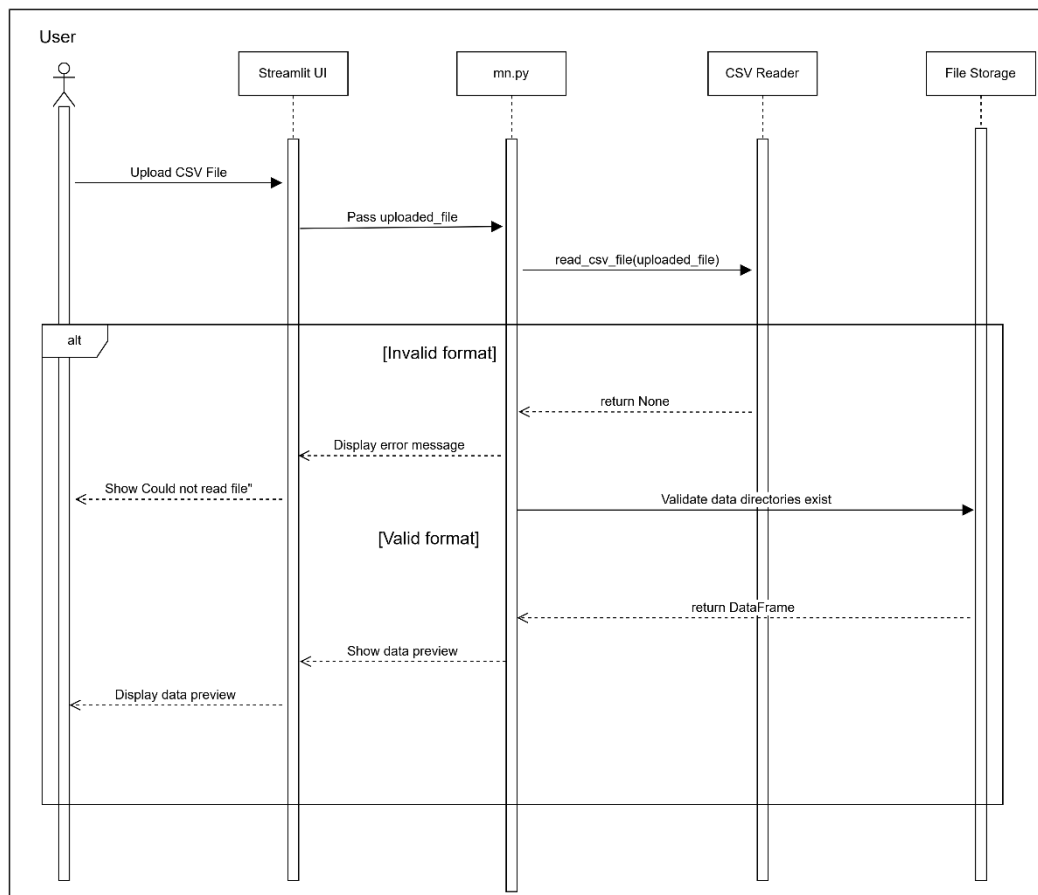
Gambar IV- 13. Activity Diagram Pemodelan Topik



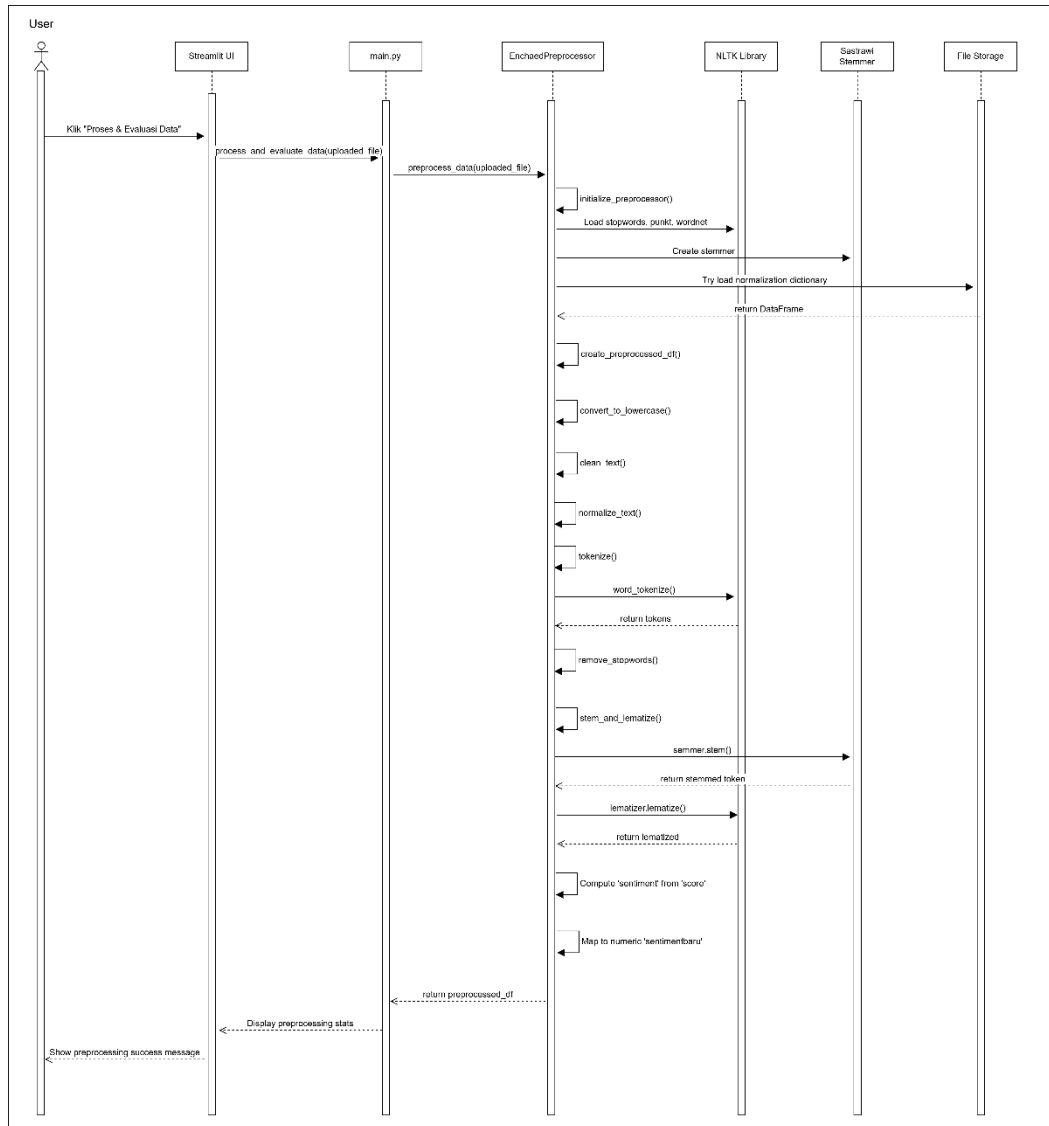
Gambar IV- 14. Activity Diagram Pelatihan Model LSTM

4.3.6 Sequence Diagram

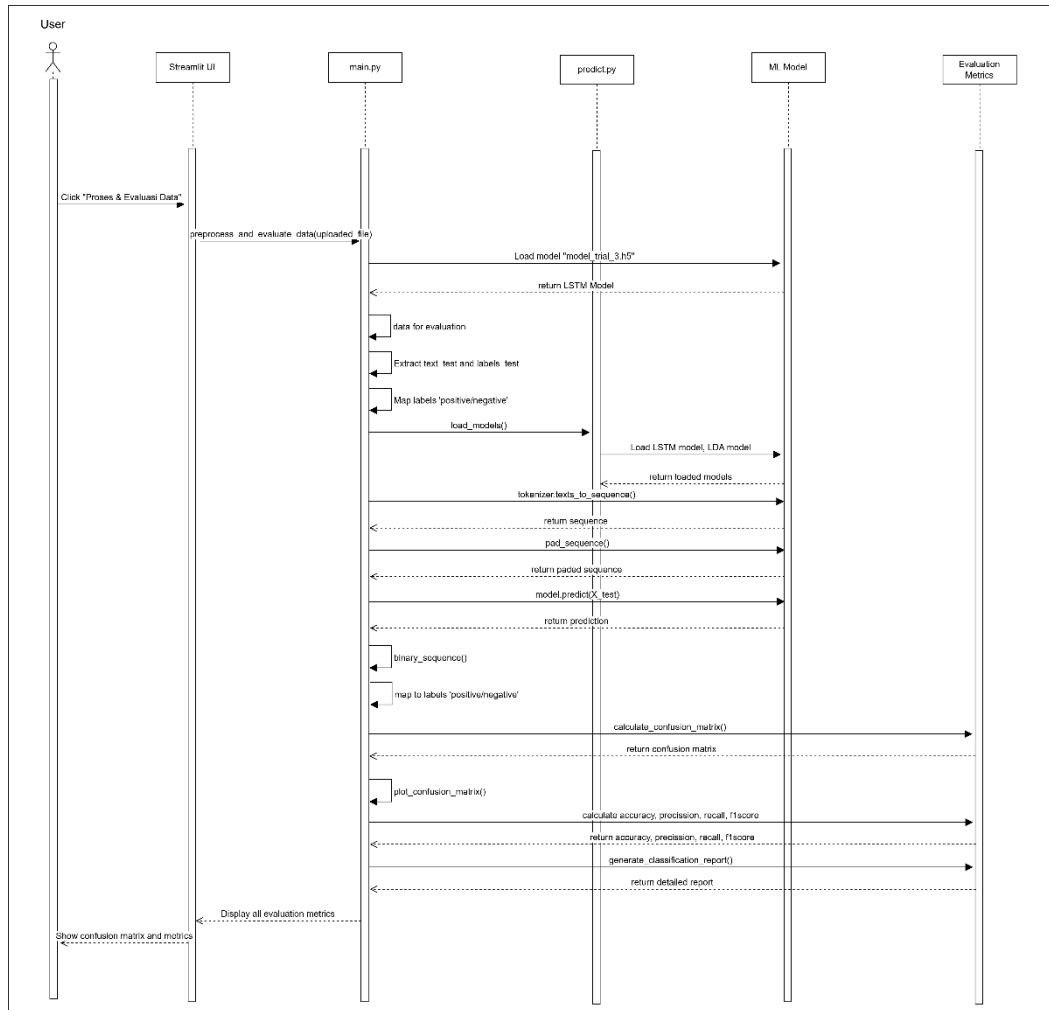
Dalam perangkat lunak ini, terdapat empat diagram urutan yang menggambarkan interaksi dan hubungan antar objek dalam berbagai skenario penggunaan. Diagram urutan digunakan untuk memodelkan bagaimana komponen sistem berkomunikasi dan berinteraksi sesuai urutan waktu. Dengan tiga diagram ini, pemahaman tentang alur dan komunikasi antar objek menjadi lebih jelas dan terstruktur. Diagram sequence dapat dilihat pada Gambar IV-12, IV-13, IV-14, IV-15, dan IV-16.



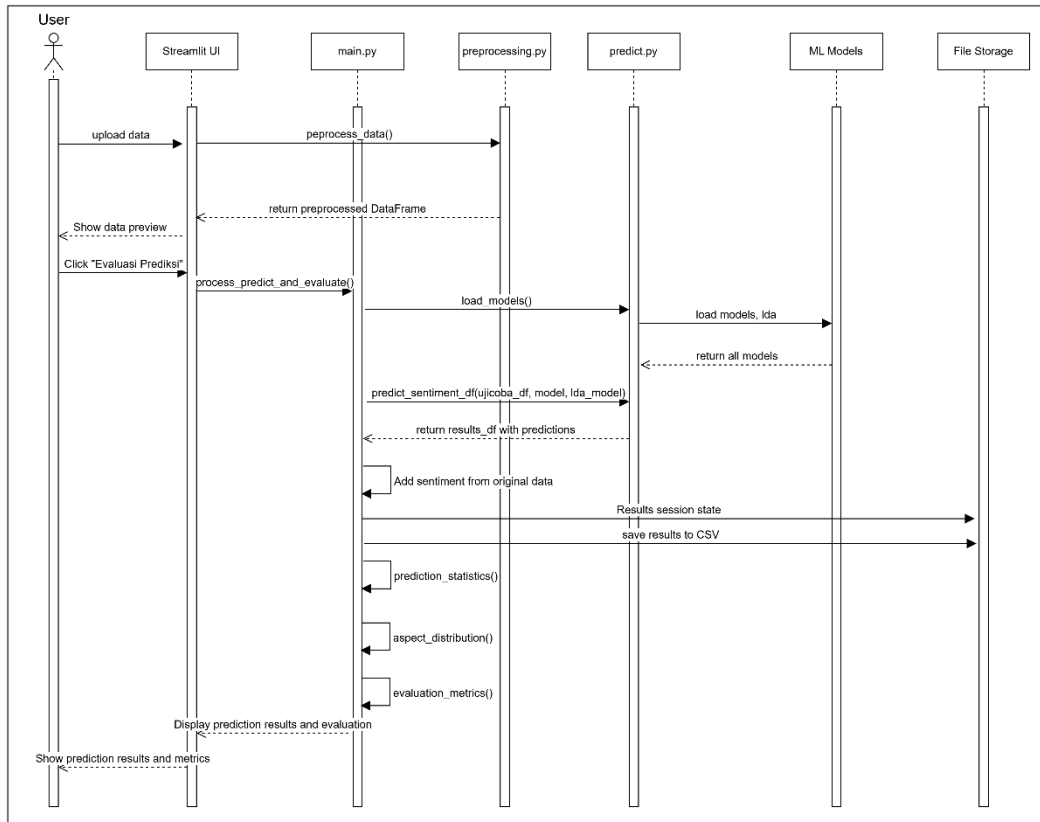
Gambar IV- 15. Diagram *Sequence Upload Data*



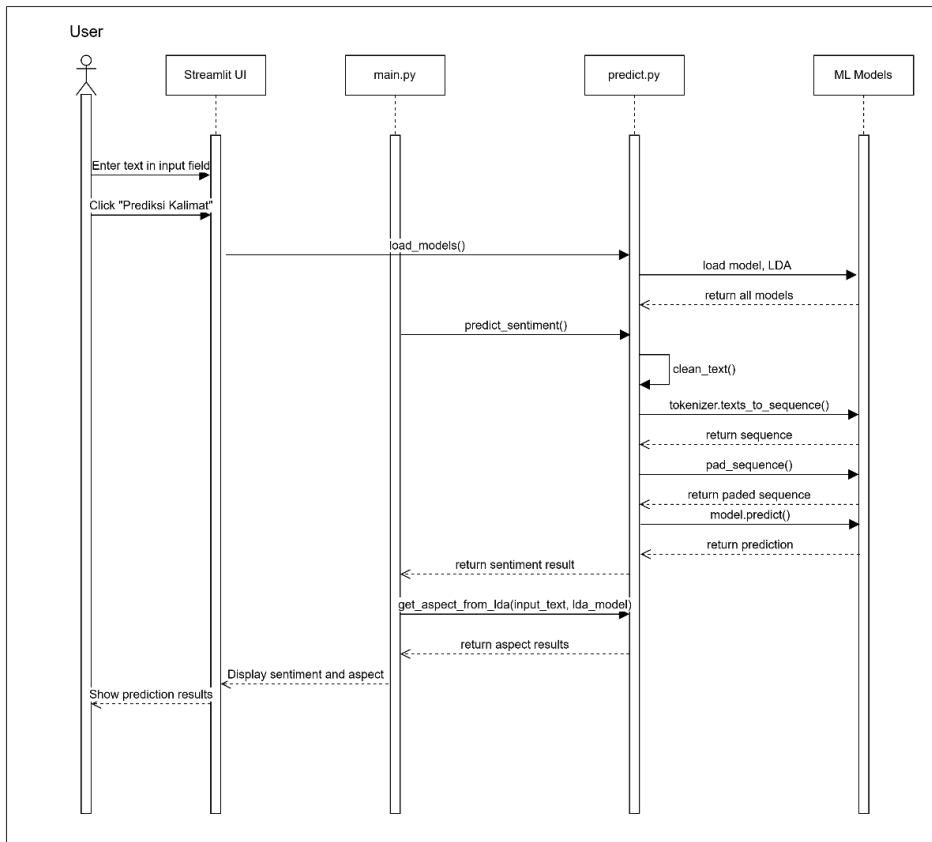
Gambar IV- 16. Diagram Sequence Preprocessing



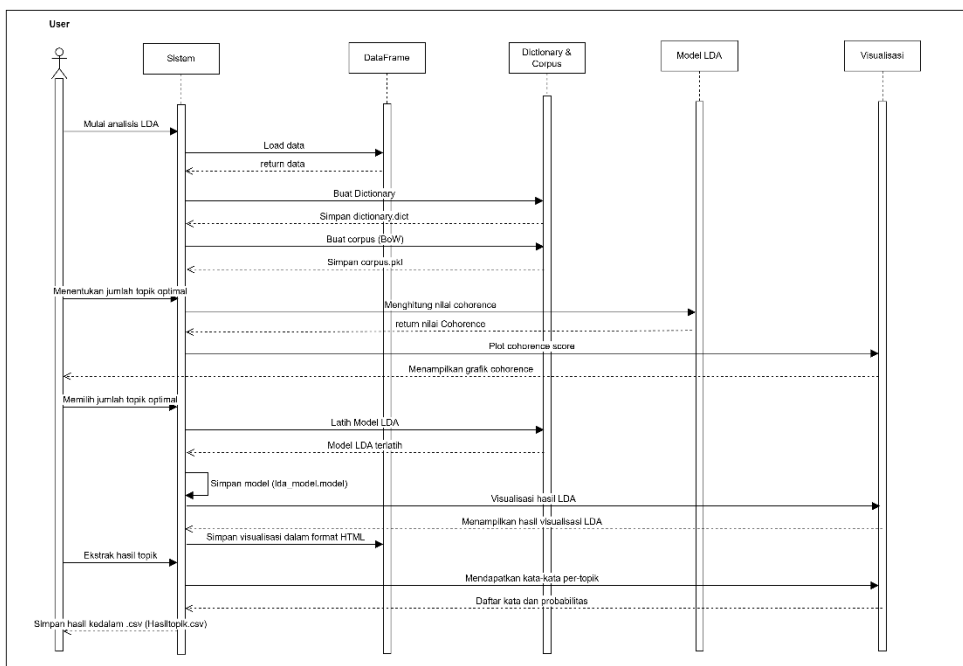
Gambar IV- 17. Diagram Sequence Evaluasi



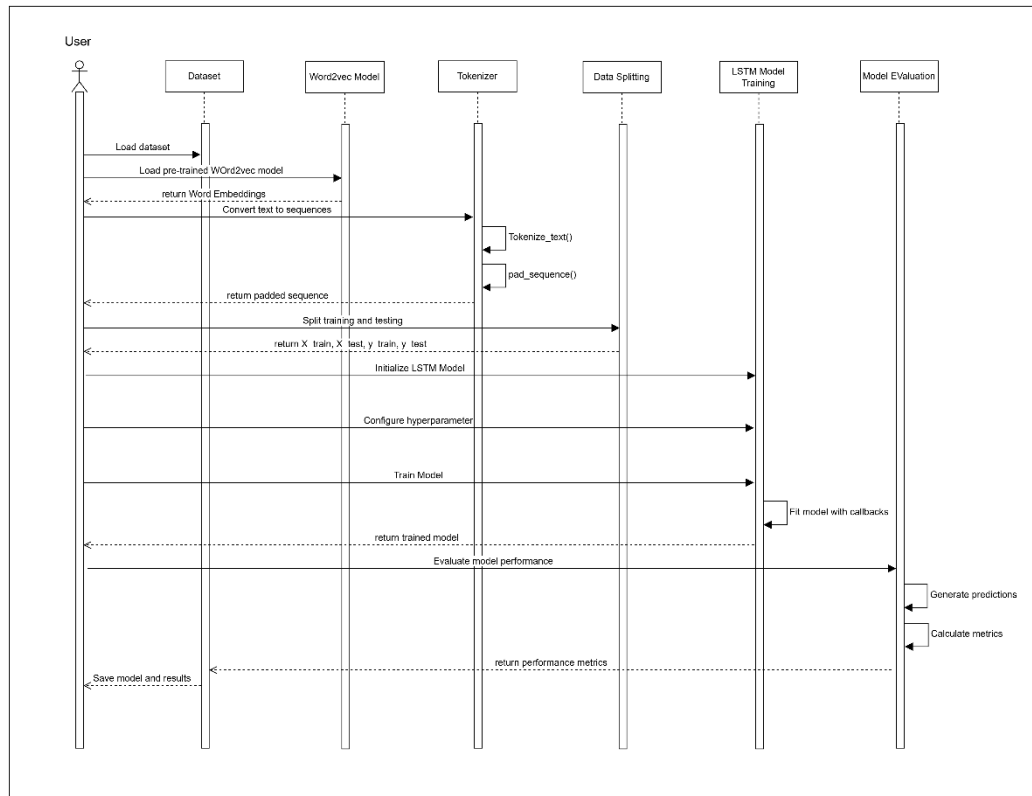
Gambar IV- 18. Diagram *Sequence* Prediksi Data



Gambar IV- 19. Diagram *Sequence* Prediksi Kalimat



Gambar IV- 20. Diagram *Sequence* Pemodelan Topik



Gambar IV- 21. Diagram *Sequence* Pelatihan Model LSTM

4.4 Fase Konstruksi

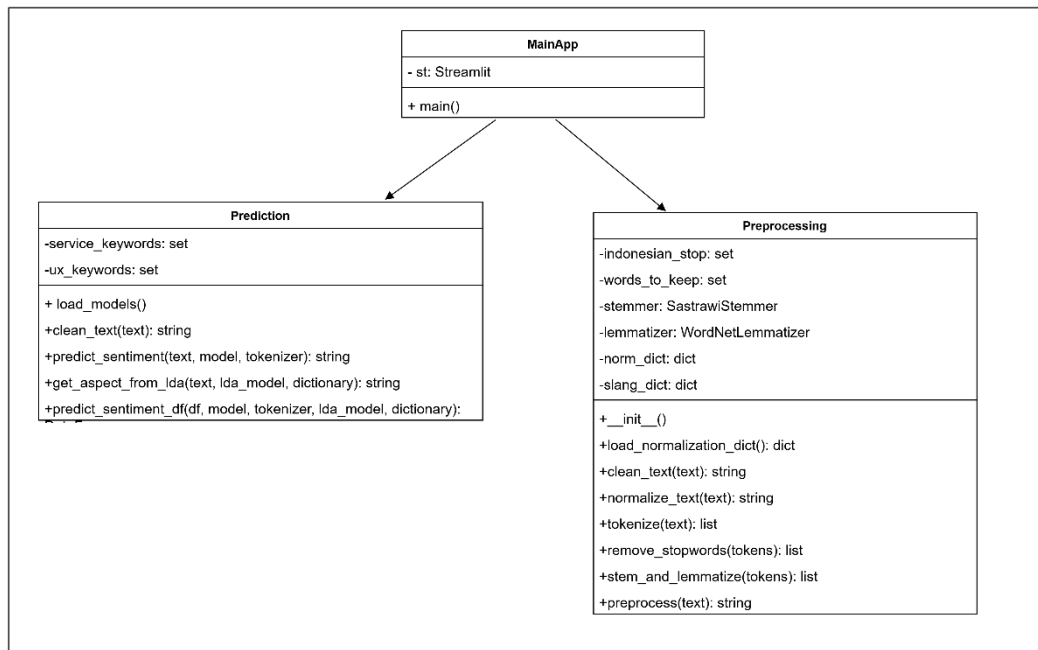
Fase konstruksi dalam rekayasa perangkat lunak adalah tahap pengembangan di mana tim mulai mengimplementasikan perangkat lunak sesuai desain perencanaan. Analisis kebutuhan sistem digunakan untuk menyusun diagram kelas yang kemudian diimplementasikan dalam perangkat lunak.

4.4.1 Kebutuhan Sistem

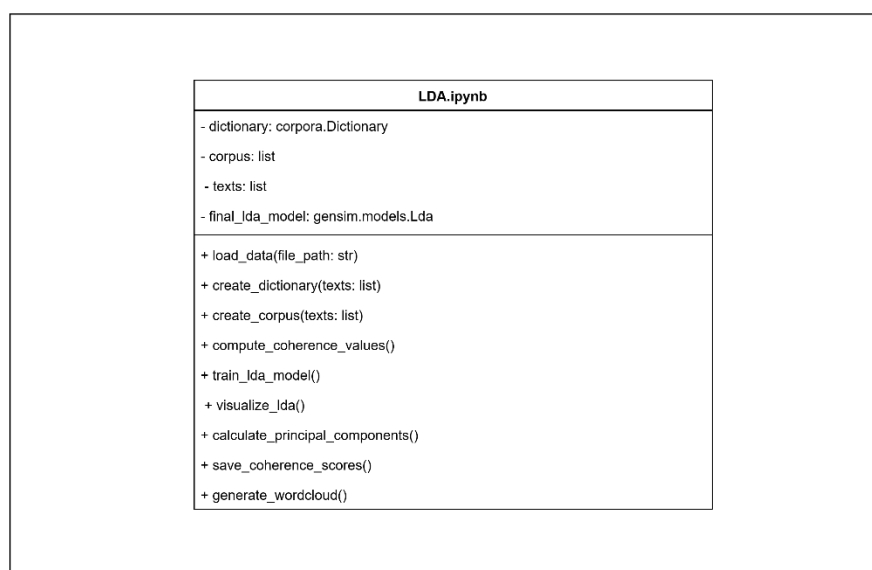
Dalam proses pengembangannya, perangkat lunak yang dikonstruksi mengandalkan beberapa *library* Python sebagai alat bantu.

4.4.2 Diagram Kelas

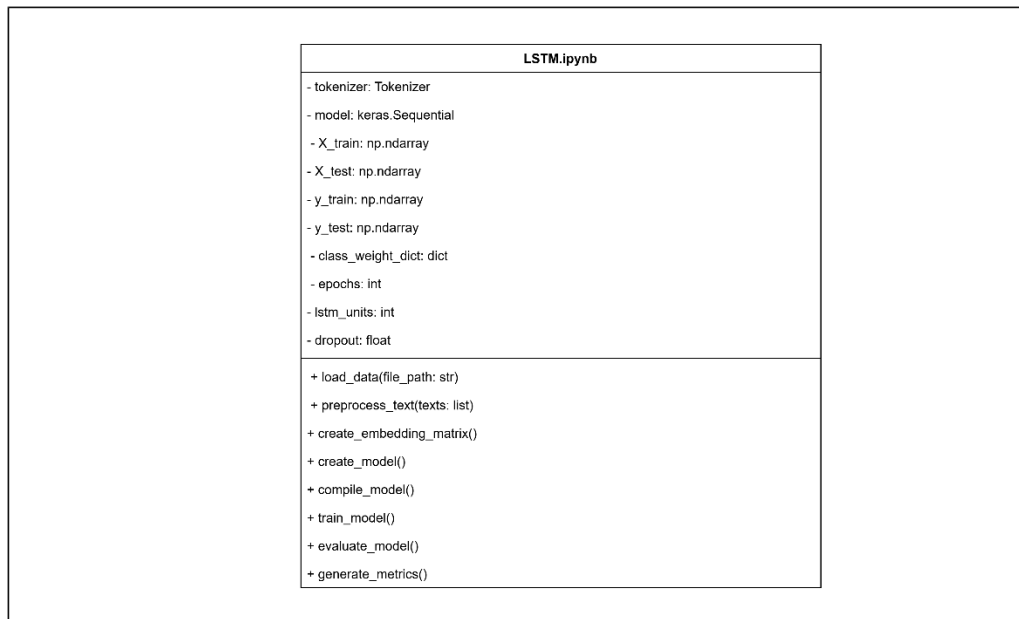
Diagram kelas dengan jelas mengilustrasikan relasi antara kelas-kelas, atribut, serta metode yang terdapat dalam perangkat lunak. Anda dapat merujuk ke Gambar IV-22, IV-23, dan IV-24 untuk melihat diagram kelas tersebut.



Gambar IV- 22. *Diagram Class Pengujian*



Gambar IV- 23. *Diagram Class Pelatihan Pemodelan Topik*



Gambar IV- 24. *Diagram Class* Pelatihan Model LSTM

4.4.3 Implementasi

Pada tahap ini, proses implementasi perangkat lunak akan dilaksanakan secara menyeluruh sesuai dengan rancangan yang telah disusun sebelumnya serta diagram kelas yang telah dibuat.

4.4.3.1 Implementasi Kelas

Penjabaran mengenai kelas-kelas yang diterapkan dalam perangkat lunak ini, yang dikembangkan menggunakan bahasa pemrograman Python, dapat dilihat secara rinci pada Tabel IV-28 dan IV-29.

Tabel IV- 28. Implementasi Kelas Pengujian

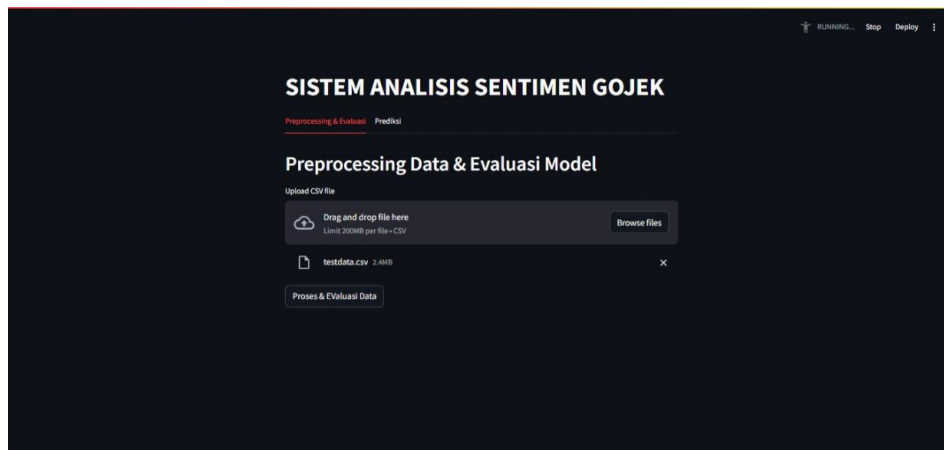
No	Nama Kelas	Nama File	Deskripsi
1.	GUI	main.py	Kelas GUI memiliki peran utama dalam menampilkan antarmuka pengguna. Kelas ini bertanggung jawab untuk menyajikan tampilan visual yang interaktif, memungkinkan pengguna untuk mengunggah file, memproses data, dan mendapatkan hasil analisis sentimen serta evaluasi model.
2.	Pra-pemrosesan Data	Preprocessing.py	Kelas ini berperan dalam menyiapkan dan membersihkan data agar dapat digunakan pada tahap selanjutnya. Proses ini mencakup berbagai teknik pemrosesan teks yang bertujuan untuk meningkatkan kualitas data sebelum dianalisis lebih lanjut. Dalam tahap ini, data mentah yang berisi teks ulasan akan diproses agar lebih terstruktur dan siap digunakan dalam model analisis sentimen.
3.	Prediksi	predict.py	Kelas prediksi bertanggung jawab untuk melakukan prediksi sentimen dan aspek berdasarkan model yang telah dilatih.

Tabel IV- 29. Implementasi Kelas Pelatihan

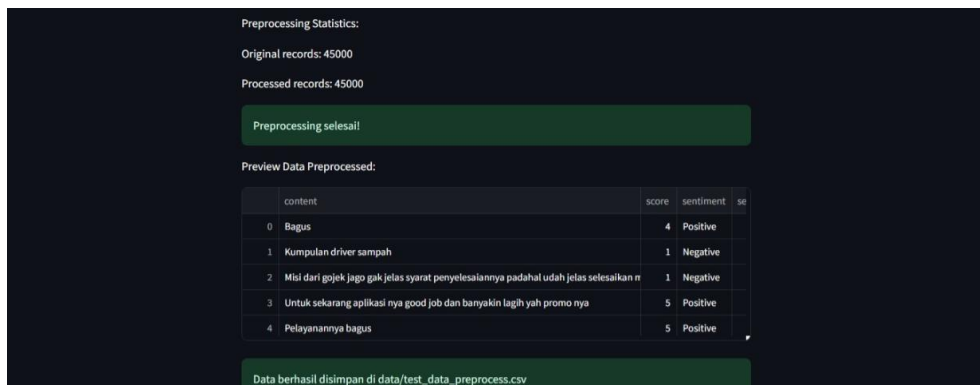
No	Nama Kelas	Nama File	Deskripsi
1.	Pemodelan Topik	lda.ipynb	Kelas ini memiliki peran untuk melakukan pemodelan topik menggunakan LDA.
2.	Pelatihan Model LSTM	LSTM.ipynb	Kelas ini berperan untuk mengatur seluruh proses <i>workflow</i> , dimulai dengan pemuatan dataset sampai pelatihan model LSTM.

4.4.3.2 Implementasi *Interface*

Antarmuka yang telah dirancang sebelumnya diterapkan dalam perangkat lunak ini. Tampilan antarmuka tersebut dapat dilihat pada Gambar IV-25, IV-26, IV-27, IV-28, IV-29, IV-30, IV-31, dan IV-32.

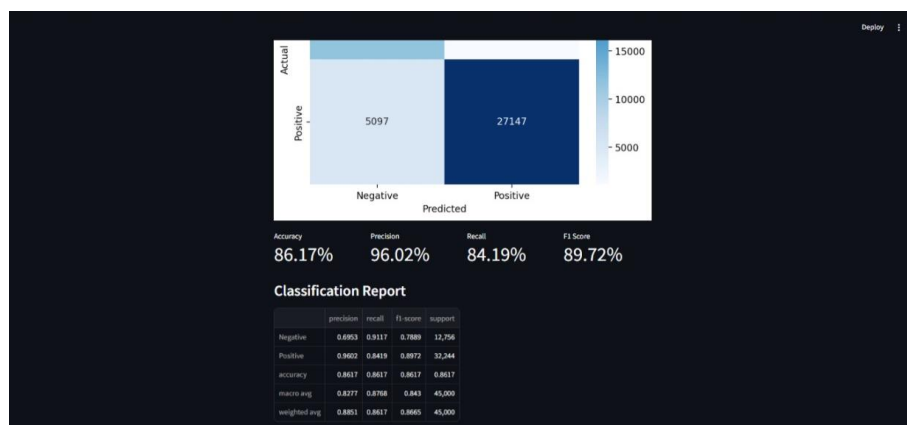


Gambar IV- 25. Implementasi *Interface Preprocessing* dan Evaluasi Data



Gambar IV- 26. Implementasi *Interface Hasil Preprocessing* dan Evaluasi

Data

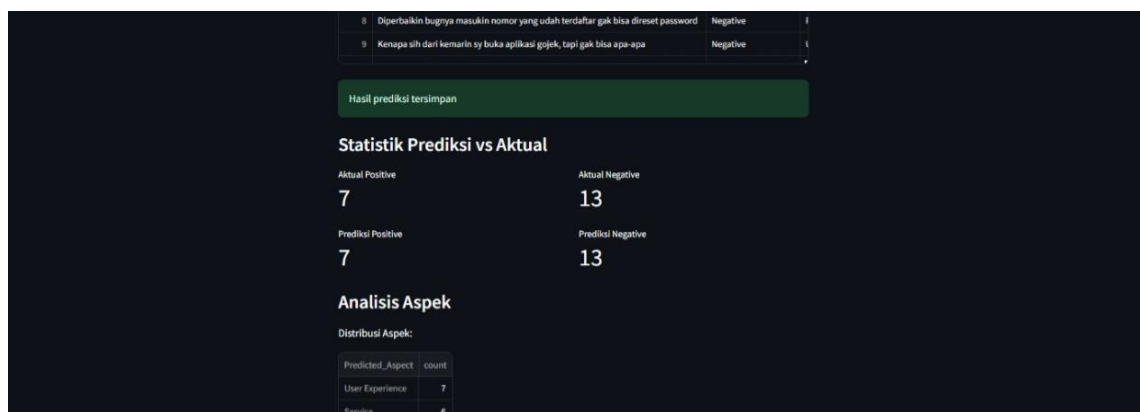


Gambar IV- 27. Implementasi *Interface Hasil Preprocessing* dan Evaluasi

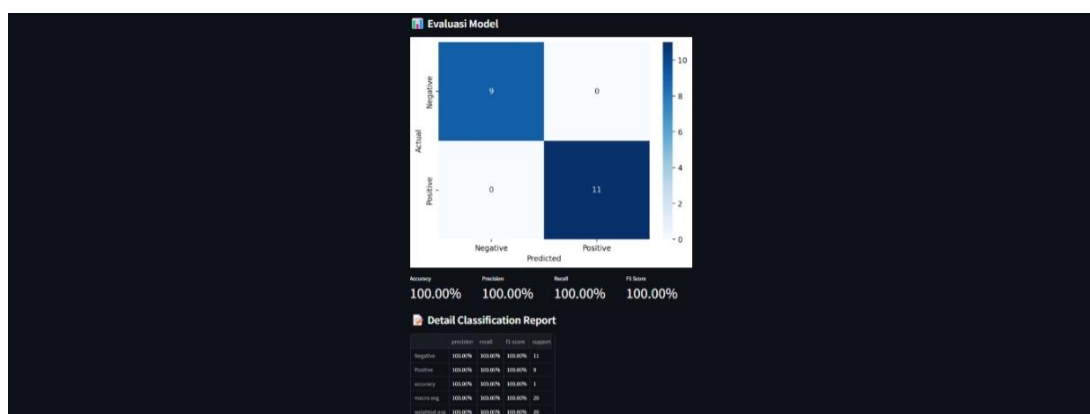
Data II



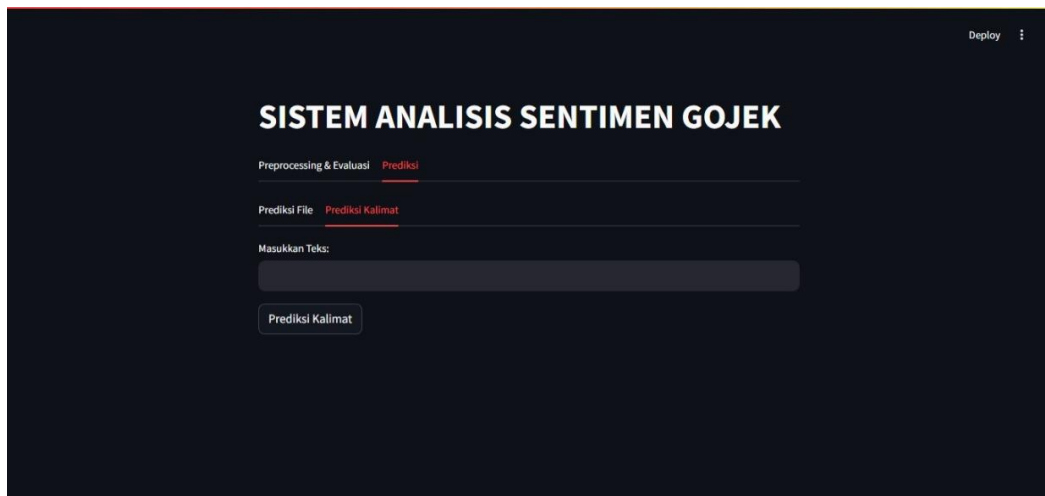
Gambar IV- 28. Implementasi *Interface* Prediksi Data



Gambar IV- 29. Implementasi *Interface* Hasil Prediksi Data



Gambar IV- 30. Implementasi *Interface* Hasil Evaluasi Prediksi Data



Gambar IV- 31. Implementasi *Interface* Prediksi Kalimat



Gambar IV- 32. Implementasi *Interface* Hasil Prediksi Kalimat

4.5 Fase Transisi

Fase transisi merupakan tahap akhir dalam proses pengembangan perangkat lunak di mana dilakukan pengujian terhadap perangkat tersebut. Pada tahap ini, pengujian dilaksanakan untuk setiap kasus uji yang telah disusun sebelumnya. Selain itu, hasil dari pengujian ini digunakan untuk memastikan bahwa perangkat lunak beroperasi dengan baik dan memenuhi spesifikasi yang telah ditetapkan sebelumnya.

4.5.1 Pemodelan Bisnis

Pengujian perangkat lunak dilakukan dengan menggunakan metode kotak hitam pada setiap use case yang telah dirancang. Metode ini menguji perangkat lunak tanpa mempertimbangkan struktur internalnya, dengan berfokus pada masukan dan keluaran untuk memastikan perangkat lunak berfungsi sesuai harapan.

4.5.2 Rencana Pengujian

Rencana pengujian bertujuan untuk menguji perangkat lunak dengan berbagai skenario yang telah ditentukan sebelumnya. Rencana ini disusun berdasarkan use case yang telah dibuat, memastikan bahwa setiap fungsi perangkat lunak diuji sesuai dengan kebutuhan dan urutan penggunaan yang diinginkan. Rencana pengujian dapat ditemukan pada Tabel IV-30, IV-31, IV-32, IV-33, IV-34, dan IV-35.

1. Rencana Pengujian Perangkat Lunak Pelatihan

Tabel IV- 30. Rencana Pengujian Pemodelan Topik

No	ID	Pengujian	Tingkat Pengujian
1.	T-001	Proses ini membuat model LDA dengan menggunakan dataset yang telah dibersihkan	Pengujian Unit

Tabel IV- 31. Rencana Pengujian Pelatihan Model LSTM

No	ID	Pengujian	Tingkat Pengujian
1.	T-002	Melakukan pelatihan model LSTM	Pengujian Unit

2. Rencana Pengujian Perangkat Lunak Pengujian

Tabel IV- 32. Rencana Pengujian *Upload* Data

No	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-001	Proses mengunggah data ke dalam sistem	Black Box	Pengujian Unit

Tabel IV- 33. Rencana Pengujian *Preprocessing*

No	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-001	Melakukan pra-pemrosesan pada data yang telah diunggah oleh pengguna	Black Box	Pengujian Unit

Tabel IV- 34. Rencana Pengujian Evaluasi

No	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-002	Melakukan <i>testing</i> dengan model LSTM	Black Box	Pengujian Unit

Tabel IV- 35. Rencana Pengujian Prediksi

No	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-003	Memprediksi teks dengan file yang diunggah oleh pengguna	Black Box	Pengujian Unit

2.	U-004	Memprediksi kalimat yang telah dimasukkan oleh pengguna.	Black Box	Pengujian Unit
----	-------	--	--------------	-------------------

4.5.3 Implementasi

Pada bagian sebelumnya, rencana pengujian telah dijelaskan secara rinci. Pada subbab ini, akan diuraikan pelaksanaan pengujian kasus secara lebih terperinci, sesuai dengan rencana pengujian yang telah disusun sebelumnya. Penjelasan ini mencakup tahapan-tahapan yang dilakukan untuk memastikan perangkat lunak dapat memenuhi setiap kriteria dan skenario yang telah ditetapkan dalam rencana pengujian.

4.5.3.1 Pengujian *Use Case* Pemodelan Topik

Tabel IV-36 berikut menampilkan hasil pengujian *use case* yang berkaitan dengan pemodelan topik.

Tabel IV- 36. Hasil Pengujian Use Case Pemodelan Topik

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
T-001	Memuat model LDA	Melakukan pemodelan topik menggunakan model LDA	Dataset	Model LDA berhasil dimuat tanpa <i>error</i>	Perangkat lunak dapat melakukan pemodelan topik	Terpenuhi

4.5.3.2 Pengujian *Use Case* Pelatihan Model LSTM

Tabel IV-37 berikut menampilkan hasil pengujian *use case* yang berkaitan dengan Pelatihan Model LSTM

Tabel IV- 37. Hasil Pengujian Use Case Model LSTM

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
T-002	Pelatihan model LSTM	Melakukan pelatihan model LSTM	Dataset	Model LSTM berhasil dimuat tanpa <i>error</i>	Perangkat lunak dapat menghasilkan model LSTM	Terpenuhi

4.5.3.3 *Upload Data*

Tabel IV-38 berikut menampilkan hasil pengujian *use case* yang berkaitan dengan *upload data*.

Tabel IV- 38. Hasil Pengujian Use Case *Upload Data*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
U-001	Memasukan data pada sistem	Menekan tombol “Pilih File” dan “Unggah File”	Dataset	Sistem menampilkan nama dataset dan <i>preview data</i>	Perangkat lunak dapat memasukan data	Terpenuhi

4.5.3.4 Pengujian *Use Case Preprocessing Data*

Tabel IV-39 berikut menampilkan hasil pengujian *use case* yang berkaitan dengan *preprocessing data*.

Tabel IV- 39. Hasil Pengujian *Use Case Preprocessing Data*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
U-002	Menjalankan tahap pra-pemrosesan data yang diunggah pengguna	Menekan tombol “Proses dan Evaluasi Data”	Dataset	Hasil dari proses pra-pemrosesan data tersaji dalam tabel.	Perangkat lunak berhasil menjalankan tahap pra-pemrosesan	Terpenuhi

4.5.3.5 Pengujian *Use Case* Prosedur Evaluasi

Tabel IV-40 berikut menampilkan hasil pengujian *use case* yang meliputi tahap analisis.

Tabel IV- 40. Hasil Pengujian *Use Case* Evaluasi

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
U-003	Menjalankan tahap evaluasi berupa data pengujian	Menekan tombol “Proses dan	Dataset yang telah di proses melalui	Hasil dari proses data tersaji dalam bentuk <i>confusion</i>	Perangkat lunak berhasil mengeksekusi tahap evaluasi	Terpenuhi

	berdasarkan model LSTM	Evaluasi Data”	tahap pra-pemrosesan	<i>matrix</i> dan <i>classification report</i>		
--	------------------------	----------------	----------------------	--	--	--

4.5.3.6 Pengujian Use Case Prosedur Prediksi

Tabel IV-41 menyajikan hasil dari pengujian use case yang berkaitan dengan prediksi data. Pengujian tersebut dilakukan untuk menganalisis dan menguji penggunaan skenario yang berfokus pada proses prediksi data, yang mencakup berbagai aspek dan variabel yang relevan.

Tabel IV- 41. Hasil Pengujian *Use Case* Prediksi

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
U-004	Melakukan prediksi dan evaluasi berupa data dalam format CSV yang diunggah pengguna .	Melakukan prediksi dan evaluasi dengan tombol yang sesuai	Data yang sudah di prediksi oleh sistem	Mauskan terdiri dari data yang diunggah dalam format CSV	Hasil dari prediksi tersaji dalam bentuk tabel dan hasil evaluasi tersaji dalam bentuk <i>confusion matrix</i> serta <i>classification report</i>	Terpenuhi

U-005	Menguji kemampuan sistem dalam memprediksi sentimen dan aspek dari kalimat yang dimasukkan oleh pengguna.	Memprediksi kalimat yang telah dimasukkan oleh pengguna.	Masukkan kalimat berupa teks tunggal	Hasil yang diperoleh dari kalimat yang dimasukkan oleh pengguna	Sistem menghasilkan prediksi sentimen dan aspek	Terpenuhi
-------	---	--	--------------------------------------	---	---	-----------

4.6 Kesimpulan

Bab ini membahas mengenai pengembangan perangkat lunak yang dirancang untuk analisis sentimen ulasan pengguna pada aplikasi Gojek dengan memanfaatkan algoritma LSTM dan metode RUP. Penjelasan secara rinci mengenai proses pengembangan telah menghasilkan perangkat lunak yang memenuhi kebutuhan penelitian.