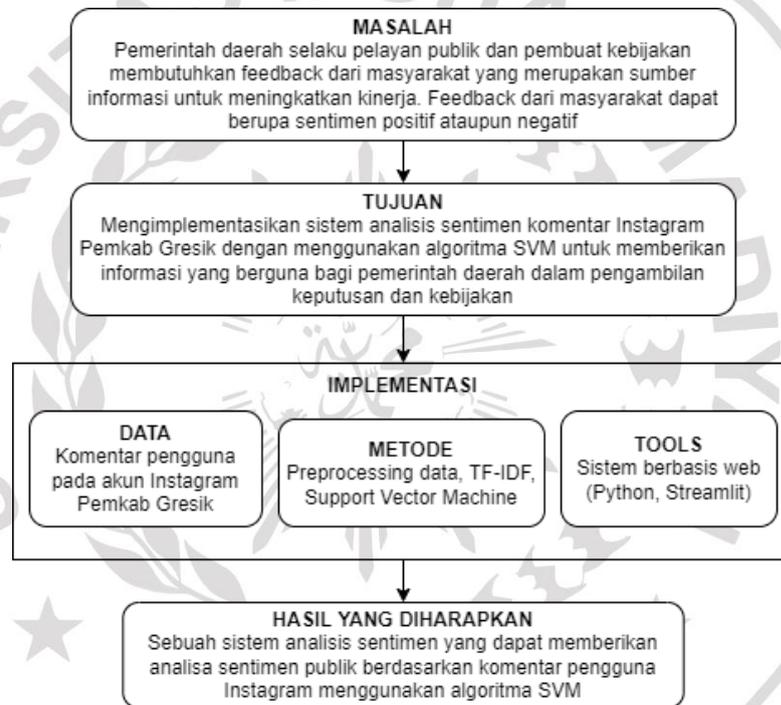


BAB III ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Setelah mengetahui dan mengenali latar belakang, tujuan, ruang lingkup dan proses yang sudah dikaji secara keseluruhan dari sistem tersebut, maka dapat disusun tahapan Analisis dan Perancangan. Dalam menganalisis sistem, diperlukan sebuah kerangka pemikiran agar pembaca dapat memahami tentang alur kinerja sistem yang akan dibuat. Berikut merupakan diagram kerangka pemikiran sistem analisis sentimen :



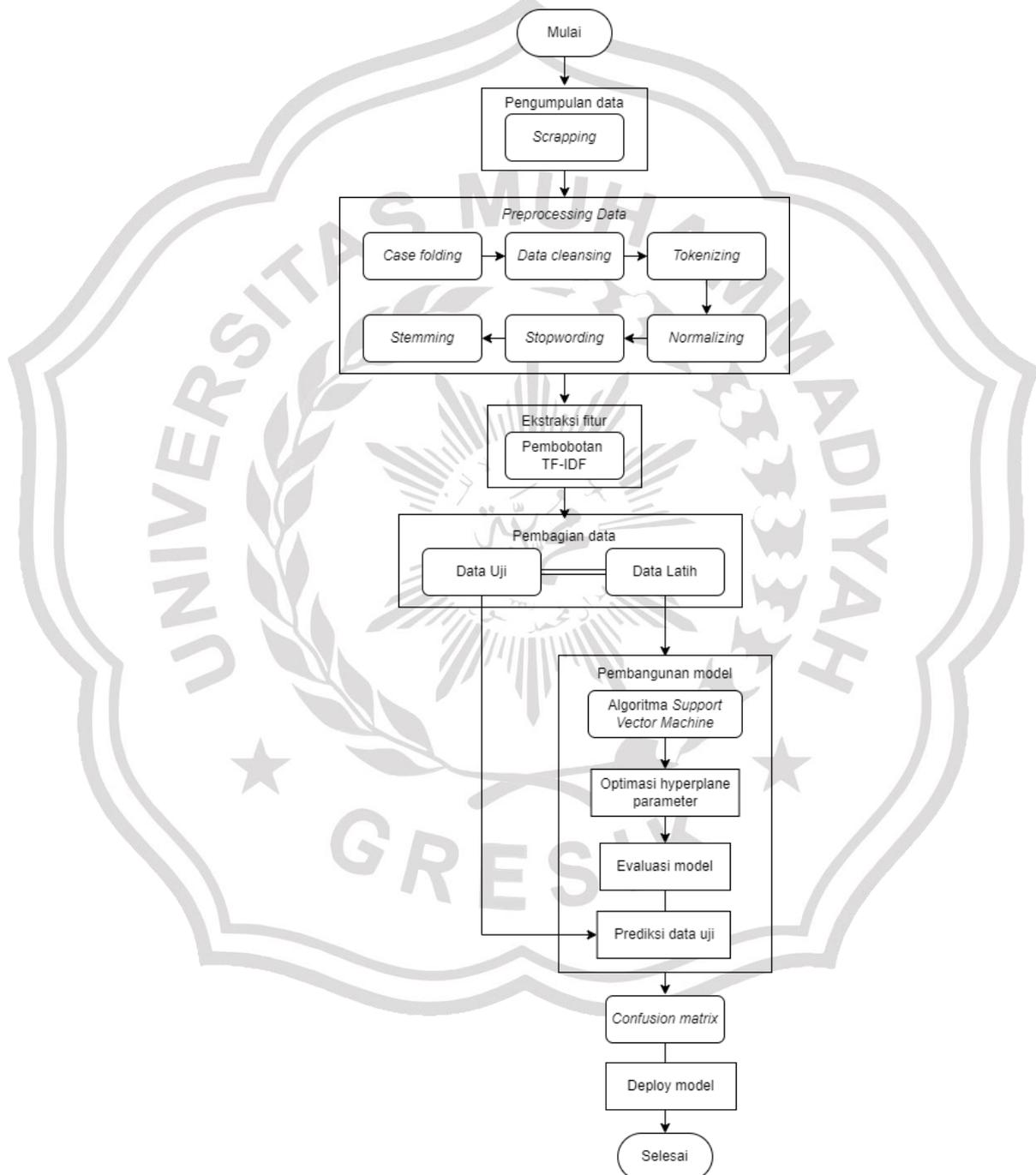
Gambar 3.1 Kerangka Pemikiran

Berdasarkan skema kerangka pemikiran yang telah disusun, maka sistem yang diinginkan dapat dirancang dan dikembangkan lebih lanjut. Skema tersebut memberikan panduan dan arah yang jelas mengenai komponen-komponen apa saja yang harus dimasukkan dalam sistem, serta bagaimana alur proses yang harus dijalankan. Dengan adanya kerangka pemikiran yang matang, proses perancangan sistem menjadi lebih terstruktur dan terarah, sehingga hasil akhirnya dapat memenuhi tujuan dan kebutuhan yang telah

ditetapkan sebelumnya.

3.2 Perancangan Sistem

Untuk mengimplementasikan sistem, dibuatlah diagram metode perancangan hasil dari analisis yang diusulkan sebagai berikut :

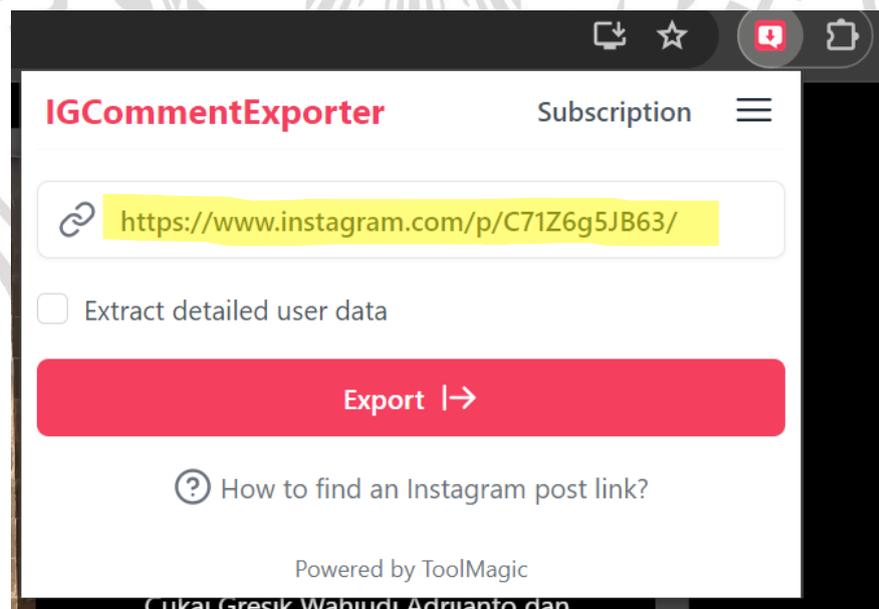


Gambar 3.2 Diagram Alur Sistem

Mulanya, data komentar Instagram dikumpulkan untuk dilakukan *preprocessing*. Setelah itu, ekstraksi fitur TF-IDF (*Term Frequency-Inverse Document Frequency*) dilakukan pada data untuk mendapatkan vektor data numerik dan pembobotan kata. Lalu, data dibagi menjadi dua bagian, yaitu data latih dan data uji. Untuk data latih, vektor akan diproses menggunakan metode SVM (*Support Vector Machine*) untuk mendapatkan garis pemisah atau *hyperplane*. Sehingga nantinya, data uji akan dilakukan perhitungan dengan membandingkannya terhadap garis *hyperplane* yang telah diperoleh dari data latih. Untuk menguji akurasi perbandingan antara data uji dan data latih, dilakukan perhitungan *confusion matrix*. Setelah itu, sistem dapat diunggah (*deploy*) agar dapat digunakan.

3.3 Pengumpulan Data

Proses pengumpulan data dilakukan dengan metode *web scrapping*. Aplikasi yang digunakan bernama IGCommentExporter yang merupakan fitur *Google Chrome Extension*. Dalam penggunaannya, mulanya salin tautan dari sebuah postingan yang akan dilakukan scrapping komentar. Setelah disalin, tautan tersebut dimasukkan ke dalam chrome extensi IGCommentExporter untuk mengunduh data.



Gambar 3.3 Aplikasi Web Scrapping Komentar Instagram

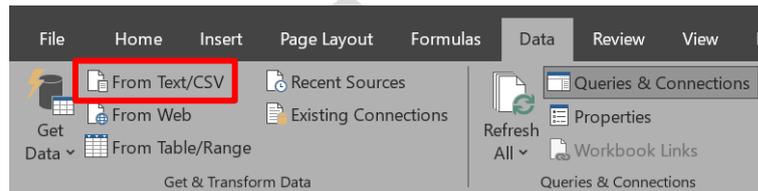
File yang diunduh akan berupa file csv yang berisi kumpulan detail komentar. Dari kumpulan data tersebut akan diambil data komentarnya saja untuk diproses dan dilakukan analisa sentimen.

Tabel 3.1 Contoh Data Hasil *Scrapping* Instagram

User ID,"Username","Comment ID","Comment","Comment Time","User Homepage","Avatar URL"
1405493283,"lesansblague_","="18053969683628496""", "Kemarin saya lewat jalan itu masih berlubang, kapan pemerintah memperbaikinya?","6/4/2024, 1:42:08 PM","https://www.instagram.com/lesansblague_","https://instagram.fsub3-1.fna.fbcdn.net/v/t51.2885-19/445887760_420042994235886_3920780153098734447_n.jpg?stp=dst-jpg_s150x150&_nc_ht=instagram.fsub3-1.fna.fbcdn.net&_nc_cat=106&_nc_ohc=nGtGEBFegQUQ7kNvgHrsuay&edm=AI-cjbYBAAAA&ccb=7-5&oh=00_AYCKC3wBaXKEhUlcy-2_l8ngOP4ePKGhu59CVh12T4cxUQ&oe=6665A546&_nc_sid=0985b0"
2071038035,"fitrah_ardi","="18003258374333051""", "Gresik semakin bagus dan maju, semoga semakin baik ke depannya ","6/4/2024, 2:42:37 PM","https://www.instagram.com/fitrah_ardi","https://instagram.fsub3-1.fna.fbcdn.net/v/t51.2885-19/1516166_1871520259738842_1265757858_a.jpg?stp=dst-jpg_s150x150&_nc_ht=instagram.fsub3-1.fna.fbcdn.net&_nc_cat=106&_nc_ohc=IMXWbRRCPjgQ7kNvgFMqgoE&edm=AI-cjbYBAAAA&ccb=7-5&oh=00_AYDOzArWwfbFzH16FQ3bdWEjMMojjn-cNFAWXJJDEtB7A&oe=66658D5B&_nc_sid=0985b0"
826816967,"tina_borneo","="17999397278621656""", "banyak jalan di Gresik yang berlubang, mohon diperhatikan ","6/4/2024, 3:08:04 PM","https://www.instagram.com/tina_borneo","https://instagram.fsub3-1.fna.fbcdn.net/v/t51.2885-19/307717350_768380607756373_9154891726271450450_n.jpg?stp=dst-jpg_s150x150&_nc_ht=instagram.fsub3-1.fna.fbcdn.net&_nc_cat=101&_nc_ohc=_MIJBzn4X88Q7kNvgHr3ByR&edm=AI-cjbYBAAAA&ccb=7-5&oh=00_AYAtWRdJDzpvpcQ-8eUgSPHW6GEc3xAqFGgv3ysC-pliIA&oe=66659FDB&_nc_sid=0985b0"
61232352212,"gresik_bahaya","="18026500159963792""", "di sekitar jalanan terate banyak menjual makanan khas Gresik yang enak ","6/4/2024, 3:53:09 PM","https://www.instagram.com/gresik_bahaya","https://instagram.fsub3-2.fna.fbcdn.net/v/t51.2885-19/361756443_2574495242701265_6358091578698500774_n.jpg?stp=dst-jpg_s150x150&_nc_ht=instagram.fsub3-2.fna.fbcdn.net&_nc_cat=105&_nc_ohc=G2dljIDn6u8Q7kNvgGvu3nF&edm=A

I-cjbYBAAAA&ccb=7-5&oh=00_AYBg7A6zLYoZC3Fcg4pwwqAdKu-dPqv26IUyzFL--86pDQ&oe=6665A793&_nc_sid=0985b0"

Data tersebut kemudian ditransformasikan menjadi bentuk tabel kolom menggunakan *tools* pada Microsoft Excel dengan cara Data > Get & Transform Data > From Text/CSV.



Gambar 3.4 Proses Transformasi Data Di Microsoft Excel

Lalu pilih file csv yang sudah diunduh tersebut dan kemudian dilakukan transformasi data untuk mengubah file delimiter koma (,) menjadi file bentuk kolom. Dari data tersebut, hanya data komentar saja yang diambil untuk kemudian diberi label sentimen secara manual oleh peneliti agar seluruh data komentar dapat lebih akurat sesuai dengan sentimen individu. Label sentimen dibagi menjadi 2 yaitu positif dan negatif. Pada Tabel 3.1 adalah contoh data komentar yang didapatkan dari proses transformasi dan pemberian label:

Tabel 3.2 Contoh Data Komentar Instagram

Komentar	Sentimen
Kemarin saya lewat jalan itu masih berlubang, kapan pemerintah memperbaikinya?	negatif
Gresik semakin bagus dan maju, semoga semakin baik ke depannya	positif
banyak jalan di Gresik yang berlubang, mohon diperhatikan	negatif
di sekitar jalanan terate banyak menjual makanan khas Gresik yang enak	positif

3.4 Preprocessing Data

Dari dataset yang telah terkumpul tersebut, selanjutnya dilakukan *preprocessing data*. Tahapan *preprocessing data* diantaranya yaitu menghapus sebutan atau *mention removing*, *case folding*, normalisasi,

tokenisasi, *stopwords* removing dan *stemming*. Berikut adalah langkah-langkah *preprocessing data*:

a. Case Folding

Proses *case folding* adalah proses mengubah semua karakter teks dari huruf besar menjadi huruf kecil, serta menghapus semua tanda baca dan angka, termasuk menghilangkan spasi berlebih.

- "kemarin saya lewat jalan itu masih berlubang, kapan pemerintah memperbaikinya?"
- "gresik semakin bagus dan maju, semoga semakin baik ke depannya"
- "banyak jalan di gresik yang berlubang, mohon diperhatikan"
- "di sekitar jalanan terate banyak menjual makanan khas gresik yang enak"

b. Tokenizing

Tokenizing adalah proses memecah sebuah kalimat utuh menjadi kumpulan kata-kata terpisah.

- ['kemarin', 'saya', 'lewat', 'jalan', 'itu', 'masih', 'berlubang', 'kapan', 'pemerintah', 'memperbaikinya']
- ['gresik', 'semakin', 'bagus', 'dan', 'maju', 'semoga', 'semakin', 'baik', 'ke', 'depannya']
- ['banyak', 'jalan', 'di', 'gresik', 'yang', 'berlubang', 'mohon', 'diperhatikan']
- ['di', 'sekitar', 'jalanan', 'terate', 'banyak', 'menjual', 'makanan', 'khas', 'gresik', 'yang', 'enak']

c. Stopwords Removing

Stopwords removing adalah proses menghilangkan kata-kata yang dianggap tidak memberikan makna penting atau kontribusi signifikan dalam sebuah analisis teks. Dengan menghilangkan stopwords, analisis teks dapat lebih fokus pada kata-kata yang dianggap lebih penting dan bermakna dalam menangkap inti dari sebuah teks. Beberapa contoh kata

yang dihilangkan dari kumpulan data di atas diantaranya:

saya di
 itu yang
 dan ke
 masih kapan

Sehingga hasil yang didapatkan dari penghapusan kata di atas adalah sebagai berikut:

- ['kemarin', 'lewat', 'jalan', 'berlubang', 'pemerintah', 'memperbaikinya']
- ['gresik', 'semakin', 'bagus', 'maju', 'semoga', 'semakin', 'baik', 'depannya']
- ['banyak', 'jalan', 'gresik', 'berlubang', 'mohon', 'diperhatikan']
- ['sekitar', 'jalanan', 'terate', 'banyak', 'menjual', 'makanan', 'khas', 'gresik', 'enak']

d. Stemming

Poses *stemming* yaitu mengubah kata berimbuhan menjadi kata dasar (*root word/stem*) dengan menghilangkan imbuhan seperti awalan dan akhiran. Sehingga didapatkan hasil sebagai berikut:

- ['kemarin', 'lewat', 'jalan', 'lubang', 'perintah', 'baik']
- ['gresik', 'makin', 'bagus', 'maju', 'moga', 'makin', 'baik', 'depan']
- ['banyak', 'jalan', 'gresik', 'lubang', 'mohon', 'perhatian']
- ['sekitar', 'jalan', 'terate', 'banyak', 'jual', 'makan', 'khas', 'gresik', 'enak']

Setelah dilakukan tahapan *preprocessing data*, data komentar tersebut telah menjadi bentuk susunan kata dasar hasil dari proses *stemming*.

3.5 Pembangunan Model Klasifikasi

Membangun model klasifikasi terdiri dari tahapan ekstraksi fitur TF-IDF dan SVM.

3.5.1 TF-IDF

Dari dataset yang telah dilakukan *preprocessing* hingga membentuk susunan kata dasar, kemudian dilakukan transformasi data teks menjadi data numerik. Proses transformasi dilakukan dengan perhitungan frekuensi kemunculan dari beberapa data tersebut dengan menerapkan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Berikut adalah langkah perhitungan TF-IDF dengan bentuk komentar yang telah dilakukan tahap *preprocessing* sebagai data latih:

Tabel 3.3 Data Komentar Hasil *Preprocessing*

Data ke-	Komentar
1	['kemarin', 'lewat', 'jalan', 'lubang', 'perintah', 'baik']
2	['gresik', 'makin', 'bagus', 'maju', 'moga', 'makin', 'baik', 'depan']
3	['banyak', 'jalan', 'gresik', 'lubang', 'mohon', 'perhatian']
4	['sekitar', 'jalan', 'terate', 'banyak', 'jual', 'makan', 'khas', 'gresik', 'enak']

Dari data komentar yang telah dilakukan stemming kemudian dilakukan perhitungan TF (*Term Frequency*) untuk setiap kata di setiap dokumen.

$$TF(\text{Term Frequency}) = \frac{\text{jumlah kemunculan kata dalam dokumen}}{\text{jumlah total kata dalam dokumen}}$$

Tabel 3.4 Data Komentar Proses TF

Dokumen 1: (total kata d1=6)	Dokumen 2: (total kata d2=8)
TF("kemarin") = 1/6 = 0.167	TF("gresik") = 1/8 = 0.125
TF("lewat") = 1/6 = 0.167	TF("makin") = 2/8 = 0.25
TF("jalan") = 1/6 = 0.167	TF("bagus") = 1/8 = 0.125
TF("lubang") = 1/6 = 0.167	TF("maju") = 1/8 = 0.125
TF("perintah") = 1/6 = 0.167	TF("moga") = 1/8 = 0.125
TF("baik") = 1/6 = 0.167	TF("baik") = 1/8 = 0.125

	TF("depan") = $1/8 = 0.125$
Dokumen 3: (total kata $d_3=6$)	Dokumen 4: (total kata $d_4=9$)
TF("banyak") = $1/6 = 0.167$	TF("sekitar") = $1/9 = 0.111$
TF("jalan") = $1/6 = 0.167$	TF("jalan") = $1/9 = 0.111$
TF("gresik") = $1/6 = 0.167$	TF("terate") = $1/9 = 0.111$
TF("lubang") = $1/6 = 0.167$	TF("banyak") = $1/9 = 0.111$
TF("mohon") = $1/6 = 0.167$	TF("jual") = $1/9 = 0.111$
TF("perhatian") = $1/6 = 0.167$	TF("makan") = $1/9 = 0.111$
	TF("khas") = $1/9 = 0.111$
	TF("gresik") = $1/9 = 0.111$
	TF("enak") = $1/9 = 0.111$

Langkah selanjutnya menghitung IDF (*Inverse Document Frequency*) untuk setiap kata. Jika terdapat 4 dokumen data, maka $N = 4$ sehingga dengan menggunakan rumus 2.1 di atas, didapatkan seperti Tabel 3.4 berikut:

Tabel 3.5 Langkah *TF-IDF*

No	Kata	d1	d2	d3	d4	df	IDF
1	Bagus		1			1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
2	Baik	1	1			2	$\log\left(\frac{4}{2}\right) + 1 = 1,301$
3	Banyak			1	1	2	$\log\left(\frac{4}{2}\right) + 1 = 1,301$
4	Depan		1			1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
5	Enak				1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
6	Gresik		1	1	1	3	$\log\left(\frac{4}{3}\right) + 1 = 1,125$
7	Jalan	1		1	1	3	$\log\left(\frac{4}{3}\right) + 1 = 1,125$
8	Jual				1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
9	Kemarin	1				1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
10	Khas				1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
11	Lewat	1				1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
12	Lubang	1		1		2	$\log\left(\frac{4}{2}\right) + 1 = 1,301$
13	Maju		1			1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
14	Makan				1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
15	Makin		2			2	$\log\left(\frac{4}{2}\right) + 1 = 1,301$

16	Moga		1		1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
17	Mohon			1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
18	Perhatian			1	1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
19	Perintah	1			1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
20	Sekitar				1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$
21	Terate				1	$\log\left(\frac{4}{1}\right) + 1 = 1,602$

Dari perhitungan IDF tersebut, nantinya kumpulan kata ini akan dibentuk menjadi sebuah korpus data latih. Kemudian menggunakan persamaan 2.2 dilakukan perhitungan TF-IDF untuk setiap kata di setiap dokumen:

Tabel 3.6 Data Komentor proses TF-IDF

<p>TF-IDF Dokumen 1:</p> <p>"kemarin" = $0.167 \times 1,602 = 0.267$</p> <p>"lewat" = $0.167 \times 1,602 = 0.267$</p> <p>"jalan" = $0.167 \times 1,125 = 0.187$</p> <p>"lubang" = $0.167 \times 1,301 = 0.217$</p> <p>"perintah" = $0.167 \times 1,602 = 0.267$</p> <p>"baik" = $0.167 \times 1,301 = 0.217$</p>	<p>TF-IDF Dokumen 2:</p> <p>"gresik" = $0.125 \times 1,125 = 0.14$</p> <p>"makin" = $0.25 \times 1,301 = 0.325$</p> <p>"bagus" = $0.125 \times 1,602 = 0.2$</p> <p>"maju" = $0.125 \times 1,602 = 0.2$</p> <p>"moga" = $0.125 \times 1,602 = 0.2$</p> <p>"baik" = $0.125 \times 1,301 = 0.163$</p> <p>"depan" = $0.125 \times 1,602 = 0.2$</p>
<p>TF-IDF Dokumen 3:</p> <p>"banyak" = $0.167 \times 1,301 = 0.217$</p> <p>"jalan" = $0.167 \times 1,125 = 0.187$</p> <p>"gresik" = $0.167 \times 1,125 = 0.187$</p> <p>"lubang" = $0.167 \times 1,301 = 0.217$</p> <p>"mohon" = $0.167 \times 1,602 = 0.267$</p> <p>"perhatian" = $0.167 \times 1,602 = 0.267$</p>	<p>TF-IDF Dokumen 4:</p> <p>"sekitar" = $0.111 \times 1,602 = 0.178$</p> <p>"jalan" = $0.111 \times 1,125 = 0.124$</p> <p>"terate" = $0.111 \times 1,602 = 0.178$</p> <p>"banyak" = $0.111 \times 1,301 = 0.144$</p> <p>"jual" = $0.111 \times 1,602 = 0.178$</p> <p>"makan" = $0.111 \times 1,602 = 0.178$</p> <p>"khas" = $0.111 \times 1,602 = 0.178$</p> <p>"gresik" = $0.111 \times 1,125 = 0.124$</p> <p>"enak" = $0.111 \times 1,602 = 0.178$</p>

Jadi, setiap dokumen sekarang memiliki vektor TF-IDF yang menunjukkan bobot setiap kata dalam dokumen tersebut. Vektor ini akan digunakan dalam proses klasifikasi menggunakan SVM.

3.5.2 SVM

Dalam menerapkan metode klasifikasi *Support Vector Machine* (SVM), dibutuhkan dua jenis data, yaitu data latih dan data uji. Pengklasifikasian data bertujuan untuk memperkirakan kelas dari suatu objek yang labelnya belum diketahui saat pembentukan model data. Data latih dimanfaatkan untuk membangun model *classifier* yang dapat digunakan untuk memprediksi kelas dari data baru. Sedangkan data uji digunakan untuk melakukan pengukuran terhadap model *classifier* yang telah dibentuk dari data latih guna menentukan hasil klasifikasi label data tersebut (Februariyanti, Firmansyah, Wibowo, & Utomo, 2020).

Perhitungan klasifikasi SVM secara manual dari hasil nilai TF-IDF pada Tabel 3.6 dengan asumsi awal sentimen kalimat untuk setiap dokumen sesuai pada Tabel 3.2 masing-masing secara berurutan yaitu negatif, positif, negatif, positif. Dari data tersebut terdapat dua kelas, yaitu kelas positif (+1) dan kelas negatif (-1). Dimana kelas positif yaitu dokumen 1 dan dokumen 3, sedangkan kelas negatif yaitu dokumen 2 dan dokumen 4.

Hasil dari perhitungan ekstraksi fitur TF-IDF kemudian disusun dalam bentuk vektor sesuai dengan urutan abjad dari korpus data latih.

Tabel 3.7 Pembentukan Vektor Hasil TF-IDF

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11
x1	0	0.217	0	0	0	0	0.187	0	0.267	0	0.267
x2	0.2	0.163	0	0.2	0	0.14	0	0	0	0	0
x3	0	0	0.217	0	0	0.187	0.187	0	0	0	0
x4	0	0	0.144	0	0.178	0.124	0.124	0.178	0	0.178	0

	t12	t13	t14	t15	t16	t17	t18	t19	t20	t21
x1	0.217	0	0	0	0	0	0	0.267	0	0
x2	0	0.2	0	0.325	0.2	0	0	0	0	0
x3	0.217	0	0	0	0	0.267	0.267	0	0	0
x4	0	0	0.178	0	0	0	0	0	0.178	0.178

1. Menentukan vektor data latih:

- Dokumen 1 (kelas -1):

$$X_1 = [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0]$$

- Dokumen 2 (kelas +1):

$$X_2 = [0.2, 0.163, 0, 0.2, 0, 0.14, 0, 0, 0, 0, 0.2, 0, 0.325, 0.2, 0, 0, 0, 0, 0, 0]$$

- Dokumen 3 (kelas -1):

$$X_3 = [0, 0, 0.217, 0, 0, 0.187, 0.187, 0, 0, 0, 0, 0.217, 0, 0, 0, 0, 0.267, 0.267, 0, 0, 0]$$

- Dokumen 4 (kelas +1):

$$X_4 = [0, 0, 0.144, 0, 0.178, 0.124, 0.124, 0.178, 0, 0.178, 0, 0, 0, 0.178, 0, 0, 0, 0, 0.178, 0.178]$$

2. Menghitung kernel data latih

Dengan menggunakan persamaan 2.10, dilakukan perhitungan fungsi kernel untuk mencari nilai $K(x_i, x_j)$.

$$\begin{aligned} K(x_1, x_1^T) &= [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0] \cdot [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0]^T \\ &= 0.343 \end{aligned}$$

$$\begin{aligned} K(x_1, x_2^T) &= [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0] \cdot [0.2, 0.163, 0, 0.2, 0, 0.14, 0, 0, 0, 0, 0.2, 0, 0.325, 0.2, 0, 0, 0, 0, 0, 0]^T \\ &= 0.0884 \end{aligned}$$

$$\begin{aligned} K(x_1, x_3^T) &= [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0] \cdot [0, 0, 0.217, 0, 0, 0.187, 0.187, 0, 0, 0, 0, 0.217, 0, 0, 0, 0, 0.267, 0.267, 0, 0]^T \\ &= 0.082 \end{aligned}$$

$$\begin{aligned} K(x_1, x_4^T) &= [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0] \cdot [0, 0, 0.144, 0, 0.178, 0.124, 0.124, 0.178, 0, 0.178, 0, 0, 0, 0.178, 0, 0, 0, 0.178, 0.178] \end{aligned}$$

$$\begin{aligned}
& 0,124, 0,178, 0, 0,178, 0, 0, 0, 0,178, 0, 0, 0, 0, 0, 0,178, \\
& 0,178]^T \\
& = 0,023
\end{aligned}$$

Perhitungan serupa dilakukan untuk iterasi $x_i = 1$ sampai 4 dan

$x_j^T = 1$ sampai 4 sehingga diperoleh matriks hasil $x_i x_j^T$ berikut ini:

$$x_i x_j^T = \begin{bmatrix} 0,343 & 0,0887 & 0,082 & 0,023 \\ 0,0887 & 0,312 & 0,0262 & 0,053 \\ 0,082 & 0,0262 & 0,3067 & 0,0776 \\ 0,023 & 0,053 & 0,0776 & 0,274 \end{bmatrix}$$

3. Menghitung Lagrange Multiplier

Nilai Lagrange Multiplier ini untuk memperoleh nilai α_i yang menghasilkan nilai Lagrange Multiplier yang maksimal dengan menggunakan persamaan 2.6 seperti berikut ini :

$$\begin{aligned}
\max_{\alpha} L_D &= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2} (0,343\alpha_1\alpha_1 + 0,0887\alpha_1\alpha_2 + \\
& 0,082\alpha_1\alpha_3 + 0,023\alpha_1\alpha_4 + 0,0887\alpha_2\alpha_1 + 0,312\alpha_2\alpha_2 + \\
& 0,0262\alpha_2\alpha_3 + 0,053\alpha_2\alpha_4 + 0,082\alpha_3\alpha_1 + 0,0262\alpha_3\alpha_2 \\
& + 0,3067\alpha_3\alpha_3 + 0,0776\alpha_3\alpha_4 + 0,023\alpha_4\alpha_1 + \\
& 0,053\alpha_4\alpha_2 + 0,0776\alpha_4\alpha_3 + 0,274\alpha_4\alpha_4) \\
&= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2} [0,343\alpha_1^2 + 0,312\alpha_2^2 + \\
& 0,3067\alpha_3^2 + 0,274\alpha_4^2 + (0,0887 + 0,0887)\alpha_1\alpha_2 + \\
& (0,082 + 0,082)\alpha_1\alpha_3 + (0,023 + 0,023)\alpha_1\alpha_4 + \\
& (0,0262 + 0,0262)\alpha_2\alpha_3 + (0,053 + 0,053)\alpha_2\alpha_4 + \\
& (0,0776 + 0,0776\alpha_4\alpha_3)] \\
&= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2} [0,343\alpha_1^2 + 0,312\alpha_2^2 + \\
& 0,3067\alpha_3^2 + 0,274\alpha_4^2 + 0,1774\alpha_1\alpha_2 + 0,164\alpha_1\alpha_3 + \\
& 0,046\alpha_1\alpha_4 + 0,0524\alpha_2\alpha_3 + 0,106\alpha_2\alpha_4 + 0,1552\alpha_4\alpha_3] \\
\max_{\alpha} L_D &= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - (0,1715\alpha_1^2 + 0,156\alpha_2^2 + \\
& 0,15335\alpha_3^2 + 0,136816\alpha_4^2 + 0,0887\alpha_1\alpha_2 + 0,082\alpha_1\alpha_3 \\
& + 0,023\alpha_1\alpha_4 + 0,0262\alpha_2\alpha_3 + 0,053\alpha_2\alpha_4 + 0,0776\alpha_4\alpha_3)
\end{aligned}$$

Dengan syarat : $\alpha_i \geq 0$ ($i = 1, 2, \dots, n$) dan $\sum_{i=1}^n \alpha_i y_i = 0$ sehingga sesuai dengan label dokumen yang telah ditentukan $y_1 = -1$,

$y_2=1, y_3 = -1, y_4 = 1$ maka untuk mendapatkan nilai α_i , dihitung turunan parsial dari L terhadap α_i dan λ dan setarakan dengan nol.

$$\frac{\partial L}{\partial \alpha} = 0, \frac{\partial L}{\partial \lambda} = 0$$

Dan kita hitung turunan parsial untuk setiap α_i :

$$\frac{\partial L}{\partial \alpha} = 1 - 0.1715\alpha_1 - 0.0887\alpha_2 - 0.082\alpha_3 - 0.023\alpha_4 + \lambda y_1 = 0$$

$$\frac{\partial L}{\partial \alpha} = 1 - 0.156\alpha_2 - 0.0887\alpha_1 - 0.0262\alpha_3 - 0.053\alpha_4 + \lambda y_2 = 0$$

$$\frac{\partial L}{\partial \alpha} = 1 - 0.15335\alpha_3 - 0.082\alpha_1 - 0.0262\alpha_2 - 0.0776\alpha_4 + \lambda y_3 = 0$$

$$\frac{\partial L}{\partial \alpha} = 1 - 0.136816\alpha_4 - 0.023\alpha_1 - 0.053\alpha_2 - 0.0776\alpha_3 + \lambda y_4 = 0$$

$$\frac{\partial L}{\partial \alpha} = -\alpha_1 + \alpha_2 - \alpha_3 + \alpha_4 = 0$$

Substitusi nilai $y_1 = -1, y_2 = 1, y_3 = -1, y_4 = 1$

Sehingga didapatkan $\max_{\alpha} L_D = 3.2356$ dengan $\alpha_1 = 2.8715, \alpha_2 = 2.6122, \alpha_3 = 3.2243, \alpha_4 = 3.4836$

4. Menghitung nilai bobot atau w

Untuk menghitung nilai bobot dapat dilakukan dengan menggunakan persamaan 2.7 yaitu:

$$\begin{aligned} w &= -\alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 - \alpha_3 y_3 x_3 + \alpha_4 y_4 x_4 \\ &= -2.8715 [0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, 0.217, 0, 0, 0, \\ &\quad 0, 0, 0, 0.267, 0, 0] + 2.6122 [0.2, 0.163, 0, 0.2, 0, 0.14, 0, 0, 0, 0.2, \\ &\quad 0, 0.325, 0.2, 0, 0, 0, 0, 0, 0, 0] - 3.2243 [0, 0, 0.217, 0, 0, 0.187, \\ &\quad 0.187, 0, 0, 0, 0, 0.217, 0, 0, 0, 0, 0.267, 0.267, 0, 0, 0] + 3.4836 [0, \\ &\quad 0, 0.144, 0, 0.178, 0.124, 0.124, 0.178, 0, 0.178, 0, 0, 0, 0.178, 0, 0, 0, \\ &\quad 0, 0, 0.178, 0.178] \\ &= [0.5225, -0.19735, -0.19823, 0.5225, 0.6201, 0.1965, -0.7058, \\ &\quad 0.6201, -0.7678, 0.6201, -0.24536, -1.32301, 0.84997, 1.14255, 0, \\ &\quad 0, -0.861, -0.86, -0.7678, 0.6201048, 0.62] \end{aligned}$$

5. Mencari nilai bias

Sedangkan untuk mencari nilai bias menggunakan persamaan

2.8 sebagai berikut:

$$\begin{aligned}
 b &= -\frac{1}{2} (x_1 w + x_2 w + x_3 w + x_4 w) \\
 &= -\frac{1}{2} w (x_1 + x_2 + x_3 + x_4 w) \\
 &= -\frac{1}{2} ((0.5225, -0.19735, -0.19823, 0.5225, 0.6201, 0.1965, \\
 &\quad -0.7058, 0.6201, -0.7678, 0.6201, -0.24536, -1.32301, \\
 &\quad 0.84997, 1.14255, 0, 0, -0.861, -0.86, -0.7678, 0.6201048, \\
 &\quad 0.62)) \cdot [-(0, 0.217, 0, 0, 0, 0, 0.187, 0, 0.267, 0, 0.267, \\
 &\quad 0.217, 0, 0, 0, 0, 0, 0.267, 0, 0) + (0.2, 0.163, 0, 0.2, 0, \\
 &\quad 0.14, 0, 0, 0, 0, 0.2, 0, 0.325, 0.2, 0, 0, 0, 0, 0, 0) - (0, 0, \\
 &\quad 0.217, 0, 0, 0.187, 0.187, 0, 0, 0, 0, 0.217, 0, 0, 0, 0, 0.267, \\
 &\quad 0.267, 0, 0, 0) + (0, 0, 0.144, 0, 0.178, 0.124, 0.124, 0.178, \\
 &\quad 0, 0.178, 0, 0, 0, 0.178, 0, 0, 0, 0, 0.178, 0.178) \\
 &= -1.4959
 \end{aligned}$$

Sehingga setelah diketahui bobot w dan nilai bias, untuk mengklasifikasikan data baru, dilakukan tahapan seperti berikut:

Contoh:

“Kemarin pulang ke Gresik, sudah banyak jalan yang diperbaiki”

Setelah di lakukan *preprocessing text*, didapatkan hasil:

“kemarin”, “pulang”, “gresik”, “sudah”, “banyak”, “jalan”, “baik”

Lalu data tersebut dihitung menggunakan TF-IDF untuk ditentukan bobot kalimat. Dan didapatkan hasil vektor:

[0, 0.2602, 0.2602, 0, 0, 0.225, 0.225, 0, 0.3204, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Untuk kata “pulang” dan “sudah” diabaikan karena tidak terdapat dalam korpus data latih. Setelah itu dilakukan perhitungan keputusan klasifikasi menggunakan persamaan 2.9 sebagai berikut:

$$\begin{aligned}
 f(x) &= \text{sign} [\vec{w} \cdot \vec{x} + b] \\
 &= \text{sign} [(0.5224, -0.1973, -0.1982, 0.5224, 0.6201, 0.1965, \\
 &\quad -0.7058, 0.6201, -0.7678, 0.6201, -0.24536, -1.3230, \\
 &\quad 0.8499, 1.14255, 0, 0, -0.8601, -0.8601, -0.7678, 0.6201, \\
 &\quad 0.6201) \cdot (0, 0.2602, 0.2602, 0, 0, 0.225, 0.225, 0, 0.3204, \\
 &\quad 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) - 1.4959] \\
 &= \text{sign} [-0.46359192 - 1.4959]
 \end{aligned}$$

$$= \text{sign}[-1.95949192]$$

$$= -1$$

Karena hasil perhitungan antara vektor data baru dengan *hyperplane* pemisah adalah -1.95949192 , maka data baru akan diklasifikasikan sebagai kelas negatif (-1). Hal ini dapat diasumsikan karena data uji melakukan klasifikasi sesuai dengan korpus data latih dimana:

Tabel 3.8 Klasifikasi Dokumen Data Latih

Kata	Klasifikasi Dokumen
kemarin	(-)
gresik	(+) (-) (+)
banyak	(+) (-)
jalan	(-) (-) (+)
baik	(+) (-)

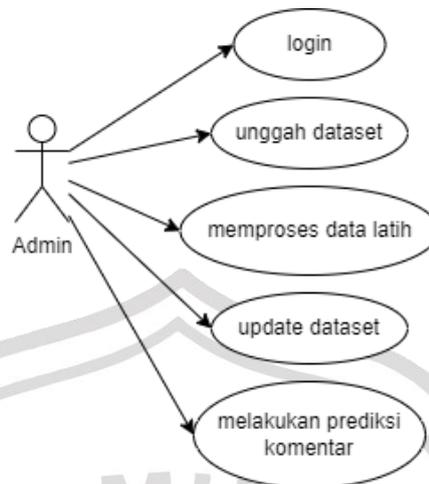
Sehingga dari Tabel 3.7 dapat ditarik kesimpulan bahwa kalimat tersebut diklasifikasikan oleh mesin SVM menjadi kalimat yang cenderung negatif. Namun hal ini akan berubah apabila terdapat lebih banyak data latih agar klasifikasi dapat menjadi lebih akurat.

3.6 Desain Perancangan Sistem

Proses perancangan sistem terdiri dari proses pembuatan diagram *use case*, dan desain tampilan *interface*.

3.6.1 Use Case Diagram

Use case diagram adalah grafik yang digunakan untuk menggambarkan interaksi antara sistem dan pengguna di dalam sistem tersebut. Diagram ini membantu mengidentifikasi kebutuhan sistem, mengorganisasi permintaan sistem, dan menunjukkan aliran event dalam sistem. *Use case diagram* tidak menggambarkan bagaimana *event* tersebut diimplementasikan, tetapi menggambarkan bagaimana pengguna interaksi dengan sistem.

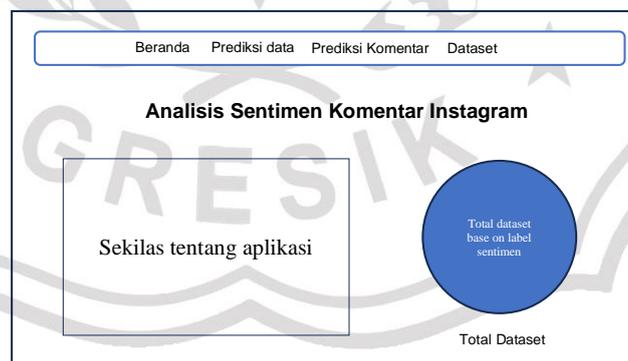


Gambar 3.5 Diagram Use Case

Dalam diagram di atas, aktor yang berperan adalah admin yang bertanggung jawab untuk *media analytic* Instagram Pemkab Gresik. Admin dapat mengunggah dataset berupa komentar untuk dijadikan acuan sebagai prediksi komentar lainnya.

3.6.2 Desain Interface Sistem Aplikasi

Dalam membuat sebuah sistem *website*, diperlukan desain *interface* agar lebih memudahkan dan menyesuaikan dengan kebutuhan pengguna. Berikut merupakan desain dari beberapa tampilan halaman website sistem analisis sentimen:



Gambar 3.6 Desain Tampilan Halaman Beranda

Halaman beranda adalah tampilan awal saat website diakses. Pada halaman ini, terdapat deskripsi singkat tentang aplikasi dan diagram total dataset yang digunakan sebagai data latih.

Beranda Prediksi data Prediksi Komentar Dataset

Upload file (.csv)

[Pilih file](#)

Username	Komentar

Gambar 3.7 Desain Tampilan Halaman Prediksi Data

Halaman Prediksi Data adalah halaman yang digunakan untuk memprediksi sejumlah komentar yang akan dianalisis dalam bentuk file CSV yang dapat diunggah. Sistem kemudian akan langsung melakukan proses prediksi untuk setiap baris data pada file yang diunggah tersebut.

Beranda Prediksi data Prediksi Komentar Dataset

Hasil prediksi [Unduh csv](#)

Username	Komentar	sentimen
	مختار	

Gambar 3.8 Lanjutan Desain Tampilan Halaman Hasil Prediksi Data
Hasil prediksi berupa label sentimen untuk setiap baris komentar. Hasil prediksi ini nantinya dapat diunduh.

Beranda Prediksi data Prediksi Komentar Dataset

Prediksi Komentar

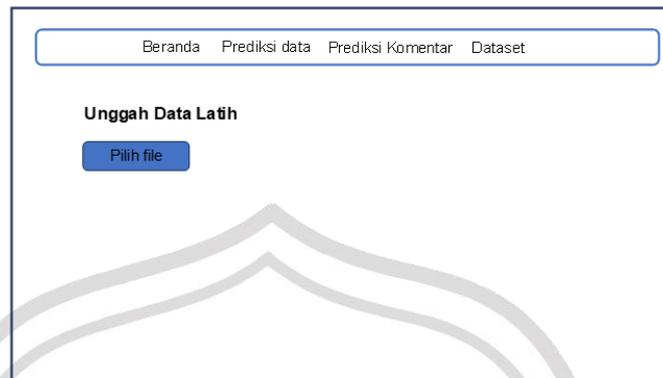
[Prediksi](#)

[Hasil Prediksi](#)

Gambar 3.9 Desain Tampilan Halaman Prediksi

Tampilan halaman prediksi untuk 1 komentar yang hasilnya dapat

langsung ditampilkan.



Gambar 3.10 Desain Tampilan Halaman Unggah Dataset

Halaman untuk mengunggah dataset guna menambah data latih yang nantinya akan dapat dilatih ulang.

3.7 Pengujian Sistem

Pengujian fungsionalitas sistem dilakukan dengan metode Confussion Matrix untuk mengukur akurasi model. Pengujian ini dilakukan untuk memastikan bahwa sistem dapat mengklasifikasikan komentar Instagram ke dalam kelas sentimen yang tepat (positif, negatif, atau netral) dengan menggunakan algoritma Support Vector Machine (SVM). Skenario pengujian fungsionalitas meliputi pengujian dengan kumpulan data komentar Instagram yang telah diketahui sentimen sebenarnya. Tujuannya untuk menguji apakah sistem dapat mengklasifikasikan sentimen komentar dengan benar sesuai dengan masukan yang diberikan.

Pengujian performa sistem antarmuka dilakukan dengan menggunakan metode pengujian alpha (Black Box testing) seperti tampilan, navigasi, dan kemudahan penggunaan dan pengujian beta. Pengujian beta dilakukan setelah pengujian alpha untuk mendapatkan umpan balik dari pengguna nyata. Tujuan utamanya adalah untuk menemukan bug atau masalah yang mungkin tidak terdeteksi selama pengujian alpha dan untuk mendapatkan umpan balik mengenai pengalaman pengguna (Menora, et al., 2023).

Seluruh proses pengujian sistem dilakukan dalam lingkungan pengujian

yang terkontrol, dengan menggunakan perangkat keras dan perangkat lunak yang sesuai dengan spesifikasi sistem yang direncanakan untuk diimplementasikan dalam lingkungan produksi.

Hasil pengujian sistem analisis sentimen dievaluasi dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengukur kinerja klasifikasi sentimen oleh algoritma SVM.

