

BAB III

ANALISA, ALUR PENELITIAN DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang Analisis kebutuhan dan perancangan sistem dalam penelitian yang mengaplikasikan Metode *Naïve bayes* untuk menganalisis Sentimen terhadap kinerja Bupati sebagai Bupati Lamongan berdasarkan komentar masyarakat.

3.1 Analisa Kebutuhan

Analisis kebutuhan adalah langkah penting dalam pengembangan sistem atau penelitian, karena membantu memastikan bahwa tujuan dan persyaratan proyek atau penelitian dipahami dengan baik sebelum memulai perancangan atau implementasi. Penting untuk mencatat bahwa analisis kebutuhan adalah proses berkelanjutan, dan kebutuhan dapat berubah seiring waktu atau sebagai tanggapan terhadap perubahan dalam lingkungan proyek atau penelitian. Oleh karena itu, komunikasi yang baik antara peneliti, pengguna, dan tim pengembangan sangat penting dalam memastikan bahwa semua kebutuhan dipahami dan dipenuhi dengan baik selama seluruh siklus pengembangan atau penelitian. Kebutuhan dapat dibagi menjadi dua kategori utama: kebutuhan fungsional dan kebutuhan non-fungsional.

3.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah spesifikasi tentang apa yang sistem atau produk perangkat lunak harus lakukan, yaitu fungsi-fungsi atau layanan-layanan yang harus disediakan oleh sistem tersebut. Ini mencakup input yang diterima oleh sistem, proses yang dilakukan oleh sistem, dan output yang dihasilkan oleh sistem berdasarkan input dan proses tersebut. Dalam konteks pengembangan perangkat lunak atau sistem lainnya, kebutuhan fungsional adalah panduan yang sangat penting untuk memahami dan merencanakan apa yang harus dikerjakan oleh tim pengembangan. Kebutuhan fungsional sering kali diuraikan dalam dokumen atau

spesifikasi yang berisi daftar rinci dari fitur-fitur, tindakan-tindakan, dan layanan-layanan yang diharapkan dari sistem. Dokumen ini membantu dalam komunikasi yang jelas antara pihak-pihak yang terlibat dalam pengembangan, seperti klien, manajer proyek, dan pengembang.

Kebutuhan fungsional pada sistem analisis sentimen ini adalah pengambilan data dengan pemakaian instrumen yang sesuai serta penghapusan pada atribut yang tidak digunakan. Penggunaan *flowchart* pada tahapan *preprocessing* yaitu tahapan penghilangan *noise* pada data yang tidak memiliki dampak pada proses klasifikasi, pengadaan tahapan ekstraksi fitur dengan metode *TF-IDF* setelah tahapan *preprocessing* agar pada tahapan klasifikasi menggunakan metode *naïve bayes* mendapat akurasi yang lebih optimal serta visualisasi akurasi sistem menggunakan matriks konfusi.

3.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional berupa perangkat keras (*Hardware*) dan perangkat lunak (*Software*) yang digunakan untuk membantu dalam penelitian hingga perancangan sistem.

1. Perangkat Keras (*Hardware*)

Sebuah perangkat keras digunakan untuk membuat suatu sistem dengan adanya perangkat keras peneliti dapat membuat program dan menjalankan sistem yang dibuat

dengan bantuan perangkat keras yang mumpuni, semakin baik perangkat keras yang dibutuhkan maka semakin baik pula keandalan perangkat yang digunakan.

Perangkat keras yang digunakan adalah sebuah komputer dengan spesifikasi minimum sebagai berikut:

- a. *Processor Intel Inside*
- b. RAM 4 GB
- c. *Hard Disk* 500 GB
- d. Monitor 14 inc
- e. Keyboard
- f. Mouse

2. Perangkat Lunak (*Software*)

Perangkat lunak merupakan komponen penting dalam pembuatan sebuah program atau sistem, perangkat lunak akan membantu dalam mempermudah proses pengerjaan dan pembuatan. Dengan menggunakan perangkat lunak yang tepat efisiensi pengerjaan juga semakin cepat dan baik, perangkat lunak yang digunakan dalam pembuatan sistem prediksi diskon adalah sebagai berikut:

- a. Sistem Operasi *Windows* 10 64 bit
- b. Web Browser
- c. *Jupyter Notebook*
- d. *Python 3*
- e. Microsoft Excel

3.4 Desain *Database*

Berikut ini merupakan *Database* dari program penelitian Analisis sentiment terhadap kinerja kepemimpinan bupati Kabupaten Lamongan periode 2021-2024 menggunakan data komentar menggunakan metode *Naïve bayes Classifier*

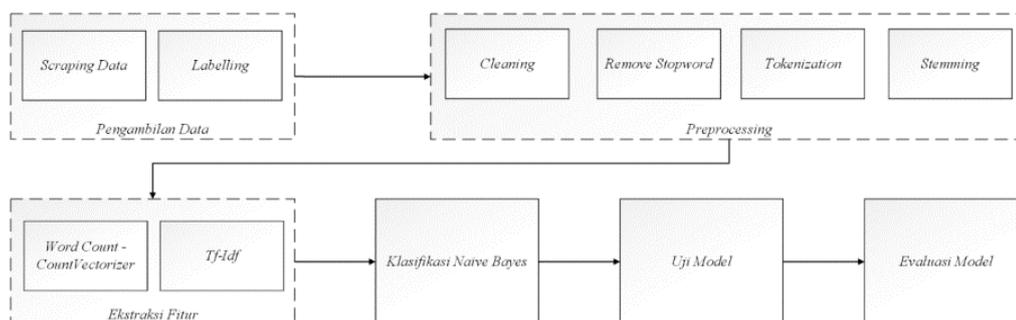
Tabel 3.1 Desain *database*

No	Field	Type	Lenght	Keterangan
1	Id	Int	200	urutan komentar
2	Usia	Int	2	Usia responden
3	Comment	Text	100	Komentar yang diambil
4	Class	Int	2	kategori kelas

3.5 Alur Penelitian

Perancangan alur penelitian adalah langkah penting dalam merencanakan dan mengatur jalannya sebuah tugas akhir atau penelitian. Alur penelitian merinci langkah-langkah yang akan diambil dalam menjalankan penelitian dari awal hingga akhir. Hal ini membantu memastikan bahwa penelitian dilaksanakan dengan cara yang sistematis dan efisien. Perancangan alur penelitian membantu menjalankan penelitian dengan lebih teratur dan terarah. Ini juga membantu menghindari

kebingungan atau kebingungan selama proses penelitian Alur penelitian sebagaimana pada gambar 3.1 berikut.



Gambar 3.1 Alur penelitian

Proses Analisis Sentimen kinerja Bupati sebagai Bupati Lamongan berdasarkan komentar dari masyarakat. Tahap pertama, akan dilakukan pengumpulan data dan pelabelannya, setelah itu data masuk tahapan praproses, kemudian tahap pemodelan, dilanjut tahap pengujian dan yang terakhir adalah tahap evaluasi hasil. Kemudian dilakukan prediksi sentimen terhadap data yang telah melewati seluruh proses tahapan. Berikut adalah perincian dari alur penelitian.:

1. Pengumpulan Data

Tahap Tahap awal penelitian, peneliti melibatkan pengumpulan data menggunakan *Google Form*. Berikut adalah deskripsi lebih lanjut tentang proses yang dijelaskan:

a. Pengumpulan Data Menggunakan *Google Form* :

- Membuat formulir menggunakan *Google Form* yang berisi pertanyaan mengenai kinerja Bupati sebagai Bupati Lamongan.
- Responden yang berasal dari berbagai kecamatan di Kabupaten Lamongan diminta untuk mengisi formulir dengan memberikan tanggapan atau komentar mengenai kinerja tersebut.
- Data yang terkumpul akan berisi komentar berbahasa Indonesia yang menjadi input untuk analisis sentimen.

b. Pembagian Data :

- Data yang terkumpul akan dibagi menjadi dua bagian: data latih (*training data*) dan data uji (*testing data*).

- Biasanya, proporsi data latih dan data uji dipilih berdasarkan perbandingan tertentu, dalam hal ini 80% data latih dan 20% data uji.
- c. Pembagian Kategori dan Sentimen :
- Data latih akan diberi label sentimen, yaitu sentimen positif atau sentimen negatif, secara manual.
 - Membagi komentar-komentar dalam data latih menjadi dua kategori berdasarkan sentimen yang terpancar dari komentar tersebut.
- d. Data Uji
- Data uji merupakan bagian dari data keseluruhan yang belum diberi label sentimen.
 - Data uji ini akan digunakan untuk menguji kinerja model yang telah dilatih pada data latih. Model akan memprediksi sentimen dari komentar-komentar dalam data uji.

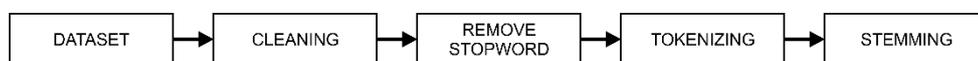
Proses ini merupakan langkah awal yang krusial dalam analisis sentimen. Pengumpulan data yang tepat dan pembagian data menjadi data latih dan data uji yang representatif akan memengaruhi kualitas analisis dan prediksi sentimen yang dihasilkan. Selanjutnya, data latih akan digunakan untuk melatih model klasifikasi sentimen, dan data uji akan digunakan untuk menguji performa model tersebut.

2. Pelabelan Data

Proses penyimpanan data setelah pengumpulan dilakukan dalam bentuk file *xlsx*, diikuti dengan tahap pelabelan untuk menentukan sentimen atau pandangan dari komentar yang telah terkumpul. Agar dapat memudahkan pada tahapan ekstraksi fitur, permodelan hingga tahapan pengklasifikasian dengan metode *naïve bayes*, maka pelabelan data perlu dilakukan. Pada tahap labeling, data dibagi menjadi dua kategori utama, yaitu positif dan negatif. Dalam konteks pelabelan, sentimen positif diganti dengan label 0, sementara sentimen negatif diwakili oleh label 1. Data kemudian dikelompokkan berdasarkan karakteristiknya, di mana kelas positif mengandung komentar-komentar dengan makna positif, persetujuan, dan dukungan, sedangkan kelas negatif mencakup komentar-komentar dengan makna negatif, kritik, atau sikap kontra.

3. *Preprocessing*

Proses *Pre-processing* merupakan langkah yang wajib dilakukan sebelum melanjutkan dengan tahapan analisis topik yang sedang dipelajari. *Pre-processing* data bertujuan untuk mengatur dan menyusun teks dalam bentuk yang seragam dan terstruktur, sehingga siap untuk diolah pada tahap berikutnya. Tujuan dari tahap *Pre-processing* adalah untuk menghilangkan elemen-elemen yang tidak diperlukan atau kata-kata yang tidak memberikan makna yang relevan. Proses ini melibatkan serangkaian langkah yang disesuaikan dengan karakteristik data komentar masyarakat. Setelah melalui tahap *Pre-processing*, diharapkan bahwa data akan menjadi lebih bersih dan terstruktur, yang pada akhirnya diharapkan akan meningkatkan akurasi analisis yang akan dilakukan. Gambaran tahapan *Preprocessing* dapat di lihat pada Gambar 3.2 :



Gambar 3.2 Alur *preprocessing*

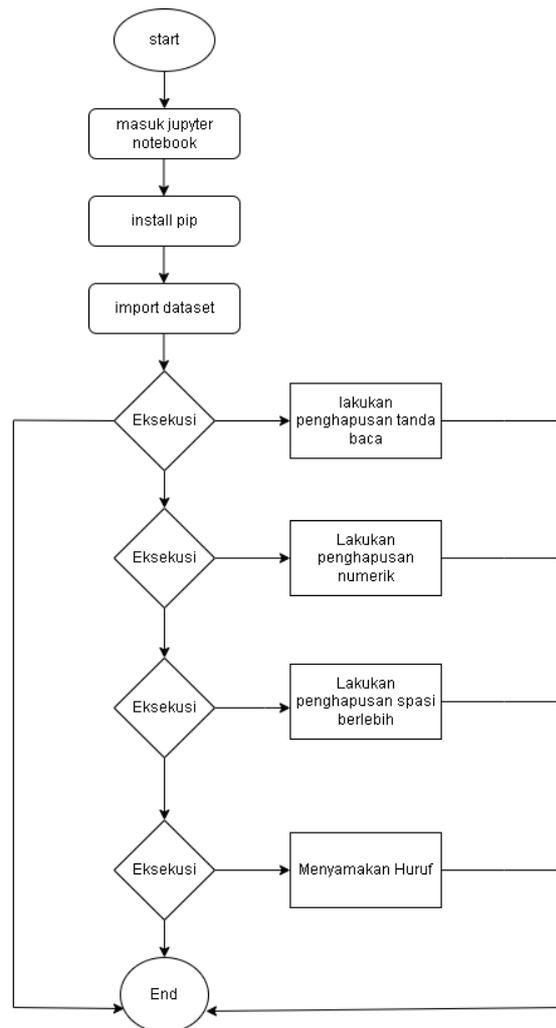
Adapun urutan dan penjelasan dari tahapan *Preprocessing* yang akan dilakukan adalah sebagai berikut:

a. *Cleansing*

Pada tahap ini, proses pembersihan (*cleaning*) dilakukan dengan tujuan untuk menghilangkan elemen-elemen yang tidak relevan dari data komentar. Ini mencakup tindakan seperti menghapus tanda baca, melakukan normalisasi unicode, dan langkah-langkah lainnya yang mendukung pengolahan data yang lebih baik. Proses pembersihan ini diimplementasikan melalui empat tahapan yang saling terkait, dengan tujuan akhir untuk mencapai hasil yang optimal., di antaranya ialah:

- a. Menghapus tanda baca
- b. Menghapus teks numerik (nomor)
- c. Menghapus spasi yang kelebihan
- d. Menyamakan huruf menjadi huruf kecil semua

Adapun *flowchart* dari tahapan *cleansing* adalah sebagai berikut :



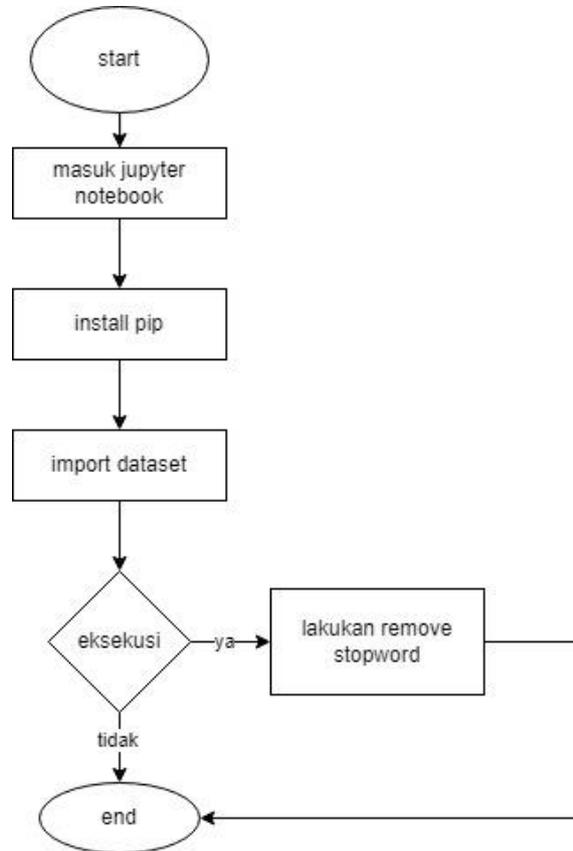
Gambar 3.3 *Flowchart* tahapan *cleansing*

Tabel 3.2 Proses *cleansing*

Sebelum	Sesudah
Bagus dan mulus sekali jalannya. Kami bersama teman2 puas sekali	bagus dan mulus sekali jalannya kami bersama puas sekali
Perawatannya jg tolong diperhatikan pak, terima kasih sudah mendengar aspirasi rakyat.	perawatannya tolong diperhatikan terima kasih sudah mendengar aspirasi rakyat
Jalanan banyak yang berlubang dan kurang perawatan	jalanan banyak yang berlubang dan kurang perawatan

b. *Remove stopwords*

Pada tahap proses ini disebut juga *filtering*. Adapun flowchart dari tahapan *remove stopwords* adalah sebagai berikut :



Gambar 3.4 *Flowchart* tahapan *removed stopwords*

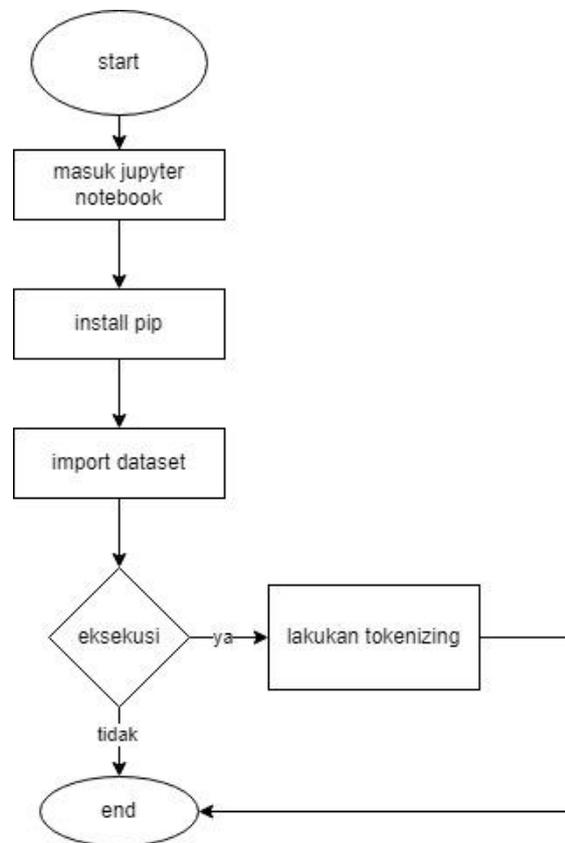
Berikut adalah contoh hasil Proses *Stopword* :

Tabel 3.3 *Proses remove stopwords*

Sebelum	Sesudah
bagus dan mulus sekali jalannya kami bersama puas sekali	bagus mulus jalannya puas
perawatannya tolong diperhatikan terima kasih sudah mendengar aspirasi rakyat	perawatannya tolong diperhatikan terima kasih mendengar aspirasi rakyat
jalanan banyak yang berlubang dan kurang perawatan	jalanan berlubang kurang perawatan

c. *Tokenizing*

Pada Tahapan ini kalimat akan dipecah menjadi kata yang dinamakan token. Adapun *flowchart* dari tahapan *tokenizing* adalah sebagai berikut :



Gambar 3.5 *Flowchart* tahapan *tokenizing*

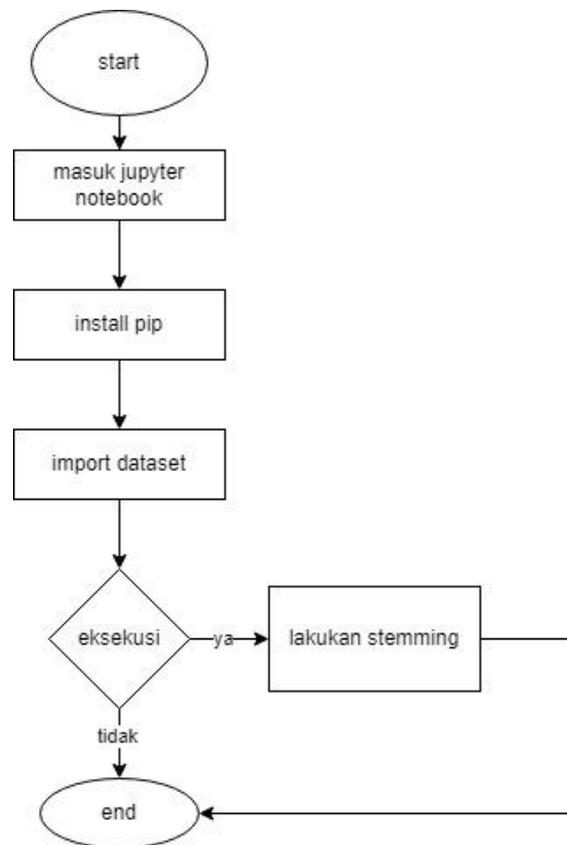
Berikut adalah contoh hasil proses *Tokenizing* :

Tabel 3.4 Proses *tokenizing*

Sebelum	Sesudah
bagus mulus jalannya puas	['bagus', 'mulus', 'jalannya', 'puas']
perawatannya tolong diperhatikan terima kasih mendengar aspirasi rakyat	['perawatannya', 'tolong', 'diperhatikan', 'terima', 'kasih', 'mendengar', 'aspirasi', 'rakyat']
jalanan berlubang kurang perawatan	['jalanan', 'berlubang', 'kurang', 'perawatan']

d. *Stemming*

Stemming adalah tahap mengembalikan sebuah bentuk kata ke kata dasarnya sesuai kamus. Adapun *flowchart* dari tahapan *stemming* adalah sebagai berikut :



Gambar 3.6 *Flowchart* tahapan *stemming*

Berikut adalah contoh hasil proses *Stemming*:

Tabel 3.5 Proses *stemming*

Sebelum	Sesudah
['bagus', 'mulus', 'jalannya', 'puas']	bagus mulus jalan puas
['perawatannya', 'tolong', 'diperhatikan', 'terima', 'kasih', 'mendengar', 'aspirasi', 'rakyat']	rawat tolong perhati terima kasih dengar aspirasi rakyat
['jalanan', 'berlubang', 'kurang', 'perawatan']	jalan lubang kurang rawat

Adapun hasil setelah melewati tahap *Preprocessing* adalah sebagai berikut :

Tabel 3.6 Hasil *preprocessing*

No	Sesudah	Label
0	bagus mulus jalan puas	0
1	rawat tolong perhati terima kasih dengar aspirasi rakyat	0
2	jalan lubang kurang rawat	1

4. Ekstraksi Fitur

Setelah melalui seluruh tahap *Preprocessing*, langkah berikutnya adalah menciptakan fitur-fitur yang akan mempermudah proses klasifikasi. Dalam tahap pembuatan fitur, terdapat dua proses penting yang dilakukan: pembuatan vektor kata (*word vector*) dan pembobotan kata dengan metode Tf-Idf. Vektor kata, juga dikenal sebagai *word vector*, merujuk pada konversi fitur teks menjadi representasi vektor. Proses ini mengubah setiap kalimat atau dokumen yang telah diolah menjadi sejumlah *array*, yang kemudian diatur menjadi matriks. Setiap baris dalam matriks ini mewakili sebuah dokumen, sedangkan kolom matriks merepresentasikan seluruh kata yang ada dalam seluruh teks yang terkumpul.

Setelah mengubah teks menjadi vektor kata, langkah selanjutnya adalah memberikan bobot kepada kata-kata di dalam setiap kalimat atau dokumen dengan menggunakan metode Unigram dan *tf-idf*. Dengan melakukan pembobotan ini, dataset akan siap digunakan dalam tahap pelatihan dengan memanfaatkan perhitungan dalam algoritma *Naïve bayes*. Semua langkah ini membantu dalam mengolah teks menjadi bentuk yang dapat diinterpretasikan oleh algoritma pembelajaran mesin dan membantu dalam mengidentifikasi pola dan karakteristik yang relevan untuk klasifikasi sentimen. Contoh :

(D1) “bagus mulus jalan puas”

(D2) “rawat tolong perhati terima kasih dengar aspirasi rakyat”

(D3) “jalan lubang kurang rawat”

Pada langkah berikutnya, setiap dokumen diubah menjadi vektor dengan elemen-elemen tertentu. Jika suatu kata terdapat dalam dokumen tersebut, nilai elemen akan diberi 1; jika tidak, nilai elemen akan diberi 0. Kemudian, data yang telah diubah menjadi vektor kata tersebut akan dihitung menggunakan rumus TF-IDF, menghasilkan vektor kata dengan bobot yang telah dihitung.

Rumus TF-IDF menggabungkan dua konsep: *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). TF mengacu pada frekuensi kemunculan sebuah kata tertentu dalam dokumen tersebut. Sementara itu, IDF mengukur sejauh mana kata tersebut didistribusikan di seluruh koleksi dokumen. Proses pembobotan kata dimulai dengan menghitung nilai TF (*Term Frequency*) untuk setiap kata dalam setiap dokumen. Dengan menggabungkan informasi tentang frekuensi kemunculan kata dalam dokumen (TF) dengan penyebaran kata di seluruh koleksi dokumen (IDF), nilai TF-IDF memberikan bobot yang lebih tepat kepada kata-kata dalam konteks spesifik dokumen (Purnama et al., n.d.).

Tabel 3.7 Proses awal TF-IDF

	(D1)	(D2)	(D3)
bagus	1	0	0
mulus	1	0	0
jalan	1	0	1
puas	1	0	0
rawat	0	1	1
tolong	0	1	0
perhati	0	1	0
terima	0	1	0
kasih	0	1	0
dengar	0	1	0
aspirasi	0	1	0
rakyat	0	1	0
lubang	0	0	1
kurang	0	0	1

Keterangan :

1. Setelah menghitung frekuensi kata dalam dokumen, langkah berikutnya adalah menghitung *Term Frequency* (TF) dengan rumus berikut ini: $tf = 0,5 * 0,5 \frac{f_{t,d}}{\max(f_{t,d})}$, misalkan pada D1 maka $tf = 0,5 * 0,5(1/1) = 1$
2. Rumus untuk menghitung nilai IDF $\log(\frac{N}{df_t})$, maka $\log 3/1 = 0,477$
3. Rumus menghitung nilai bobot setiap dokumen berdasarkan hasil perhitungan nilai dari $tf-idf$ yaitu dengan $tf * idf$, maka $1 * 0,477 = 0,477$
Setelah nilai TF (*Term Frequency*) diketahui, selanjutnya menghitung DF (*Document Frequency*).

Tabel 3.8 Menentukan *document frequency*

T (Term)	DF(Document Frequency)
bagus	1
mulus	1
jalan	2
puas	1
rawat	2
tolong	1
perhati	1
terima	1
kasih	1
dengar	1
aspirasi	1
rakyat	1
lubang	1
kurang	1

Kemudian menentukan nilai IDF (*Inverse Document Frequency*) dengan mencari nilai \log dari D (jumlah dokumen) dibagi nilai DF (*Document Frequency*).

Tabel 3.9 Menentukan *inverse document frequency*

T (Term)	DF(Document Frequency)	D/DF	IDF (Inverse Document Frequency)
bagus	1	3	$\log 3 = 0,477$
mulus	1	3	$\log 3 = 0,477$
jalan	2	1,5	$\log 1,5 = 0,176$
puas	1	3	$\log 3 = 0,477$
rawat	2	1,5	$\log 1,5 = 0,176$
tolong	1	3	$\log 3 = 0,477$
perhati	1	3	$\log 3 = 0,477$
terima	1	3	$\log 3 = 0,477$
kasih	1	3	$\log 3 = 0,477$
dengar	1	3	$\log 3 = 0,477$
aspirasi	1	3	$\log 3 = 0,477$
rakyat	1	3	$\log 3 = 0,477$
lubang	1	3	$\log 3 = 0,477$
kurang	1	3	$\log 3 = 0,477$

Setelah nilai IDF (*Inverse Document Frequency*) diketahui, selanjutnya menghitung nilai TF-IDF.

Tabel 3.10 Menghitung *TF-IDF*

Q	TF			DF	D/DF	IDF	IDF+1	W=TF*(IDF)		
	D1	D2	D3					D1	D2	D3
bagus	1	0	0	1	3	0,477	1,477	0,477	0	0
mulus	1	0	0	1	3	0,477	1,477	0,477	0	0
jalan	1	0	1	2	1,5	0,176	1,176	0,176	0	0,176
puas	1	0	0	1	3	0,477	1,477	0,477	0	0
rawat	0	1	1	2	1,5	0,176	1,176	0	0,176	0,176
tolong	0	1	0	1	3	0,477	1,477	0	0,477	0
perhati	0	1	0	1	3	0,477	1,477	0	0,477	0

Tabel 3.10 Lanjutan

terima	0	1	0	1	3	0,477	1,477	0	0,477	0
kasih	0	1	0	1	3	0,477	1,477	0	0,477	0
dengar	0	1	0	1	3	0,477	1,477	0	0,477	0
aspirasi	0	1	0	1	3	0,477	1,477	0	0,477	0
rakyat	0	1	0	1	3	0,477	1,477	0	0,477	0
lubang	0	0	1	1	3	0,477	1,477	0	0	0,477
kurang	0	0	1	1	3	0,477	1,477	0	0	0,477
Nilai bobot setiap dokumen								1,607	3,515	1,306

5. Implementasi Klasifikasi *Naïve bayes*

Perhitungan klasifikasi *Naïve bayes* memperhitungkan frekuensi setiap kata yang muncul. Pada tabel berikut D1 dan D2 merupakan kelas positif serta D3 merupakan kelas negatif, didapatkan hasil W untuk kelas positif adalah $W(+)=5,122$ dan hasil W bagi kelas negatif adalah $W(-)=1,306$ dan jumlah seluruh nilai idf pada semua kelas adalah 18,2. Berikut merupakan bobot dari masing-masing kelas :

Tabel 3.11 Pembobotan kalimat

Q	W Positif	W Negatif
bagus	0,477	0
mulus	0,477	0
jalan	0,176	0,176
puas	0,477	0
rawat	0,176	0,176
tolong	0,477	0
perhati	0,477	0
terima	0,477	0
kasih	0,477	0
dengar	0,477	0
aspirasi	0,477	0
rakyat	0,477	0
lubang	0	0,477
kurang	0	0,477
Bobot	5,122	1,306

Langkah selanjutnya mencari probabilitas *prior* dengan persamaan 2.5 :

$$P(c1) = P(c1) = \frac{2}{a} = 0,666$$

$$P(c2) = P(c2) = \frac{1}{a} = 0,333$$

Setelah probabilitas diketahui, selanjutnya menghitung *Likelihood* pada setiap kata dengan persamaan 2.6, berikut contoh perhitungannya.

$$\text{Bagus}(c1) = \frac{W_{ct+1}}{(\sum W' \in VW_{ct'}) + B'} = \frac{0,477+1}{5,122+18,2} = \frac{1,477}{23,322} = 0,063$$

$$\text{Bagus}(c2) = \frac{W_{ct+1}}{(\sum W' \in VW_{ct'}) + B'} = \frac{0+1}{1,306+18,2} = \frac{1}{19,506} = 0,051$$

Hasil perhitungan bisa dilihat pada tabel 3.12 berikut :

Tabel 3.12 Hasil perhitungan *likelihood*

P(w c)	Positif	Negatif
bagus	0.063	0,051
mulus	0.063	0,051
jalan	0,05	0,05
puas	0.063	0,051
rawat	0,05	0,05
tolong	0.063	0,051
perhati	0.063	0,051
terima	0.063	0,051
kasih	0.063	0,051
dengar	0.063	0,051
aspirasi	0.063	0,051
rakyat	0.063	0,051
lubang	0,051	0.063
kurang	0,051	0.063

Lalu hitung nilai V kelas positif dan negatif pada kalimat untuk mengetahui apakah kalimat tersebut masuk pada kelas positif atau negatif, contoh pada kalimat “jalan rakyat kurang” berikut perhitungannya :

$$\begin{aligned} V_{\text{map}}(c1) &= P(\text{jalan}) * P(\text{rakyat}) * P(\text{kurang}) * P(C) \\ &= 0,05 * 0,063 * 0,051 * 0,666 \\ &= 0,0001069 \end{aligned}$$

$$\begin{aligned}
 V_{\text{map}}(c2) &= P(\text{jalan}) * P(\text{rakyat}) * P(\text{kurang}) * P(C) \\
 &= 0,05 * 0,051 * 0,063 * 0,333 \\
 &= 0,0000534
 \end{aligned}$$

Setelah itu menentukan *determinan* kalimat tersebut dengan membandingkan $V_{\text{map}}(c1)$ dan $V_{\text{map}}(c2)$, pada kalimat “jalan rakyat kurang” nilai $V_{\text{map}}(c2)$ yang merupakan kelas negatif lebih tinggi dari pada $V_{\text{map}}(c1)$ yang merupakan kelas positif, maka *determinan* dari kalimat tersebut adalah positif

6. Uji Model

Setelah melalui proses pelatihan data, tahap berikutnya adalah melakukan uji model. Uji model dilakukan untuk mengevaluasi kinerja model yang telah dilatih. Sejumlah data diambil dari data pelatihan, biasanya sekitar 20% dari total data pelatihan. Proses pengambilan data dilakukan secara acak dengan menggunakan pustaka atau *library Python* yang sesuai. Setelah uji model selesai, hasilnya akan menunjukkan seberapa besar akurasi dari metode yang telah diterapkan pada data yang tidak pernah dilihat sebelumnya.

7. Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja model yang telah dibuat. Proses evaluasi melibatkan berbagai metrik termasuk akurasi, serta menggunakan matriks konfusi (*confusion matrix*) dan tabel akurasi serta presisi untuk setiap model. Setelah data pengujian (test) dijalankan melalui model yang telah dihasilkan oleh data pelatihan (train), prediksi kelas akan dihasilkan dari data pengujian. Prediksi kelas ini akan dibandingkan dengan kelas sebenarnya dari data pengujian yang telah disembunyikan sebelumnya. Hal ini memungkinkan perhitungan nilai akurasi, presisi, *Recall*, dan *F1-Score*.

Akurasi mewakili persentase dari total sentimen yang berhasil dikenali dengan benar oleh model. Perhitungan akurasi dilakukan dengan membagi jumlah sentimen yang terklasifikasikan dengan benar dengan total jumlah data pengujian. Akurasi adalah salah satu metrik penting untuk menilai seberapa baik performa model dalam mengklasifikasikan data. Rumus akurasi adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (3.1)$$

Presisi adalah rasio antara jumlah data yang relevan yang berhasil diidentifikasi dan jumlah keseluruhan data yang diidentifikasi sebagai positif. Formula untuk presisi adalah sebagai berikut :

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

Recall merupakan metrik evaluasi yang mengukur sejauh mana model mampu mengidentifikasi data yang relevan dari keseluruhan data yang benar relevan. Dengan kata lain, *Recall* mengukur seberapa baik model dalam menangkap data positif yang sebenarnya. Rumus *Recall* adalah berikut :

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

Recall memberikan pandangan tentang sejauh mana model dapat mengidentifikasi dan mendeteksi data yang sebenarnya positif. Semakin tinggi nilai *Recall*, semakin sedikit data positif yang terlewatkan oleh model.

F1-Score adalah parameter tunggal yang mengukur keseimbangan antara presisi (*precision*) dan *Recall* dalam suatu model. *F1-Score* berguna untuk memberikan gambaran keseluruhan tentang kinerja model dalam mengklasifikasikan data, dengan mempertimbangkan baik jumlah data yang relevan yang diidentifikasi (*precision*) maupun kemampuan model dalam mendeteksi data yang relevan (*Recall*). Rumus *F1-Score* adalah sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision * recall}{Precision + recall} \quad (3.4)$$