

**PENERAPAN METODE KLASIFIKASI *K-NEAREST NEIGHBOR* DAN  
*LEXICON BASED* UNTUK ANALISIS SENTIMEN PUBLIK  
TERHADAP PEMINDAHAN IBU KOTA NEGARA DENGAN  
EKSTRAKSI FITUR *TF-IDF***

**TUGAS AKHIR**

Sebagai syarat untuk memperoleh gelar sarjana S-1 di Program Studi Informatika, Jurusan  
Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran”

Yogyakarta



Disusun oleh:

Rifqi Maulana

123200128

**PROGRAM STUDI INFORMATIKA**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**

**YOGYAKARTA**

**2024**

## HALAMAN PENGESAHAN PEMBIMBING

**PENERAPAN METODE KLASIFIKASI *K-NEAREST NEIGHBOR* DAN  
*LEXICON BASED* UNTUK ANALISIS SENTIMEN PUBLIK TERHADAP  
PEMINDAHAN IBU KOTA NEGARA DENGAN EKSTRAKSI FITUR *TF-IDF***

Disusun Oleh:

Rifqi Maulana

123200128

Telah diperiksa dan disetujui oleh pembimbing untuk diseminarkan  
pada tanggal : .....

Menyetujui,  
Pembimbing

Rifki Indra Perwira, S.Kom., M.Eng.

NIP. 19830708 202121 1001

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>1</b>
<b>HALAMAN PENGESAHAN PEMBIMBING.....</b>	<b>2</b>
<b>DAFTAR ISI .....</b>	<b>3</b>
<b>DAFTAR GAMBAR .....</b>	<b>5</b>
<b>DAFTAR TABEL .....</b>	<b>6</b>
<b>BAB 1 .....</b>	<b>7</b>
<b>PENDAHULUAN.....</b>	<b>7</b>
1.1 Latar Belakang .....	7
1.2 Rumusan Masalah.....	9
1.3 Batasan Masalah .....	9
1.4 Tujuan Penelitian.....	9
1.5 Manfaat Penelitian .....	9
1.6 Metodologi Penelitian dan Pengembangan Sistem.....	10
1.6.1 Metode Penelitian .....	10
1.6.2 Metode Pengembangan Sistem .....	11
1.7 Sistematika Penulisan .....	12
<b>BAB II .....</b>	<b>13</b>
<b>TINJAUAN LITERATUR.....</b>	<b>13</b>
2.1 Analisis Sentimen .....	13
2.2 Analisis <i>Term Frequency Inverse Document Frequency (TF-IDF)</i> .....	14
2.3 <i>X (Twitter)</i> .....	14
2.4 <i>Python</i> .....	15
2.5 <i>Web Scraping</i> .....	15
2.6 <i>SMOTE (Synthetic Minority Oversampling Technique)</i> .....	15
2.7 <i>Text Preprocessing</i> .....	16
2.7.1 <i>Text Cleaning</i> .....	16
2.7.2 <i>Tokenization</i> .....	16
2.7.3 <i>Pembakuan Kata</i> .....	16
2.7.4 <i>Negation Handling</i> .....	16
2.7.5 <i>Stopwords Removal</i> .....	16
2.7.6 <i>Stemming</i> .....	17
2.8 <i>Lexicon Based</i> .....	17
2.9 <i>K-Nearest Neighbors (KNN)</i> .....	19

2.10 Validasi dan Pengujian .....	20
2.11 Studi Pustaka .....	23
<b>BAB III</b> .....	25
<b>METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM</b> .....	25
3.1 Metodologi Penelitian .....	25
3.1.1 Pengumpulan Data .....	26
3.1.2 <i>Preprocessing</i> .....	27
3.1.3 <i>Sentiment Labelling</i> .....	33
3.1.4 Pembobotan <i>TF-IDF</i> .....	33
3.1.5 Penerapan Teknik <i>SMOTE</i> .....	39
3.1.6 Analisis Sentimen dengan Model <i>KNN</i> .....	45
3.1.7 Pembuatan Model Sentimen .....	46
3.1.8 Pengujian .....	47
3.2 Metode Pengembangan Sistem .....	47
3.2.1 <i>Requirements Analysis</i> .....	47
3.2.2 <i>System and Software Design</i> .....	48
3.2.3 <i>Implementation</i> .....	55
3.2.4 <i>System Testing</i> .....	55
<b>DAFTAR PUSTAKA</b> .....	56

## DAFTAR GAMBAR

Gambar 3.1 Metodologi Penelitian .....	25
Gambar 3.2 Flowchart Preprocessing .....	27
Gambar 3.3 Flowchart Text Cleaning .....	28
Gambar 3.4 Flowchart Pembakuan Kata .....	29
Gambar 3.5 Flowchart Negation Handling .....	30
Gambar 3.6 Flowchart Stopwords Removal .....	31
Gambar 3.7 Flowchart Stemming .....	32
Gambar 3.8 Flowchart TF-IDF .....	34
Gambar 3.9 Flowchart Jarak Nearest Neighbor SMOTE .....	39
Gambar 3.10 Flowchart Membuat Data Sintesis Baru SMOTE .....	41
Gambar 3.11 Flowchart Klasifikasi Model KNN .....	45
Gambar 3.12 Flowchart Pembuatan Model Sentimen .....	46
Gambar 3.13 Arsitektur Sistem .....	49
Gambar 3.14 Rancangan Halaman Beranda .....	50
Gambar 3.15 Rancangan Halaman Prediksi Sentimen .....	51
Gambar 3.16 Rancangan Halaman Dataset .....	51
Gambar 3.17 Rancangan Halaman Evaluasi .....	52
Gambar 3.18 Rancangan Halaman Sampel Tweet .....	53
Gambar 3.19 Rancangan Halaman Preprocessing .....	53
Gambar 3.20 Rancangan Halaman Sentimen Leksikon .....	54

## DAFTAR TABEL

Tabel 2.1 Confusion Matrix .....	20
Tabel 2.2 Studi Pustaka .....	23
Tabel 2.3 Lanjutan Studi Pustaka .....	24
Tabel 3.1 Kata Kunci.....	26
Tabel 3.2 Contoh Dokumen Perhitungan Term Frequency .....	35
Tabel 3.3 Hasil Term Frequency .....	35
Tabel 3.4 Lanjutan Hasil Term Frequency .....	36
Tabel 3.5 Hasil IDF .....	36
Tabel 3.6 Lanjutan Hasil IDF .....	37
Tabel 3.7 Hasil TF-IDF .....	38
Tabel 3.8 Hasil Jarak Nearest Neighbor SMOTE .....	40
Tabel 3.9 Hasil Data Sintesis SMOTE .....	42
Tabel 3.10 Dokumen Data Sintesis SMOTE .....	43
Tabel 3.11 Hasil Perubahan TF-IDF Setelah SMOTE .....	44
Tabel 3.12 Spesifikasi Hardware.....	48
Tabel 3.13 Software .....	48
Tabel 3.14 Rancangan Pengujian Black Box .....	55

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Keputusan Presiden Indonesia untuk memindahkan ibu kota negara ke luar Pulau Jawa menjadi salah satu proyek strategis yang tertuang dalam Rencana Pembangunan Jangka Menengah Nasional Tahun Anggaran 2020-2024. Pada 26 Agustus 2019, Presiden yang pada saat itu sedang menjabat mengumumkan ibu kota negara baru ini akan dibangun tepatnya di Kabupaten Penajam Paser Utara dan sebagian Kutai Kartanegara, Kalimantan Timur (Hadi, 2020). Pemindahan ibu kota negara Indonesia merupakan salah satu proyek mega infrastruktur yang paling signifikan dalam beberapa dekade terakhir. Keputusan berani ini tidak hanya memicu perdebatan di kalangan para pengambil kebijakan, tetapi juga memicu beragam reaksi dari masyarakat luas.

Transisi menuju ibu kota baru negara Indonesia merupakan hal yang sangat sensitif sehingga banyak dibicarakan di media sosial, tidak terkecuali pada media sosial *X* (sebelumnya dikenal dengan nama *Twitter*). *X* merupakan media sosial yang seringkali menjadi pusat *trending* mengenai isu di dunia baik itu skala nasional maupun internasional yang dijadikan warganet sebagai media untuk menyuarakan opini terkait sentimen terhadap apa pun yang terkini diperbincangkan di jejaring sosial yang begitu kompleks (Sandi et al., 2023). Sosial media *X* menjadi salah satu platform yang sering digunakan oleh masyarakat Indonesia untuk mengekspresikan pendapat dan respons terhadap peristiwa-peristiwa penting seperti pemindahan ibu kota negara baru. Indonesia menempati posisi ke-5 di dunia sebagai pengguna media sosial *X* terbanyak menurut data dari *Statista* pada bulan Januari 2023. *Statista* merupakan salah satu situs *web* yang menyediakan data statistik yang dikenal di seluruh dunia.

Pada penelitian ini akan dilakukan analisa sentimen publik mengenai pemindahan ibu kota negara Indonesia di media sosial *X*. Teknik yang akan digunakan adalah *web scraping* untuk mengumpulkan data teks dari media sosial *X*. Penelitian ini akan dapat mengakses data teks yang mencakup berbagai macam opini, komentar, dan persepsi dari pengguna *X* terkait topik pemindahan ibu kota negara. Hal ini akan memungkinkan penulis untuk memiliki isi *dataset* dalam bentuk teks dan memadai untuk dilakukan analisis sentimen menggunakan metode klasifikasi *K-Nearest Neighbors (KNN)* dengan pelabelan sentimen *Lexicon Based*.

Metode *KNN* memberikan hasil yang kurang baik pada proses klasifikasi data jika terdapat *noise* atau informasi tambahan yang tidak berarti (Pratomo et al., 2021), metode tersebut juga sensitif terhadap data tidak seimbang (*imbalance data*) yang dapat menyebabkan kesalahan klasifikasi pada kelas minoritas (Pramayasa et al., 2023). Oleh karena itu, terdapat beberapa penelitian yang menyebutkan bahwa performansi pada metode *machine learning* dapat menghasilkan performa yang baik ketika dikombinasikan dengan ekstraksi fitur yang tepat (Pratomo et al., 2021). Teknik *resampling* juga dapat menjadi salah satu solusi dalam menangani kasus data yang tidak seimbang (Pramayasa et al., 2023). Penggabungan tersebut menawarkan pendekatan yang memadai dalam memproses dan menganalisis data teks untuk mengekstraksi informasi sentimen. Penerapan metode ini membuat kita dapat mengidentifikasi pandangan, sikap, dan respon yang diberikan publik terkait dengan topik pemindahan ibu kota negara tersebut.

Berdasarkan paparan sebelumnya, beberapa solusi yang penulis gunakan adalah metode klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen berbasis *lexicon*, penyeimbang kelas dengan *SMOTE*, serta ekstraksi fitur *TF-IDF*. Tahapan awal pada penelitian ini adalah *data preprocessing* bertujuan agar data yang nanti diproses lebih terstruktur, selanjutnya proses pelabelan dan ekstraksi fitur menggunakan metode berbasis *lexicon* dan *TF-IDF*, tidak lupa juga penyeimbangan kelas dengan *SMOTE*, lalu terakhir yaitu tahap klasifikasi. Beberapa teknologi yang telah disebutkan sebelumnya dipilih karena sudah dilakukannya peninjauan pustaka dari beberapa metode sejenis sebelumnya. Penelitian yang dilakukan oleh Setyo Adji Pratomo, dkk pada tahun 2021, dengan judul “Analisis Sentimen Pengaruh Kombinasi Ekstraksi Fitur TF-IDF dan Lexicon Pada Ulasan Film Menggunakan Metode KNN” berkesimpulan bahwa penggabungan fitur ekstraksi *TF-IDF* dengan leksikon *SentiWordNet* memiliki hasil akurasi 73.31%. Lalu, penelitian yang dilakukan oleh Azhar pada tahun 2018, dengan judul “Analisis Kinerja Algoritma Naïve Bayes dan K-Nearest Neighbor Pada Sentimen Analisis dengan Pendekatan Lexicon di Media Twitter”. *Dataset* yang digunakan merupakan data dari sosial media *X (Twitter)* dengan menggunakan *Twitter API*. Proses *Natural Language Processing* yang digunakan adalah *case folding*, *filtering*, *tokenizing*, normalisasi, *stopwords*, dan *stemming*. Hasilnya nilai *KNN* pada  $k=5$  dengan tingkat akurasi mencapai 77%. Terakhir, penelitian yang dilakukan oleh Muhammad Rayhan Elfansyah, dkk pada tahun 2024 dengan judul “Perbandingan Metode K-Nearest Neighbor (KNN) Dan Naïve Bayes Terhadap Analisis Sentimen Pada Pengguna E-Wallet Aplikasi Dana Menggunakan Fitur Ekstraksi TF-IDF” berkesimpulan bahwa metode *KNN* dan *Naïve Bayes* memiliki akurasi yang berbeda berdasarkan sumber label data. Pada data yang diberi label model *lexicon*, akurasi *KNN* mencapai 78% dan *Naïve Bayes* 74%.

Berdasarkan latar belakang yang telah dipaparkan, diharapkan penelitian ini dapat memberikan kontribusi dalam memahami bagaimana mengoptimalkan metode klasifikasi *K-Nearest Neighbors* dengan melakukan beberapa penyesuaian yang tepat dari mulai *preprocessing* sampai ke tahap klasifikasi sentimen untuk memahami reaksi publik, khususnya warganet di *X (Twitter)*, terhadap pemindahan ibu kota negara Indonesia, apakah lebih cenderung positif, netral, atau negatif. Hal ini dilakukan dengan mengintegrasikan data teks yang telah dibersihkan untuk diproses lebih lanjut oleh teknik pelabelan sentimen berbasis *lexicon* yang telah dikustomisasi, serta mencari nilai  $k$  yang optimal sebagai variabel penting dalam pembuatan model klasifikasi *KNN*, sebelum itu dilakukan penyeimbangan kelas dengan *SMOTE*, serta ekstraksi fitur *TF-IDF* untuk menghasilkan model analisis sentimen dengan akurasi tinggi. Selain itu, diharapkan penelitian ini dapat memberikan kontribusi dalam memahami bagaimana tanggapan serta dinamika sosial yang muncul saat terjadi perubahan besar, seperti pemindahan ibu kota. Hasil penelitian ini diharapkan dapat menjadi referensi bagi penelitian serupa di masa depan.



## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang sebelumnya, masalah yang dapat dirumuskan mencakup bagaimana tingkat akurasi dari analisis sentimen publik terkait pemindahan ibu kota negara Indonesia dengan model berbasis *KNN*.

## 1.3 Batasan Masalah

Agar masalah yang diteliti menghasilkan sasaran yang jelas, maka dibuatlah batasan masalah untuk menghindari adanya perluasan pembahasan ke depannya sebagai berikut:

1. Data penelitian yang digunakan merupakan data yang dihasilkan dari *web scraping* pada unggahan teks di media sosial *X (Twitter)*, mengenai sentimen publik atas keputusan Presiden terkait pemindahan ibu kota negara.
2. Dataset yang digunakan dalam penelitian berjumlah total 5341 baris unggahan teks memakai format *.csv*.
3. Kategori sentimen dibagi menjadi 3 yaitu positif, negatif, serta netral.
4. Leksikon yang digunakan adalah leksikon berbahasa Indonesia yaitu *InSet (Indonesia Sentiment Lexicon)* dari literatur yang disusun oleh Fajri Koto dan Gemala Y. Rahmanningtyas pada tahun 2017 dengan judul "InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs". Leksikon *InSet* terdiri atas 3,609 kata positif dan 6,609 kata negatif dengan bobot antara -5 sampai +5.

## 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah mengetahui nilai akurasi dari model analisis sentimen berbasis *KNN* berdasarkan unggahan teks di media sosial *X (Twitter)* mengenai keputusan Presiden terkait pemindahan ibu kota negara.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat untuk memberikan kontribusi dalam pengembangan metodologi analisis sentimen berbasis teks, khususnya dalam penerapan algoritma klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen berbasis *lexicon* dalam konteks bahasa Indonesia. Penelitian ini juga diharapkan tidak hanya bermanfaat secara ilmiah sebagai referensi bagi penelitian-penelitian selanjutnya yang menggunakan pendekatan serupa, penelitian ini juga mendukung pengembangan algoritma *Natural Language Processing (NLP)* untuk teks berbahasa Indonesia. Selain itu, penelitian ini juga merupakan bagian dari pemenuhan salah satu syarat kelulusan strata satu (S1) Program Studi Informatika Fakultas Teknik Industri.

## 1.6 Metodologi Penelitian dan Pengembangan Sistem

Metode penelitian ini menggunakan metode penelitian kuantitatif. Metode penelitian kuantitatif merupakan penelitian empiris dimana data dalam bentuk sesuatu yang dapat dihitung atau angka (Punch, 1988). Berikut merupakan tahapan-tahapan penelitian yang dilakukan:

### 1.6.1 Metode Penelitian

#### 1. Pengumpulan Data

Data yang akan digunakan dalam penelitian ini merupakan data yang diperoleh dari proses *web scraping* dari media sosial *X (Twitter)*. Data yang telah didapatkan akan disimpan dalam format *.csv*.

#### 2. Preprocessing Data

*Text Preprocessing* memiliki tujuan untuk mengubah bentuk data yang tidak terstruktur menjadi data terstruktur. Pada tahapan ini, akan dilakukan *preprocessing* data mencakup *case folding*, *text cleaning*, pembakuan kata dengan kamus slang atau tidak baku, *negation handling*, *stopwords removal*, dan *stemming*.

#### 3. Sentiment Labelling

Data yang telah melalui tahap *preprocessing*, kemudian dilakukan pelabelan sentimen terhadap data tersebut yang mencakup 3 kelas sentimen yaitu positif, negatif, dan netral. Pelabelan sentimen ini dilakukan menggunakan kamus *Lexicon*.

#### 4. Ekstraksi Fitur

Tahapan ini akan membobotkan kata menggunakan metode *TF-IDF* untuk mengubah data yang sebelumnya berbentuk teks atau *string* ke bentuk matriks numerik.

#### 5. Klasifikasi KNN

Pembuatan model *KNN* dan menguji performanya untuk mengetahui hasil analisis sentimen menggunakan salah satu *python library* yaitu *scikit-learn*.

#### 6. Pengujian

Tahapan ini melakukan pengujian kinerja model yang telah dikembangkan dengan menggunakan *confusion matrix* untuk menemukan nilai akurasi, presisi, dan *recall*.

### 1.6.2 Metode Pengembangan Sistem

Metode *Waterfall* merupakan salah satu model *SDLC* (*Software Development Life Cycle*) yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Metode ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (*maintenance*) dan dilakukan secara bertahap (Abdul Wahid, 2020). Tahapan metode *waterfall* adalah sebagai berikut:

1. *Requirements Analysis and Definition*

Merupakan tahapan awal yang melibatkan identifikasi dan pemahaman yang mendalam terhadap kebutuhan. Tujuan utamanya yaitu mengumpulkan persyaratan fungsional dan non-fungsional yang nantinya akan menjadi dasar dari pengembangan sistem.

2. *System and Software Design*

Tahapan perancangan sistem ini mengalokasikan kebutuhan-kebutuhan sistem pada perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program.

4. *System Testing*

Merupakan tahap pengujian terhadap sistem yang telah dibuat yang bertujuan untuk mengetahui apakah sistem yang dibuat udah siap digunakan atau belum.

## 1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut:

- |         |   |
|---------|---|
| BAB I   | <b>PENDAHULUAN</b><br>Pada bagian pendahuluan membahas mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.  |
| BAB II  | <b>TINJAUAN PUSTAKA</b><br>Tinjauan pustaka merupakan bagian yang memuat mengenai dasar teori yang digunakan untuk analisis serta perancangan sistem dan juga implementasi pada penelitian ini. Selain itu juga digunakan sebagai bahan referensi serta pondasi untuk memperkuat argumentasi pada penelitian ini. |
| BAB III | <b>METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM</b><br>Pada bab ini membahas mengenai metodologi penelitian, analisis sistem dan perancangan sistem analisis sentimen.   |
| BAB IV  | <b>HASIL, PENGUJIAN, DAN PEMBAHASAN</b><br>Bab ini menyajikan hasil dari penelitian yang berisi hasil implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Selain itu berisi pengujian terhadap hasil penelitian beserta pembahasannya.   |
| BAB V   | <b>KESIMPULAN DAN SARAN</b><br>Bab ini berisi kesimpulan dari hasil penelitian serta saran yang diajukan oleh penulis untuk pengembangan pada penelitian selanjutnya.   |

## **BAB II**

### **TINJAUAN LITERATUR**

#### **2.1 Analisis Sentimen**

Analisis sentimen adalah bidang studi yang menganalisis pendapat, sentimen, evaluasi, penilaian, sikap dan emosi seseorang terhadap sebuah produk, organisasi, individu, masalah, peristiwa atau topik (Liu, 2012). Tujuan dari analisis sentimen yaitu untuk memahami opini, perasaan, serta pandangan yang terkandung pada teks atau data unstruktural lainnya. Pengaruh dan manfaat dari analisis sentimen menyebabkan penelitian mengenai analisis sentimen berkembang pesat, serta kurang lebih 20-30 perusahaan di Amerika berfokus pada layanan analisis sentimen (Liu, 2012). Manfaat sentimen analisis dalam dunia usaha antara lain untuk melakukan pemantauan terhadap suatu produk. Secara cepat dapat digunakan sebagai alat bantu untuk melihat respon masyarakat terhadap suatu produk sehingga dapat diambil langkah strategis berikutnya. Garis besar analisis sentimen itu sendiri bertujuan untuk mengekstrak atribut dan komponen dari beberapa komentar yang ada di media sosial dan sehingga dapat menentukan beberapa kelas positif, negatif dan netral (Permatasari et al., 2021).

Pada umumnya, sentimen analisis merupakan klasifikasi tetapi kenyataannya tidak semudah proses kualifikasi biasa karena terkait penggunaan bahasa, dimana terdapat ambiguitas dalam penggunaan kata serta perkembangan bahasa itu sendiri.

Menurut Liu (2012), analisis sentimen memiliki beberapa tahap untuk melakukan analisis sentimen, yaitu:

##### **1. Level Dokumen**

Level dokumen menganalisis satu dokumen penuh dan mengklasifikasikan dokumen tersebut memiliki sentimen positif atau negatif. Level analisis ini berasumsi bahwa keseluruhan dokumen hanya berisi opini tentang satu entitas saja. Level analisis ini tidak cocok diterapkan pada dokumen yang membandingkan lebih dari satu entitas.

##### **2. Level Kalimat**

Level kalimat menganalisis satu kalimat dan menentukan tiap kalimat sentimen bernilai positif, netral, atau negatif. Sentimen netral berarti kalimat tersebut bukan opini.

##### **3. Level Aspek**

Level aspek tidak melakukan analisis pada konstruksi bahasa (dokumen, paragraf, kalimat, klausa, atau frasa) melainkan melakukan langsung pada opini itu sendiri. Hal ini didasari bahwa opini terdiri dari sentimen (positif dan negatif) dan target dari opini tersebut. Tujuan level analisis ini adalah untuk menemukan sentimen entitas pada tiap aspek yang dibahas.

## 2.2 Analisis *Term Frequency Inverse Document Frequency (TF-IDF)*

*TF-IDF* atau *Term Frequency Inverse Document Frequency* merupakan metode pembobotan dengan menggabungkan metode *TF* dan *IDF*, metode ini memberikan bobot hubungan suatu kata terhadap dokumen (Wahyuni et al., 2017).

Proses frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen.

**Nilai *TF* dapat dihitung dengan rumus:**

$$TF = \frac{\text{jumlah kata terpilih}}{\text{jumlah kata}} \dots\dots\dots(2.1)$$

**Nilai *IDF* dapat dihitung dengan rumus:**

$$IDF = \frac{\text{jumlah dokumen}}{\text{jumlah frekuensi kata terpilih}} \dots\dots\dots(2.2)$$

**Nilai *TF-IDF*:**

$$TFIDF = TF \times IDF \dots\dots\dots(2.3)$$

## 2.3 *X (Twitter)*

*X* adalah sebuah situs *web* yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa *microblog* sehingga memungkinkan penggunaanya untuk mengirim dan membaca pesan dengan sebutan *tweet* atau kicauan (Akbar et al., 2013).

*Microblog* adalah jenis alat komunikasi daring dengan manfaat agar pengguna dapat memperbarui status tentang mereka yang sedang memikirkan dan melakukan sesuatu, apa pendapat mereka tentang suatu objek atau fenomena tertentu. *Tweet* atau kicauan adalah teks tulisan hingga 140 karakter (atau lebih jika berlangganan fitur khusus pada platform tersebut) yang ditampilkan pada halaman profil pengguna. *Tweet* bisa dilihat secara publik atau dapat dibatasi pengiriman pesan ke daftar pengguna lain tertentu saja. Pengguna dapat melihat *tweet* pengguna lain yang dikenal dengan sebutan pengikut atau *followers* (Ramadhon, 2020).

## 2.4 Python

*Python* adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek dengan semantik dinamis. Struktur data bawaan tingkat tinggi, dikombinasikan dengan pengetikan dan pengikatan dinamis membuatnya sangat menarik untuk *Rapid Application Development*, serta digunakan sebagai bahasa skrip atau lem untuk menghubungkan komponen yang ada bersama-sama.

Sintaks *python* yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket yang mendorong modularitas program dan penggunaan kembali kode. Penerjemah *python* dan perpustakaan standar yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua *platform* utama, dan dapat didistribusikan secara bebas (*Python – Wikipedia, 2017*).

## 2.5 Web Scraping

*Web scraping* merupakan sebuah teknik untuk mendapatkan informasi dari situs web secara otomatis tanpa harus menyalinnya secara manual (Ayani et al., 2019). Teknik *web scraping* memungkinkan konten utama yang terdapat pada situs dapat diekstraksi, dihimpun, kemudian dapat diproses. *Web scraping* akan melakukan ekstraksi data pada *World Wide Web* (*www*), kemudian data yang didapat akan disimpan pada *file system* atau basis data yang nantinya bisa diambil kembali atau dianalisis (Setiawan et al., 2020).

Cara kerja *web scraping* adalah dengan mengakses halaman pada *web*, menentukan data yang dalam halaman tersebut, melakukan ekstraksi, dan transformasi bila diperlukan, kemudian menyimpan data tersebut menjadi dataset terstruktur (Boeing, 2016).

## 2.6 SMOTE (Synthetic Minority Oversampling Technique)

Metode *SMOTE* (*Synthetic Minority Oversampling Technique*) adalah metode populer yang digunakan untuk menangani ketidakseimbangan kelas (*imbalanced class*), teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan kumpulan data dengan melakukan *resampling* kelas minoritas (Siringoringo, 2018). Contoh proses *SMOTE* (Barus, 2022):

1. Ambil sampel acak contohnya P1
2. Kemudian terapkan algoritma *KNN* pada P1
3. Ambil jarak tetangga terdekat P1, contohnya P2
4. Lalu *generate* data baru dengan persamaan:  
$$P1' = P1 + rand(0,1) \times (P2 - P1) \dots \dots \dots (2.4)$$
5. Ulang terus proses tersebut hingga jumlah data minoritas sebanyak data mayoritas

## **2.7 Text Preprocessing**

Tahapan *preprocessing* merupakan tahapan awal untuk mempersiapkan dokumen agar lebih mudah untuk diproses, tahapan *preprocessing* sebelum proses klasterisasi meliputi *cleansing*, *case folding*, *tokenizing*, *filtering*, dan *stemming* (Amalia et al., 2018). Berdasarkan hal tersebut, mengubah data yang sebelumnya tidak terstruktur memerlukan proses pengubahan menjadi data yang terstruktur untuk diproses pada langkah berikutnya. Berikut adalah beberapa tahapan dari *preprocessing*:

### **2.7.1 Text Cleaning**

*Text cleaning* atau *cleansing* merupakan proses pembersihan data yang bertujuan untuk menghilangkan elemen-elemen yang tidak dibutuhkan pada sebuah dokumen. *Cleansing* bertujuan untuk memperbaiki kualitas data. Pada penelitian ini tahapan *cleansing* berfungsi untuk menghilangkan *mention*, *hashtag*, *url* dan *uri*, tanda baca, *emoji*, menghilangkan angka-angka, serta *case folding* atau mengubah huruf besar ke kecil.

### **2.7.2 Tokenization**

*Tokenization* atau tokenisasi merupakan tahapan pemisahan teks menjadi potongan-potongan berupa huruf, kata, atau kalimat menjadi kata yang tidak terhubung. Data teks yang masuk pada tahap *tokenization* akan diubah menjadi potongan-potongan kata. Pada umumnya, karakter spasi membedakan atau mengidentifikasi setiap kata satu sama lain. Oleh karena itu, proses *tokenization* pada dokumen bergantung pada karakter spasi. Sebagai contoh, tokenisasi pada kalimat “ibu kota pindah” menghasilkan tiga *token*, yaitu “ibu”, “kota”, dan “pindah”.

### **2.7.3 Pembakuan Kata**

Proses ini mengganti kata slang atau gaul menjadi kata baku dengan menggunakan kamus kata slang yang didapatkan dari beberapa sumber di internet agar lebih mudah terbaca oleh kamus *lexicon* saat dilakukan pelabelan sentimen.

### **2.7.4 Negation Handling**

proses *negation handling* mengubah bentuk kata yang memiliki makna yang sama tanpa kata negasi seperti “tidak” dan “belum”. Contohnya seperti “tidak bagus” yang maknanya sama dengan “jelek”, atau “tidak susah” yang sama maknanya dengan “mudah”.

### **2.7.5 Stopwords Removal**

Penghapusan *stopwords* bertujuan untuk menghilangkan kata yang dianggap tidak memiliki makna atau tidak relevan yang terdapat pada dokumen. Contoh *stopwords* yang ada pada Bahasa Indonesia seperti “yang”, “dan”, “di”, “itu”, “adapun”, “agak” dan sebagainya. Kata-kata umum tersebut tidak mempunyai nilai pada sebuah dokumen sehingga kata-kata yang termasuk dalam kamus *stopwords* dihilangkan sehingga ukuran data juga akan berkurang.



### 2.7.6 Stemming

*Stemming* merupakan proses yang bertujuan untuk menemukan kata dasar dari sebuah kata dengan menghapus imbuhan (*affixes*), termasuk awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*), serta kombinasi dari awalan dan akhiran (*confixes*) pada kata turunan. Tujuan utama dari proses *stemming* ini adalah mengubah bentuk kata menjadi kata dasar sesuai dengan bahasa Indonesia yang baik dan benar. Menurut algoritma Nazief & Adriani memiliki tahap-tahap sebagai berikut:

1. Mencari kata yang akan dilakukan *stemming* pada kamus, apabila ditemukan maka diasumsikan bahwa kata tersebut merupakan kata akar (*root word*), jika terbukti maka proses diberhentikan pada tahap pertama.
2. Menghilangkan *inflection suffixes*, sebuah kata yang mengandung *inflection suffixes* yaitu apabila memiliki imbuhan “-lah”, “-ku”, “-kah”, “-mu”, atau “-nya”, apabila kata berupa *particles* atau dalam kata lain yang mengandung “-lah”, “-kah”, atau “-pun”, maka langkah untuk menghilangkan *inflection suffixes* diulangi agar dapat menghilangkan *possessive pronouns* yang termasuk imbuhan *possessive pronouns* adalah “-ku”, “-mu”, atau “-nya”.
3. Menghapus *derivation suffixes* atau imbuhan turunan yang termasuk pada kata imbuhan turunan adalah “-i”, “-an”, atau “-kan”.
4. Menghapus *derivational prefix* atau imbuhan yang berada pada awal kata yang dimaksud dengan *derivational prefix* adalah “be-”, “di-”, “ke-”, “me-”, “pe-”, “se-”, dan “te-”.
5. Apabila 4 langkah tersebut telah dilakukan tetapi belum berhasil menemukan kata dasar maka algoritma ini akan melakukan analisis apakah kata tersebut termasuk ke dalam tabel ambiguitas kolom terakhir.
6. Apabila belum berhasil maka algoritma akan dikembalikan pada kata aslinya.

*Stemming* pada *python* dilakukan melalui kelas *StemmerFactory* yaitu sebuah kelas yang terdapat pada *library* yang bernama “*Sastrawi*” dan kompatibel dengan *input* berbahasa Indonesia yang mana *library* ini akan lebih dulu di-*import* sebelum meng-*import* kelas *StemmerFactory*.

### 2.8 Lexicon Based

Metode berbasis *Lexicon* merupakan metode yang sederhana, layak, dan praktis untuk analisis sentimen dari data media sosial. Data yang cocok dengan metode *Lexicon Based* yaitu data kuesioner, data *Twitter*, data *Facebook*, atau media sosial lainnya yang berupa opini pelanggan tentang suatu produk atau pelayanan jasa (Matulatuwa et al., 2017).

Metode *Lexicon Based* didasarkan pada asumsi bahwa orientasi sentimen kontekstual adalah jumlah dari orientasi sentimen setiap kata atau frasa. Metode ini dapat digunakan untuk melakukan ekstraksi sentimen dari blog dengan mengombinasikan *lexical knowledge* dan klasifikasi teks. Metode *Lexicon* dapat dibuat secara manual atau diperluas secara otomatis dari *seed of words* (Matulatuwa et al., 2017).

Penentuan label sentimen dilakukan pada data teks berupa kalimat yang memiliki kata pada kamus *lexicon* yang terdiri dari kata negatif dan positif. Kata yang teridentifikasi dalam kamus *lexicon* akan dihitung skornya sesuai dengan jumlah kata pada setiap teks atau kalimat (Ismail et al., 2023).

$$S_{positive} = \sum_{i \in t}^{n} positive\ score_i \dots \dots \dots (2.5)$$

$$S_{negative} = \sum_{i \in t}^{n} negative\ score_i \dots \dots \dots (2.6)$$

$S_{positive}$  adalah bobot dari kalimat yang didapatkan melalui penjumlahan  $n$  skor polaritas kata opini positif dan  $S_{negative}$  adalah bobot dari kalimat yang didapatkan melalui penjumlahan  $n$  skor polaritas kata opini negatif. Oleh karena itu, dari persamaan nilai sentimen dalam satu kalimat diperoleh persamaan untuk menentukan orientasi sentimen dengan perbandingan jumlah nilai positif, negatif, dan netral (Ismail et al., 2023).

$$Sentence_{sentiment} = \begin{cases} positive & \text{if } S_{positive} > S_{negative} \\ neutral & \text{if } S_{positive} = S_{negative} \\ negative & \text{if } S_{positive} < S_{negative} \end{cases} \dots \dots \dots (2.7)$$

Jika dalam suatu teks memiliki jumlah kata positif lebih banyak dari kata negatif, maka data teks tersebut akan dilabeli sentimen positif. Jika dalam suatu teks memiliki jumlah kata positif lebih sedikit dari kata negatif, maka data teks tersebut akan dilabeli sentimen negatif. Jika dalam suatu teks memiliki jumlah kata positif sama dengan kata negatif, maka data teks tersebut akan dilabeli sentimen netral (Ismail et al., 2023).

## 2.9 K-Nearest Neighbors (KNN)

*KNN* adalah algoritma *machine learning classifier* populer yang paling sederhana yang pertama kali diperkenalkan oleh T. Cover dan P. Hart pada tahun 1967 dimana algoritma ini mengklasifikasikan kelas sampel berdasarkan kelas tetangga terdekatnya (Fajri et al., 2020). *K-Nearest Neighbors* sendiri memiliki prinsip sederhana, bekerja berdasarkan jarak terpendek dari sampel uji ke sampel latih (Sari, 2020). Algoritma *KNN* bekerja dengan cara menghitung jarak tiap titik pada data tes dengan data latihan tiap kelas. Lalu, diurutkan dari jarak terdekat ke jarak terjauh dan akan dipilih jarak terdekat antara data tes dengan data latihan sejumlah  $k$ . Kelas yang memiliki jarak terdekat dengan data tes akan menjadi kelas data tes tersebut (Raschka, 2018). Adapun tahapan proses yang dilakukan pada *KNN* menurut Abdillah (2023) adalah sebagai berikut:

1. Hitung jarak antara sampel yang tidak diketahui dengan semua sampel pada set data pelatihan menggunakan rumus jarak yang dipilih, didalam kasus ini digunakan *Cosine Similarity*.
2. Pilih  $k$  tetangga terdekat dari sampel yang tidak diketahui berdasarkan jarak yang telah dihitung.
3. Hitung label kelas mayoritas dari  $k$  tetangga terdekat. Dalam kasus klasifikasi biner, label mayoritas dapat dihitung dengan menghitung frekuensi masing-masing kelas pada  $k$  tetangga terdekat dan memilih kelas dengan frekuensi yang paling tinggi. Dalam kasus klasifikasi multikelas, label mayoritas dihitung dengan metode voting, yaitu dengan menghitung jumlah suara setiap kelas pada  $k$  tetangga terdekat dan memilih kelas dengan jumlah suara terbanyak.
4. Kembalikan label kelas mayoritas sebagai hasil klasifikasi untuk sampel yang tidak diketahui.

Berdasarkan langkah sebelumnya telah diketahui bobot tiap kata. Langkah selanjutnya adalah menghitung jarak atau tingkat kemiripan data dengan setiap data latih yang ada menggunakan rumus jarak *Cosine Similarity*. Lalu, sistem akan mengurutkan nilai jarak dari yang tertinggi sampai terendah. Kelebihan dari algoritma *Cosine Similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen dan memiliki tingkat akurasi yang tinggi. Tahapan pada *Cosine similarity* adalah sebagai berikut (Abdillah, 2023):

1. Kalikan bobot dari setiap *term* pada D1 dengan setiap *term* dari semua dokumen data latih yang ada.
2. Hasil perkalian D1 dengan setiap dokumen kemudian dijumlahkan.
3. Hitung hasil kuadrat dari masing-masing *term* dalam setiap dokumen (termasuk D1) kemudian jumlahkan lalu diakarkan.
4. Lakukan pembagian antara hasil dari langkah nomor 2 dengan langkah nomor 3. Maka, didapatkan nilai *Cosine Similarity*. Berikut adalah rumus *Cosine Similarity*:

$$\cos(\theta_{QD}) = \frac{\sum_{i=1}^n Q_i D_i}{\sqrt{\sum_{i=1}^n (Q_i)^2} \sqrt{\sum_{i=1}^n (D_i)^2}} \dots\dots\dots (2.8)$$

**Keterangan:**

$\cos(\theta_{QD})$	: kemiripan Q terhadap dokumen D
Q	: data uji
D	: data latih
n	: jumlah data latih

**2.10 Validasi dan Pengujian**

Validasi dan pengujian sangat diperlukan untuk menilai kinerja dari sebuah sistem. Kinerja proses klasifikasi menggambarkan seberapa baik sistem dalam melakukan klasifikasi data. Pengukuran kinerja proses tersebut dapat dilakukan menggunakan *confusion matrix* (Abdillah, 2023).

*Confusion matrix* merupakan suatu matriks yang digunakan untuk menganalisa keakuratan dari model klasifikasi yang dibuat untuk mengidentifikasi data dengan kelas yang berbeda (Afrillia et al., 2022). Pengujian dengan *confusion matrix* ini digunakan untuk menghitung nilai *true positive*, *false positive*, *true negative*, serta *false negative* tergantung banyaknya kelas klasifikasi yang nantinya dapat digunakan untuk pengukuran nilai akurasi, presisi, serta *recall*. Melakukan pengukuran tingkat akurasi dapat mengetahui seberapa baik performa model klasifikasi tersebut. Bentuk dari *confusion matrix* adalah tabel dengan empat atau lebih kombinasi berdasarkan kelas-kelas pada penelitian ini yang berbeda antara nilai prediksi dan nilai aktual. Berikut merupakan tabel *confusion matrix*:

**Tabel 2.1 Confusion Matrix**

		<i>Predicted</i>		
		-1 ( <i>Negative</i> )	+1 ( <i>Positive</i> )	0 ( <i>Neutral</i> )
<i>Actual</i>	-1 ( <i>Negative</i> )	TN	FP	FL
	+1 ( <i>Positive</i> )	FN	TP	FL2
	0 ( <i>Neutral</i> )	FN2	FP2	TL

**Keterangan:**

TN ( <i>True Negative</i> )	: Prediksi benar bernilai negatif
TP ( <i>True Positive</i> )	: Prediksi benar bernilai positif
TL ( <i>True Neutral</i> )	: Prediksi benar bernilai netral
FN ( <i>False Negative</i> )	: Positif terprediksi negatif
FN2 ( <i>False Negative 2</i> )	: Netral terprediksi negatif
FP ( <i>False Positive</i> )	: Negatif terprediksi positif
FP2 ( <i>False Positive 2</i> )	: Netral terprediksi positif
FL ( <i>False Neutral</i> )	: Negatif terprediksi netral
FL2 ( <i>False Neutral 2</i> )	: Positif terprediksi netral

Tabel *confusion matrix* di atas dapat digunakan dalam perhitungan *performance matrix* yang bertujuan untuk mengukur model yang digunakan untuk dapat memperoleh nilai *accuracy*, *recall*, *precision*, dan *f1-score*.

### 1. Accuracy

Akurasi merupakan nilai yang menunjukkan kedekatan antar nilai prediksi dan nilai aktual. Perhitungan akurasi dengan cara membagi jumlah data yang akan diklasifikasi secara tepat dengan total sampel data *testing* yang diuji. Berikut merupakan rumus perhitungan nilai akurasi:

$$Accuracy = \frac{TP+TN+TL}{TN+FP+FL+FN+TP+FL2+FN2+FP2+TL} \dots\dots\dots(2.9)$$

### 2. Recall

*Recall* merupakan perbandingan jumlah data yang dilakukan prediksi pada kelas yang benar dengan jumlah data yang diharapkan berada pada kelas yang benar. *Recall* dikatakan sebagai tingkat keberhasilan model dalam menemukan informasi. Berikut merupakan rumus perhitungan nilai *recall*:

$$Recall\ Positive = \frac{TP}{TP+FN+FL2} \dots\dots\dots(2.10)$$

$$Recall\ Negative = \frac{TN}{TN+FP+FL} \dots\dots\dots (2.11)$$

$$Recall\ Neutral = \frac{TL}{TL+FN2+FP2} \dots\dots\dots (2.12)$$

$$Recall = \frac{Recall\ Positive+Recall\ Negative+Recall\ Neutral}{3} \times 100\% \dots\dots\dots(2.13)$$

### 3. *Precision*

*Precision* merupakan tingkat akurasi antar informasi yang diinginkan pengguna serta respon sistem. Nilai presisi menunjukkan data positif yang diklasifikasi dengan tepat kemudian dilakukan pembagian dengan jumlah data positif yang diklasifikasi. Berikut merupakan rumus perhitungan nilai presisi:

$$Preci\ Positive = \frac{TP}{TP+FP+FP2} \dots\dots\dots(2.14)$$

$$Preci\ Negative = \frac{TN}{TN+FN+FN2} \dots\dots\dots(2.15)$$

$$Preci\ Neutral = \frac{TL}{TL+FL+FL2} \dots\dots\dots(2.16)$$

$$Precision = \frac{Preci\ Positive+Preci\ Negative+Preci\ Neutral}{3} \times 100\% \dots\dots\dots(2.17)$$

### 4. *F1-Score*

*F1-Score* merupakan perbandingan rata-rata *precision* dan *recall* yang telah dibobotkan. Nilai terbaik *F1-Score* adalah 1 dan nilai terburuknya yaitu 0. Perhitungan yang didapatkan berupa informasi bahwa model klasifikasi memiliki *precision* dan *recall* yang baik. Berikut merupakan rumus perhitungan nilai *F1-Score*:

$$F1\ Positive = 2 \times \frac{Preci\ Positive \times Recall\ Positive}{Preci\ Positive+Recall\ Positive} \dots\dots\dots(2.18)$$

$$F1\ Negative = 2 \times \frac{Preci\ Negative \times Recall\ Negative}{Preci\ Negative+Recall\ Negative} \dots\dots\dots(2.19)$$

$$F1\ Neutral = 2 \times \frac{Preci\ Neutral \times Recall\ Neutral}{Preci\ Neutral+Recall\ Neutral} \dots\dots\dots(2.20)$$

$$F1\ Score = \frac{F1\ Positive+F1\ Negative+F1\ Neutral}{3} \times 100\% \dots\dots\dots(2.21)$$

## 2.11 Studi Pustaka

Penelitian di bawah ini merupakan penelitian-penelitian yang telah dilakukan sebelumnya dan berkaitan dengan penelitian tugas akhir ini, sehingga menjadi referensi dalam penelitian ini.

**Tabel 2.2 Studi Pustaka**

No	Penulis	Judul	Metode	Hasil
1.	Elfansyah et al., 2024.	Perbandingan Metode K-Nearest Neighbor (KNN) Dan Naïve Bayes Terhadap Analisis Sentimen Pada Pengguna E-Wallet Aplikasi Dana Menggunakan Fitur Ekstraksi TF-IDF	Klasifikasi <i>KNN</i> dan <i>Naïve Bayes</i> . Pelabelan sentimen berbasis <i>Lexicon</i> dengan acuan kamus label oleh tenaga ahli bahasa ( <i>expert</i> ) dan <i>library Lexicon via Python</i> . Pembobotan kata menggunakan <i>TF-IDF</i> .	Data yang diberi label model <i>Lexicon</i> , akurasi <i>KNN</i> 78% dan <i>Naïve Bayes</i> 74%.  Terkait nilai <i>k</i> , keputusan akhir sebagai parameter final yang diambil menurut penulis adalah <i>k</i> = 5 dikarenakan memberikan kinerja optimal tanpa penurunan akurasi yang signifikan.
2.	Alamsyah & Mulyati, 2023.	Implementasi Algoritme K-Nearest Neighbour Dan Lexicon Based Untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Gramedia Digital Pada Media Sosial Twitter	Klasifikasi <i>KNN</i> . Pelabelan sentimen berbasis <i>Lexicon</i> dengan acuan kamus label <i>InSet</i> . Pembobotan kata menggunakan <i>TF-IDF</i> .	<i>KNN</i> berhasil mencapai akurasi tertinggi sebesar 75.97% dengan nilai <i>k</i> = 3 dengan rasio pembagian data 60:40.
3.	Putri et al., 2023.	Analisis Sentimen dan Pemodelan Ulasan Aplikasi AdaKami Menggunakan Algoritma SVM dan KNN	Klasifikasi <i>KNN</i> dan <i>SVM</i> . Pelabelan sentimen berbasis <i>library Lexicon via Python</i> dan manual. Pembobotan kata menggunakan <i>TF-IDF</i> .	Rasio pembagian data 90:10, 80:20, 70:30, 60:40, 50:50. Model analisis sentimen dengan performa paling optimal menggunakan algoritma <i>SVM</i> dengan metode pelabelan manual dan proporsi pembagian data 90:10 dengan akurasi sebesar 93%, presisi 93%, <i>recall</i> 93%, dan <i>f1-score</i> 92%. Penelitian ini berkesimpulan bahwa <i>SVM</i> menghasilkan model dengan performa lebih optimal dibanding <i>KNN</i>

**Tabel 2.3 Lanjutan Studi Pustaka**

4.	Rahayu et al., 2022.	Implementasi Metode K-Nearest Neighbor (K-NN) untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP	Klasifikasi <i>KNN</i> . Pembobotan kata menggunakan <i>TF-IDF</i> . Pelabelan sentimen berbasis <i>Lexicon</i> dengan acuan kamus label <i>SentiWordnet</i> .	Rasio pembagian data 80:20 dengan algoritma klasifikasi <i>KNN</i> memperoleh akurasi sebesar 76.68%.
5.	Angel et al., 2024.	Analisis Sentimen dan Emosi dari Ulasan Google Maps untuk Layanan Rumah Sakit di Palangka Raya Menggunakan Machine Learning	Klasifikasi <i>KNN</i> , <i>Logistic Regression</i> , dan <i>Decision Tree</i> . Pelabelan sentimen berbasis <i>Lexicon Vader via Python</i> dan <i>NRC Lexicon</i> (emosi). Pembobotan kata menggunakan <i>TF-IDF</i> .	Akurasi tertinggi didapat oleh algoritma klasifikasi <i>Decision Tree</i> sebesar 92%, diikuti dengan <i>Logistic Regression</i> dengan akurasi 86%, dan <i>KNN</i> dengan akurasi 48%.
6.	Sholeha et al., 2022.	Analisis Sentimen Pada Agen Perjalanan Online Menggunakan Naïve Bayes dan K-Nearest Neighbor	Klasifikasi <i>KNN</i> dan <i>Naïve Bayes</i> . Pelabelan sentimen berbasis manual. Penghitungan frekuensi kata dengan <i>TF</i> .	Menurut penulis, berdasarkan 1500 data komentar dari 3 <i>fanpage Facebook</i> agen perjalanan online, ditemukan bahwa algoritma <i>KNN</i> memiliki akurasi yang lebih baik pada rata-rata dibandingkan <i>Naïve Bayes</i> dengan akurasi tertinggi 52.35%.  Akurasi tertinggi kedua algoritma klasifikasi tersebut didapatkan saat seluruh data menggunakan huruf kecil.
7	Pramayasa et al, 2023.	Analisis Sentimen Program Mbkm Pada Media Sosial Twitter Menggunakan KNN Dan SMOTE	Klasifikasi <i>KNN</i> , penyeimbang kelas <i>SMOTE</i> .	Hasil performa metode <i>KNN</i> dengan nilai akurasi 68.81%, presisi 70.73%, <i>recall</i> 68.81%, serta <i>f1-score</i> 68.42%.  Hasil optimasi menggunakan metode <i>SMOTE</i> mendapatkan hasil akurasi 76.13%, presisi 76.03%, <i>recall</i> 76.13%, serta <i>f1-score</i> 76.01%.

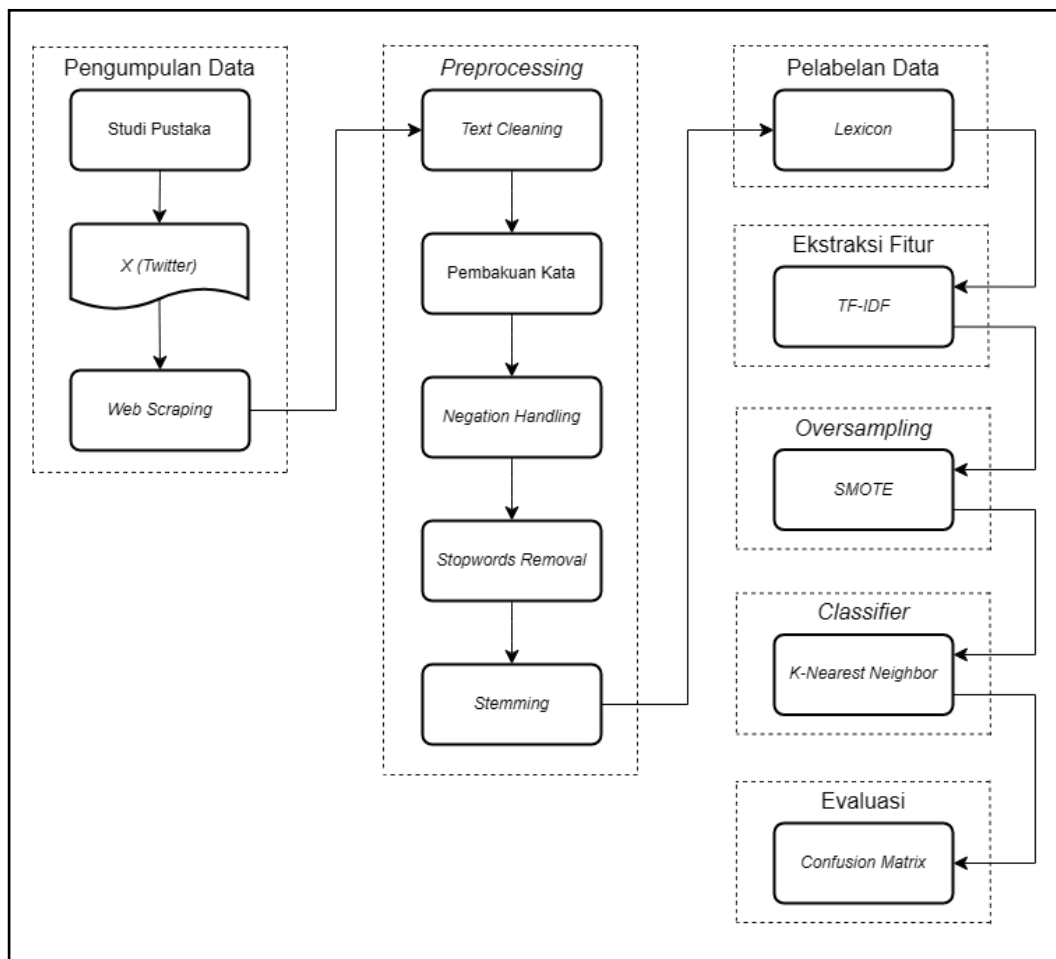


## BAB III

### METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

#### 3.1 Metodologi Penelitian

Bagian ini membahas mengenai metodologi penelitian serta pengembangan sistem yang akan dilakukan dalam penelitian ini yaitu menganalisis sentimen publik terhadap topik pemindahan ibu kota negara berdasarkan unggahan dalam bentuk teks di sosial media *X (Twitter)* dengan metode klasifikasi *KNN* dan pelabelan sentimen berbasis leksikon serta ekstraksi fitur *TF-IDF*. Metodologi penelitian merupakan sub bab yang menggambarkan alur kerja serta tahapan pada penelitian ini. Tahapan metodologi penelitian pada tugas akhir ini yaitu pengumpulan data, *preprocessing* data, pelabelan data, pembobotan kata, penyeimbangan kelas, pembuatan model, pengujian model, pengujian sistem, serta analisis dan visualisasi hasil. Berikut merupakan tahapan pada metodologi penelitian ini dapat dilihat pada gambar 3.1.



**Gambar 3.1 Metodologi Penelitian**

### 3.1.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data yang nantinya akan diolah menjadi informasi yang mudah dipahami dan dapat digunakan sebagai dasar pengambilan keputusan. Data yang terdapat dalam penelitian ini dikumpulkan dengan melakukan *web scraping* dari media sosial *X (Twitter)*.

a. Studi Pustaka

Studi pustaka merupakan langkah awal dalam proses penelitian. Kegiatan ini bertujuan untuk mencari dan memperoleh informasi terkait metode, topik, serta masalah yang menjadi fokus penelitian. Studi pustaka ini didasarkan pada referensi dari jurnal, makalah, prosiding, buku, dan sumber terpercaya lainnya. Studi pustaka dilakukan untuk memperkuat argumentasi dalam penelitian yang sedang dilakukan.

b. *X (Twitter)*

Data yang digunakan dalam penelitian ini diambil dari media sosial *X*. *X* atau yang sebelumnya dikenal dengan sebutan *Twitter* merupakan media sosial yang memungkinkan pengguna untuk berbagi pesan singkat yang disebut *tweet* atau cuitan. Media sosial ini digunakan secara luas oleh berbagai kalangan, mulai dari individu hingga organisasi besar untuk menyampaikan pendapat, berita, dan informasi lainnya. Platform ini memberikan peluang untuk dilakukannya analisis sentimen publik terhadap pemindahan ibu kota negara dan mengamati reaksi pengguna terhadap berbagai topik, termasuk ulasan tentang layanan tertentu.

c. *Web Scraping*

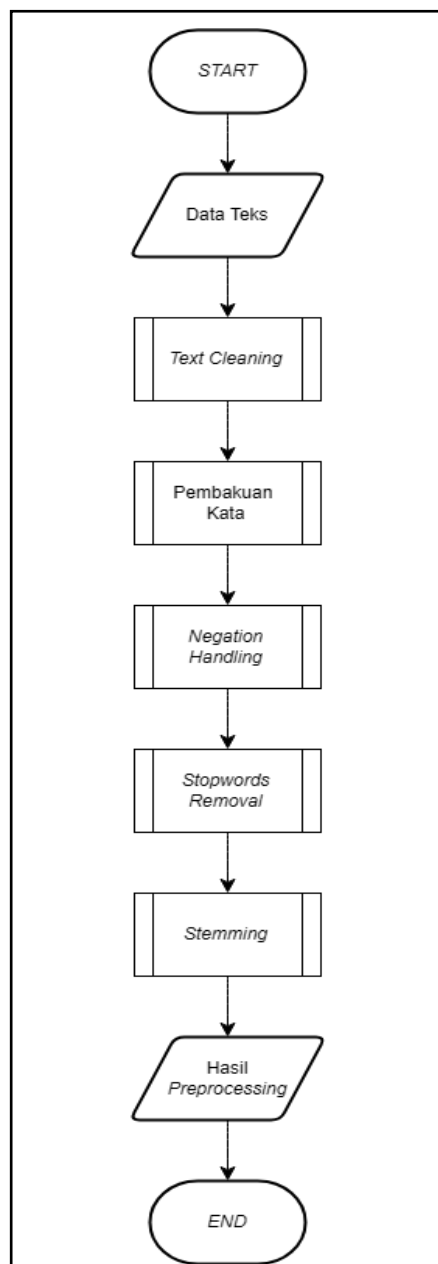
Pengumpulan data dilakukan dengan cara *web scraping* terhadap media sosial *X (Twitter)*. Data yang berhasil dikumpulkan akan dilakukan *preprocessing*. Data yang digunakan pada penelitian ini merupakan data berbahasa Indonesia dengan total data berjumlah 5341 baris *tweet* menggunakan 9 kata kunci yang tertera pada tabel 3.1.

**Tabel 3.1 Kata Kunci**

No.	Kata Kunci	Jumlah Data
1	<i>ikn</i>	1005
2	<i>ibu kota baru</i>	223
3	<i>ibu kota nusantara</i>	15
4	<i>ibu kota pindah</i>	1005
5	<i>pemindahan ibu kota</i>	1010
6	<i>ibukota baru</i>	212
7	<i>ibukota nusantara</i>	804
8	<i>ibukota pindah</i>	554
9	<i>pemindahan ibukota</i>	513

### 3.1.2 Preprocessing

Tahap penting dalam pengolahan data yang bertujuan untuk membersihkan dan mempersiapkan data agar siap digunakan dalam analisis sentimen. Pada tahapan ini, berbagai langkah dilakukan, seperti menghapus data yang tidak relevan (misalnya tanda baca, angka, dan simbol khusus), mengubah teks menjadi format standar (seperti konversi huruf besar menjadi huruf kecil), menghilangkan kata-kata umum yang tidak memiliki makna khusus (*stopwords*), serta melakukan *stemming* untuk mengembalikan kata ke bentuk dasarnya. *Preprocessing* sangat penting untuk meningkatkan akurasi model dalam memahami dan menganalisis data secara efektif, terutama dalam tugas-tugas yang melibatkan analisis teks seperti analisis sentimen. Berikut merupakan *flowchart* dari *preprocessing* dapat dilihat pada gambar 3.2.

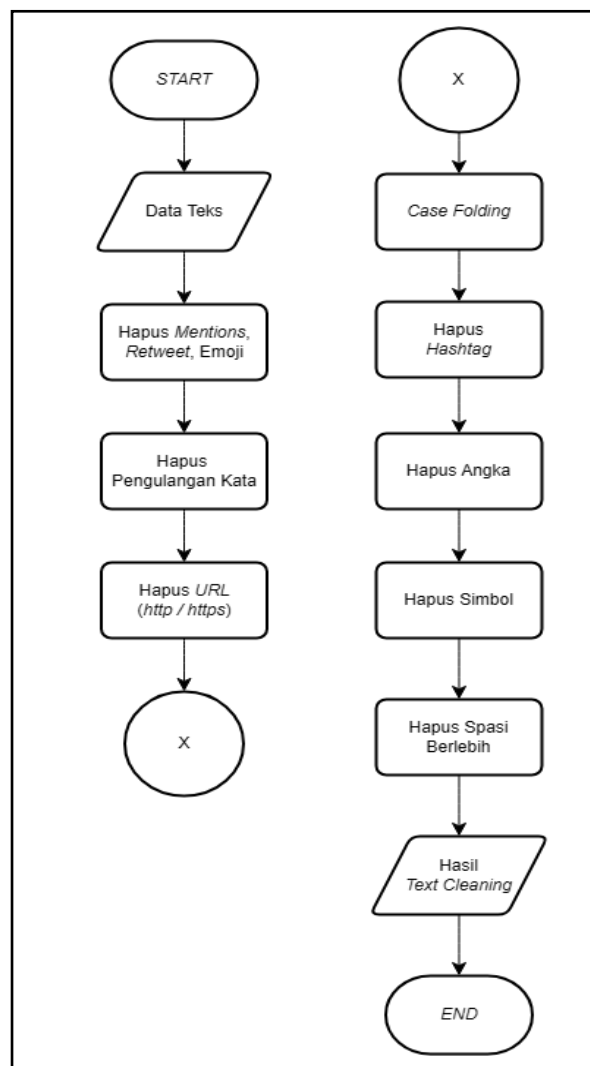


Gambar 3.2 Flowchart Preprocessing

a. *Text Cleaning*

Proses ini bertujuan untuk membersihkan data dengan menghapus elemen-elemen yang tidak relevan, seperti tanda baca, angka, emoji, *url*, atau karakter khusus lainnya yang tidak memiliki makna signifikan untuk analisis teks. *Cleansing* sangat membantu untuk fokus pada elemen-elemen yang relevan pada data untuk analisis lebih lanjut.

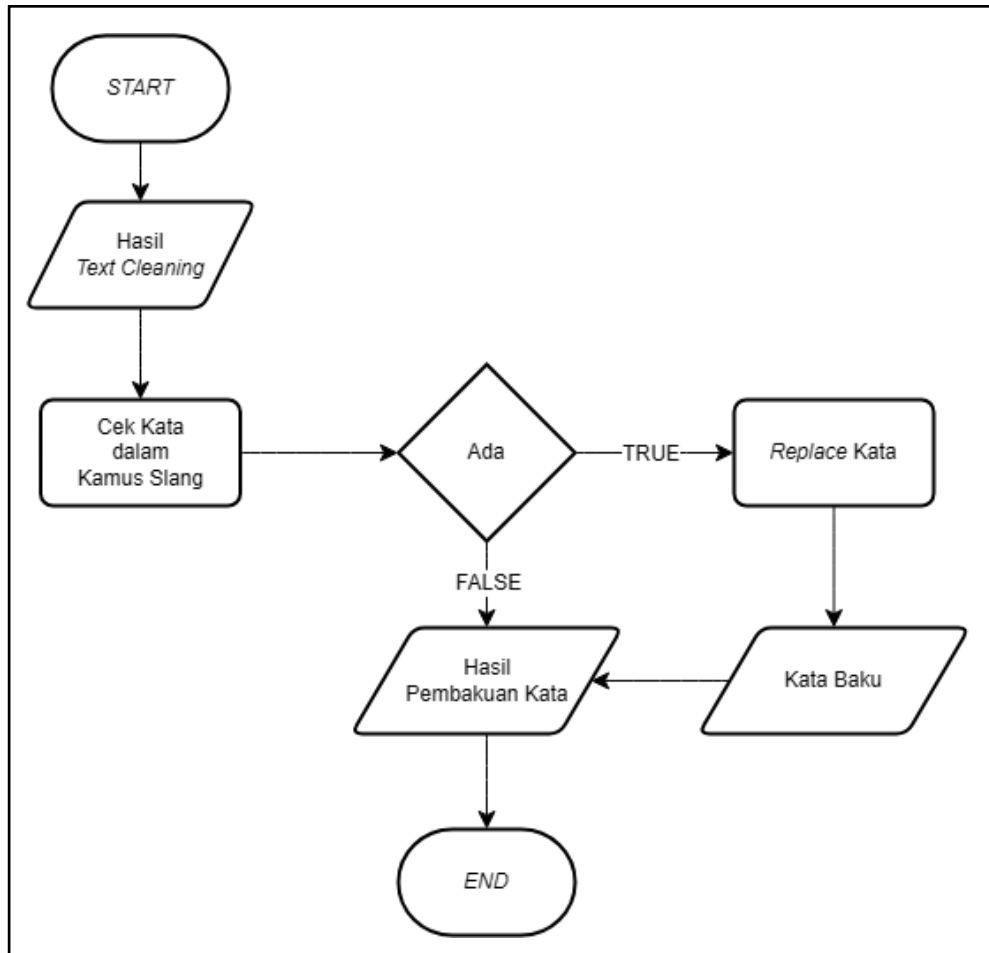
Tahap ini juga mengubah seluruh teks menjadi huruf kecil (*lower case*). Hal tersebut dilakukan untuk menghilangkan perbedaan antara huruf besar dan huruf kecil, karena dalam analisis berbasis teks, kata dengan huruf besar dan kecil dianggap berbeda.



**Gambar 3.3** *Flowchart Text Cleaning*

b. Pembakuan Kata

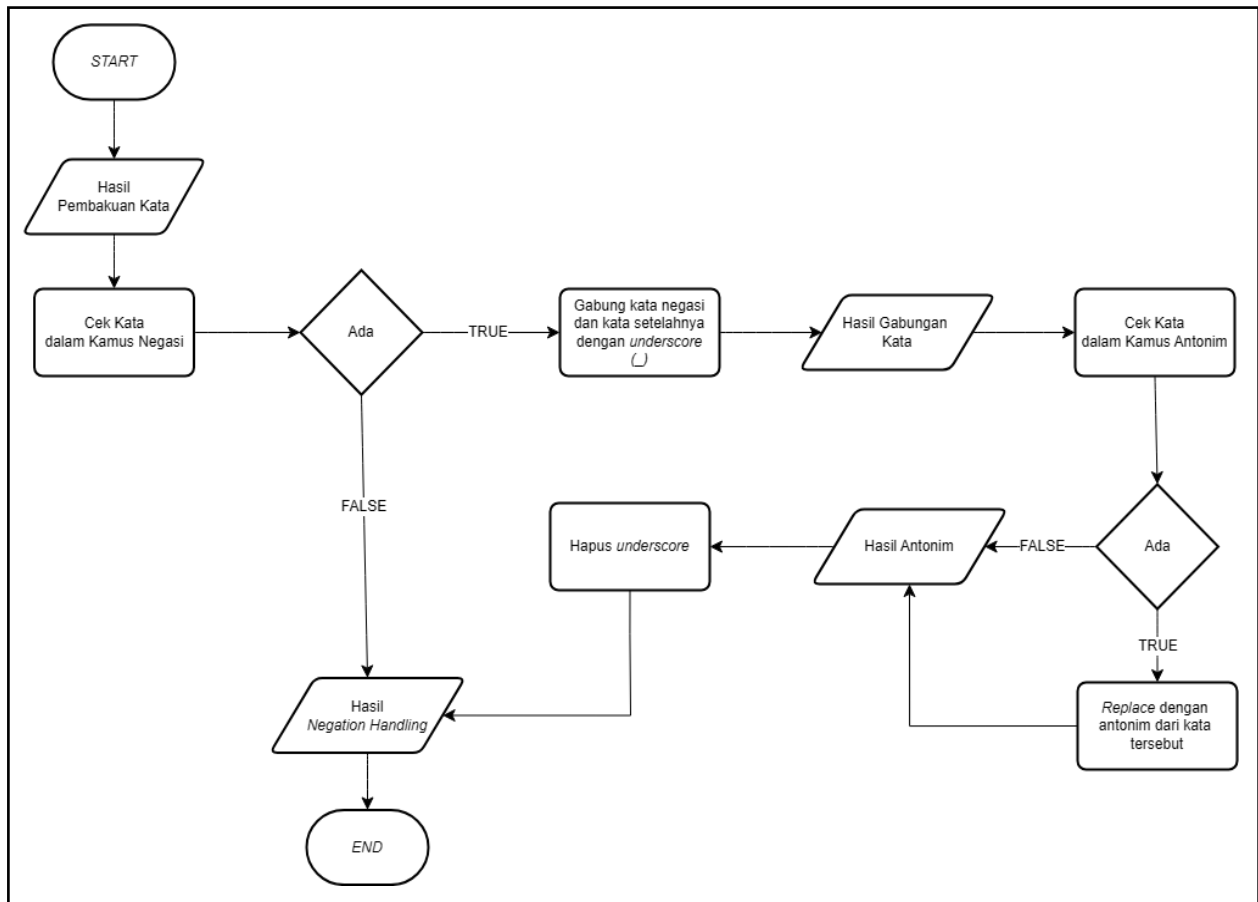
Tahapan ini mengganti kata slang atau gaul menjadi kata baku dengan menggunakan kamus kata slang yang didapatkan dari beberapa sumber di internet agar lebih mudah terbaca oleh kamus *lexicon* saat dilakukan pelabelan sentimen.



**Gambar 3.4 Flowchart Pembakuan Kata**

c. *Negation Handling*

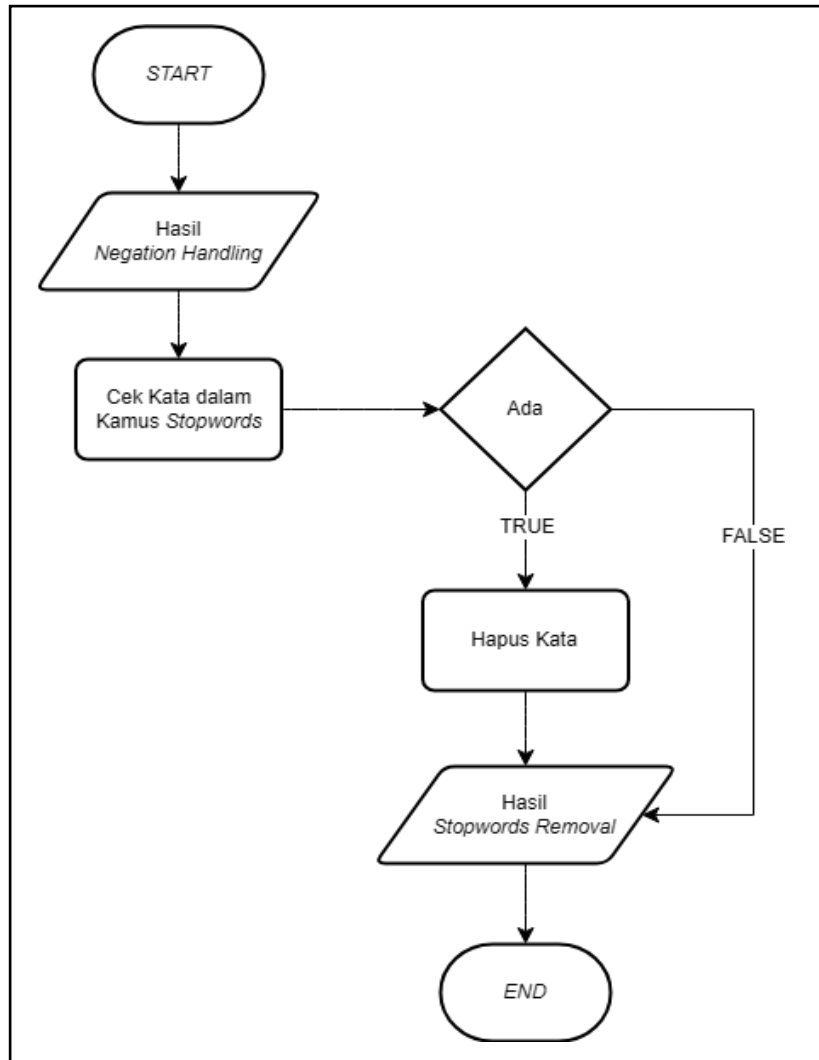
Tahapan *negation handling* mengubah bentuk kata yang memiliki makna yang sama tanpa kata negasi seperti “tidak” dan “belum”. Contohnya seperti “tidak bagus” yang maknanya sama dengan “jelek”, atau “tidak susah” yang sama maknanya dengan “mudah”.



**Gambar 3.5** *Flowchart Negation Handling*

d. *Stopwords Removal*

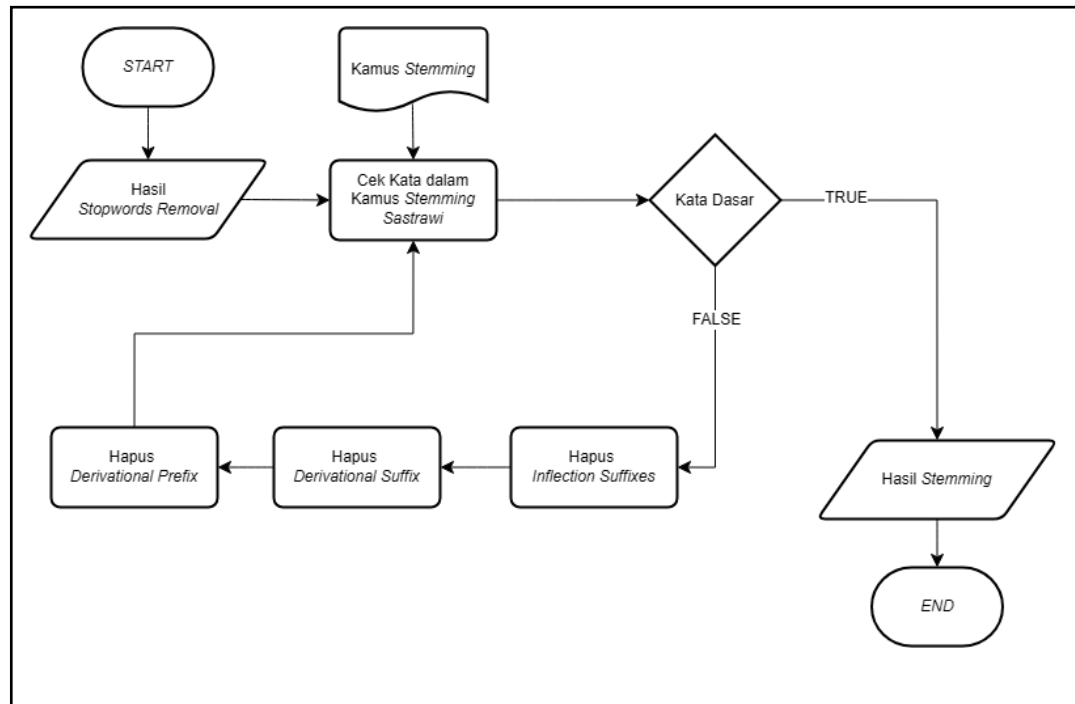
Pada tahap ini, kata-kata umum yang sering muncul tetapi tidak memiliki makna khusus dalam konteks analisis, seperti “dan”, “atau”, “dengan” akan dihilangkan. Kata-kata ini disebut *stopwords* dan biasanya tidak memberikan kontribusi signifikan dalam penentuan makna teks.



**Gambar 3.6 Flowchart Stopwords Removal**

e. *Stemming*

Proses *stemming* merupakan proses mengubah kata-kata dalam teks menjadi bentuk dasarnya atau kata dasar, dengan membuang awalan, akhiran, atau sisipan. Seperti *inflection suffixes* (-lah, -kah, -tah, -pun, -ku, dsb), *derivational suffix* (-an, -kan, -i), dan *derivational prefix* (-be, -di, -ke, -me, -pe, -se, -te). Proses *stemming* yang ada dalam penelitian ini menggunakan algoritma Nazief & Adriani untuk teks berbahasa Indonesia.



**Gambar 3.7 Flowchart Stemming**



### 3.1.3 *Sentiment Labelling*

Data yang telah dilakukan *preprocessing* selanjutnya akan diberi label dengan menggunakan pelabelan *lexicon*. Pelabelan dengan menggunakan pendekatan *lexicon* merupakan langkah penting dalam implementasi metode *lexicon based* dan *K-Nearest Neighbor* untuk menganalisis sentimen publik terhadap pemindahan ibu kota negara. Metode ini menggunakan daftar kata (*lexicon*) yang memiliki nilai sentimen tertentu, baik positif, negatif, maupun netral, untuk menentukan kategori dari setiap teks yang akan dianalisis.

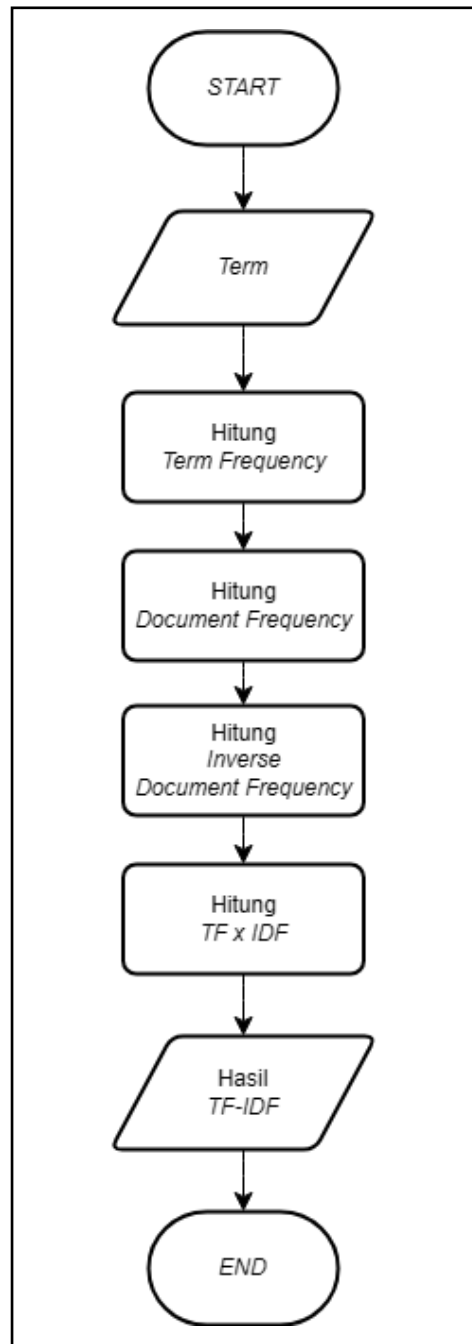
Pada tahapan ini, setiap kata yang terdapat dalam data dicocokkan dengan daftar kata pada *lexicon*, jika kata tersebut ditemukan dalam *lexicon*, maka akan diberikan nilai sentimen sesuai dengan yang tertera pada daftar tersebut. Misalnya, kata-kata seperti “bagus” atau “terpuji” dapat diberi label positif, sedangkan kata seperti “buruk” atau “kecewa” diberi label negatif. Skor total kemudian dihitung berdasarkan nilai dari setiap kata dalam teks, jika skor total lebih besar dari nol, maka teks diberi label sentimen positif. Selain itu, jika skor total kurang dari nol, maka teks diberi label sentimen negatif. Apabila skor adalah nol, maka label sentimen dianggap netral.

Setelah pelabelan menggunakan metode *lexicon*, hasil pelabelan ini kemudian digunakan untuk melatih model *KNN*. Data yang sudah dilabeli akan digunakan oleh *KNN* untuk belajar membedakan pola dan karakteristik teks positif, negatif, atau netral. Pelabelan menggunakan *lexicon* memungkinkan model *KNN* dapat lebih efektif dalam melakukan klasifikasi sentimen pada data baru, dikarenakan sudah dilatih menggunakan data yang dilabeli secara manual atau semi-otomatis berdasarkan *lexicon*. Penggabungan metode *lexicon based* dengan *KNN* ini memanfaatkan keunggulan dari kedua pendekatan yaitu interpretasi langsung dari *lexicon* dan kemampuan pembelajaran dari *KNN* sehingga menghasilkan model yang lebih akurat untuk analisis sentimen publik terhadap pemindahan ibu kota negara.

### 3.1.4 Pembobotan *TF-IDF*

Pembobotan kata dengan menggunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)* adalah teknik yang digunakan untuk mengekstraksi fitur dalam analisis berbasis teks, termasuk analisis sentimen. *TF-IDF* adalah cara untuk mengukur seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap keseluruhan kumpulan dokumen atau korpus. Metode ini memberikan bobot lebih tinggi pada kata-kata yang sering muncul dalam suatu dokumen, tetapi jarang ditemukan dalam dokumen lain di korpus, karena dianggap lebih signifikan.

Setiap kata dalam dokumen akan diberi bobot menggunakan nilai *TF-IDF*. Kata-kata dengan bobot *TF-IDF* tinggi dianggap lebih relevan dan memiliki peran lebih besar dalam mendeskripsikan dokumen tersebut. Pembobotan ini membantu mengurangi dampak dari kata-kata umum yang sering muncul tetapi tidak memberikan banyak informasi (*stopwords*), sementara memberi fokus lebih pada kata-kata unik yang lebih menggambarkan isi dari dokumen.



**Gambar 3.8** *Flowchart TF-IDF*

Gambar 3.8 menunjukkan tahapan perhitungan *TF-IDF*, yang dimulai dengan memilih *term* yang akan dihitung. Selanjutnya, dilakukan penghitungan frekuensi kemunculan kata dalam dokumen (*TF*), diikuti dengan menghitung jumlah dokumen yang mengandung kata tersebut (*DF*). Setelah itu, nilai *IDF* dihitung, lalu tahap terakhir adalah menghitung nilai *TF-IDF* dengan mengalikan hasil *TF* dan *IDF*.

1. Menghitung *Term Frequency (TF)*

*Term Frequency (TF)* adalah frekuensi kemunculan suatu istilah dalam sebuah dokumen. Semakin sering istilah tersebut muncul dalam dokumen, semakin tinggi bobotnya. Berikut merupakan contoh data yang akan digunakan untuk perhitungan manual *TF-IDF* dapat dilihat pada tabel 3.2.

**Tabel 3.2 Contoh Dokumen Perhitungan *Term Frequency***

ID	Dokumen	Sentimen
D1	pindah kota semangat indonesia maju	Netral
D2	jokowi daerah ibukota negara tulang punggung tahan pangan kota peluang daerah kalimantan timur kembang sektor tani ikan integrasi	Positif
D3	gara paksa bikin kota negara paksa sih	Negatif
D4	ditawarin sosok bikin taman analisisnya pikir bikin park garden pilih ngelanjutin mindah kota pusing	Negatif

Selanjutnya yaitu menghitung nilai *Term Frequency (TF)* dari masing-masing dokumen yang telah dilakukan *preprocessing* pada setiap *term* yang ada. Proses perhitungan *TF* dengan memberikan nilai 1 apabila *term* tersebut terdapat pada komentar dan sebaliknya. Proses dilakukan pada dokumen 1 (*D1*) hingga dokumen 4 (*D4*). Hasil *TF* dapat dilihat pada tabel 3.3 dan tabel 3.4.

**Tabel 3.3 Hasil *Term Frequency***

No.	<i>Term</i>	<i>TF</i>			
		D1	D2	D3	D4
1	analisisnya	0	0	0	1
2	bikin	0	0	1	1
3	daerah	0	1	0	0
4	ditawarin	0	0	0	1
5	gara	0	0	1	0
6	garden	0	0	0	1
7	ibukota	0	1	0	0
8	ikan	0	1	0	0
9	indonesia	1	0	0	0
10	integrasi	0	1	0	0
11	jokowi	0	1	0	0
12	kalimantan	0	1	0	0
13	kembang	0	1	0	0
14	kota	1	1	1	1

**Tabel 3.4 Lanjutan Hasil *Term Frequency***

15	maju	1	0	0	0
16	mindah	0	0	0	1
17	negara	0	1	1	0
18	ngelanjutin	0	0	0	1
19	paksa	0	0	1	0
20	pangan	0	1	0	0
21	park	0	0	0	1
22	peluang	0	1	0	0
23	pikir	0	0	0	1
24	pilih	0	0	0	1
25	pindah	1	0	0	0
26	punggung	0	1	0	0
27	pusing	0	0	0	1
28	sektor	0	1	0	0
29	semangat	1	0	0	0
30	sih	0	0	1	0
31	sosok	0	0	0	1
32	tahan	0	1	0	0
33	taman	0	0	0	1
34	tani	0	1	0	0
35	timur	0	1	0	0
36	tulang	0	1	0	0

Dapat dilihat pada tabel 3.3 dan tabel 3.4 merupakan *term* yang telah diproses perhitungan *TF*, dengan diberi nilai 1 apabila *term* tersebut terdapat pada dokumen, dan nilai 0 apabila *term* tersebut tidak ada dalam dokumen.

## 2. Menghitung Nilai *Inverse Document Frequency (IDF)*

Setelah perhitungan *Term Frequency (TF)* selesai, langkah berikutnya adalah menghitung *Inverse Document Frequency (IDF)*, yaitu menghitung seberapa banyak *term* muncul di seluruh dokumen. Rumus *IDF* dihitung menggunakan persamaan:

$$idf_t = \log \frac{N}{df_t} \dots\dots\dots(3.1)$$

Berikut merupakan hasil perhitungan *IDF* dapat dilihat pada tabel 3.5 dan tabel 3.6.

**Tabel 3.5 Hasil *IDF***

No.	Term	TF				DF	N/df(t)	IDF
		D1	D2	D3	D4			
1	analisnya	0	0	0	1	1	4.0	1.916291
2	bikin	0	0	1	1	2	2.0	1.510826
3	daerah	0	1	0	0	1	4.0	1.916291
4	ditawarin	0	0	0	1	1	4.0	1.916291

**Tabel 3.6 Lanjutan Hasil *IDF***

5	gara	0	0	1	0	1	4.0	1.916291
6	garden	0	0	0	1	1	4.0	1.916291
7	ibukota	0	1	0	0	1	4.0	1.916291
8	ikan	0	1	0	0	1	4.0	1.916291
9	indonesia	1	0	0	0	1	4.0	1.916291
10	integrasi	0	1	0	0	1	4.0	1.916291
11	jokowi	0	1	0	0	1	4.0	1.916291
12	kalimantan	0	1	0	0	1	4.0	1.916291
13	kembang	0	1	0	0	1	4.0	1.916291
14	kota	1	1	1	1	4	1.0	1.000000
15	maju	1	0	0	0	1	4.0	1.916291
16	mindah	0	0	0	1	1	4.0	1.916291
17	negara	0	1	1	0	2	2.0	1.510826
18	ngelanjutin	0	0	0	1	1	4.0	1.916291
19	paksa	0	0	1	0	1	4.0	1.916291
20	pangan	0	1	0	0	1	4.0	1.916291
21	park	0	0	0	1	1	4.0	1.916291
22	peluang	0	1	0	0	1	4.0	1.916291
23	pikir	0	0	0	1	1	4.0	1.916291
24	pilih	0	0	0	1	1	4.0	1.916291
25	pindah	1	0	0	0	1	4.0	1.916291
26	punggung	0	1	0	0	1	4.0	1.916291
27	pusing	0	0	0	1	1	4.0	1.916291
28	sektor	0	1	0	0	1	4.0	1.916291
29	semangat	1	0	0	0	1	4.0	1.916291
30	sih	0	0	1	0	1	4.0	1.916291
31	sosok	0	0	0	1	1	4.0	1.916291
32	tahan	0	1	0	0	1	4.0	1.916291
33	taman	0	0	0	1	1	4.0	1.916291
34	tani	0	1	0	0	1	4.0	1.916291
35	timur	0	1	0	0	1	4.0	1.916291
36	tulang	0	1	0	0	1	4.0	1.916291

Tabel 3.5 dan tabel 3.6 merupakan hasil perhitungan nilai *IDF* atau seberapa sering suatu *term* muncul pada dokumen-dokumen tersebut.

### 3. Menghitung Nilai *TF-IDF*

Setelah didapatkan nilai *TF*, *DF*, dan *IDF* maka langkah selanjutnya yaitu menghitung nilai *TF-IDF* melalui persamaan berikut:

$$W_{dt} = tf_{dt} \times idf_t \dots\dots\dots(3.2)$$

*W* merupakan hasil dari perhitungan *TF* dikalikan dengan *IDF* (*TF* × *IDF*), maka hasil perhitungan *TF-IDF* dari dokumen diatas dapat dilihat pada tabel 3.7.

**Tabel 3.7 Hasil *TF-IDF***

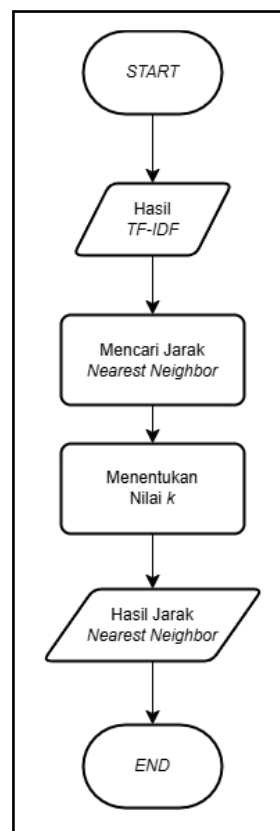
No.	Term	$W_{dt}$			
		<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>D4</i>
1.	analisnya	0	0	0	0.269594
2.	bikin	0	0	0.287590	0.425103
3.	daerah	0	0.460117	0	0
4.	ditawarin	0	0	0	0.269594
5.	gara	0	0	0.364771	0
6.	garden	0	0	0	0.269594
7.	ibukota	0	0.230058	0	0
8.	ikan	0	0.230058	0	0
9.	indonesia	0.483802	0	0	0
10.	integrasi	0	0.230058	0	0
11.	jokowi	0	0.230058	0	0
12.	kalimantan	0	0.230058	0	0
13.	kembang	0	0.230058	0	0
14.	kota	0.252468	0.120054	0.190352	0.140685
15.	maju	0.483802	0	0	0
16.	mindah	0	0	0	0.269594
17.	negara	0	0.181381	0.287590	0
18.	ngelanjutin	0	0	0	0.269594
19.	paksa	0	0	0.729543	0
20.	pangan	0	0.230058	0	0
21.	park	0	0	0	0.269594
22.	peluang	0	0.230058	0	0
23.	pikir	0	0	0	0.269594
24.	pilih	0	0	0	0.269594
25.	pindah	0.483802	0	0	0
26.	punggung	0	0.230058	0	0
27.	pusing	0	0	0	0.269594
28.	sektor	0	0.230058	0	0
29.	semangat	0.483802	0	0	0
30.	sih	0	0	0.364771	0
31.	sosok	0	0	0	0.269594
32.	tahan	0	0.230058	0	0
33.	taman	0	0	0	0.269594
34.	tani	0	0.230058	0	0
35.	timur	0	0.230058	0	0
36.	tulang	0	0.230058	0	0

### 3.1.5 Penerapan Teknik *SMOTE*

Tahapan *Synthetic Minority Oversampling Technique* atau *SMOTE* adalah teknik yang digunakan untuk mengatasi ketidakseimbangan kelas dalam *dataset*. Teknik ini fokus pada kelas minoritas dalam *dataset*, di mana jumlah sampelnya jauh lebih sedikit dibandingkan dengan kelas mayoritas. Proses *SMOTE* melibatkan dua tahap yaitu mencari jarak *Nearest Neighbor*, dan menghasilkan data sintesis.

#### 1. Mencari Jarak *Nearest Neighbor*

Pada setiap data dalam kelas minoritas, tetangga (*neighbor*) terdekatnya diidentifikasi dengan metode *KNN* (*K-Nearest Neighbor*) dengan nilai *k* yang ditentukan dan dengan nilai bobot kelas minoritas yang dihasilkan pada tahap *TF-IDF* sebelumnya. Untuk mencari nilai tetangga terdekat digunakan matriks *Euclidean Distance*. *Flowchart* untuk mencari jarak *nearest neighbor* dapat dilihat pada gambar 3.9 berikut:



**Gambar 3.9 Flowchart Jarak *Nearest Neighbor* *SMOTE***

Contoh perhitungan mencari *nearest neighbor* pada *D2* ke *D3* pada *term* ‘paksa’ adalah sebagai berikut:

$$X_{knn} = \sqrt{(0 - 0.729543)^2} \dots\dots\dots(3.3)$$

$$X_{knn} = \sqrt{0.532232} \dots\dots\dots(3.4)$$

Proses perhitungan jarak *nearest neighbor* pada semua *term* pada kelas minor positif dengan contoh  $k = 2$ . Hasil jarak *nearest neighbor* dapat dilihat pada tabel 3.8 berikut:

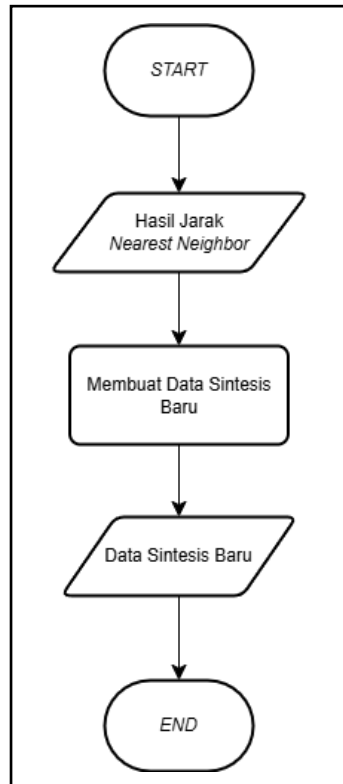
**Tabel 3.8 Hasil Jarak Nearest Neighbor SMOTE**

No.	Term	Jarak Kelas Minor Positif					
		D2 ke data		D3 ke data		D4 ke data	
		D3	D4	D2	D4	D2	D3
1	analisnya	0	0.073	0	0.073	0.073	0.073
2	bikin	0.083	0.181	0.083	0.019	0.181	0.019
3	daerah	0.212	0.212	0.212	0	0.212	0
4	ditawarin	0	0.073	0	0.073	0.073	0.073
5	gara	0.133	0	0.133	0.133	0	0.133
6	garden	0	0.073	0	0.073	0.073	0.073
7	ibukota	0.053	0.053	0.053	0	0.053	0
8	ikan	0.053	0.053	0.053	0	0.053	0
9	indonesia	0	0	0	0	0	0
10	integrasi	0.053	0.053	0.053	0	0.053	0
11	jokowi	0.053	0.053	0.053	0	0.053	0
12	kalimantan	0.053	0.053	0.053	0	0.053	0
13	kembang	0.053	0.053	0.053	0	0.053	0
14	kota	0.005	0.0004	0.005	0.002	0.0004	0.002
15	maju	0	0	0	0	0	0
16	mindah	0	0.073	0	0.073	0.073	0.073
17	negara	0.011	0.033	0.011	0.083	0.033	0.083
18	ngelanjutin	0	0.073	0	0.073	0.073	0.073
19	paksa	0.532	0	0.532	0.532	0	0.532
20	pangan	0.053	0.053	0.053	0	0.053	0
21	park	0	0.073	0	0.073	0.073	0.073
22	peluang	0.053	0.053	0.053	0	0.053	0
23	pikir	0	0.073	0	0.073	0.073	0.073
24	pilih	0	0.073	0	0.073	0.073	0.073
25	pindah	0	0	0	0	0	0
26	punggung	0.053	0.053	0.053	0	0.053	0
27	pusing	0	0.073	0	0.073	0.073	0.073
28	sektor	0.053	0.053	0.053	0	0.053	0
29	semangat	0	0	0	0	0	0
30	sih	0.133	0	0.133	0.133	0	0.133
31	sosok	0	0.073	0	0.073	0.073	0.073
32	tahan	0.053	0.053	0.053	0	0.053	0
33	taman	0	0.073	0	0.073	0.073	0.073
34	tani	0.053	0.053	0.053	0	0.053	0
35	timur	0.053	0.053	0.053	0	0.053	0
36	tulang	0.053	0.053	0.053	0	0.053	0
Jumlah		1.851	1.971	1.851	1.705	1.971	1.705
Akar		1.361	1.404	1.361	1.306	1.404	1.306
$k = 2$		NN1	NN2	NN2	NN1	NN2	NN1



## 2. Pembentukan Data Sintesis

Pada tahap pembentukan data sintesis, dilakukan dengan cara memilih nilai *Nearest Neighbor* secara acak yang paling dekat dari setiap data. Data sintesis dibuat sepanjang garis vektor yang menghubungkan data asli dengan tetangga terdekatnya. Jumlah data sintesis yang dihasilkan bergantung pada nilai  $N$  yang telah ditentukan atau perbedaan antara jumlah data kelas minor dan mayor. *Flowchart* untuk membuat data sintesis baru dapat dilihat pada gambar 3.10 berikut:



**Gambar 3.10 Flowchart Membuat Data Sintesis Baru SMOTE**

Dalam mencari nilai data sintesis baru dilakukan persamaan:

$$X_{syn} = x_i + (x_{knn} - x_i) \times \delta \dots\dots\dots(3.5)$$

Dengan  $\delta$  adalah nilai acak antara 0 sampai 1, digunakan nilainya 0,5. Hasil perhitungan data sintesis baru dapat dilihat pada tabel 3.9 berikut:

**Tabel 3.9 Hasil Data Sintesis *SMOTE***

No.	Term	Data Sintesis Kelas Minor Positif		
		<i>D2syn</i>	<i>D3syn</i>	<i>D4syn</i>
1	analisnya	0.0365	0.0365	0.171297
2	bikin	0.0655	0.153295	0.222052
3	daerah	0.336058	0	0
4	ditawarin	0.0365	0.0365	0.171297
5	gara	0.0665	0.248886	0.0665
6	garden	0.0365	0.0365	0.171297
7	ibukota	0.141529	0.0265	0
8	ikan	0.141529	0.0265	0
9	indonesia	0	0	0
10	integrasi	0.141529	0.0265	0
11	jokowi	0.141529	0.0265	0
12	kalimantan	0.141529	0.0265	0
13	kembang	0.141529	0.0265	0
14	kota	0.060227	0.096176	0.071343
15	maju	0	0	0
16	mindah	0.0365	0.0365	0.171297
17	negara	0.10819	0.160295	0.0165
18	ngelanjutin	0.0365	0.0365	0.171297
19	paksa	0.266	0.630772	0.266
20	pangan	0.141529	0.0265	0
21	park	0.0365	0.0365	0.171297
22	peluang	0.141529	0.0265	0
23	pikir	0.0365	0.0365	0.171297
24	pilih	0.0365	0.0365	0.171297
25	pindah	0	0	0
26	punggung	0.141529	0.0265	0
27	pusing	0.0365	0.0365	0.171297
28	sektor	0.141529	0.0265	0
29	semangat	0	0	0
30	sih	0.0665	0.248886	0.0665
31	sosok	0.0365	0.0365	0.171297
32	tahan	0.141529	0.0265	0
33	taman	0.0365	0.0365	0.171297
34	tani	0.141529	0.0265	0
35	timur	0.141529	0.0265	0
36	tulang	0.141529	0.0265	0

Sehingga hasil dokumen dari data sintesis untuk kelas minor positif dapat dilihat pada tabel 3.10 berikut:

**Tabel 3.10 Dokumen Data Sintesis *SMOTE***

<b>Data Sintesis Minor Positif</b>		
<b><i>D2syn</i></b>	<b><i>D3syn</i></b>	<b><i>D4syn</i></b>
analisnya	analisnya	analisnya
bikin	bikin	bikin
daerah	ditawarin	ditawarin
ditawarin	gara	gara
gara	garden	garden
garden	ibukota	kota
ibukota	ikan	mindah
ikan	integrasi	negara
integrasi	jokowi	ngelanjutin
jokowi	kalimantan	paksa
kalimantan	kembang	park
kembang	kota	pikir
kota	mindah	pilih
mindah	negara	pusing
negara	ngelanjutin	sih
ngelanjutin	paksa	sosok
paksa	pangan	taman
pangan	park	
park	peluang	
peluang	pikir	
pikir	pilih	
pilih	punggung	
punggung	pusing	
pusing	sektor	
sektor	sih	
sih	sosok	
sosok	tahan	
tahan	taman	
taman	tani	
tani	timur	
timur	tulang	
tulang		

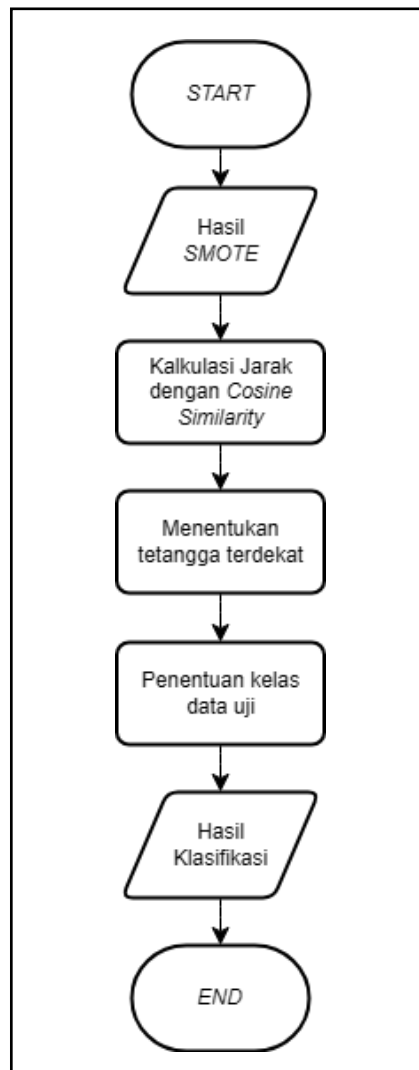
Setelah semua data sintesis diketahui hasil dan nilainya, maka berikut adalah perubahan tabel *TF-IDF* yang terdapat data sintesis:

**Tabel 3.11 Hasil Perubahan *TF-IDF* Setelah *SMOTE***

No.	Term	TF-IDF						
		D1	D2	D3	D4	D5	D6	D7
1	analisnya	0	0	0	0.27	0.036	0.036	0.171
2	bikin	0	0	0.288	0.425	0.066	0.153	0.222
3	daerah	0	0.46	0	0	0.336	0	0
4	ditawarin	0	0	0	0.27	0.036	0.036	0.171
5	gara	0	0	0.365	0	0.067	0.249	0.067
6	garden	0	0	0	0.27	0.036	0.036	0.171
7	ibukota	0	0.23	0	0	0.142	0.027	0
8	ikan	0	0.23	0	0	0.142	0.027	0
9	indonesia	0.484	0	0	0	0	0	0
10	integrasi	0	0.23	0	0	0.142	0.027	0
11	jokowi	0	0.23	0	0	0.142	0.027	0
12	kalimantan	0	0.23	0	0	0.142	0.027	0
13	kembang	0	0.23	0	0	0.142	0.027	0
14	kota	0.252	0.12	0.19	0.141	0.06	0.096	0.071
15	maju	0.484	0	0	0	0	0	0
16	mindah	0	0	0	0.27	0.036	0.036	0.171
17	negara	0	0.183	0.288	0	0.108	0.16	0.017
18	ngelanjutin	0	0	0	0.27	0.036	0.036	0.171
19	paksa	0	0	0.73	0	0.266	0.631	0.266
20	pangan	0	0.23	0	0	0.142	0.027	0
21	park	0	0	0	0.27	0.036	0.036	0.171
22	peluang	0	0.23	0	0	0.142	0.027	0
23	pikir	0	0	0	0.27	0.036	0.036	0.171
24	pilih	0	0	0	0.27	0.036	0.036	0.171
25	pindah	0.484	0	0	0	0	0	0
26	punggung	0	0.23	0	0	0.142	0.027	0
27	pusing	0	0	0	0.27	0.036	0.036	0.171
28	sektor	0	0.23	0	0	0.142	0.027	0
29	semangat	0.484	0	0	0	0	0	0
30	sih	0	0	0.365	0	0.067	0.249	0.067
31	sosok	0	0	0	0.27	0.036	0.036	0.171
32	tahan	0	0.23	0	0	0.142	0.027	0
33	taman	0	0	0	0.27	0.036	0.036	0.171
34	tani	0	0.23	0	0	0.142	0.027	0
35	timur	0	0.23	0	0	0.142	0.027	0
36	tulang	0	0.23	0	0	0.142	0.027	0

### 3.1.6 Analisis Sentimen dengan Model *KNN*

Tahapan ini menentukan hasil analisis sentimen berdasarkan data yang telah melalui proses sebelumnya untuk mendapatkan hasil analisis sentimen menggunakan metode *K-Nearest Neighbor*. Berikut merupakan *flowchart* dari proses analisis sentimen dengan *KNN*.

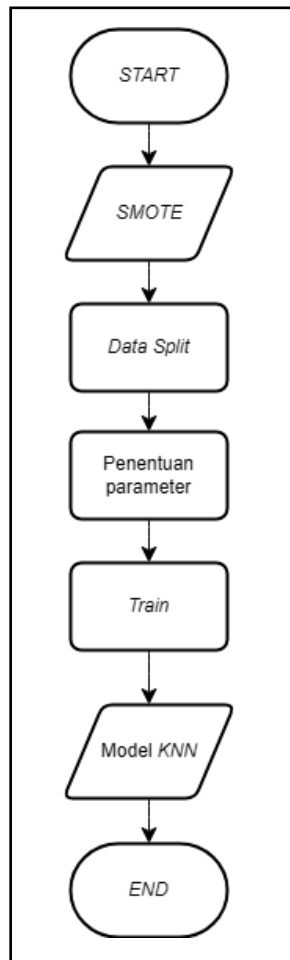


**Gambar 3.11** *Flowchart* Klasifikasi Model *KNN*

Analisis sentimen menggunakan model *KNN* berawal dari *input* data yang sudah melalui tahapan transformasi kata dengan *TF-IDF* untuk membobotkan data teks, sehingga setiap kata diwakili dalam bentuk vektor. Selanjutnya, dilakukan penghitungan jarak antara setiap sampel data uji terhadap seluruh data latih menggunakan *Cosine Similarity* untuk mengukur kemiripan. Berdasarkan hasil dari jarak yang sudah dihitung, data uji akan dikelompokkan dengan tetangga terdekat sejumlah  $k$ . Lalu, dari tetangga terdekat yang telah terpilih, kelas mayoritas akan ditetapkan sebagai prediksi kelas untuk data uji.

### 3.1.7 Pembuatan Model Sentimen

Sebagai cara untuk mendapatkan hasil akhir dari penelitian ini, yaitu klasifikasi sentimen publik terhadap pemindahan ibu kota negara yang terdapat pada media sosial *X (Twitter)*, langkah yang diambil adalah melakukan klasifikasi menggunakan metode *K-Nearest Neighbor (KNN)*. Berikut merupakan *flowchart* pembuatan model sentimen dapat dilihat pada gambar 3.12.



**Gambar 3.12 Flowchart Pembuatan Model Sentimen**

*Flowchart* proses klasifikasi sentimen menggunakan metode *K-Nearest Neighbor* dimulai dengan mengambil hasil dari proses *TF-IDF*, yang digunakan untuk mengekstraksi fitur dari data teks yang akan diklasifikasikan. Setelah itu, dilakukan pembagian data latih dan data uji masing-masing 80% dan 20% dari jumlah keseluruhan data. Kemudian, proses selanjutnya adalah penentuan parameter  $k$  terbaik sebagai variabel penting dalam pembuatan model sentimen dengan *K-Nearest Neighbor*. Hasil akhirnya adalah pengelompokan data ke dalam kategori sentimen yang tepat, yakni positif, netral, atau negatif. Pengulangan proses ini dari awal kembali bisa terjadi jika ada data baru yang perlu diklasifikasikan.

### 3.1.8 Pengujian

Pengujian sistem pada penelitian ini dilakukan menggunakan *confusion matrix* untuk mengevaluasi performa model yang diterapkan. Pengujian ini bertujuan untuk mengukur tingkat kinerja metode klasifikasi *K-Nearest Neighbor* yang digunakan dalam penelitian ini. Beberapa nilai kinerja yang diperoleh dari *confusion matrix* meliputi akurasi, presisi, dan *recall* model. Pada penelitian ini, data dibagi menjadi 80% data latih (*train*) dan 20% data uji (*test*). Pengujian dilakukan dengan menggunakan *library scikit-learn* yang terdapat pada *environment* bahasa pemrograman *python*, yang mana *library* tersebut dapat menghasilkan *confusion matrix* beserta nilai akurasi, presisi, dan *recall* secara otomatis.

## 3.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *Waterfall*. Metode ini merupakan metode yang terdiri dari tahap *requirements analysis*, *system and software design*, *implementation*, dan *system testing*. Berikut merupakan uraian terkait dengan masing-masing proses dari metode *Waterfall* yang digunakan sebagai metode pengembangan sistem:

### 3.2.1 Requirements Analysis

Tahapan awal ini merupakan tahapan yang berfokus pada pencarian kebutuhan apa saja yang bisa digunakan dalam proses pembuatan sistem ini diawali dengan mengidentifikasi kebutuhan keseluruhan yang akan diterapkan serta diimplementasikan menjadi sebuah perangkat lunak atau *software*. Tahap awal ini melibatkan proses pengumpulan data. Opini pengguna *X* yang terdapat pada kolom komentar media sosial *X* terkait dengan *keyword* yang digunakan dikumpulkan menggunakan teknik *scraping*. Kumpulan data opini yang diperoleh kemudian disimpan ke dalam suatu file untuk dianalisis lebih lanjut. Selain itu, pada tahapan ini juga dilakukan pengumpulan referensi studi literatur yang relevan dengan penelitian yang dilakukan.

#### 1. Kebutuhan Fungsional

- a. Kemampuan sistem untuk melakukan *preprocessing* terhadap *input* yang diberikan.
- b. Sistem berhasil melakukan pelabelan data dengan *lexicon*.
- c. Sistem berhasil melakukan vektorisasi teks dengan *TF-IDF*.
- d. Sistem berhasil melakukan klasifikasi data terhadap kelas-kelas analisis sentimen dengan algoritma *K-Nearest Neighbor*.
- e. Sistem mampu untuk memberikan *output* nilai akurasi dari operasi perhitungan dengan seluruh *input* yang didapatkan.

## 2. Kebutuhan Non Fungsional

Implementasi sistem yang akan dibangun sesuai dengan rancangan yang telah dibuat memerlukan beberapa perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna sebagai dukungan dalam pengembangan sistem analisis. Berikut adalah analisis kebutuhan non-fungsional yang diperlukan untuk merancang sistem:

### a. Kebutuhan *Hardware*

Berikut merupakan spesifikasi *hardware* yang akan digunakan dalam pembuatan sistem pada penelitian ini:

**Tabel 3.12 Spesifikasi *Hardware***

No.	Hardware	Keterangan
1.	<i>Processor</i>	AMD Ryzen 5 5600H
2.	<i>RAM</i>	16 GB
3.	<i>SSD</i>	512 GB
4.	Perangkat Input dan Output	<i>Mouse, Keyboard, Trackpad</i>
5.	Koneksi	<i>WiFi, Seluler, LAN</i>

### b. Kebutuhan *Software*

Berikut merupakan daftar *software* yang dipakai saat berjalannya proses penelitian:

**Tabel 3.13 *Software***

No.	Software	Keterangan
1.	<i>Operating System</i>	Windows 11 64 bit, Ubuntu Linux
2.	<i>IDE</i>	Visual Studio Code, Google Colab, Kaggle Notebook
3.	Peramban	Google Chrome
4.	<i>Programming Language</i>	Python 3.12.4, Python 3.11.7
5.	Grafis Diagram	<a href="https://draw.io">draw.io</a>

## 3.2.2 System and Software Design

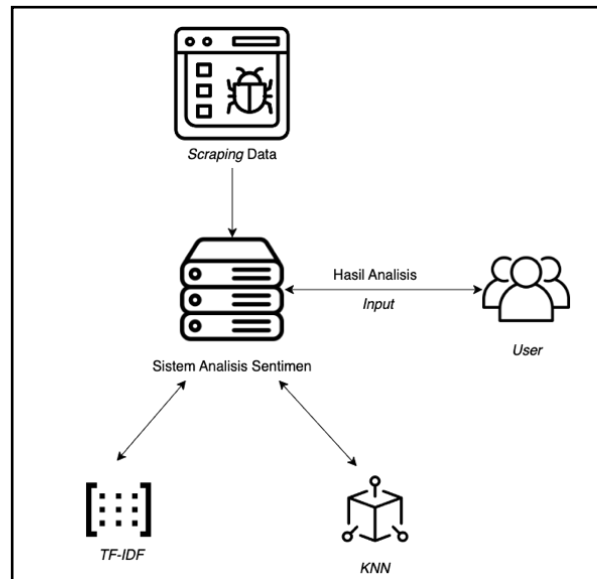
Merupakan tahapan untuk melakukan perancangan mengenai sistem dan juga perangkat lunak yang akan dibangun dan digunakan dalam penelitian ini. Penting dan krusial merupakan alasan utama mengapa diperlukannya tahap ini. Adapun tahapannya sebagai berikut:

### 1. Perancangan Arsitektur

Beberapa arsitektur sistem yang akan dikembangkan dalam penelitian ini mencakup elemen-elemen seperti *user*, *web scraper*, model *KNN*, pelabelan sentimen berbasis *lexicon*, serta sistem analisis sentimen terhadap pemindahan ibu kota negara Indonesia. *User* dalam sistem ini adalah individu yang dapat menggunakan atau mengoperasikan sistem secara umum serta melakukan pengecekan terhadap *input* untuk mengetahui hasil analisis sentimennya.



Pada sistem ini, terdapat beberapa tahap di mana *dataset* komentar publik terhadap pemindahan ibu kota negara diperoleh melalui proses *scraping* dari media sosial *X (Twitter)*. Data tersebut kemudian melalui tahapan *preprocessing* dan pelabelan sentimen menggunakan kamus *lexicon*. Setelah tahapan sebelumnya selesai, sistem melanjutkan dengan proses penganalisisan sentimen dengan model *KNN* yang telah melalui proses pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*). Ketika model *KNN* sudah siap digunakan, sistem akan menampilkan hasil analisis sentimen untuk *user*. Ilustrasi arsitektur sistem dapat dilihat pada Gambar 3.13 berikut.



**Gambar 3.13 Arsitektur Sistem**

## 2. Perancangan Proses

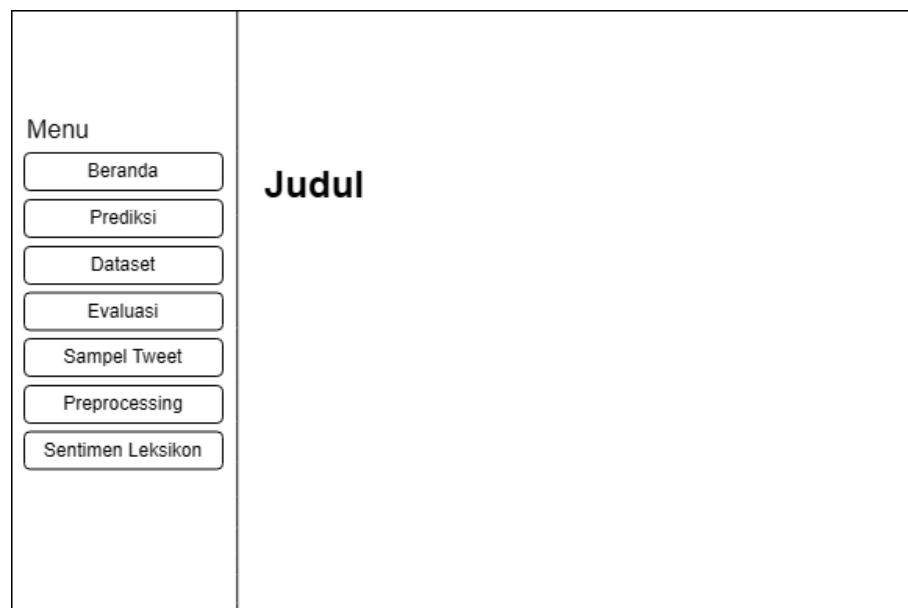
Perancangan ini menggambarkan tahapan dalam analisis sentimen pada data teks menggunakan algoritma klasifikasi *K-Nearest Neighbor*, pelabelan sentimen berbasis *lexicon*, serta pembobotan kata dengan *TF-IDF*. Tahapan diawali dengan pengumpulan data unggahan teks dari media sosial *X* melalui teknik *scraping*. Setelah data didapatkan, dilakukan *preprocessing* terhadap data berfungsi sebagai penghilangan elemen-elemen yang tidak diperlukan atau berpotensi menjadi variabel yang membingungkan model. Kemudian, data yang telah melalui *preprocessing* diberi bobot menggunakan metode *TF-IDF*, yang menonjolkan kata-kata yang relevan dengan memberikan bobot yang lebih tinggi pada kata yang sering muncul di dokumen tertentu tetapi jarang di seluruh korpus. Hasil dari pembobotan tersebut kemudian digunakan sebagai *input* bagi model *KNN* untuk mengklasifikasikan sentimen dalam data tersebut. Tahapan akhir adalah klasifikasi sentimen, dimana model *KNN* yang telah dilatih mengelompokkan data ke dalam sentimen positif, netral, atau negatif. Setelah klasifikasi selesai, seluruh prosesnya berakhir.

### 3. Perancangan *Interface*

Rancangan *interface* atau antarmuka pengguna adalah proses desain yang berfokus pada pengembangan model komunikasi antara pengguna dan sistem. Rancangan ini mencakup berbagai aspek, seperti tata letak dan navigasi, agar pengguna dapat berinteraksi dengan sistem secara mudah dan efisien. Terdapat satu aktor dalam sistem ini, yaitu *user* atau pengguna. Berikut ini merupakan gambaran rancangan antarmuka pengguna yang terbagi ke dalam beberapa menu sebagai berikut:

#### a. Rancangan Halaman Beranda

Halaman ini merupakan halaman awal sistem, dapat dilihat pada gambar 3.14 berikut:



**Gambar 3.14 Rancangan Halaman Beranda**

b. Rancangan Halaman Prediksi

Halaman ini digunakan untuk melakukan prediksi sentimen dari *input* yang diberikan oleh *user* atau pengguna, dapat dilihat pada gambar 3.15 berikut:

Menu

- Beranda
- Prediksi
- Dataset
- Evaluasi
- Sampel Tweet
- Preprocessing
- Sentimen Leksikon

Lorem Ipsum

Prediksi

**Gambar 3.15 Rancangan Halaman Prediksi Sentimen**

c. Rancangan Halaman *Dataset*

Halaman *dataset* memungkinkan pengguna untuk melihat hasil dari setiap langkah yang dilewati *dataset*, dari mulai *dataset* yang masih mentah sampai ke *dataset* yang sudah dilakukan pelabelan sentimen. Antarmuka halaman *dataset* dapat dilihat pada gambar 3.16.

Menu

- Beranda
- Prediksi
- Dataset
- Evaluasi
- Sampel Tweet
- Preprocessing
- Sentimen Leksikon

Lorem Ipsum

Tabel		

**Gambar 3.16 Rancangan Halaman *Dataset***

d. Rancangan Halaman Evaluasi

Halaman ini menampilkan hasil evaluasi model *KNN* dari parameter yang bisa ditentukan di dalam halaman.

The wireframe shows a web page layout for evaluation. On the left is a vertical sidebar with a 'Menu' header and eight buttons: 'Beranda', 'Prediksi', 'Dataset', 'Evaluasi', 'Sampel Tweet', 'Preprocessing', and 'Sentimen Leksikon'. The 'Evaluasi' button is highlighted with a red border. The main content area on the right contains a 'Lorem Ipsum' text block, a 'Parameter' label above a dropdown menu, a 'Metriks' label above three empty rectangular boxes, and labels 'Akurasi', 'Presisi', and 'Recall' positioned below each box. At the bottom of the main area is a 'Confusion Matrix' label above a large empty rectangular box.

**Gambar 3.17 Rancangan Halaman Evaluasi**

e. Rancangan Halaman Sampel *Tweet*

Menu ini memungkinkan pengguna untuk dapat mengambil kurang lebih 5 sampel data unggahan teks dari media sosial *X* sesuai dengan kueri atau kata kunci yang dimasukkan pengguna pada kolom *text box* yang tersedia.

<div>Menu</div> <div><div>Beranda</div><div>Prediksi</div><div>Dataset</div><div>Evaluasi</div><div>Sampel Tweet</div><div>Preprocessing</div><div>Sentimen Leksikon</div></div>	<div> Lorem Ipsum</div> <div><div></div></div> <div><div>Fetch</div></div>
--	--

**Gambar 3.18 Rancangan Halaman Sampel *Tweet***

f. Rancangan Halaman *Preprocessing*

Halaman ini berfungsi sebagai simulasi tahapan-tahapan *preprocessing* data teks dengan cara memberikan *input* pada kolom *text box* yang tersedia.

<div>Menu</div> <div><div>Beranda</div><div>Prediksi</div><div>Dataset</div><div>Evaluasi</div><div>Sampel Tweet</div><div>Preprocessing</div><div>Sentimen Leksikon</div></div>	<div> Lorem Ipsum</div> <div><div></div></div> <div><div>Proses</div></div>
--	---

**Gambar 3.19 Rancangan Halaman *Preprocessing***

g. Rancangan Halaman Sentimen Leksikon

Halaman ini berfungsi sebagai simulasi pelabelan sentimen suatu kalimat dengan cara memberikan *input* pada kolom *text box* yang tersedia.

<p>Menu</p> <p>Beranda</p> <p>Prediksi</p> <p>Dataset</p> <p>Evaluasi</p> <p>Sampel Tweet</p> <p>Preprocessing</p> <p>Sentimen Leksikon</p>	<p>Lorem Ipsum</p> <p></p> <p>Proses</p>
---	--

**Gambar 3.20 Rancangan Halaman Sentimen Leksikon**

### 3.2.3 Implementation

Tahapan ini merupakan fase ketika desain dan perancangan sistem yang telah dibuat akan direalisasikan menjadi kode program untuk menghasilkan sebuah sistem yang berfungsi. Sistem analisis sentimen ini diimplementasikan dan dirancang sesuai dengan metode yang digunakan berdasarkan tahapan analisis sentimen untuk memprediksi atau klasifikasi sentimen publik terhadap pemindahan ibu kota negara ke dalam tiga kelas yaitu positif, netral, dan negatif.

### 3.2.4 System Testing

Tahapan akhir ini merupakan tahapan untuk pengujian sistem. Pengujian ini bertujuan untuk menguji rancangan sistem yang sudah dibangun sebelumnya agar dapat melakukan evaluasi dari model yang digunakan untuk proses klasifikasi yang sudah di jalankan. *Black box* merupakan metode yang digunakan untuk tahapan *system testing* ini.

Metode *black box testing* memiliki tujuan untuk mengetahui efektivitas, keberhasilan serta kemampuan kerja sistem yang dikembangkan. Selain itu juga digunakannya metode *black box* bertujuan untuk memastikan fungsi setiap fitur yang terdapat pada sistem yang sudah dibangun sesuai dengan yang seharusnya. Berikut detail dari hasil pengujian sistem menggunakan metode *black box testing*:

**Tabel 3.14 Rancangan Pengujian *Black Box***

Halaman	Detail Pengujian	Berhasil	Tidak Berhasil
<i>Dataset</i>	Menampilkan perubahan <i>dataset</i> dari setiap proses		
Evaluasi	Menampilkan informasi hasil dari evaluasi model masing-masing parameter		
Prediksi	Menampilkan prediksi sentimen dari <i>input</i> yang diberikan		
Sampel <i>Tweet</i>	Menampilkan 5 <i>tweets</i> berdasarkan kueri pencarian yang diberikan		
<i>Preprocessing</i>	Menampilkan hasil <i>preprocessing</i> dari <i>input</i> yang diberikan		
Sentimen Leksikon	Menampilkan hasil pelabelan sentimen <i>InSet</i> dari <i>input</i> yang diberikan		

## DAFTAR PUSTAKA

- Abdillah, T. (2023). *KOMPARASI METODE NAÏVE BAYES DAN K-NEAREST NEIGHBORS TERHADAP ANALISIS SENTIMEN PENGGUNA APLIKASI ZENIUS*.
- Afrillia, Y., Rosnita, L., Siska, D., Rigayatsyah, M., & Nurqamarina, N. (2022). Analisis Sentimen Ciutan Twitter Terkait Penerapan Permendikbudristek Nomor 30 Tahun 2021 Menggunakan TextBlob dan Support Vector Machine. *G-Tech: Jurnal Teknologi Terapan*, 6(2), 387–394. <https://doi.org/10.33379/gtech.v6i2.1778>
- Akbar, W., Astuti, E. S., & Riyadi, R. (2013). *PENERIMAAN DAN PENGGUNAAN SITUS JEJARING SOSIAL TWITTER DI LINGKUNGAN MAHASISWA DENGAN PENDEKATAN TECHNOLOGY ACCEPTANCE MODEL (TAM)*. 6(1).
- Alamsyah, A. A. F., & Mulyati, S. (2023). *IMPLEMENTASI ALGORITME K-NEAREST NEIGHBOUR DAN LEXICON BASED UNTUK ANALISIS SENTIMEN KEPUASAN PENGGUNA APLIKASI GRAMEDIA DIGITAL PADA MEDIA SOSIAL TWITTER* (Vol. 2, Issue 2).
- Amalia, A., Lydia, M. S., Fadilla, S. D., & Huda, M. (2018). Perbandingan Metode Kluster dan Preprocessing untuk Dokumen Berbahasa Indonesia. *Jurnal Rekayasa ElektriKa*, 14(1), 35–42. <https://doi.org/10.17529/jre.v14i1.9027>
- Angel, A. C. T., Pranatawijaya, V. H., & Widiatry, W. (2024). *Analisis Sentimen dan Emosi dari Ulasan Google Maps untuk Layanan Rumah Sakit di Palangka Raya Menggunakan Machine Learning* (Vol. 4, Issue 1).
- Ayani, D. D., Pratiwi, H. S., & Muhandi, H. (2019). *Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace*. 7(4), 257–262.
- Azhar. (2019). *ANALISIS KINERJA ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBOR PADA SENTIMEN ANALISIS DENGAN PENDEKATAN LEXICON DI MEDIA TWITTER*.
- Barus, S. G. (2022). *KLASIFIKASI SENTIMEN DATA TIDAK SEIMBANG MENGGUNAKAN ALGORITMA SMOTE DAN K-NEAREST NEIGHBOR PADA ULASAN PENGGUNA APLIKASI PEDULILINDUNGI*. 162–173.
- Boeing, G., & Waddell, P. (2016). New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research*, 37(4), 457–476. <https://doi.org/10.1177/0739456X16664789>



- Elfansyah, M. R., Rudiman, R., & Yulianto, F. (2024). *PERBANDINGAN METODE K – NEAREST NEIGHBOR (KNN) DAN NAIVE BAYES TERHADAP ANALISIS SENTIMEN PADA PENGGUNA E-WALLET APLIKASI DANA MENGGUNAKAN FITUR EKSTRAKSI TF-IDF*. 18(2), 139–159. <https://doi.org/https://doi.org/10.47111/JTI>
- Fajri, M. S., Septian, N., & Sanjaya, E. (2020). Evaluasi Implementasi Algoritma Machine Learning K-Nearest Neighbors (kNN) pada Data Spektroskopi Gamma Resolusi Rendah. *Al-Fiziya: Journal of Materials Science, Geophysics, Instrumentation and Theoretical Physics*, 3(1), 9–14. <https://doi.org/10.15408/fiziya.v3i1.16180>
- Hadi, F., & Ristawati, R. (2020). Pemindahan Ibu Kota Indonesia dan Kekuasaan Presiden dalam Perspektif Konstitusi. *Jurnal Konstitusi*, 17(3), 530–557. <https://doi.org/10.31078/jk1734>
- Ismail, A. R., & Hakim, R. B. F. (2023). Implementasi Lexicon Based Untuk Analisis Sentimen Dalam Mengetahui Trend Wisata Pantai Di DI Yogyakarta Berdasarkan Data Twitter. In *Emerging Statistics and Data Science Journal* (Vol. 1, Issue 1).
- Koto, F., & Rahmaningtyas, G. Y. (2017). InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs. *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017, 2018-January*, 391–394. <https://doi.org/10.1109/IALP.2017.8300625>
- Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
- Matulatuwa, F. M., Sedyono, E., & Iriani, A. (2017). *TEXT MINING DENGAN METODE LEXICON BASED UNTUK SENTIMENT ANALYSIS PELAYANAN PT. POS INDONESIA MELALUI MEDIA SOSIAL TWITTER* (Vol. 2, Issue 3).
- Permatasari, P. A., Linawati, L., & Jasa, L. (2021). Survei Tentang Analisis Sentimen Pada Media Sosial. *Majalah Ilmiah Teknologi Elektro*, 20(2), 177–186. <https://doi.org/10.24843/mite.2021.v20i02.p01>
- Pramayasa, K., Maysanjaya, I. M. D., & Indradewi, I. G. A. A. D. (2023). Analisis Sentimen Program Mbkm Pada Media Sosial Twitter Menggunakan KNN Dan SMOTE. *SINTECH JOURNAL*, 6(2), 89–98. <https://doi.org/doi.org/10.31598>
- Pratomo, S. A., Faraby, S. al, & Purbolaksono, M. D. (2021). *Analisis Sentimen Pengaruh Kombinasi Ekstraksi Fitur TF-IDF dan Lexicon Pada Ulasan Film Menggunakan Metode KNN*.
- Punch, K. F. (1998). *Introduction to Social Research: Quantitative & Qualitative Approaches*.

- Putri, D. S., Sulistiyowati, N., & Voutama, A. (2023). Analisis Sentimen dan Pemodelan Ulasan Aplikasi AdaKami Menggunakan Algoritma SVM dan KNN. *Journal Sensi*, 9(2), 209–225.
- Rahayu, S., MZ, Y., Bororing, J. E., & Hadiyat, R. (2022). Implementasi Metode K-Nearest Neighbor (K-NN) untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP. *Edumatic: Jurnal Pendidikan Informatika*, 6(1), 98–106. <https://doi.org/10.29408/edumatic.v6i1.5433>
- Ramadhon, M. I. (2020). *ANALISIS SENTIMEN TERHADAP PEMINDAHAN IBU KOTA INDONESIA PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE ALGORITMA K-NEAREST NEIGHBOR (K-NN)*.
- Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. <http://arxiv.org/abs/1811.12808>
- Sandi, D., Utami, E., & Kusnawi, K. (2023). Analisis Sentimen Publik Terhadap Elektabilitas Ganjar Pranowo di Tahun Politik 2024 di Twitter dengan Algoritma KNN dan Naïve Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 7(3), 1097–1108. <https://doi.org/10.30865/mib.v7i3.6298>
- Sari, R. (2020). Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi menggunakan Algoritma K-Nearest Neighbor (K-NN). *Jurnal Sains Dan Manajemen*, 8(1), 10–17. [www.tripadvisor.com](http://www.tripadvisor.com).
- Setiawan, A., Santoso, L. W., & Adipranata, R. (2020). *Klasifikasi Artikel Berita Bahasa Indonesia Dengan Naive Bayes Classifier*.
- Sholeha, E. W., Yunita, S., Hammad, R., Hardita, V. C., & Kaharuddin, K. (2022). Analisis Sentimen Pada Agen Perjalanan Online Menggunakan Naïve Bayes dan K-Nearest Neighbor. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, 3(4), 203–208.
- Siringoringo, R. (2018). *KLASIFIKASI DATA TIDAK SEIMBANG MENGGUNAKAN ALGORITMA SMOTE DAN k-NEAREST NEIGHBOR* (Vol. 3, Issue 1).
- Wahid, A. A. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*. <https://www.researchgate.net/publication/346397070>
- Wahyuni, R. T., Prastiyanto, D., & Suprpto, E. (2017). *Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi*.