

**ANALISIS REAKSI PUBLIK TERHADAP PEMINDAHAN IBU KOTA NEGARA
MENGGUNAKAN METODE *LEXICON BASED* DAN *KNN***

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

ANALISIS REAKSI PUBLIK TERHADAP PEMINDAHAN IBU KOTA NEGARA MENGGUNAKAN METODE *LEXICON BASED* DAN *KNN*

Disusun Oleh:

Rifqi Maulana

123200128

Telah diperiksa dan disetujui oleh pembimbing

pada tanggal :

Menyetujui,
Pembimbing

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

NIP 19830708 202121 1001

Mengetahui,
Koordinator Program Studi

Wilis Kaswidjanti, S.Si., M.Kom

NIDN. 0513047601

DAFTAR ISI

COVER	1
HALAMAN PENGESAHAN	2
DAFTAR ISI	3
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian dan Pengembangan Sistem	3
1.6.1 Metode Pengumpulan Data	4
1.6.2 Metode Pengembangan Sistem	4
1.7 Sistematika Penulisan	4
BAB II TINJAUAN LITERATUR	6
2.1 Analisis Sentimen	6
2.2 Analisis Penerapan <i>K-Nearest Neighbor</i>	6
2.3 <i>X (Twitter)</i>	6
2.4 <i>Scraping</i>	7
2.5 <i>Pre-processing</i>	7
2.5.1 <i>Cleansing</i> dan <i>Case Folding</i>	4
2.5.2 <i>Tokenization</i>	4
2.5.3 <i>Stopwords Removal</i>	4
2.5.4 <i>Stemming</i>	4
2.6 <i>Lexicon Based</i>	8
2.7 <i>K-Nearest Neighbor (KNN)</i>	9
2.8 Validasi dan Pengujian	12
2.9 Studi Pustaka	14
BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM	16
3.1 Metodologi Penelitian	16
3.1.1 Metode Analisis Data	17
3.1.1.1 Analisis Masalah.....	17
3.1.1.2 Pengumpulan Data.....	17

3.1.1.3	<i>Pre-processing Data</i>	17
3.1.1.4	Pelabelan Data.....	17
3.1.1.5	Pembobotan <i>Term-Invers Document Frequency</i> (TF-IDF)	17
3.1.1.6	Klasifikasi Model dengan <i>K-Nearest Neighbor (KNN)</i>	17
3.1.1.6	Pengujian.....	17
3.1.2	Metode Pengembangan Sistem	30
3.1.2.1	<i>Requirement Analysis</i>	17
3.1.2.2	<i>System and Software Design</i>	17
3.1.2.3	<i>Implementation</i>	17
3.1.2.4	<i>System Testing</i>	17

DAFTAR GAMBAR

Gambar 3. 1 Tahapan Penelitian	16
Gambar 3. 2 Flowchart Preprocessing	18
Gambar 3. 3 Flowchart <i>Cleansing</i> dan <i>Case Folding</i>	19
Gambar 3. 4 Flowchart <i>Tokenizing</i>	20
Gambar 3. 5 Flowchart <i>Stopword Removal</i>	21
Gambar 3. 6 Flowchart <i>Stemming</i>	22
Gambar 3. 7 Flowchart TF-IDF	23
Gambar 3. 8 Flowchart Pelatihan SVM	27
Gambar 3. 9 Arsitektur Sistem	31
Gambar 3. 10 Flowchart Perancangan Proses	32
Gambar 3. 11 Rancangan Halaman List Rekomendasi Coffee Shop.....	33
Gambar 3. 12 Rancangan halaman Detail Ulasan	33
Gambar 3. 13 Rancangan Halaman Informasi Coffee Shop	34
Gambar 3. 14 Rancangan Halaman Login	34
Gambar 3. 15 Rancangan Halaman Dashboard	35
Gambar 3. 16 Rancangan Halaman Data <i>Coffee Shop</i>	35
Gambar 3. 17 Rancangan Halaman Fitur Tambah Data <i>Coffee Shop</i>	36
Gambar 3. 18 Rancangan Halaman Dataset	36
Gambar 3. 19 Rancangan Halaman Uji Klasifikasi	37
Gambar 3. 20 Rancangan Halaman Data Hasil Pengujian	37
Gambar 3. 21 Rancangan Halaman Model SVM	38

DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i>	12
Tabel 2.2 Studi Pustaka	14
Tabel 3.1 Contoh Penggunaan <i>Cleansing</i> dan <i>Case Folding</i>	19
Tabel 3.2 Contoh Penggunaan <i>Tokenizing</i>	20
Tabel 3.3 Contoh Penggunaan <i>Stopword Removal</i>	21
Tabel 3.4 Contoh Penggunaan <i>Stemming</i>	22
Tabel 3.5 Contoh Hasil Pelabelan Data	23
Tabel 3.6 Dokumen	23
Tabel 3.8 Contoh Hasil Perhitungan TF	24
Tabel 3.9 Contoh Hasil Perhitungan Bobot IDF	25
Tabel 3.10 Contoh Hasil Perhitungan TF-IDF	25
Tabel 3.11 Lanjutan Contoh Hasil Perhitungan TF-IDF	26
Tabel 3.12 Nilai pada Matriks K	27
Tabel 3.13 Nilai pada Matriks Hessian	28
Tabel 3.14 Hasil Perhitungan Ei	28
Tabel 3.15 Hasil Perhitungan $\delta\alpha_i$	28
Tabel 3.16 Hasil Perhitungan α_i baru	28
Tabel 3.17 Hasil Perhitungan Ei iterasi ke-2	29
Tabel 3.18 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke-2	29
Tabel 3.19 Hasil Perhitungan α_i baru	29
Tabel 3.20 Rancangan <i>Confusion Matrix</i>	30
Tabel 3.21 Rancangan <i>K-Fold Cross Validation</i>	30
Tabel 3.22 Kebutuhan <i>Hardware</i>	31
Tabel 3.23 Kebutuhan <i>Software</i>	31
Tabel 3.24 Rancangan Pengujian <i>Black Box</i>	38
Tabel 3.25 Lanjutan Rancangan Pengujian <i>Black Box</i>	39

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*, 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 karena terdapat fitur *noise*, namun terdapat beberapa penelitian yang menyebutkan bahwa performansi pada metode *machine learning* dapat menghasilkan performa yang baik ketika dikombinasikan dengan ekstraksi fitur dan seleksi fitur yang tepat (Pratomo et al., 2021). 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, solusi yang penulis gunakan adalah metode klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen *Lexicon Based* 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 leksikon dan *TF-IDF*, lalu terakhir yaitu tahap klasifikasi. Penulis memilih beberapa teknologi yang telah disebutkan sebelumnya karena penulis sudah melakukan tinjauan 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 yang tidak lebih tinggi dibandingkan dengan hanya menggunakan fitur ekstraksi *TF-IDF* yaitu 73.31%, sedangkan dengan *TF-IDF* saja mendapatkan 81.04%, dan penggunaan fitur seleksi *Information Gain (IG)* dengan *threshold* yang tepat mampu mengoptimasi hasil performansi pada metode klasifikasi *KNN*. 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, penulis berharap penelitian ini dapat memberikan kontribusi dalam memahami bagaimana mengoptimalkan metode klasifikasi *K-Nearest Neighbors* dengan melakukan beberapa penyesuaian yang tepat dari mulai pra pengolahan dataset 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 leksikon yang telah dikustomisasi, serta mencari nilai k yang optimal sebagai variabel penting dalam optimasi metode klasifikasi *KNN*, tidak lupa juga menyertakan ekstraksi fitur *TF-IDF* untuk memperoleh prediksi sentimen dengan akurasi tinggi. Selain itu, penulis berharap penelitian ini dapat memberikan kontribusi kecil 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 penulis mencakup bagaimana melakukan analisis sentimen menggunakan metode klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen *Lexicon Based* dengan beberapa penyesuaian dapat membantu dalam memahami bagaimana optimasi metode tersebut untuk mendapatkan hasil akhir akurasi yang tinggi serta pandangan dan sikap masyarakat atau publik terhadap keputusan Presiden terkait pemindahan ibu kota negara Indonesia?

1.3 Batasan Masalah

Agar masalah yang diteliti menghasilkan sasaran yang jelas, maka dibuatlah batasan masalah untuk menghindari adanya perluasan pembahasan kedepannya 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. *Web scraping* terhadap unggahan teks terbaru pada media sosial *X (Twitter)* dilakukan pada tanggal 21 Agustus 2024 dengan kata kunci: ikn, pemindahan ibu kota, ibu kota nusantara, ibu kota baru, dan ibu kota pindah.
3. *Web scraping* dilakukan dengan kueri seperti “*since:2023-01-01*”, “*lang:id*”, serta fitur “*Latest*” pada kolom pencarian teks yang terdapat di media sosial *X (Twitter)*
4. Dataset yang digunakan dalam penelitian berjumlah total 4593 baris unggahan teks memakai format *.csv*.
5. *Preprocessing* data mencakup pembersihan duplikasi baris data, pembersihan teks, integrasi pembakuan kata dari kata slang menjadi kata formal, integrasi *stopwords*, dan *stemming*.
6. Kategori sentimen dibagi menjadi 3 yaitu positif, negatif, serta netral.
7. Kategori sentimen teks ditentukan oleh skor yang dihasilkan dari proses pelabelan menggunakan leksikon yang mana teks yang memiliki skor sentimen lebih dari 0 akan bernilai positif, kurang dari 0 akan bernilai negatif, dan sama dengan 0 akan bernilai netral.
8. Pelabelan sentimen dilakukan dengan pendekatan *Lexicon Based*.
9. Leksikon yang digunakan adalah 2 leksikon yang berbahasa Indonesia yaitu *InSet (Indonesia Sentiment Lexicon)* dari literatur yang disusun oleh Fajri Koto dan Gemala Y. Rahmaningtyas pada tahun 2017 dengan judul “*InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs*” serta leksikon *sentistrength_id* dari literatur yang disusun oleh Wahid dan Azhari pada tahun 2016 dengan judul “*Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity*”. Leksikon *InSet* terdiri atas 3,609 kata positif dan 6,609 kata negatif dengan bobot antara -5 sampai +5.
10. Menggunakan fitur pembobotan kata *TF-IDF*.
11. Rasio perbandingan pembagian data *train* dan data *test* adalah 80:20. Data *train* digunakan untuk pembentukan model klasifikasi (Arifiyanti & Wahyuni, 2020).

12. Percobaan beberapa nilai k yakni $k1, k3, k5, k7$, dan $k9$ untuk melatih model *K-Nearest Neighbors*.
13. Metode evaluasi model yang digunakan adalah *k-fold cross validation, confusion matrix, accuracy, precision, recall*, serta *f1-score*.
14. Model *KNN* yang dilatih mencakup 2 leksikon yaitu *InSet* dan *sentistrength_id* dengan masing-masing leksikon memiliki model *KNN* dari nilai k yang berbeda-beda ($k1, k3, k5, k7$, dan $k9$).

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah mengimplementasikan serta mengoptimasi metode klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen *Lexicon Based* dalam melakukan analisa sentimen masyarakat 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 kecil dalam pengembangan metodologi analisis sentimen berbasis teks, khususnya dalam penerapan algoritma klasifikasi *K-Nearest Neighbors* dengan pelabelan sentimen *Lexicon Based* 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 Pengumpulan Data

Penelitian ini menggunakan teknik *web scraping* untuk mengumpulkan data teks yang berkaitan dengan pemindahan ibu kota negara dari platform media sosial *X (Twitter)*.

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	PENDAHULUAN Pada bagian pendahuluan membahas mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.
BAB II	TINJAUAN PUSTAKA 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	METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM Pada bab ini membahas mengenai metodologi penelitian, analisis sistem dan perancangan sistem analisis sentimen.
BAB IV	HASIL, PENGUJIAN, DAN PEMBAHASAN 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	KESIMPULAN DAN SARAN 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 ambigu 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)

K-Nearest 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:

$$TF = \frac{\text{jumlah kata terpilih}}{\text{jumlah kata}}$$

Nilai *IDF* dapat dihitung dengan rumus:

$$IDF = \frac{\text{jumlah dokumen}}{\text{jumlah frekuensi kata terpilih}}$$

Nilai *TF-IDF*:

$$TFIDF = TF \times IDF$$

2.3 X (*Twitter*)

X adalah sebuah situs *web* yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa *microblog* sehingga memungkinkan pengguna 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* (Ramadhan, 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 dinamis 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)*, lalu 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 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*:

2.6.1 Cleansing

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*, serta menghilangkan angka-angka.

2.6.2 Case Folding

Case folding merupakan tahapan yang bertujuan untuk mengubah semua huruf yang terdapat pada dokumen menjadi huruf kecil. Huruf ‘a’ sampai ‘z’ yang akan diterima, apabila pada dokumen terdapat karakter selain huruf maka akan dihilangkan dan dianggap *delimiter*. Tahap *case folding* akan menghasilkan kalimat yang sudah rapi dan akan memudahkan proses selanjutnya.

2.6.3 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. Sehingga, proses *tokenization* pada dokumen bergantung pada karakter spasi. Sebagai contoh, tokenisasi pada kalimat “ibu kota pindah” menghasilkan empat *token*, yaitu “ibu”, “kota”, dan “pindah”.

2.6.4 Normalisasi

Proses ini dilakukan untuk merubah kata-kata singkatan atau slang dalam bahasa Indonesia menjadi kata baku.

2.6.5 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.

2.6.6 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*.

2.6 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).

$$S_{positive} \sum_{i \in t}^n positive score_i$$

$$S_{negative} \sum_{i \in t}^n negative score_i$$

$S_{positive}$ 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).

$$Sentence_{sentiment} \left\{ \begin{array}{l} positive \text{ if } S_{positive} > S_{negative} \\ neutral \text{ if } S_{positive} = S_{negative} \\ negative \text{ if } S_{positive} < S_{negative} \end{array} \right\}$$

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).

2.7 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, 2016). Adapun tahapan proses yang dilakukan pada *KNN* adalah sebagai berikut (Abdillah, 2023):

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*:

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

Keterangan:

$\cos(\theta_{QD})$: kemiripan Q terhadap dokumen D

Q : data uji

D : data latih

n : jumlah data latih

2.8 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* yang nantinya dapat digunakan untuk pengukuran nilai akurasi, presisi, serta *recall*. Dengan melakukan pengukuran Tingkat akurasi maka dapat mengetahui seberapa baik performa model klasifikasi tersebut. Bentuk dari *confusion matrix* adalah tabel dengan empat kombinasi yang berbeda antara nilai prediksi dan nilai aktual. Metode *confusion matrix* mempunyai empat kemungkinan yang merepresentasikan hasil dari proses klasifikasi, yaitu TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*).

Berikut merupakan tabel *confusion matrix*:

Tabel 2.1 Confusion Matrix

		Predicted		
		-1 (Negative)	+1 (Positive)	0 (Neutral)
Actual	-1 (Negative)	TN	FP	FL2
	+1 (Positive)	FN	TP	FP2
	0 (Neutral)	FN2	FP2	TL

Keterangan:

TN (*True Negative*) : Prediksi benar bernilai negatif

TP (*True Positive*) : Prediksi benar bernilai positif

TL (*True Neutral*) : Prediksi benar bernilai netral

FN (<i>False Negative</i>)	: Negatif terprediksi positif
FP (<i>False Positive</i>)	: Positif terprediksi negatif
FL (<i>False Neutral</i>)	: Positif terprediksi netral
FN2 (<i>False Negative 2</i>)	: Netral terprediksi negatif
FP2 (<i>False Positive 2</i>)	: Netral terprediksi positif
FL2 (<i>False Neutral 2</i>)	: Negatif 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:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + T}$$

2. Recall

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

$$Recall\ Positive = \frac{TP}{TP + FN + FL}$$

$$Recall\ Negative = \frac{TP}{TP + FN + FL2}$$

$$Recall\ Neutral = \frac{TP}{TP + FN + FP2}$$

$$Recall = \frac{Recall\ Positive + Recall\ Negative + Recall\ Neutral}{3} \times 100\%$$

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:

$$\text{Precision Positive} = \frac{TP}{TP + FP + FN}$$

$$\text{Precision Negative} = \frac{TP}{TP + FP + FN}$$

$$\text{Precision Neutral} = \frac{TP}{TP + FP + FN}$$

$$\text{Precision} = \frac{\text{Precision Positive} + \text{Precision Negative} + \text{Precision Neutral}}{3} \times 100\%$$

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*:

$$F1 \text{ Score Positive} = \frac{\text{Precision Positive} \times \text{Recall Positive}}{\text{Precision Positive} + \text{Recall Positive}}$$

$$F1 \text{ Score Negative} = \frac{\text{Precision Negative} \times \text{Recall Negative}}{\text{Precision Negative} + \text{Recall Negative}}$$

$$F1 \text{ Score Neutral} = \frac{\text{Precision Neutral} \times \text{Recall Neutral}}{\text{Precision Neutral} + \text{Recall Neutral}}$$

$$F1 \text{ Score} = \frac{F1 \text{ Score Positive} + F1 \text{ Score Negative} + F1 \text{ Score Neutral}}{3} \times 100\%$$

2.9 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> .	<p>Data yang diberi label model <i>Lexicon</i>, akurasi <i>KNN</i> 78% dan <i>Naïve Bayes</i> 74%. Data yang diberi label oleh <i>expert</i>, akurasi kedua metode klasifikasi mencapai 96%.</p> <p>Rasio pembagian data 70:30 memberikan akurasi terbaik sebesar 96%.</p> <p>Rasio pembagian data 90:10 dan 80:20 masing-masing mencapai akurasi 95%.</p> <p>Nilai $k = 1$ mencapai akurasi 95.24%.</p> <p>Nilai $k = 2, \dots, k = 20$ mencapai akurasi yang sama yaitu 96.19%.</p> <p>Terkait nilai k, keputusan akhir sebagai parameter final yang diambil menurut penulis adalah $k = 5$ dikarenakan memberikan kinerja optimal tanpa penurunan akurasi yang signifikan.</p>
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 $k = 3$ dengan rasio pembagian data 60:40.
3.	Diwandanu & Wisudawati, 2023.	Analisis Sentimen Terhadap Twit Maxim Pada Twitter Menggunakan R Programming Dan K Nearest Neighbors	Klasifikasi <i>KNN</i> . Pelabelan sentimen berbasis <i>library Lexicon</i> .	<p>Rasio pembagian data 80:20, 75:25, 70:30 dengan nilai $k = 1, \dots, k = 10$.</p> <p>Hasil akurasi terbaik dengan rasio pembagian data 80:20 dan nilai $k = 1$, yaitu 95.43%.</p>

4.	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 library <i>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>
5.	Arifin & Nugroho, 2023.	Uji Akurasi Penggunaan Metode KNN dalam Analisis Sentimen Kenaikan Harga BBM pada Media Twitter	Klasifikasi <i>KNN</i> . Pelabelan sentimen berbasis manual.	Dengan nilai $k = 5$, didapatkan akurasi, presisi, dan <i>recall</i> dari evaluasi terhadap algoritma klasifikasi <i>KNN</i> masing-masing 94.33%, 91.3%, dan 84%.
6.	Handoko et al., 2024.	Analisis Sentimen Ulasan Penumpang Maskapai Penerbangan di Indonesia Dengan Algoritma Random Forest Dan KNN	Klasifikasi <i>KNN</i> dan <i>Random Forest</i> . Pelabelan sentimen berbasis <i>Lexicon Afinn</i> . Pembobotan kata menggunakan <i>TF-IDF</i> .	Hasil evaluasi model menunjukkan tingkat akurasi dari <i>Random Forest</i> sebesar 83% dan <i>KNN</i> mencapai 82%.
7.	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%. 77.67% dari data uji, menurut penulis sudah benar terkласifikasi ke dalam kelas ulasan positif dengan nilai presisi dan <i>recall</i> masing-masing 82.67% dan 86.92%.
8.	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%.

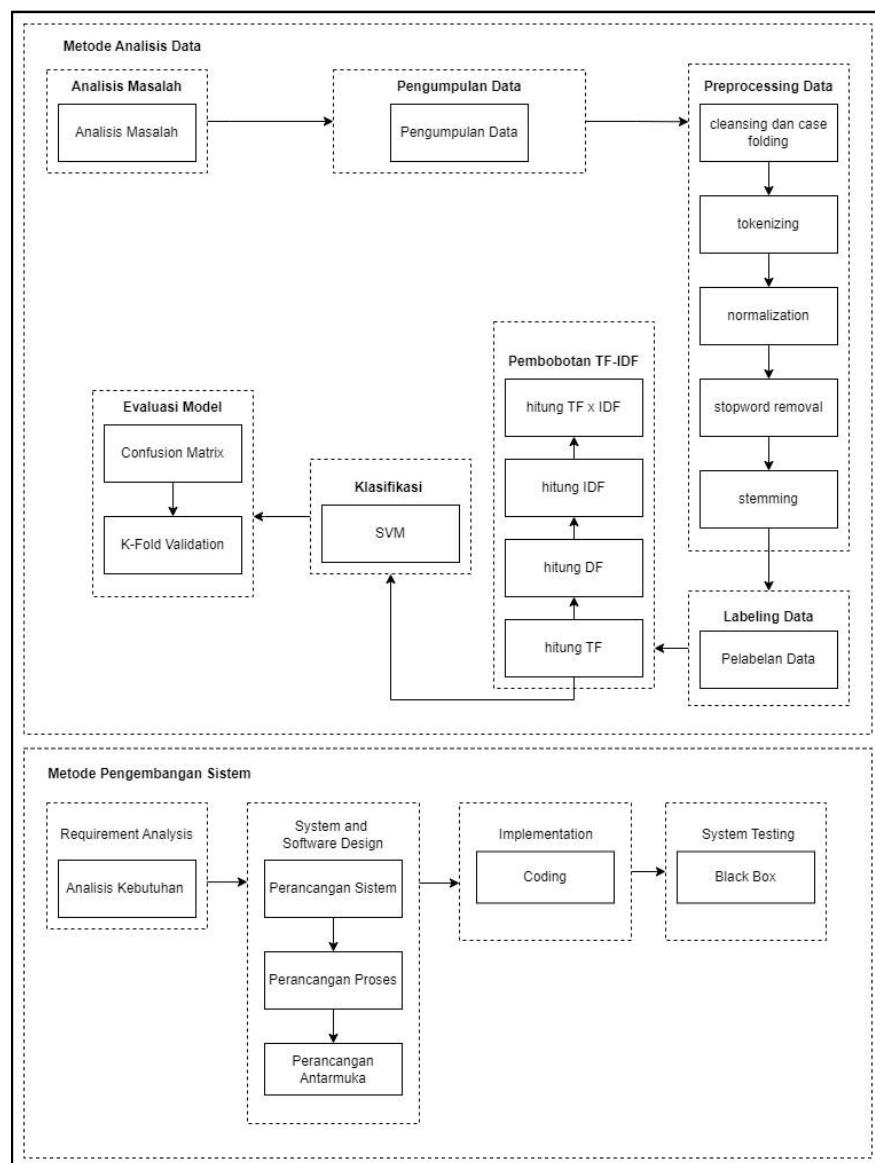
9.	Sholeha et al., 2022.	Analisis Sentimen Pada Agen Perjalanan Online Menggunakan Naïve Bayes dan K-Nearest Neighbor	Klasifikasi <i>KNN</i> dan <i>Naïve Bayes</i> . Pelabelan sentimen berbasis manual. Penghitungan frekuensi kata dengan <i>TF</i> .	Menurut penulis, berdasarkan 1500 data komentar dari 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.
10.	Mustaqim et al., 2024.	Analisis Sentimen Ulasan Aplikasi PosPay untuk Meningkatkan Kepuasan Pengguna dengan Metode K-Nearest Neighbor (KNN)	Klasifikasi <i>KNN</i> . Pembobotan kata menggunakan <i>TF-IDF</i> . Pelabelan sentimen berbasis <i>library Lexicon</i> .	Menurut penulis, hasil akhir dari penelitian menunjukkan bahwa sentimen pengguna aplikasi Pospay cenderung positif. Model klasifikasi <i>KNN</i> menghasilkan akurasi sebesar 91%, presisi sebesar 90%, dan <i>recall</i> sebesar 99%.

BAB III

METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1 Metodologi Penelitian

Pada bagian ini akan dibahas mengenai metodologi penelitian serta pengembangan sistem yang akan dilakukan dalam penelitian ini. Dalam penelitian ini yaitu menentukan analisis sentimen *coffee shop* di Yogyakarta berdasarkan ulasan yang ada pada *google maps* yang nantinya akan dijadikan sebagai rekomendasi *coffee shop* di Yogyakarta dengan menggunakan metode *Support Vector Machine* dan *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, *text preprocessing*, pelabelan data, pembobotan kata (*TF-IDF*), pengujian model dengan *Support Vector Machine*, pengujian sistem, model terbaik dan analisis serta visualisasi hasil. Berikut merupakan tahapan pada metodologi penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

3.1.1. Metode Analisis Data

3.1.1.1 Analisis Masalah

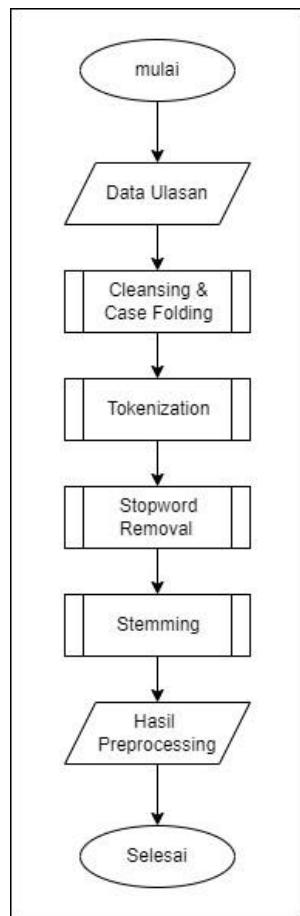
Analisis masalah merupakan Gambaran masalah dari penelitian tugas akhir Analisis Sentimen untuk Rekomendasi Tempat Kuliner di Yogyakarta menggunakan metode *Support Vector Machine* dan TF-IDF. Penelitian ini dilakukan untuk merancang sistem yang dapat menghasilkan klasifikasi berdasarkan *review* atau ulasan *coffee shop* yang tedapat pada *google maps* yang bersentimen positif, negatif, ataupun netral. Pada sistem ini juga akan menampilkan *list* rekomendasi *coffee shop* yang ada di Yogyakarta.

3.1.1.2 Pengumpulan Data

Pada penelitian ini data yang dibutuhkan dikumpulkan dengan cara melakukan scraping data pada *google maps*. Tahapan scraping pada penelitian ini bertujuan untuk pengambilan data serta informasi secara spesifik dan juga otomatis dari *google maps*. Proses *scraping* pada *google maps* dilakukan menggunakan website *Outscapper*. Data yang diambil secara spesifik yaitu data nama *coffee shop*, alamat *coffee shop*, *review text*, serta *rating coffee shop* yang berada di Yogyakarta. Data yang berhasil diambil dari tahapan scraping disimpan kedalam file berformat csv.

3.1.1.3 Preprocessing Data

Tahap *preprocessing* merupakan serangkaian tahap yang sangat penting dalam klasifikasi. Tahap *preprocessing* bertujuan untuk membersihkan data, serta mempersiapkan data agar data yang akan digunakan tidak mengandung *noise* dan juga menghasilkan teks yang lebih jelas, singkat, serta padat tanpa mengurangi makna yang ada pada teks tersebut. Pada tahap preprocessing data terdapat beberapa tahapan seperti *cleansing*, *case folding*, *tokenizing*, *stopwords removal*, serta *stemming*. Berikut merupakan flowchart dari tahap preprocessing yang dapat dilihat pada Gambar 3.2.

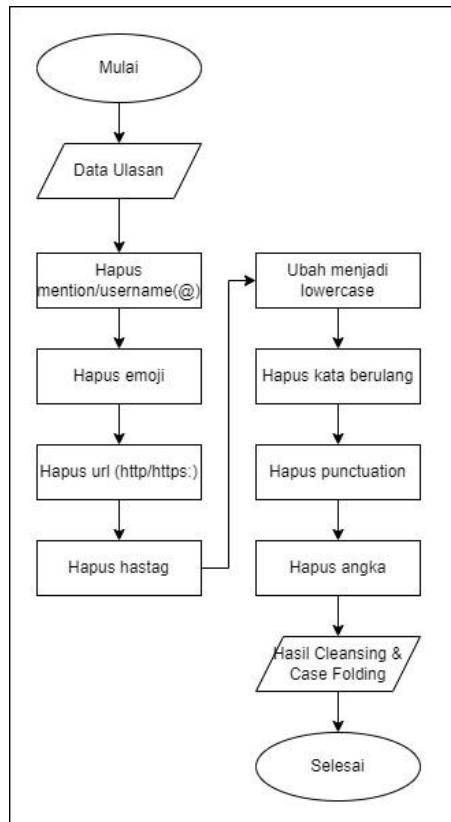


Gambar 3.2 Flowchart Preprocessing

Gambar 3.2 merupakan *flowchart* dari tahapan *preprocessing*, berdasarkan *flowchart* tersebut berikut merupakan penjelasan dari tahapan *preprocessing* yang dilakukan pada penelitian ini :

a. *Cleansing* dan *Case Folding*

Cleansing merupakan proses membersihkan data teks yang berupa tanda baca, link angka atau nomor, serta menghapus spasi untuk menghilangkan terjadinya *noise* pada data teks. *Case folding* merupakan tahap dimana data akan diubah menjadi huruf kecil. Berikut merupakan alur serta contoh hasil dari *cleansing* dan *case folding* dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Flowchart Cleansing dan Case Folding

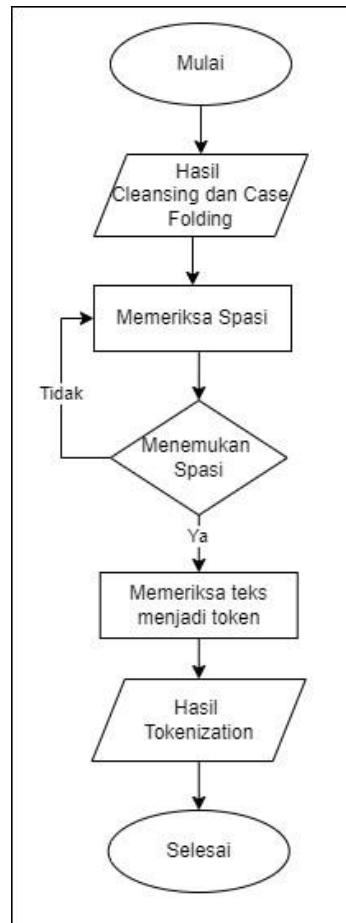
Berikut contoh hasil dari tahap *cleansing* dan *case folding* dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Contoh Penggunaan *Cleansing* dan *Case Folding*

No.	Data Review	Hasil <i>Cleansing</i> dan <i>Case Folding</i>
1.	Kopi ENAK POL, tempat nyaman Untuk ngerjain tugas, pelayanan Ramah	kopi enak tempat nyaman untuk ngerjain tugas pelayanan ramah
2.	Tempat nyaman, banyak tanaman, suasana asik, di Jogja Selatan. Yang paling PENTING kopinya cucok!!!	tempat nyaman banyak tanaman suasana asik di jogja selatan yang paling penting kopinya cucok

b. Tokenizing

Setelah dilakukan tahapan *cleansing* dan *case folding*, tahap berikutnya yaitu tokenization Dimana pada tahap ini berfungsi untuk memisahkan kalimat pada teks menjadi satu kata berdasarkan spasi. Berikut merupakan alur serta contoh hasil *tokenizing* dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Flowchart Tokenizing

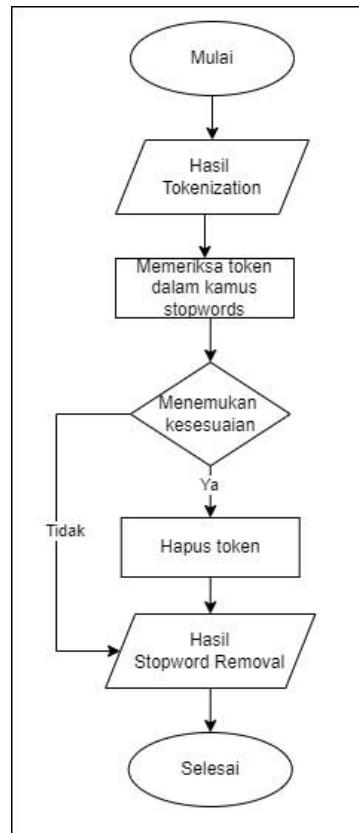
Berikut contoh hasil dari tahap *tokenizing* dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Contoh Penggunaan Tokenizing

No.	Data Review	Hasil Tokenizing
1.	kopi enak tempat nyaman untuk ngerjain tugas pelayanan ramah	“kopi” “enak” “tempat” “nyaman” “untuk” “ngerjain” “tugas” “pelayanan” “ramah”
2.	tempat nyaman banyak tanaman suasana asik di jogja selatan yang paling penting kopinya cucok	“tempat” “nyaman” “banyak” “tanaman” “suasana” “asik” “di” “jogja” “selatan” “yang” “paling” “penting” “kopinya” “cucok”

c. Stopword Removal

Tahapan ini merupakan tahapan yang berfungsi untuk menghilangkan kata yang tidak memiliki definisi yang jelas. apabila kata tersebut masuk dalam daftar *stopword* maka kata tersebut tidak bisa diproses, sedangkan apabila kata tidak masuk dalam daftar *stopword* maka akan masuk pada proses selanjutnya. Berikut merupakan alur dan contoh hasil dari tahap *stopword removal* dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Flowchart Stopword Removal

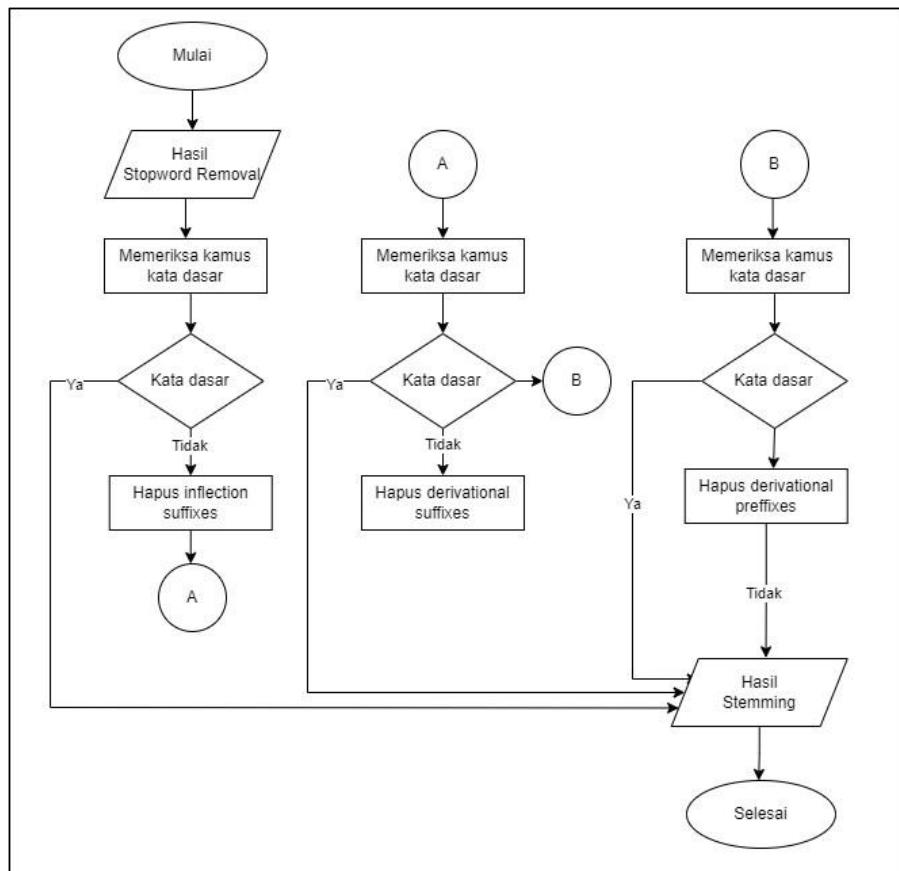
Berikut contoh hasil dari tahap *stopword removal* dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Contoh Penggunaan Stopword Removal

No.	Data Review	Hasil Stopword Removal
1.	“kopi” “enak” “tempat” “nyaman” “untuk” “ngerjain” “tugas” “pelayanan” “ramah”	“kopi” “enak” “tempat” “nyaman” “ngerjain” “tugas” “pelayanan” “ramah”
2.	“tempat” “nyaman” “banyak” “tanaman” “suasana” “asik” “di” “jogja” “selatan” “yang” “paling” “penting” “kopinya” “cucok”	“tempat” “nyaman” “banyak” “tanaman” “suasana” “asik” “jogja” “selatan” “penting” “kopinya” “cucok”

d. *Stemming*

Tahap *stemming* merupakan tahap untuk pembentukan kata dasar. Pada tahapan ini menerapkan *library* dari sastrawi. Berikut merupakan alur dan contoh hasil pada tahap *stemming* dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Flowchart Stemming

Berikut contoh hasil dari tahap *stemming* dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Contoh Penggunaan Stemming

No.	Data Review	Hasil Stemming
1.	“kopi” “enak” “tempat” “nyaman” “ngerjain” “tugas” “pelayanan” “ramah”	“kopi” “enak” “tempat” “nyaman” “ngerjain” “tugas” “pelayanan” “ramah”
2.	“tempat” “nyaman” “banyak” “tanaman” “suasana” “asik” “jogja” “selatan” “penting” “kopinya” “cucok”	“tempat” “nyaman” “banyak” “tanaman” “suasana” “asik” “jogja” “selatan” “penting” “kopi” “cucok”

3.1.1.4 Pelabelan Data

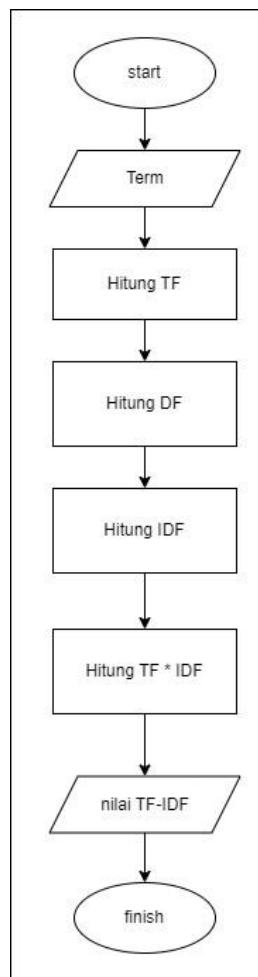
Data yang didapat dari proses *pre-processing* akan diberi label dengan menggunakan metode *lexicon based* menjadi beberapa kelas sentimen yang terdiri dari kelas positif, negatif, serta netral berdasarkan ulasan. Label positif akan direpresentasikan dengan nilai 1, label negatif dengan nilai -1, dan label netral dengan label 0. Berikut merupakan contoh pelabelan data dapat dilihat pada Tabel 3.5 berikut :

Tabel 3. 5 Contoh Hasil Pelabelan Data

No.	Ulasan	Label
1.	Tempat nya asri sejuk dan tenang cocok kalau berkunjung pagi hari cobain best seller nya pasti ketagihan	1
2.	Manual <i>brewing</i> bener bener gak enak sampe gak bisa diminum gak sesuai test noted yang tertera tapi harusnya minimal bisa diminum gak yang rasanya mengganggu banget	-1
3.	Tempat nongkrong ngopi pas di tiga terban jl solo pilih kopi dan snacknya lumayan di sini juga dapat food court dengan agam pilih makan tradisional jawa steak dll	0

3.1.1.5 Pembobotan *Term-Invers Document Frequency* (TF-IDF)

Pembobotan TF-IDF merupakan proses yang dilakukan apabila tahapan preprocessing telah dilakukan. Tujuan dari pembobotan kata dengan TF-IDF ini yaitu untuk memberikan bobot pada kata yang terdapat pada dokumen. Berikut merupakan *flowchart* proses perhitungan TF-IDF dapat dilihat pada Gambar 3.7.

**Gambar 3. 7 Flowchart TF-IDF**

Berikut merupakan beberapa contoh data yang akan digunakan untuk perhitungan manual pada TF-IDF dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Dokumen

ID	Dokumen	Sentimen	Label
D1	kopi enak makan enak tempat cocok nongkrong kota hening nugas	Positif	1
D2	tempat asyik banget harga menu standar gak mahal	Netral	0

D3	rasa masakan gak banget menu minuman tawarkan pembayaran terlalu lama saat melakukan transaksi	Negatif	-1
D4	kopi enak makan enak layan oke strategis pandang tugu jogja	Positif	1

1. Hitung *Term Frequency* (TF)

Term frequency bertujuan untuk menghitung kemunculan suatu *term* dalam suatu dokumen, semakin sering kemunculan term maka nilai kesesuaian yang dihasilkan bobot yang dihasilkan dari kata tersebut akan menjadi semakin besar. Pada tabel 3.5 Jumlah kemunculan pada kata tiap dokumen direpresentasikan sebagai D1 yaitu ulasan 1, D2 yaitu ulasan 2, D3 yaitu ulasan 3, dan D4 yaitu ulasan 4. Langkah awal dalam perhitungan TF-IDF yaitu menentukan TF, dengan persamaan sebagai berikut: $tf = 1 + \log_{10}$

Maka hasil perhitungan TF berdasarkan dokumen diatas dapat dilihat pada Tabel 3.7 berikut :

Tabel 3. 7 Contoh Hasil Perhitungan TF

No.	Term	TF			
		D1	D2	D3	D4
1.	asyik	0	1	0	0
2.	banget	0	1	1	0
3.	cocok	1	0	0	0
4.	enak	2	0	0	2
5.	gak	0	1	1	0
6.	harga	0	1	0	0
7.	hening	1	0	0	0
8.	jogja	0	0	0	1
9.	kopi	1	0	0	1
10.	kota	1	0	0	0
11.	lama	0	0	1	0
12.	layan	0	0	0	1
13.	mahal	0	1	0	0
14.	makan	1	0	0	1
15.	masakan	0	0	1	0
16.	melakukan	0	0	1	0
17.	menu	0	1	1	0
18.	minuman	0	0	1	0
19.	nongkrong	1	0	0	0
20.	nugas	1	0	0	0
21.	oke	0	0	0	1
22.	pandang	0	0	0	1
23.	pembayaran	0	0	1	0
24.	rasa	0	0	1	0
25.	saat	0	0	1	0
26.	standar	0	1	0	0
27.	strategis	0	0	0	1
28.	tawarkan	0	0	1	0
29.	tempat	1	1	0	0

30.	transaksi	0	0	1	0
31.	tugu	0	0	0	1

2. Hitung Nilai *Inverse Document Frequency* (IDF)

Apabila sudah dilakukan perhitungan *term frequency* (TF), maka tahap selanjutnya yaitu menghitung (IDF) atau menghitung banyaknya term pada seluruh dokumen.

Perhitungan IDF didapat dari ($\log(N/DF)$), Dimana N adalah jumlah dokumen.

Perhitungan IDF dapat dilakukan melalui persamaan sebagai berikut:

$$idf_t = \log \frac{N}{df_t}$$

Berikut merupakan hasil perhitungan IDF dapat dilihat pada Tabel 3.9.

Tabel 3.8 Contoh Hasil Perhitungan Bobot IDF

No.	Term	Term Frequency				DF	Nd/df(t)	IDF
		D1	D2	D3	D4			
1.	asyik	0	1	0	0	1	2	1,916
2.	banget	0	1	1	0	2	1,2	1,511
3.	cocok	1	0	0	0	1	2	1,916
4.	enak	2	0	0	2	2	1,5	1,511
5.	gak	0	1	1	0	2	3	1,511
6.	harga	0	1	0	0	1	3	1,916
7.	hening	1	0	0	0	1	3	1,916
8.	jogja	0	0	0	1	1	3	1,916
9.	kopi	1	0	0	1	2	3	1,511
10.	kota	1	0	0	0	1	3	1,916
11.	lama	0	0	1	0	1	3	1,916
12.	layan	0	0	0	1	1	3	1,916
13.	mahal	0	1	0	0	1	3	1,916
14.	makan	1	0	0	1	2	3	1,511
15.	masakan	0	0	1	0	1	3	1,916
16.	melakukan	0	0	1	0	1	3	1,916
17.	menu	0	1	1	0	2	3	1,511
18.	minuman	0	0	1	0	1	3	1,916
19.	nongkrong	1	0	0	0	1	3	1,916
20.	nugas	1	0	0	0	1	3	1,916
21.	oke	0	0	0	1	1	3	1,916
22.	pandang	0	0	0	1	1	3	1,916
23.	pembayaran	0	0	1	0	1	3	1,916
24.	rasa	0	0	1	0	1	3	1,916
25.	saat	0	0	1	0	1	3	1,916
26.	standar	0	1	0	0	1	3	1,916
27.	strategis	0	0	0	1	1	3	1,916
28.	tawarkan	0	0	1	0	1	3	1,916
29.	tempat	1	1	0	0	2	3	1,511
30.	transaksi	0	0	1	0	1	3	1,916
31.	tugu	0	0	0	1	1	3	1,916

3. Hitung nilai TF-IDF

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

$$W_{dt} = tf_{dt} \cdot idf_t$$

Dengan W merupakan hasil dari perhitungan TF dikalikan dengan IDF (TF*IDF)

Maka, hasil perhitungan TF-IDF dari dokumen diatas dapat dilihat pada Tabel 3.10.

Tabel 3. 9 Contoh Hasil Perhitungan TF-IDF

No.	Term	Term Frequency				DF	IDF	W			
		D1	D2	D3	D4			D1	D2	D3	D4
1.	asyik	0	1	0	0	1	1,916	0	1,916	0	0
2.	banget	0	1	1	0	2	1,511	0	1,511	1,511	0
3.	cocok	1	0	0	0	1	1,916	1,916	0	0	0
4.	enak	2	0	0	2	2	1,511	3,022	0	0	3,022
5.	gak	0	1	1	0	2	1,511	0	1,511	1,511	0
6.	harga	0	1	0	0	1	1,916	0	1,916	0	0

Tabel 3. 10 Lanjutan Contoh Hasil Perhitungan TF-IDF

No.	Term	Term Frequency				DF	IDF	W			
		D1	D2	D3	D4			D1	D2	D3	D4
7.	hening	1	0	0	0	1	1,916	1,916	0	0	0
8.	jogja	0	0	0	1	1	1,916	0	0	0	1,916
9.	kopi	1	0	0	1	2	1,511	1,511	0	0	1,511
10.	kota	1	0	0	0	1	1,916	1,916	0	0	0
11.	lama	0	0	1	0	1	1,916	0	0	1,916	0
12.	layan	0	0	0	1	1	1,916	0	0	0	1,916
13.	mahal	0	1	0	0	1	1,916	0	1,916	0	0
14.	makan	1	0	0	1	2	1,511	1,511	0	0	1,511
15.	masakan	0	0	1	0	1	1,916	0	0	1,916	0
16.	melakukan	0	0	1	0	1	1,916	0	0	1,916	0
17.	menu	0	1	1	0	2	1,511	0	1,511	1,511	0
18.	minuman	0	0	1	0	1	1,916	0	0	1,916	0
19.	nongkrong	1	0	0	0	1	1,916	1,916	0	0	0
20.	nugas	1	0	0	0	1	1,916	1,916	0	0	0
21.	oke	0	0	0	1	1	1,916	0	0	0	1,916
22.	pandang	0	0	0	1	1	1,916	0	0	0	1,916
23.	pembayaran	0	0	1	0	1	1,916	0	0	1,916	0
24.	rasa	0	0	1	0	1	1,916	0	0	1,916	0
25.	saat	0	0	1	0	1	1,916	0	0	1,916	0
26.	standar	0	1	0	0	1	1,916	0	1,916	0	0
27.	strategis	0	0	0	1	1	1,916	0	0	0	1,916
28.	tawarkan	0	0	1	0	1	1,916	0	0	1,916	0
29.	tempat	1	1	0	0	2	1,511	1,511	1,511	0	0
30.	transaksi	0	0	1	0	1	1,916	0	0	1,916	0
31.	tugu	0	0	0	1	1	1,916	0	0	0	1,916

Berdasarkan hasil perhitungan TF-IDF selanjutnya dituliskan dalam bentuk notasi *vector* yang akan digunakan untuk melakukan *training* dengan menggunakan metode SVM.

$X_1 = [0, 0, 4.790, 5.036, 0, 0, 4.790, 0, 2.518, 4.790, 0, 0, 0, 2.518, 0, 0, 0, 0, 4.790, 4.790, 0, 0, 0, 0, 0, 0, 0, 2.518, 0, 0]$

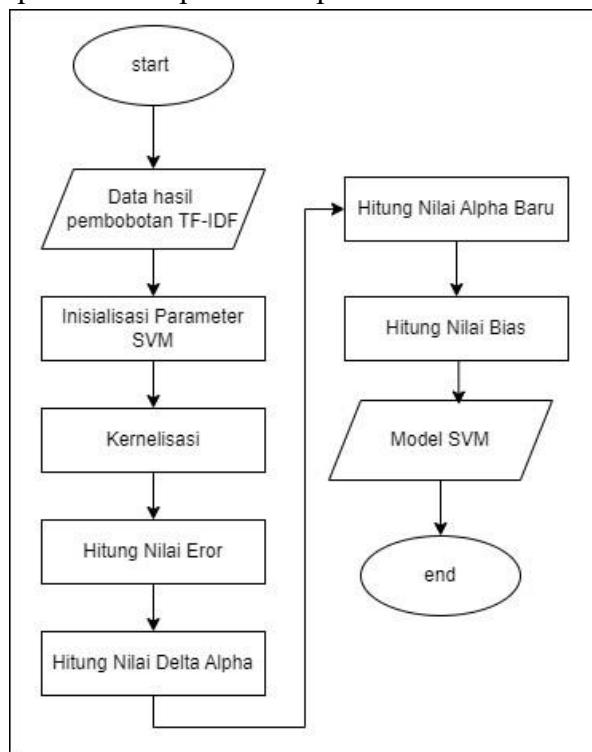
$X_2 = [1.916, 1.510, 0, 0, 1.510, 1.916, 0, 0, 0, 0, 0, 0, 1.916, 0, 0, 1.510, 0, 0, 0, 0, 0, 0, 0, 0, 1.916, 0, 0, 1.510, 0]$

$X_3 = [0, 1.510, 0, 0, 1.510, 0, 0, 0, 0, 1.916, 0, 0, 0, 1.916, 1.916, 1.510, 1.916, 0, 0, 0, 1.916, 1.916, 0, 0, 1.916, 0, 0, 1.916, 0, 0, 1.916, 0, 0, 1.916, 0]$

$X_4 = [0, 0, 0, 3.021, 0, 0, 0, 1.916, 1.510, 0, 0, 1.916, 0, 1.510, 0, 0, 0, 0, 0, 1.916, 1.916, 0, 0, 0, 0, 0, 1.916]$

3.1.1.6 Klasifikasi Model dengan *Support Vector Machine (SVM)*

Tahap klasifikasi dengan model SVM ini bertujuan untuk memperoleh hasil akhir penelitian ini dengan mencari nilai klasifikasi sentimen berdasarkan data ulasan dari google maps. Alur tahapan SVM dapat dilihat pada Gambar 3.8 berikut :



Gambar 3.8 *Flowchart Pelatihan SVM*

Metode yang digunakan dalam perhitungan ini yaitu *sequential SVM* dengan kernel linear. Berikut merupakan alur proses SVM:

1. Melakukan inisialisasi pada parameter (Vijayakumar, 1998), :
 - a. $\lambda = 0,5$
 - b. $\gamma = 0,001$
 - c. $\varepsilon = 0,0001$
 - d. $C = 1$
 - e. $\alpha = 0$

2. Proses setelah inisialisasi parameter yaitu mencari nilai kernel pada matriks K beserta matriks *hessian* yang berukuran n x n yaitu sebanyak data training.

Berikut merupakan contoh perhitungannya.

$$k(x_i, x_j) = (x_i, x_j + 1)^2$$

$$\begin{aligned} k(x_i, x_j) &= ((0 \times 1,916) + (0 \times 1,511) + (1,916 \times 0) + (3,022 \times 0) + (0 \times 1,511) \\ &+ (0 \times 1,916) + (1,916 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1,916) + (1,511 \times 0) + (0 \\ &\times 0) + (1,916 \times 0) + (1,916 \times 0) + (0 \\ &\times 0) + (0 \times 0) + (0 \times 0) + (1,511 \times 0) + (0 \times 0) + (0 \times 0 + 1)^2 \end{aligned}$$

$$k(x_i, x_j) = 10,78$$

Perhitungan tersebut terus dilakukan pada dokumen lainnya sehingga memperoleh hasil matriks K seperti pada Tabel 3.11.

Tabel 3. 11 Nilai pada Matriks K

Dokumen	X1	X2	X3	X4
X1	1248,85	10,78	1,00	215,96
X2	10,78	615,99	61,59	1,00
X3	1,00	61,59	1672,59	1,00
X4	215,96	1,00	1,00	1348,99

Untuk memperoleh nilai matriks hessian, perlu dilakukan perhitungan dengan proses perhitungan sebagai berikut

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$$

$$D_{12} = (1)(0)((10,78) + 0,5^2))$$

$$D_{12} = (0)(10,78 + 0,25)$$

$$D_{12} = 0$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan hasil matriks hessian yang terdapat pada Tabel 3.12.

Tabel 3. 12 Nilai pada Matriks Hessian

Dokumen	X1	X2	X3	X4
X1	1249,096	0	-1,25	216,210
X2	0	0	0	0
X3	-1,25	0	1672,840	-1,25
X4	216,210	0	-1,25	1349,238986

3. Apabila sudah didapatkan nilai pada matriks hessian, proses selanjutnya yaitu mencari nilai eror (E_i). Berikut merupakan persamaan yang digunakan yaitu $E_i = \sum nj = 1 \alpha_i D_{ij}$

$$E_i = (0 \times 1249,096) + (0 \times 0) + (0 \times (-1,25)) + (0 \times 216,210) = 0$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan hasil nilai E_i yang terdapat pada Tabel 3.13.

Tabel 3. 13 Hasil Perhitungan E_i

Dokumen	E_i
D1	0
D2	0
D3	0
D4	0

4. Proses selanjutnya yaitu mencari nilai $\delta\alpha_i$ melalui persamaan

$\delta\alpha_i = \min\{\max[\gamma(1-E_i), \alpha_i], C - \alpha_i\}$. Berikut merupakan contoh perhitungannya: $\delta\alpha_i = \min\{\max[0,001(1-0), 0]1-0\} = 0,001$.

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan nilai $\delta\alpha_i$ seperti pada Tabel 3.14.

Tabel 3. 14 Hasil Perhitungan $\delta\alpha_i$

Dokumen	$\delta\alpha_i$
D1	0,001
D2	0,001
D3	0,001
D4	0,001

5. Tahap selanjutnya yaitu mencari nilai baru dengan menggunakan persamaan

$$\begin{aligned}\alpha_i &= \alpha_i + \delta\alpha_i \\ &= 0 + 0,001 \\ &= 0,001\end{aligned}$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan nilai α_i seperti pada Tabel 3.15.

Tabel 3. 15 Hasil Perhitungan α_i baru

Dokumen	α_i baru
D1	0,001
D2	0,001
D3	0,001
D4	0,001

6. Pada tahap ini yaitu mengulang perhitungan pada tahap ke-3 sampai tahap ke5 hingga mencapai batas maksimal iterasi. Berikut merupakan contoh perhitungan iterasi ke-2.

$$\begin{aligned}E_i &= (0,001 \times 1249,096) + (0,001 \times 0) + (0,001 \times (-1,25)) + (0,001 \times 216,210) \\ &= 1,4640\end{aligned}$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan hasil nilai E_i yang terdapat pada Tabel 3.16.

Tabel 3. 16 Hasil Perhitungan E_i iterasi ke-2

Dokumen	E_i
D1	1,4641
D2	0
D3	1,6703
D4	1,5642

7. Tahapan berikutnya yaitu menghitung $\delta\alpha_i$ pada iterasi ke-2. Berikut merupakan contoh perhitungan $\delta\alpha_i$ pada iterasi ke-2

$$\delta\alpha_i = \min\{\max[\gamma(1- E_i), \alpha_i], C - \alpha_i\}$$

$$\begin{aligned}\delta\alpha_i &= \min\{\max[0,001(1-1,4641), 0,001], 1-0,001\} \\ &= 0,001\end{aligned}$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan hasil nilai $\delta\alpha_i$ yang terdapat pada Tabel 3.17.

Tabel 3. 17 Hasil Perhitungan $\delta\alpha_i$ Iterasi ke-2

Dokumen	$\delta\alpha_i$
D1	-0,001000464
D2	-0,000999
D3	-0,00100067
D4	-0,001000564

$$\begin{aligned}\alpha_i &= \alpha_i + \delta\alpha_i \\ &= 0,001 + -0,001000464 \\ &= -0,0000004641\end{aligned}$$

Ulangi perhitungan tersebut pada dokumen lainnya sehingga mendapatkan hasil nilai $\delta\alpha_i$ yang terdapat pada Tabel 3.19

Tabel 3. 18 Hasil Perhitungan α_i baru

Dokumen	α_i baru
D1	-0,001000464
D2	-0,000999
D3	-0,00100067
D4	-0,001000564

8. Tahapan selanjutnya adalah perhitungan untuk mencari nilai bias dengan persamaan $b = -((w \cdot x_{-1}) + (w \cdot x_{+1})) \cdot x_{-1}$ masuk pada golongan kelas negatif yang dimana memiliki nilai alpha paling besar, sedangkan x_{+1} masuk pada golongan kelas positif yang Dimana memiliki nilai alpha paling besar.

$$\begin{aligned}w \cdot x_{-1} &= (1 \times (-0,001000464)) \times 1 + (0 \times (-0,000999)) \times 61,59 \\ &\quad + (-1 \times (-0,00100067)) \times 1671,59 + (1 \times (-0,001000564)) \times 1 \\ &= 1,672534942\end{aligned}$$

$$\begin{aligned}w \cdot x_{+1} &= (1 \times (-0,001000464)) \times 1248,85 + (0 \times (-0,000999)) \times 10,78 \\ &\quad + (-1 \times (-0,00100067)) \times 1 + (1 \times (-0,001000564)) \times 215,96 \\ &= -1,464505898\end{aligned}$$

Dengan diperolehnya kedua nilai tersebut, maka untuk mendapatkan nilai bias dapat dilakukan dengan perhitungan sebagai berikut:

$$\begin{aligned}b &= -\frac{1}{2} (1,672534942 - (-1,464505898)) \\ &= -1,56852042\end{aligned}$$

3.1.1.7 Pengujian

Pengujian pada sistem dilakukan dengan menggunakan *k-fold cross validation* serta *confusion matrix* yang bertujuan untuk mengetahui performa model yang diterapkan. Pengujian dengan *confusion matrix* ini digunakan untuk mengukur Tingkat akurasi, presisi, serta *recall* dari suatu model. Rancangan pengujian *confusion matrix* dapat dilihat pada Tabel 3.19.

Tabel 3. 19 Rancangan *Confusion Matrix*

Actual	Classified		
	Negative	Positive	Neutral
Negative	<i>TN (True Negative)</i>	<i>FP (False Positive)</i>	<i>FN (False Negative)</i>
Positive	<i>TP (True Positive)</i>	<i>FP2 (False Positive 2)</i>	<i>FN2 (False Neutral 2)</i>

<i>Neutral</i>	<i>FN2(False Negative 2)</i>	<i>FP2(False Positive 2)</i>	<i>TL (True Neutral)</i>
----------------	------------------------------	------------------------------	--------------------------

Selanjutnya pengujian *k-fold cross validation* bertujuan untuk melakukan validasi dari suatu model dengan melakukan pengujian sebanyak K, Dimana pada penelitian ini menggunakan K sebanyak 5. Berikut merupakan tabel rancangan *k-fold cross validation* dapat dilihat pada Tabel 3.20.

Tabel 3. 20 Rancangan K-Fold Cross Validation

Fold	SVM		
	Akurasi	Presisi	Recall
1			
2			
3			
4			
5			

3.1.2. Metode Pengembangan Sistem

3.1.2.1 Requirement analysis

Analisis kebutuhan sistem merupakan proses analisis terhadap kebutuhan-kebutuhan yang digunakan pada penelitian ini yang meliputi kebutuhan fungsional dan kebutuhan non-fungsional.

A. Kebutuhan Fungsional

Adapun kebutuhan-kebutuhan yang termasuk kebutuhan fungsional dari sistem yang akan dibangun adalah sebagai berikut:

1. Sistem mampu melakukan proses *preprocessing* terhadap data ulasan yang dikumpulkan
2. Sistem mampu menampilkan hasil klasifikasi sentimen menggunakan metode *Support Vector Machine*
3. Sistem mampu melakukan penginputan kalimat untuk pengujian sistem
4. Sistem mampu menunjukkan nilai akurasi dari hasil akurasi perhitungan terhadap data yang diproses.
5. Sistem mampu menampilkan list rekomendasi *coffee shop*
6. Sistem mampu melakukan proses *login* dan *logout* untuk admin

B. Kebutuhan Non-Fungsional

Dalam pengimplementasian sistem yang akan dibangun agar sistem sesuai dengan apa yang telah dirancang maka dibutuhkan beberapa perangkat keras atau *hardware*, perangkat lunak atau *software*. Berikut merupakan kebutuhan non-fungsional untuk merancang sistem ini:

1. Kebutuhan *Hardware*

Berikut merupakan spesifikasi *hardware* yang digunakan dalam pembuatan sistem pada penelitian ini terdapat pada Tabel 3.21 berikut :

Tabel 3. 21 Kebutuhan Hardware

No.	Hardware	Keterangan
1.	<i>Processor</i>	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
2.	RAM	16,0 GB

3.	Koneksi internet	WIFI
4.	Perangkat <i>input</i> dan <i>output</i>	<i>Keyboard</i> , dan <i>Mouse</i>
5.	Koneksi Internet	WIFI

2. Kebutuhan *Software*

Berikut merupakan spesifikasi *software* yang digunakan dalam pembuatan sistem pada penelitian ini terdapat pada Tabel 3.22 berikut :

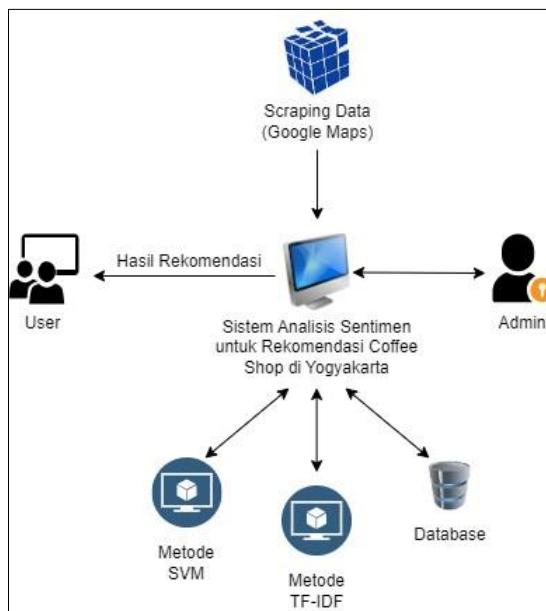
Tabel 3. 22 Kebutuhan *Software*

No.	Software	Keterangan
1.	<i>Operating system</i>	Windows 11, 64 bit
2.	<i>Google Colab</i>	Code Editor Online
3.	<i>Web Browser</i>	<i>Google Chrome, Microsoft Edge</i>
4.	<i>Draw.io</i>	<i>Software</i> untuk membuat <i>design diagram</i>
5.	Bahasa Pemrograman	<i>Python, PHP</i>
6.	IDE	<i>Visual Studio Code</i>
7.	Figma	<i>Design Sistem</i>

3.1.2.2 System and Software Design

A. Perancangan Arsitektur

Pada bagian perancangan sistem yang akan dibangun pada penelitian ini dibuat untuk menganalisis sentimen ulasan serta merekomendasikan *coffee shop* di Yogyakarta. Rancangan arsitektur sistem pada penelitian ini dapat dilihat pada Gambar 3.9 berikut:



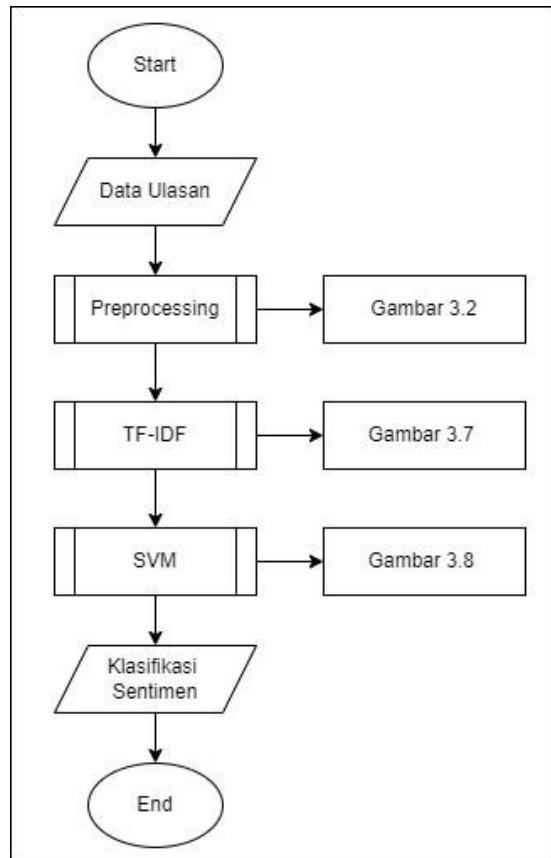
Gambar 3. 9 Arsitektur Sistem

Arsitektur yang akan dibangun terdiri dari *admin* dan *user*. Dimana *admin* akan menginputkan data pada sistem, data tersebut didapatkan dari proses *scraping* dari *google maps*. Kemudian data tersebut akan masuk pada tahap *preprocessing* yang bertujuan agar data yang telah disimpan dapat dipahami oleh sistem. Tujuan utama dari tahap *preprocessing* yaitu untuk menghasilkan data yang siap diproses oleh sistem serta membersihkan dan juga penyeragaman kata yang nantinya akan di ekstrasi pada tahap

berikutnya. Tahap berikutnya yaitu tahap ekstrasi fitur dimana tahapan ini bertujuan untuk menghitung kemunculan term pada sebuah dokumen. Selanjutnya terks akan diklasifikasikan dengan menggunakan Metode *Support Vector Machine* yang akan menghasilkan data dengan kelas positif, negatif, dan juga netral. Data yang telah diklasifikasikan kemudian akan digunakan untuk pembobotan, dan nantinya akan ditampilkan rekomendasi *coffee shop* berdasarkan analisis sentimen.

B. Perancangan Proses

Rancangan proses digambarkan melalui *flowchart* yang merupakan langkah-langkah atau tahapan tindakan terhadap sistem yang akan dibuat. Perancangan proses direpresentasikan melalui *flowchart*. Berikut merupakan *flowchart* dari perancangan proses dapat dilihat pada Gambar 3.10 berikut :



Gambar 3. 10 Flowchart Perancangan Proses

C. Perancangan *User Interface* (Antarmuka)

Rancangan *user interface* merupakan suatu proses *design* yang berfokus pada model komunikasi antara pengguna dengan sistem. Rancangan *user interface* mencakup beberapa aspek seperti, tata letak, dan navigasi yang bertujuan untuk memastikan pengguna dapat menggunakan sistem dengan mudah dan efektif. Pada sistem ini memiliki dua aktor yaitu, *admin* serta *user*. Berikut merupakan gambaran rancangan *user interface* yang terdiri dari beberapa bagian adalah sebagai berikut:

1. Rancangan Halaman List Rekomendasi *Coffee Shop*

Halaman ini merupakan halaman yang akan tampil saat user pertama kali mengakses sistem. Dimana pada halaman ini terdapat list rekomendasi beberapa *coffee shop* di Yogyakarta. Rancangan halaman utama pengunjung dapat dilihat pada Gambar 3.11 berikut :

COFFEE SHOP		Beranda	Informasi Coffee Shop	LOGIN
Rank	COFFEE SHOP	Sentimen Positif	Ulasan	
			Ulasan	

Gambar 3. 11 Rancangan Halaman List Rekomendasi Coffee Shop

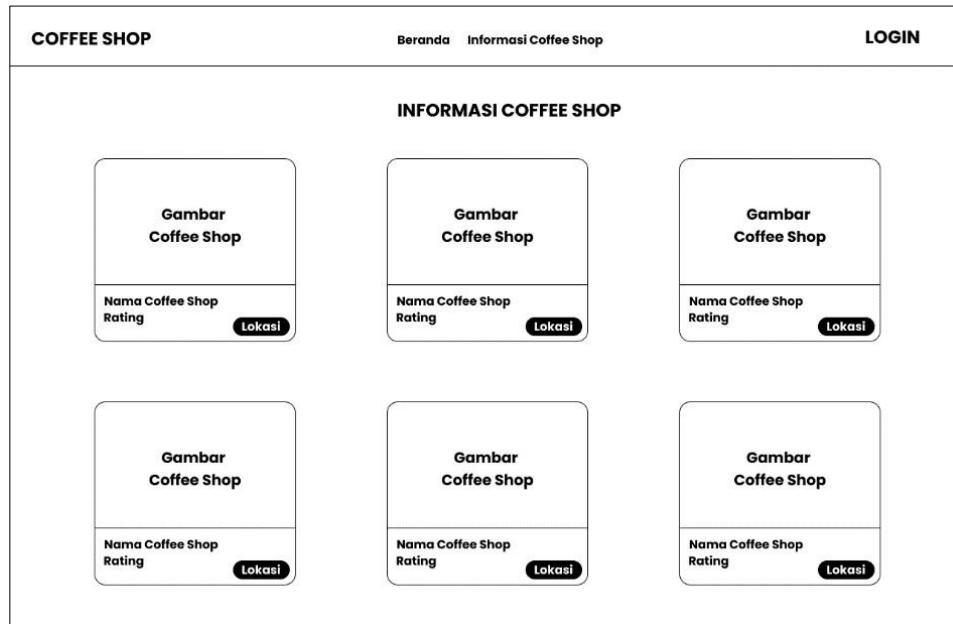
Pada fitur ulasan user dapat melihat detail review beserta sentimen dari *coffee shop* tersebut. Rancangan halaman detail ulasan dapat dilihat pada Gambar 3.12 berikut :

COFFEE SHOP		Beranda	Informasi Coffee Shop	LOGIN
NO	Review	Sentimen		

Gambar 3. 12 Rancangan halaman Detail Ulasan

2. Rancangan Halaman Informasi *Coffee Shop*

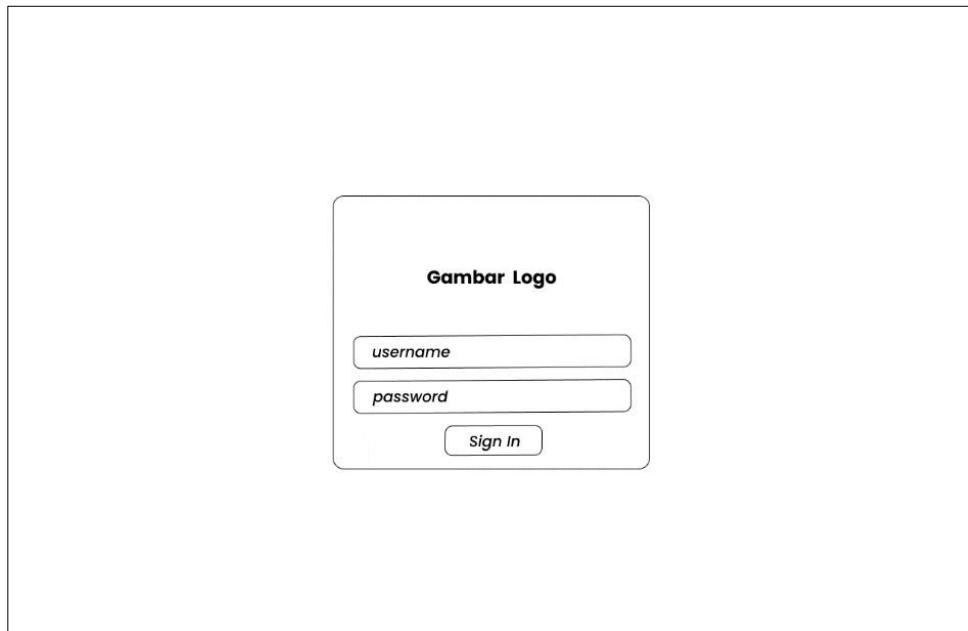
Halaman ini merupakan halaman yang menampilkan mengenai beberapa coffee shop yang ada di Yogyakarta. Rancangan halaman informasi coffee shop dapat dilihat pada Gambar 3.13 berikut :



Gambar 3. 13 Rancangan Halaman Informasi Coffee Shop

3. Rancangan Halaman *Login Admin*

Halaman *login admin* merupakan halaman yang akan tampil apabila masuk pada sistem sebagai *admin*. Pada halaman ini *admin* harus menginputkan *username* serta *password* agar bisa masuk ke dalam sistem. Rancangan halaman *login admin* dapat dilihat pada Gambar 3.14 berikut :



Gambar 3. 14 Rancangan Halaman Login

4. Rancangan Halaman *Dashboard Admin*

Halaman *dashboard admin* ini merupakan halaman yang pertama kali tampil apabila berhasil *login* sebagai *admin*. Dimana pada halaman dashboard ini menampilkan jumlah data coffee shop, dataset, serta data sentimen. Pada halaman dashboard ini memiliki beberapa menu diantaranya menu data coffee shop, dataset, data sentimen, halaman pengujian serta halaman model SVM, yang dimana pada setiap menu tersebut memiliki fungsi masing-masing. Rancangan halaman *dashboard* dapat dilihat pada Gambar 3.15 berikut :

COFFEE SHOP	 
Dashboard Data Coffe Shop Dataset Data Sentimen Halaman Pengujian Uji Coba Klasifikasi Data Hasil Pengujian Model SVM	Dashboard   

Gambar 3. 15 Rancangan Halaman Dashboard

5. Rancangan Halaman Data *Coffee Shop*

Pada halaman ini menampilkan data *coffee shop* yang digunakan pada sistem ini dalam bentuk tabel. Pada halaman data coffee shop ini menampilkan data berupa gambar, nama, rating dari coffee shop. Halaman ini memiliki fitur edit data, tambah data, serta hapus data untuk mengolah data. Rancangan halaman data *coffee shop* dapat dilihat pada Gambar 3.16 berikut :

COFFEE SHOP	 																												
Dashboard Data Coffe Shop Dataset Data Sentimen Halaman Pengujian Uji Coba Klasifikasi Data Hasil Pengujian Model SVM	Data Coffee Shop  <table border="1"> <thead> <tr> <th>Gambar</th> <th>Nama Coffee Shop</th> <th>Rata-Rata Rating</th> <th>Action</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Gambar	Nama Coffee Shop	Rata-Rata Rating	Action																								
Gambar	Nama Coffee Shop	Rata-Rata Rating	Action																										

Gambar 3. 16 Rancangan Halaman Data *Coffee Shop*

Pada halaman ini juga terdapat opsi tambah data yang dapat diakses oleh admin. Pada halaman tambah data ini berisi form nama *coffee shop*, *rating*, serta foto *coffee shop*, rancangan halaman tambah data *coffee shop* dapat dilihat pada Gambar 3.17 berikut :

Gambar 3.17 Rancangan Halaman Fitur Tambah Data *Coffee Shop*

6. Rancangan Halaman Data Sentimen

Rancangan pada halaman ini merupakan halaman yang menampilkan ulasan yang akan digunakan dataset dalam model SVM. Rancangan halaman dataset dapat dilihat pada Gambar 3.18 berikut :

Gambar 3. 18 Rancangan Halaman Dataset

7. Rancangan Halaman Pengujian

Rancangan halaman pengujian ini merupakan halaman yang akan memproses data, serta menampilkan data yang telah diuji. Pada halaman pengujian ini admin dapat menginputkan kalimat yang nantinya akan diuji oleh sistem serta mendapatkan hasil prediksi sentiment kalimat yang telah diinputkan. Rancangan halaman uji klasifikasi dapat dilihat pada Gambar 3.19 berikut :

COFFEE SHOP Dashboard Data Coffe Shop Dataset Data Sentimen Halaman Pengujian Uji Coba Klasifikasi Data Hasil Pengujian Model SVM	<p style="text-align: right;">☰</p> <p style="text-align: right;"></p> <p>Uji Coba Klasifikasi</p> <p>Ulasan</p> <div style="border: 1px solid black; height: 50px; margin-top: 10px;"></div> <div style="display: flex; justify-content: space-around; width: 100%;"> Uji Cancel </div>
--	---

Gambar 3. 19 Rancangan Halaman Uji Klasifikasi

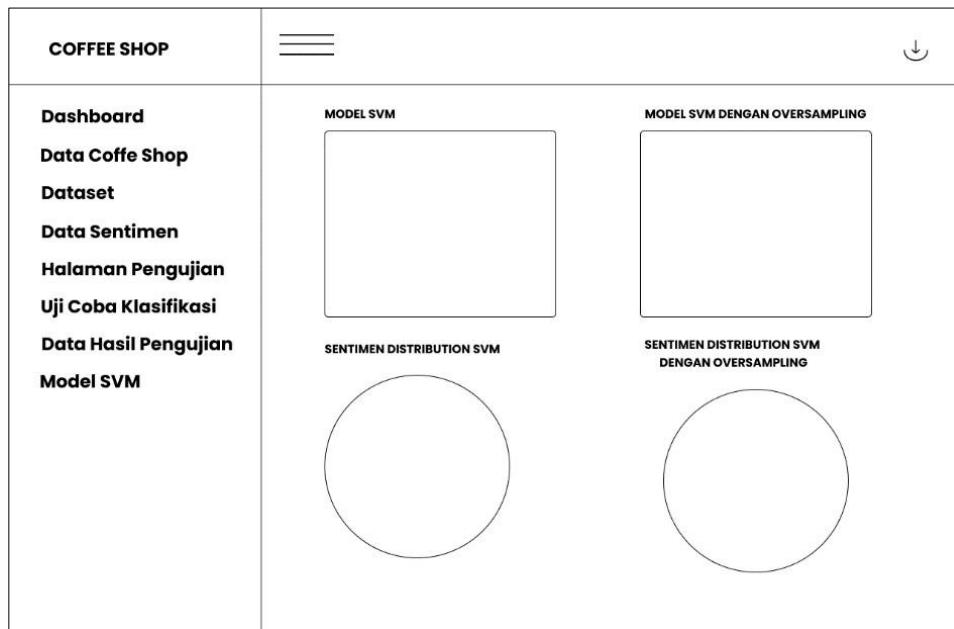
Hasil pengujian sebelumnya dapat dilihat pada fitur Data Hasil Pengujian, Dimana pada halaman ini berisikan proses ulasan berdasarkan kalimat yang telah diinputkan sebelumnya. Rancangan halaman data hasil pengujian dapat dilihat pada Gambar 3.20 berikut :

ID	Ulasan	Tokenisasi	Normalisasi	Stopwords	Text Preprocessing	Label SVM	Probabilitas Positif	Probabilitas Negatif	Probabilitas Netral

Gambar 3. 20 Rancangan Halaman Data Hasil Pengujian

8. Rancangan Halaman Model SVM

Rancangan halaman ini nantinya akan menampilkan mengenai hasil akurasi, presisi, dan recall dari model SVM ke dalam bentuk pie chart serta diagram. Pada halaman model SVM ini juga ditampilkan hasil pemodelan SVM dengan oversampling. Rancangan halaman model SVM dapat dilihat pada Gambar 3.21 berikut :



Gambar 3. 21 Rancangan Halaman Model SVM

3.1.2.3 *Implementation*

Pada tahap *implementation*, merupakan tahapan dimana perancangan sistem dan *design* yang telah dibuat akan direalisasikan menjadi kode program yang akan menghasilkan suatu sistem. Implementasi sistem analisis sentimen dibuat dan dirancang sesuai dengan metode yang digunakan berdasarkan tahapan analisis sentimen untuk rekomendasi *coffee shop* di Yogyakarta.

3.1.2.4 *System Testing*

Pada tahap ini dibuat suatu rancangan pengujian. Rancangan pengujian bertujuan untuk mempersiapkan konsep uji yang akan dilakukan untuk mengevaluasi tingkat keberhasilan proses klasifikasi yang telah dijalankan. Metode yang digunakan pada penelitian ini adalah *black box testing*.

Pengujian dengan metode *black box testing* bertujuan untuk menilai kinerja sistem yang telah dikembangkan. Serta pada pengujian *black box* ini berfokus pada fungsionalitas sistem, sehingga memastikan bahwa setiap fungsi berjalan sesuai dengan yang diharapkan. Detail mengenai rancangan pengujian dengan metode *black box testing* dapat dilihat pada Tabel 3.22 berikut :

Tabel 3. 23 Rancangan Pengujian Black Box

Aktor	Halaman	Detail Pengujian	Nilai	
			Berhasil	Tidak Berhasil
	Login	Melakukan proses login		
	Dashboard	Menampilkan jumlah tiap data		
	Data Coffee Shop	Menampilkan data <i>coffee shop</i>		
		Menambahkan data <i>coffee shop</i>		
		Mengedit data <i>coffee shop</i>		
		Menghapus data <i>coffee shop</i>		
	Dataset	Menampilkan dataset		
	Data Sentimen	Menampilkan data sentimen		

Admin	Halaman Model SVM	Menampilkan grafik model SVM		
-------	-------------------	------------------------------	--	--

Tabel 3. 24 Lanjutan Rancangan Pengujian *Black Box*

Aktor	Halaman	Detail Pengujian	Nilai	
			Berhasil	Tidak Berhasil
Admin	<i>Logout</i>	Fungsi <i>Logout</i>		
User	Rekomendasi <i>Coffee Shop</i>	Menampilkan list rekomendasi <i>coffee shop</i>		
	Informasi <i>Coffee Shop</i>	Menampilkan informasi alamat <i>coffee shop</i>		