# BAB III
# METODE PENELITIAN

## 3.1 Objek Penelitian

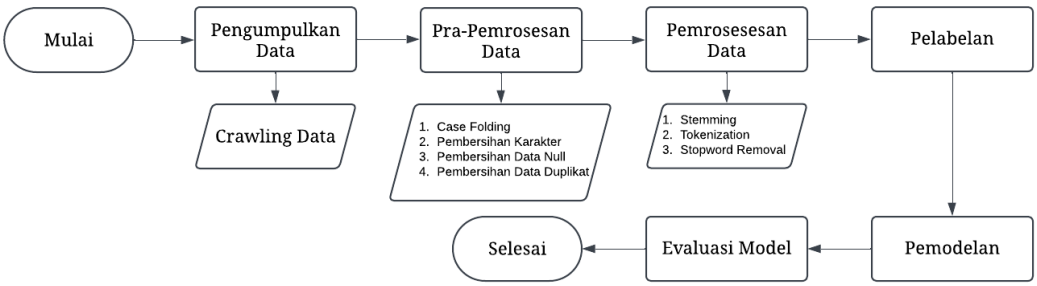
Objek yang digunakan dalam Penelitian adalah komentar dari platform Tiktok yang dikumpulkan tanggal 31 Maret 2024 pada jam 11:11 - 20:44 WIB. Menggunakan Metode *Crawling*, terkumpul data sebanyak 1032 komentar. Kemudian pada tanggal 11-15 Desember 2024 dikumpulkan data tambahan sebanyak 1375 komentar. Melalui *keyword* atau kata kunci "mobil listrik", data komentar tersebut disimpan kedalam format Excel dengan total data sebanyak 2407 komentar. Rincian dataset diperlihatkan pada Tabel 3.1.

Tabel 3. 1 Dataset Mobil listrik

| Username | Full_text |
|---|---|
| M0ND1S_4N15 | lagian kalo pake mobil listrik gak bisa braktaktak 🗿 |
| メ | daripada membuat mobil listrik atau sepeda listrik mending buat banyak sepeda aja,sepeda lebih ramah lingkungan + menyehatkan 👍🗿 |
| hiuga93 | di cina harga mobil listrik wuling gak sampe 100 juta di indo harga selangit😂 |
| … | … |
| rip•zioles | tanpa pohon emang kita bisa hidup |
| Faishal Martak | kl di jakarta sih mobil listrik ga kena gage, artinya setara dgn 2 mobil bbm. 🤣🤣🤣🤣 |
| Usep Suganda | rata2 yg beli mobil atau motor listrik kebanyakan FOMO doang,cmiw😂 |

Tabel 3.1 menunjukan dataset dari komentar platform TikTok yang akan digunakan sebagai objek penelitian. Tabel tersebut berisi *Username* dan *Full_text*. Kolom *Username* merupakan nama dari akun Tiktok Tersebut, sementara kolom *Full_text* berisi cuitan atau komentar dari postingan di platform TikTok.

## 3.2 Prosedur Penelitian




Gambar 3. 1 Prosedur Penelitian

Gambar 3.1 menyajikan prosedur penelitian yang akan dilaksanakan dalam analisis sentimen mobil listrik. Proses ini dirancang untuk memastikan analisis berjalan sistematis dan menghasilkan hasil yang relevan. Terdapat enam tahapan utama diantaranya sebagai berikut.

### 3.2.1 Pengumpulan Data

Tahap pengumpulan data dilakukan menggunakan metode *Crawling* dengan memanfaatkan *tools ekstension* yang ada pada Chrome, yaitu *Instant Data Scrapper*. Metode *crawling* diterapkan untuk mengumpulkan informasi mengenai suatu topik dari *website* (Muliawan & Dazki, 2023). Pada tahap ini, metode *crawling* diterapkan untuk mengumpulkan data komentar pada platform Tiktok. Data yang diperoleh tersebut menggambarkan pandangan publik terhadap mobil listrik. Jumlah total komentar yang berhasil didapatkan ialah sebanyak 2407 data Gambar data yang sudah terkumpul dapat dilihat pada Gambar 3.2.

| | full_text | username |
|---|---|---|
| 0 | cuan terus cuannya di pake buat benerin lingku... | Marriz \| 𝔐𝔦𝔠𝔠 |
| 1 | lah gw tinggal di Halmahera 🗿 | BG aris |
| 2 | anjay daerah gw tuh | blum ada bulu |
| 3 | guwah sih lingkungan bang dari padq cuan | zan ART ⚡ |
| 4 | gua pilih lingkungan aja bg | marvel |
| ... | ... | ... |
| 2402 | kantor ke apart cm 7 mnit? tp mau lebih cepet ... | 👁️👁️ |
| 2403 | fix orkay, | Gita Prayudi |
| 2404 | Plat nomer cash ygy 😍 | Dewi putri Kartikasa |
| 2405 | Kak dengan gaji mandiri vs life style gini ema... | youknowmelahya |
| 2406 | Mobil terjelek sepanjang masa , tp gw ga mampu... | Homeboy91 |

2407 rows × 2 columns

Gambar 3. 2 Total Data yang Terkumpul

### 3.2.2 Pra-Pemrosesan Data

Tahap pra-pemrosesan data bertujuan untuk mengonversi data mentah menjadi kumpulan data yang siap untuk analisis (Riyadi et al., 2021). Pada tahap ini, berbagai prosedur pembersihan data akan diterapkan agar data menjadi lebih konsisten dan bersih. Proses yang dilakukan meliputi konversi teks menjadi huruf kecil (*case folding*), pembersihan karakter tidak relevan seperti emoji, angka, simbol dan lain sebagainya. Pra-pemrosesan juga akan melakukan penghapusan data kosong *(null)* dan juga data duplikat yang dapat berpotensi spam. Rincian setiap tahapan disajikan sebagai berikut.

#### 3.2.2.1 *Case Folding*

Pada tahap ini, seluruh karakter dalam komentar diubah menjadi huruf kecil atau *lowercase*. Langkah ini bertujuan untuk menyeragamkan format teks sehingga perbedaan kapitalisasi tidak memengaruhi proses analisis. Penerapan *case folding* dilakukan untuk memastikan bahwa kata yang sama dapat dikenali sebagai satu kata yang konsisten. Berikut adalah contoh penerapan *case folding* dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Contoh *Case Folding*

| Sebelum | Sesudah |
|---|---|
| Kalau macet total, bikin was2 lowbat nggak ada tempat charge | kalau macet total, bikin was2 lowbat nggak ada tempat charge |

#### 3.2.2.2 Pembersihan Karakter

Pada tahap ini, data akan dilakukan pembersihan teks komentar dengan menghapus karakter-karakter yang dianggap tidak relevan terhadap proses analisis. Karakter yang dihilangkan mencakup emoji, angka, simbol khusus, URL, serta alamat *email* yang terdapat dalam komentar. Pembersihan akan menggunakan pendekatan berbasis pola (*regex*) untuk mendeteksi dan menghapus karakter-karakter tersebut secara otomatis dari setiap komentar. Melalui pembersihan karakter yang tidak relevan, hasil analisis akan menjadi lebih fokus pada konten utama yaitu komentar. Berikut merupakan contoh hasil pembersihan karakter yang dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Contoh Pembersihan Karakter

| Sebelum | Sesudah |
|---|---|
| makanya mobil listrik harus murah bgt supaya org yg dr mobil bbm pindah😁 negara juga hemat ngga perlu subsidi2 | makanya mobil listrik harus murah bgt supaya org yg dr mobil bbm pindah negara juga hemat ngga perlu subsidi |

### 3.2.2.3 Pembersihan Data *Null*

Pada tahap ini, data komentar yang kosong atau tidak memiliki konten (*null*) akan dihapus dari dataset. Tujuan pembersihan data kosong adalah untuk memastikan hanya data yang memiliki nilai informasi saja yang dipertahankan. Proses pembersihan dilakukan dengan melakukan pengecekan baris per baris data lalu kemudian mengeliminasi seluruh entri yang kosong atau *null*. Data kosong ini bisa disebabkan karena komentar yang hanya memunculkan emoji, angka, atau simbol pada tahap sebelumnya sudah dihapus, sehingga komentarnya sudah tidak ada namun datanya tidak terhapus. Berikut adalah contoh pembersihan data *null* yang disajikan pada Tabel 3.4.

Tabel 3. 4 Contoh Pembersihan Data *Null*

| Sebelum | | Sesudah | |
|---|---|---|---|
| username | full_text | username | full_text |
| *NULL* | saya mendukung kalo bisa beli | - | - |
| neta🌙 | *NULL* | - | - |

### 3.2.2.4 Pembersihan Data Duplikat

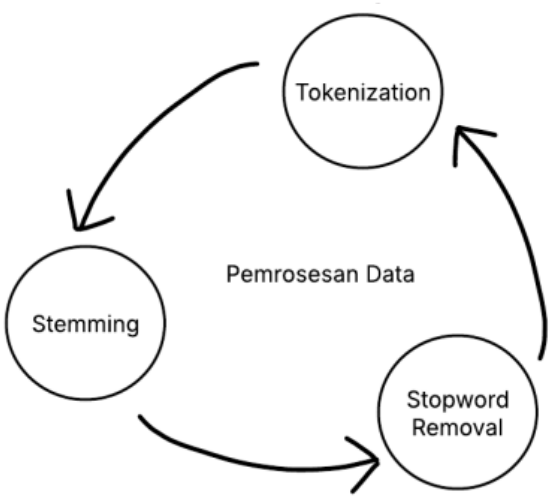
Komentar yang terdeteksi sebagai duplikat atau pengulangan akan dihapus dari dataset penelitian. Proses ini akan dilakukan dengan membandingkan setiap baris komentar berdasarkan kolom tabel *username* kemudian akan mengeliminasi baris yang terindikasi pengulangan. Pembersihan data duplikat ini bertujuan untuk memastikan bahwa setiap opini yang dianalisis berasal dari data yang unik dan tidak berulang. Data yang duplikat dapat berpotensi sebagai komentar spam sehingga proses analisis akan

menjadi kurang relevan. Berikut contoh dari pembersihan data duplikat yang disajikan pada Tabel 3.5.

Tabel 3. 5 Contoh Pembersihan Data Duplikat

| Sebelum | | Sesudah | |
|---|---|---|---|
| username | full_text | username | full_text |
| ullzs | darimana duit nya ,darimana duit nyaaaa | ullzs | darimana duit nya ,darimana duit nyaaaa |
| ullzs | darimana duit nya ,darimana duit nyaaaa | - | - |

### 3.2.3 Pemrosesan Data




Gambar 3. 3 Pemrosesan Data

Pada Gambar 3.3, tahap pemrosesan menggunakan data yang berbentuk teks mempunyai tiga langkah, yaitu *stemming*, *tokenization*, dan *stoword removal*. Tahap pemrosesan data dimaksudkan untuk memaksimalkan kualitas dari data sehingga siap dianalisis lebih lanjut. Tahap ini akan menghasilkan data teks yang lebih relevan dan terstruktur untuk meningkatkan akurasi dalam analisis sentimen.

#### 3.2.3.1 *Stemming*

Tahap *stemming* bertujuan untuk mengurangi jumlah indeks dalam satu data dengan mengubah kata yang memiliki awalan dan akhiran menjadi bentuk dasar (Syihabudin et al., 2023). *Stemming* dilakukan untuk menormalkan kata dengan mengonversinya ke bentuk dasarnya (Riskawati et al., 2024). Termasuk

menghapus kata umum yang kurang relevan, sehingga memaksimalkan proses analisis.

*Stemming* penting dilakukan agar kata yang berasal dari turunan bentuk yang sama dapat dihitung sebagai satu kata dalam analisis. Pada penelitian ini Proses *stemming* akan diterapkan menggunakan fungsi *stemmer* yang ada pada *library sastrawi* yang mampu menyesuaikan kata baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Melalui penggunaan *stemming*, variasi kata yang sejenis dapat diminimalkan, sehingga jumlah indeks kata dalam dataset berkurang secara signifikan. Berikut contoh dari proses *stemming* disajikan pada Tabel 3.6.

Tabel 3. 6 Contoh *Stemming*

| Sebelum | Sesudah |
|---|---|
| ngecas nya lama, ada kebutuhan mendadak jadi repot masih ngecas | cas lama, ada butuh mendadak jadi repot masih cas |

### 3.2.3.2 *Tokenization*

*Tokenization* dilaksanakan dengan memecah teks kedalam unit-unit yang lebih kecil. Proses ini membagi teks dari satu kalimat menjadi beberapa potongan kata (Fikri et al., 2020). Pada penelitian ini, tokenisasi akan dilakukan dengan cara membagi setiap komentar yang awalnya berupa satu baris kalimat menjadi daftar kata, menggunakan spasi, tanda baca, atau karakter pemisah tertentu sebagai batasan pemisahan.

Hasil dari tahap *tokenization* ini akan digunakan dalam pembobotan TF-IDF saat pelabelan sentimen. Tahap tokenisasi ini sangat penting karena kesalahan dalam pemecahan teks dapat berdampak pada ketidakakuratan dalam penghitungan bobot kata dan klasifikasi sentimen. Contoh dari proses *tokenizing* disajikan pada Tabel 3.7.

Tabel 3. 7 Contoh *Tokenizing*

| Sebelum | Sesudah |
|---|---|
| spklu udah banyak setiap kantor PLN | ['spklu', 'udah', 'banyak', 'setiap', 'kantor', 'PLN'] |

#### 3.2.3.3 *Stopword Removal*

Tahap *stopword removal* dilakukan dengan penyaringan untuk mengidentifikasi kata-kata yang tidak memiliki nilai atau relevansi, karena tidak memberikan informasi pada kalimat (Alin et al., 2023). Kata-kata yang terindikasi tidak memiliki *value* atau makna sentimen seperti "yang", "di", dan lainnya akan dihapus dari dataset. Proses *stopword removal* bertujuan untuk menjaga agar hanya kata bermakna yang dipertahankan dalam analisis.

Melalui kata yang sudah dihilangkan, proses analisis teks menjadi lebih fokus pada informasi yang lebih bermakna. Lalu akan dilakukan pengecekan kembali terhadap kata-kata yang tersisa untuk memastikan tidak ada kesalahan ketik (*typo*) yang berpotensi mengganggu analisis. Jika ditemukan kata *typo*, kata tersebut akan dinormalisasi ke bentuk yang sesuai dengan ejaan baku menurut KBBI. Dari hasil pemrosesan ini, maka data akan menjadi lebih bersih sehingga dapat mendukung ketepatan dalam tahap klasifikasi sentimen. Contoh stopword removal disajikan pada Tabel 3.8.

Tabel 3. 8 Contoh *stopword removal*

| Sebelum | Sesudah |
|---|---|
| mobstrik hanya baik utk dlm kota, luar kota jarak jauh... ribet dag-dig-dug, ngecas lamaaa dijalaaan | mobil listrik hanya baik untuk dalam kota, luar kota jarak jauh... ribet dag-dig-dug, ngecas lama dijalan |

### 3.2.4 Pelabelan

Pelabelan digunakan untuk menunjukan respon dari komentar dalam dataset (Santoso et al., 2022). Pada tahap ini, komentar akan diberi label untuk diklasifikasi menjadi tiga kategori yaitu respon positif, negatif, dan juga netral. Proses ini juga akan digunakan dengan menetapkan nilai bobot pada setiap kata dengan metode pembobotan TF-IDF kemudian divalidasi pakar bahasa. Hasil dari proses pembobotan dan validasi tersebut nantinya akan dikerjakan oleh algoritma *Machine Learning* untuk proses klasifikasi (Noviana & Rasal, 2023). Langkah-langkah yang digunakan dalam pelabelan dalam pembobotan TF-IDF disajikan sebagai berikut.

#### 3.2.4.1 Prosedur *Term Frequency* (TF)

Pada tahap *Term Frequency* (TF) Proses diawali dengan menerapkan proses tokenisasi dari tahap sebelumnya. Kemudian dilakukan perhitungan terhadap jumlah kemunculan setiap kata dalam satu dokumen. Nilai TF diperoleh dengan membagi frekuensi kemunculan kata dengan jumlah total kata dalam dokumen tersebut. Hasil dari perhitungan ini memberikan informasi mengenai tingkat kemunculan relatif suatu kata dalam konteks dokumen. Pseudocode perhitungan TF dapat dilihat pada Tabel 3.9.

Tabel 3. 9 pseudocode *Term Frequency* (TF)

| ***Term Frequency (TF)*** |
|---|
| BEGIN |
| INPUT: Dokumen D yang terdiri dari teks |
| OUTPUT: Daftar kata dengan nilai TF masing-masing |
| FOR setiap dokumen di dalam dataset sentimen: |
| Pisahkan teks menjadi daftar kata (tokenization) |
| CALCULATE total kata dalam dokumen |
| FOR setiap kata dalam dokumen: |
| CALCULATE frekuensi kemunculan kata dalam dokumen |
| TF(kata) = jumlah kemunculan kata / total kata pada dokumen |
| SAVE hasil TF ke dalam daftar |
| END |

#### 3.2.4.2 Prosedur *Document Frequency* (DF)

Tahap *Document Frequency* (DF) dalam penelitian ini akan menghitung jumlah dokumen dalam dataset yang memuat masing-masing kata unik. Proses ini dilakukan dengan mengidentifikasi setiap kata yang muncul dalam dokumen. Kemudian menghitung berapa banyak dokumen berbeda yang mengandung kata tersebut. Nilai DF dari suatu kata mencerminkan seberapa umum kata tersebut digunakan di seluruh dokumen yang tersedia. Pseudocode perhitungan DF dapat dilihat pada Tabel 3.10.

Tabel 3. 10 pseudocode *Document Frequency* (DF)

| ***Document Frequency* (DF)** |
|---|
| BEGIN<br>INPUT: Semua dokumen dalam dataset<br>OUTPUT: DF untuk setiap kata<br><br>INISIALISASI dictionary kosong pada df_manual<br>df_manual = { … }<br>FOR setiap kata unik dalam dataset:<br>CALCULATE banyaknya dokumen yang memuat kata tersebut<br>DF(kata) = jumlah dokumen yang mengandung kata<br>SAVE hasil DF ke dalam df_manual<br><br>END |

### 3.2.4.3 Prosedur *Inverse Document Frequency* (IDF)

Prosedur *Inverse Document Frequency* (IDF) dilakukan untuk menghitung tingkat keunikan atau kelangkaan suatu kata dalam keseluruhan dokumen yang tersedia. Nilai IDF dihitung berdasarkan rasio antara jumlah total dokumen dengan jumlah dokumen yang mengandung kata tersebut. Kata yang muncul lebih sedikit dari dokumen akan memperoleh nilai IDF yang lebih tinggi. Pseudocode perhitungan IDF dapat dilihat pada Tabel 3.11.

Tabel 3. 11 pseudocode Inverse Document Frequency (IDF)

| ***Invers Document Frequency* (IDF)** |
|---|
| BEGIN<br>INPUT: Total jumlah dokumen (N) dalam dataset<br>OUTPUT: Nilai IDF untuk setiap kata unik dalam dataset<br><br>FOR setiap kata unik dalam dataset:<br>1. Periksa berapa banyak dokumen yang terdapat kata tersebut (DF). |

| ***Invers Document Frequency* (IDF)** |
|---|
| 2. Bandingkan jumlah total dokumen dengan jumlah dokumen yang mengandung kata tersebut.<br>3. Gunakan perhitungan untuk menentukan seberapa jarang kata tersebut muncul dalam seluruh dokumen.<br>SAVE hasil IDF ke dalam dictionary idf_manual.<br><br>END |

### 3.2.4.4 Prosedur TF-IDF

Pada tahap *Term Frequency–Inverse Document Frequency* (TF-IDF) dilakukan proses pembobotan kata dengan menggabungkan nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) yang telah diperoleh sebelumnya. Kedua nilai tersebut dikalikan untuk menghasilkan bobot akhir TF-IDF. Hasil pembobotan ini disimpan dalam struktur data khusus dan digunakan sebagai dasar dalam proses pelabelan. Pseudocode perhitungan TF-IDF dapat dilihat pada Tabel 3.12.

Tabel 3. 12 pseudocode TF-IDF

| ***Term Frequency-Invers Document Frequency* (TF-IDF)** |
|---|
| BEGIN<br>INPUT: TF dan IDF dari setiap kata dalam dokumen<br>OUTPUT: Nilai bobot TF-IDF untuk setiap kata pada dokumen<br><br>FOR setiap kata pada dokumen:<br>1. Ambil nilai TF dari kata tersebut, yang menunjukkan tingkat frekuensi kemunculan kata dalam dokumen.<br>2. Ambil nilai IDF dari kata tersebut, yang menunjukkan seberapa unik kata tersebut dalam seluruh dataset.<br>3. Kombinasikan kedua nilai ini untuk menentukan frekuensi pentingnya kata pada dokumen.<br>SAVE hasil TF-IDF kedalam df_tfidf |

| ***Term Frequency-Invers Document Frequency* (TF-IDF)** |
|---|
| END |

### 3.2.4.5 Prosedur Pelabelan

Pada tahap ini, komentar akan diberi label sentimen dengan membobotkan setiap kata menggunakan metode TF-IDF yang telah dihitung sebelumnya. Komentar-komentar yang telah melalui tahapan pembobotan kata akan diklasifikasikan ke dalam kategori sentimen, yaitu positif, negatif, atau netral berdasarkan skor akhir yang dihitung. Skor tersebut diperoleh dari akumulasi bobot TF-IDF kata-kata dalam komentar dengan mengacu pada kata yang memiliki kecendrungan positif maupun negatif, sementara kata yang tidak memiliki kecendrungan akan menjadi nilai netral. Pseudocode perhitungan TF-IDF dapat dilihat pada Tabel 3.13.

Tabel 3. 13 Pseudocode Pelabelan Sentimen

| ***Pelabelan Sentimen Analisis*** |
|---|
| BEGIN<br>INPUT: Dataset komentar, Data TF-IDF, Daftar Kata Positif dan Negatif<br>OUTPUT: Skor Sentimen dan Klasifikasi Sentimen pada total dataset<br>DEFINE positive_words = ['bagus', 'murah', … ,'hemat', 'garansi']<br>DEFINE negative_words = ['kampung', 'rusak', … ,'baterai', ‘fomo’]<br><br>FOR setiap komentar dalam dataset:<br>INISIALISASI score = 0<br>Pisahkan teks menjadi daftar kata<br><br>FOR setiap kata dalam komentar:<br>SEARCH nilai TF-IDF dari kata dalam df_tfidf<br>IF kata ada dalam daftar kata positive_words THEN<br>score += TF-IDF(kata)<br>ELSE IF kata ada dalam daftar kata negative_words THEN<br>score -= TF-IDF(kata)<br><br>IF score > 0.1 THEN |

***Pelabelan Sentimen Analisis***

```
        Label Sentimen = "Positif"
    ELSE IF score < -0.1 THEN
        Label Sentimen = "Negatif"
    ELSE
        Label Sentimen = "Netral"
    SAVE score dan label sentimen ke dalam dataset
END
```

Pada Tabel 3.13 ditunjukan pseudocode dari proses pelabelan sentimen berdasarkan bobot TF-IDF. Pada tahap ini, setiap komentar diproses dengan membandingkan kata-kata yang terdapat di dalamnya terhadap daftar kata yang memiliki positif dan negatif yang telah ditentukan. Apabila suatu kata termasuk dalam daftar positif, maka nilai TF-IDF dari kata tersebut akan ditambahkan ke skor total komentar. Sebaliknya, apabila kata termasuk dalam daftar negatif, nilai TF-IDF dikurangkan dari skor total.

Setelah seluruh kata dalam komentar diproses, skor akhir yang diperoleh menjadi dasar dalam menentukan klasifikasi sentimen. Skor tersebut merupakan ambang batas atau threshold dalam penentuan sentimen. Jika skor lebih besar dari 0,1 maka komentar dikategorikan sebagai positif. Jika skor lebih kecil dari -0,1, maka dikategorikan sebagai negatif. Sedangkan jika skor berada di antara -0,1 dan 0,1, komentar dikategorikan sebagai netral. Proses ini memungkinkan klasifikasi sentimen dilakukan secara sistematis dan berbasis pembobotan kata yang relevan terhadap opini yang dianalisis.

### 3.2.5 Pemodelan

Tahap pemodelan dilakukan dengan model klasifikasi diuji dengan data komentar yang telah melewati tahap *labelling* dan di validasi pakar bahasa. Rasio dari tahap pemodelan akan berkisar 80% data latih serta 20% data uji. Pemodelan ini melibatkan algoritma *Naïve Bayes* dan *Support Vector Machine* kernel linear. Algoritma tersebut dipilih dalam analisis sentimen karena mempunyai kelebihan saat menangani serta mengklasifikasikan data yang kompleks, di dasari penelitian terkait. Proses pemodelan bertujuan untuk melatih model dengan algoritma agar

dapat mengenali pola sentimen berdasarkan bobot kata yang telah dihitung menggunakan TF-IDF. Hasil dari proses pemodelan akan menghasilkan model klasifikasi yang mampu membedakan sentimen positif, negatif, dan netral dengan tingkat akurasi yang optimal.

#### 3.2.5.1 Prosedur *Naïve Bayes*

Pada tahap ini, algoritma *Naïve Bayes* akan digunakan untuk membangun model klasifikasi sentimen berdasarkan prinsip probabilitas yang mengacu pada Teorema Bayes. Proses *Naïve Bayes* melibatkan perhitungan probabilitas *prior* untuk setiap kelas, dan *likelihood* atau *conditional probability* untuk setiap kata dalam dataset. Setelah model terbentuk, data uji akan diprediksi dengan memilih kelas yang memiliki probabilitas tertinggi berdasarkan kombinasi probabilitas kata-kata dalam komentar. Adapun pseudocode algoritma *Naïve Bayes* disajikan pada Tabel 3.14.

Tabel 3. 14 Pseudocode *Naive Bayes*

| ***Naïve Bayes*** |
|---|
| BEGIN<br>INPUT: Data latih (TF-IDF divalidasi pakar), Label sentimen<br>OUTPUT: Model klasifikasi *Naïve Bayes*<br><br>1. Tentukan daftar kelas sentimen C = {positif, negatif, netral}.<br>2. Hitung probabilitas *prior* P(Ck) untuk setiap kelas Ck.<br>3. Untuk setiap kata x dalam dataset :<br>a. Hitung probabilitas likelihood P(x \| Ck) untuk setiap kelas.<br>4. Untuk setiap komentar baru X :<br>a. Hitung skor P(Ck \| komentar) = P(Ck) × P(x1 \| Ck) × P(x2 \| Ck) × ... × P(xn \| Ck)<br>b. Pilih kelas Ck dengan skor probabilitas tertinggi sebagai hasil klasifikasi.<br>5. Evaluasi performa model menggunakan *confusion matrix*.<br><br>END |

Sumber : (Riyadi et al., 2021)

Tabel 3.14 menunjukan pseudocode prosedur penggunaan algoritma *Naïve Bayes* untuk klasifikasi sentimen. Proses diawali dengan menentukan kategori sentimen, menghitung probabilitas masing-masing kelas, lalu menghitung probabilitas setiap kata terhadap tiap kelas. Pada komentar baru, dihitung skor probabilitas dari tiap kelas, dan kelas dengan skor tertinggi dipilih sebagai hasil prediksi. Kemudian evaluasi model dilakukan menggunakan *confusion matrix*.

#### 3.2.5.2 Prosedur *Support Vector Machine*

Pada tahap ini, algoritma *Support Vector Machine* (SVM) dengan kernel linear akan diterapkan untuk membangun model klasifikasi sentimen. SVM akan bekerja dengan mencari *hyperplane* terbaik yang memisahkan data dari masing-masing kategori sentimen berdasarkan bobot kata dari hasil TF-IDF. Proses SVM difokuskan pada pencarian margin terbesar antara kelas berbeda untuk memastikan generalisasi model yang optimal. Setiap komentar akan diubah menjadi representasi vektor dan dipetakan ke ruang dimensi tinggi. Model kemudian memutuskan sentimen berdasarkan sisi *hyperplane* tempat komentar tersebut berada. Adapun pseudocode algoritma SVM disajikan pada Tabel 3.15.

Tabel 3. 15 Pseudocode *Support Vector Machine*

| ***Support Vector Machine*** |
|---|
| BEGIN<br>INPUT: Data latih (TF-IDF divalidasi pakar), Label sentimen<br>OUTPUT: Model klasifikasi SVM Linear<br><br>1. Representasikan setiap komentar sebagai vektor fitur berbasis TF-IDF.<br>2. Tentukan *hyperplane* optimal yang memisahkan kelas sentimen dengan margin maksimal.<br>3. Gunakan fungsi kernel linear untuk menghitung margin antar kelas.<br>4. Untuk setiap data uji:<br>a. Hitung posisi vektor terhadap *hyperplane*.<br>b. Prediksi kelas sentimen berdasarkan sisi *hyperplane* tempat vektor tersebut berada. |

| ***Support Vector Machine*** |
|---|
| 5. Evaluasi performa model menggunakan *confusion matrix*. |
| END |

Sumber : (Geeksforgeeks, 2025)

Pada Tabel 3.15 menunjukan pseudocode dari proses klasifikasi sentimen menggunakan *Support Vector Machine* dengan kernel linear. Komentar dikonversi ke vektor TF-IDF, lalu dipisahkan oleh *hyperplane* dengan margin terbesar menggunakan kernel linear. *Hyperplane* dibentuk sedemikian rupa agar margin antara data dari masing-masing kelas sentimen menjadi maksimal, sehingga model dapat membedakan sentimen dengan akurasi tinggi. Kemudian setiap komentar data uji, posisi vektor terhadap *hyperplane* dihitung guna memprediksi label sentimen berdasarkan sisi mana vektor tersebut berada. Setelah proses klasifikasi, kemudian hasilnya dievaluasi menggunakan *confusion matrix*.

### 3.2.6 Evaluasi Model

Tahap evaluasi model bertujuan dengan mengevaluasi perfoma dari model klasifikasi yang telah dibangun pada tahap pemodelan. Evaluasi dari hasil pemodelan dilakukan dengan *confusion matrix* untuk menilai dari perfoma model pada *machine learning* (Musfiroh et al., 2021). Pada penelitian ini, *confusion matrix* berfungsi menjadi alat ukur sejumlah metrik evaluasi perfoma model, seperti akurasi, presisi, dan *recall*, serta *f1-score* (Safitri et al., 2024).

Dari hasil *confusion matrix*, akurasi akan dihitung sebagai rasio prediksi benar terhadap seluruh prediksi, presisi akan berperan sebagai ketepatan prediksi untuk masing-masing kelas, *recall* bekerja dengan mengukur kemampuan model menangkap semua data yang benar, dan *f1-score* berperan sebagai keseimbangan antara presisi dan *recall*. Setiap metrik evaluasi ini akan dibandingkan antara algoritma *Naïve Bayes* dan *Support Vector Machine* untuk menentukan model mana yang paling optimal untuk memahami persepsi masyarakat terhadap mobil listrik di platfom TikTok.