BAB 3
METODE PENELITIAN

Penelitian ini menganalisis sentimen komentar pengguna TikTok dalam menggunakan ChatGPT di skripsi. Data yang digunakan diperoleh dari komentar pada konten TikTok yang membahas hal tersebut. Untuk proses klasifikasi sentimen, penelitian in menggunakan metode Naïve Baye guna membagi komentar ke dalam tiga kategori, yaitu sentimen positif, negatif, dan netral. Uraian selanjutnya mencakup penjelasan mengenai bahan, alat, metode, serta tahapan dalam melakukan analisis sentimen terhadap komentar-komentar tersebut.

3.1 BAHAN DAN ALAT PENELITIAN

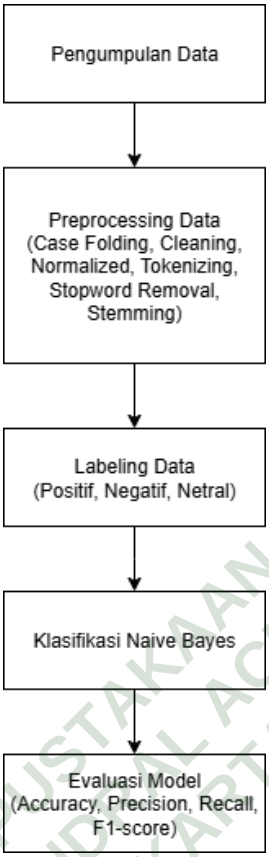
Bahan dalam penelitian ini, berupa komentar pengguna TikTok pada postingan yang terkait dengan penggunaan ChatGPT dalam penyusunan skripsi. Untuk menjaga privasi, nama akun pengguna tiktok yang berkomentar tidak dicantumkan dan hanya teks komentar yang dikumpulkan. Komentar diperoleh melalui scraping menggunakan Tapicker, yang memungkinkan pengambilam komentar secara otomatis dari postingan terkait.

Sistem operasi dan program-program aplikasi yang dipergunakan dalam penelitian ini adalah:

1. Sistem Operasi: Sistem Operasi Windows 11 Pro
2. Bahasa pemrograman: Python
3. Framework: Jupyter, Streamlit
4. Tools: TikTok, Tapicker

3.2 JALAN PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan utama yang dimulai dari proses pengumpulan data hingga evaluasi model klasifikasi. Data yang digunakan adalah komentar pengguna TikTok yang berkaitan dengan penggunaan ChatGPT dalam skripsi. Tahapan-tahapan dalam penelitan ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Pada Gambar 3.1 menunjukkan tahapan-tahapan dalam proses penelitian:

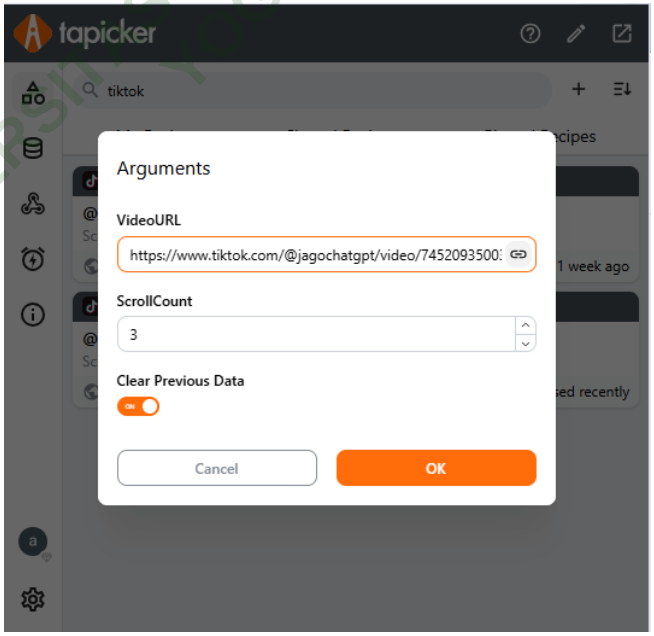
1. Pengumpulan Data: Data dikumpulkan dari komentar pada video TikTok yang membahas topik penggunaan ChatGPT dalam penulisan skripsi. Pengambilan data dilakukan dengan metode scraping menggunakan tools seperti Tapicker.
2. Preprocessing Data: Data mentah yang telah dikumpulkan selanjutnya melalui tahap praproses yang meliputi:
   - Case Folding: mengubah semua huruf menjadi huruf kecil.
   - Cleaning: menghapus karakter khusus, simbol, angka, atau link.
   - Normalisasi: mengganti kata tidak baku dengan kata baku.
   - Tokenizing: memisahkan kalimat menjadi kata-kata.
   - Stopword removal: menghapus kata-kata umum yang tidak memiliki makna.

- Stemming: mengubah kata menjadi bentuk dasarnya.

3. Labeling: Masing-masing komentar diberi label sentimen sesuai dengan isi teksnya, yaitu berupa sentimen positif, negatif, dan netral.
4. Klasifikasi Naïve Bayes: Komentar yang telah diberikan lebel selanjutnya dianalisis menggunakan Naïve Bayes guna mengidentifikasi jenis sentimen yang terkandung dalam setiap komentar.
5. Evaluasi Model: Kinerja Naïve Bayes dalam mengklasifikasikan data sentimen dinilai menggunakan beberapa metrik evaluasi, seperti *Accuracy*, *Precision*, *Recall*, dan *F1-score*.

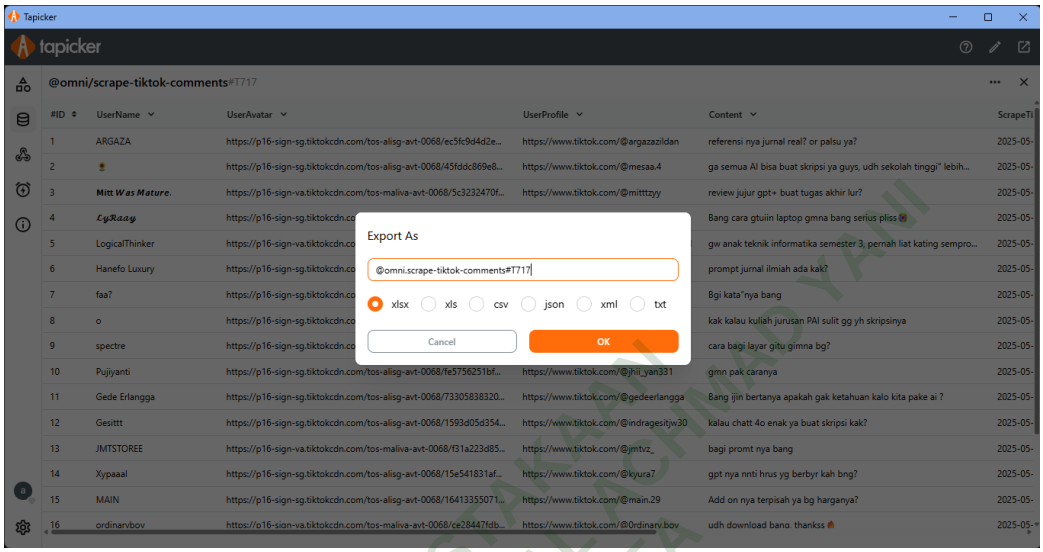
3.2.1 Pengumpulan Data

Proses pengumpulan data dilakukan dengan mengekstraksi komentar dari video TikTok yang membahas topik penggunaan ChatGPT dalam penyusunan skripsi. Komenatar dikumpulkan menggunakan tools scraping bernama Tapicker, yang memungkinkan pengambilan komentar secara otomatis dari suatu postingan. Proses pengambilan data dilakukan dengan cara memasukkan tautan (link) video yang akan diekstraksi, seperti ditunjukkan pada Gambar 3.2.



Gambar 3.2 Masukkan Link Video ke Tapicker

Setelah proses pengambilan data selesai, data komentar TikTok dapat diekspor atau diunduh dalam berbagai format, salah satunya adalah format file Excel yaitu .xlxs yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Export Data Komentar Ke File Excel

Hasil dari data komentar yang telah diunduh dapat dilihat pada Tabel 3.1

Tabel 3.1 Hasil Ekspor Data Komentar

No	Komentar
1	pkony chatgpt trbaik dah ngbntu bgttt
2	tapi kutipan dari referensinya kurang sesuai dan ada yg ga sesuai juga
3	tapi kebanyakan dosen sekarang tau tulisan bot sama tulisan mahasiswa
4	Dosenku di itb, ngasih tau kalo mau nyari ide boleh2 aja pake chatgpt. Tapi harus di cek berkali2 dan sumberny akurat.
5	tpi kan kena plagiasi woyy
6	ttp bijak menggunakan ai
7	Saat ini saya sedang membuat proposal tesis dan belum merasakan manfaat chat gpt, apalagi cabang ilmu kedokteran dan biomedik
8	kadang jurnalnya ga sesuai
9	gpt enak dipake asal tahu kalimat perintah yang benar
10	palingan revisi lagi

Pada Tabel 3.1 menunjukkan contoh data komentar yang telah diunduh dari video TikTok terkait penggunaan ChatGPT dalam skripsi. Komentar-komentar tersebut diperoleh tanpa menyertakan nama pengguna TikTok, sebagai upaya menjaga privasi pengguna. Seluruh data yang dikumpulkan hanya berisi teks komentar.

Secara keseluruhan, jumlah data yang berhasil diperoleh sebanyak 1.200 komentar. Data yang telah dikumpulkan kemudian digunakan sebagai dataset utama dalam proses analisis sentimen. Sebelum dilakukan klasifikasi sentimen, preprocessing untuk meningkatkan kualitas data dan akurasi hasil analisis.

3.2.2 Preprocessing Data

Pra-pemrosesan data merupakan tahap penting dalam analisis sentimen yang bertujuan memastikan bahwa data teks yang digunakan sudah bersih, relevan, dan siap untuk diproses pada tahap analisis berikutnya. Pada tahap ini, komentar yang diperoleh dari TikTok diproses melalui berbagai tahapan pembersihan dan normalisasi teks guna meningkatkan ketepatan serta efisiensi dalam proses klasifikasi sentimen.

1. **Case Folding**

   Mengubah semua teks komentar menjadi huruf kecil, perbedaan interprestasi antara huruf besar dan huruf kecil dapat dihindari dalam proses analisis. Berikut kode untuk melakukan tahapan *case folding.*

```
df_data = pd.read_excel('data_komentar.xlsx')
def lowercase():
    lower_word = df_data['Komentar'].str.lower()
    return lower_word
df_data['Case_Folding'] = lowercase()
```

   Kode diatas berguna untuk membuat data teks pada kolom komentar menjadi seragam dalam huruf kecil semua.

2. Cleaning

Cleaning adalah menghilangkan karakter atau elemen yang tidak diperlukan dari teks guna mencegah gangguan dalam proses analisis lebih lanjut. Berikut kode untuk tahap *cleaning*.

```
df = pd.read_excel('case_folding.xlsx')
def clean_text(text):
    if pd.isnull(text):
       return ""
    text = str(text)
    text = re.sub(r'http\S+|www.\S+', '', text)
    text = re.sub(r"[^\x00-\x7F]+", " ", text)
    text = re.sub(r'[^a-zA-Z\s-]', '', text)
    text = text.replace("-", " ")
    text = re.sub(r'\s+', ' ',  text).strip()
    return text
df['Cleaned'] = df['Case_Folding'].apply(clean_text)
```

Kode diatas berfungsi untuk membersihkan teks dengan menghapus URL, emoji, angka, simbol, dan menghapus spasi berlebih pada kolom *Case_Folding*.

3. Normalisasi

Normalisasi adalah proses mengubah kata-kata yang tidak dibakukan, termasuk kata-kata slang atau salah eja, menjadi bentuk yang sesuai dengan aturan ejaan yang benar. Daftar kalimat yang mengandung kata-kata yang tidak dibakukan dan hasil normalisasinya ditunjukkan pada Tabel 3.2.

Tabel 3.2 Daftar Kalimat Normalisasi

Tidak Baku	Baku
Copas	Copy Paste
Enggak	Tidak
Gua	Saya
Jgn	Jangan
Klo	Kalau
Liat	Lihat

Make	Pakai
Mmg	Memang
Mna	Mana

Pada Tabel 3.2 merupakan daftar kalimat normalisasi yang nantinya digunakan sebagai acuan dalam proses normalisasi teks komentar. Kata-kata tidak baku dalam teks akan diganti dengan bentuk baku sesuai daftar tersebut. Berikut kode untuk proses normalisasi.

```
   df_normalisasi = pd.read_excel('kalimat_normalisasi.xlsx')
   kamus_normalisasi = dict(zip(df_normalisasi['Kata_Asli'], df_normalisasi['Normalisasi']))
   def normalize_text(text):
       words = text.split()
       normalized_words = [kamus_normalisasi.get(word, word) for word in words]
       return ' '.join(normalized_words)
   df['Normalized'] = df['Cleaning'].apply(normalize_text)
```

Kode di atas merupakan implementasi proses normalisasi pada kolom *Cleaned*, yaitu kolom yang berisi teks komentar yang sudah melalui tahap pembersihan (*cleaning*).

4. Tokenizing

Proses membagi teks menjadi unit yang lebih kecil, yang berfungsi sebagai dasar untuk analisis lebih lanjut dari data teks. Berikut kode untuk melakukan *tokenizing*.

```
   def tokenize(text):
       if pd.isnull(text):
           return []
       return text.split()
   df['Tokenizing'] = df['Normalized'].apply(tokenize)
```

Kode di atas berfungsi untuk melakukan proses tokenizing, yaitu memecah suatu kalimat atau teks menjadi setiap kata secara terpisah, sehingga masing-

masing kata dapat dianalisis secara individual dalam tahapan pemrosesan data selanjutnya.

5. Stopword Removal

Stopword adalah penghapusan kata-kata yang sering muncul dan tidak relevan dengan analisis teks. Daftar kata yang akan dihapus dapat dilihat pada Tabel 3.3.

Tabel 3.3 Daftar Kata *Stopword Library* Sastrawi

Daftar Kata Stopword Library Sastrawi
[‘ada’, ‘adalah’, ‘agar’, ‘agak’, ‘amat’, ‘anda’, ‘anu’, ‘antara’, ‘apalagi’, ‘apakah’, ‘atau’, ‘bahwa’, ‘bagi’, ‘bagaimanapun’, ‘bisa’, ‘belum’, ‘boleh’, ‘begitu’, ‘dahulu’, ‘dalam’, ‘daripada’, ‘dari’, ‘demi’, ‘demikian’, ‘dengan’, ‘di’, ‘dia’, ‘dll’, ‘dua’, ‘dsb’, ‘dst’, ‘dulunya’, ‘guna’, ‘hal’, ‘hanya’, ‘harus’, ‘ingin’, ‘ini’, ‘itu’, ‘itulah’, ‘iya’, ‘jangan’, ‘jika’, ‘juga’, ‘kah’, ‘kami’, ‘karena’, ‘kecuali’, ‘kemana’, ‘kepada’, ‘ketika’, ‘kembali’, ‘kita’, ‘lah’, ‘lain’, ‘lagi’, ‘maka’, ‘mari’, ‘masih’, ‘melainkan’, ‘mengapa’, ‘menurut’, ‘mereka’, ‘nanti’, ‘namun, ‘nggak’, ‘oh’, ‘ok’, ‘oleh’, ‘pada’, ‘para’, ‘pasti’, ‘pun’, ‘saja’, ‘sambil’, ‘sampai’, ‘saya’, ‘sebagai’, ‘sebab’, ‘sebelum’, ‘sebetulnya’, ‘secara’, ‘sedangkan’, ‘seharusnya’, ‘sehingga’, ‘sejak’, ‘selagi’, ‘selain’, ‘selalu’, ‘sementara’, ‘semua’, ‘seolah’, ‘seperti’, ‘seraya’, ‘serta’, ‘sesuatu’, ‘sesudah’, ‘setelah’, ‘setiap’, ‘seterusny’, ‘setidaknya’, ‘sudah’, ‘supaya’ ‘sungguh’, ‘tapi’, ‘tanpa’, ‘tentang’, ‘tentu’, ‘terhadap’, ‘tidak’, ‘toh’, ‘tolong’, ‘untuk’, ‘walau’, ‘ya’, ‘yaitu’, ‘yakni’, ‘yang’]

Pada Tabel 3.3 merupakan daftar kata *stopwords* yang digunakan dalam proses *Stopword Removal*. Kata-kata tersebut diambil dari *library* Sastrawi. Berikut kode untuk proses *stopword removal.*

```
df = pd.read_excel('tokenizing.xlsx')
factory = StopWordRemoverFactory()
stopwords = set(factory.get_stop_words())
def remove_stopwords_from_tokenizing(text):
    if pd.isnull(text):
        return ""
    tokens = text.lower().split()
```

```
        filtered = [word for word in tokens if word not in
stopwords]
        return ' '.join(filtered)
    df['Stopword_Removed']=
df['Tokenizing'].apply(remove_stopwords_from_tokenizing)
```

Kode di atas merupakan kode untuk melakukan stopword removal pada kolom *Tokenizing*. Setiap token dalam kalimat akan dibandingkan dengan daftar stopword dari library Sastrawi, dan jika termasuk dalam daftar tersebut maka akan dihapus.

6. Stemming

Stemming melibatkan pengubahan kata-kata dengan imbuhan atau bentuk turunan kembali ke bentuk aslinya. Tujuannya adalah untuk menstandarisasi varian kata dengan makna dasar yang sama. Berikut kode untuk tahapan tersebut.

```
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    def stemmed_wrapper(term):
        return stemmer.stem(term)
    df = pd.read_excel('stopword_removed.xlsx')
    stopwords_tweet = df['Stopword_Removed'].astype(str)
    term_dict = {}
    for document in stopwords_tweet:
        for term in document.split():
            if term not in term_dict:
                term_dict[term] = ""
    for term in term_dict:
        stemmed = stemmed_wrapper(term)
        term_dict[term] = stemmed
    def get_stemmed_term(document):
        return ' '.join([term_dict.get(term, term) for term in
document.split()])
    df['Stemming'] = stopwords_tweet.apply(get_stemmed_term)
```

Kode di atas merupakan proses stemming yang dilakukan terhadap data komentar yang telah melalui tahap *stopword removal*. Proses ini menggunakan

pustaka Sastrawi untuk mengubah setiap kata dalam teks menjadi bentuk dasar atau kata dasarnya.

3.2.3 Labeling Data

Pada tahap ini, proses pelabelan dilakukan secara langsung oleh manusia (manual) tanpa bantuan sistem otomatis, dengan membaca dan memahami setiap komentar TikTok yang telah melalui tahap preprocessing. Jumlah data komentar yang diberi label dalam penelitian ini adalah 1.200 komentar. Pelabelan dilakukan berdasarkan makna atau sentimen yang terkandung dalam setiap komentar. Adapun kategori label yang digunakan dalam penelitian ini meliputi:

- Positif: Komentar yang menunjukkan dukungan, atau pengalaman menyenangkan terkait penggunaan ChatGPT dalam skripsi.
- Negatif: Komentar yang mengandung keluhan, ketidakpuasan, kritik, atau pandangan pesimis terhadap penggunaan ChatGPT dalam skripsi.
- Netral: Komentar yang bersifat informative, tidak berpihak, atau tidak menunjukkan ekspresi emosi secara jelas.

Tujuan dari proses pelabelan ini adalah untuk menghasilkan data yang siap digunakan dalam proses pelatihan model klasifikasi sentimen. Dengan data yang telah diberi label, model dapat mempelajari serta mengenali pola-pola sentimen secara lebih efektif. Contoh data komentar yang sudah dilakukan pelabelan manual ditunjukkan pada Tabel 3.4.

Tabel 3.4 Pelabelan Manual

No	Komentar	Kategori
1	pokok chatgpt baik ngebantu banget	positif
2	kutip referensi kurang sesuai sesuai	negatif
3	banyak dosen sekarang erti tulis bot sama tulis mahasiswa	netral
4	dosen itb ngasih erti kalau mau cari ide aja pakai chatgpt cek kali sumber akurat	netral
5	kan kena plagiarisme woy	negatif
6	tetap bijak guna chatgpt	positif

7	sedang buat proposal tesis rasa manfaat chat gpt cabang ilmu dokter biomedik	netral
8	kadang jurnal sesuai	negatif
9	gpt enak guna asal erti kalimat perintah benar	positif
10	paling revisi	netral

Tabel 3.4 menunjukan hasil pelabelan manual, setiap komentar dianalisis berdasarkan konteks dan emosi yang tersirat, kemudian dikategorikan ke dalam salah satu dari tiga label sentimen, yaitu positif, negatif, atau netral.

3.2.4 Klasifikasi Naïve Bayes

Setelah data komentar TikTok melalui tahap preprocessing dan pelabelan, langkah selanjutnya adalah melakukan klasifikasi sentimen menggunakan algortima Naïve Bayes. Berikut kode untuk melakukan klasifikasi naïve bayes.

```
df = pd.read_excel("labeling.xlsx")
df_cleaned = df.dropna(subset=["Kategori"])
texts = df_cleaned["Stemmed"].astype(str).tolist()
labels = df_cleaned["Kategori"].astype(str).tolist()
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)
X_train, X_test, y_train, y_test = train_test_split(X,
labels, test_size=0.2, random_state=42)
model = MultinomialNB()
model.fit(X_train, y_train)
```

Kode tersebut merupakan implementasi algoritma Naïve Bayes, yang digunakan setelah prapemrosesan dan pelabelan manual untuk mengklasifikasikan sentimen dari komentar pengguna TikTok. Data teks yang telah dikonversi menjadi reprensentasi numeric menggunakan teknik TF-IDF melalui TfidfVectorizer. Data tersebut kemudian dipecah menggunakan *train-test-split*, dengan 80% data digunakan untuk pelatihan model (data pelatihan) dan 20% sisanya untuk data uji (data uji)

Total data latih yang digunakan sebanyak 960 komentar, dengan rincian 436 data berlabel positif, 272 data negatif, dan 252 data netral. Sementara itu, jumlah

data uji sebanyak 240 komentar, yang terdiri dari 123 data positif, 64 data negatif, dan 53 data netral.

3.2.5 Evaluasi Model

Evaluasi terhadap model dilakukan guna menilai kinerja algoritma Naïve Bayes dalam mengelompokkan sentimen komentar pengguna TikTok terkait penggunaan ChatGPT dalam penulisan skripsi. Adapun berikut ini merupakan potongan kode yang digunakan untuk melakukan proses evaluasi tersebut.

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
print("Akurasi model:", accuracy)
print("Laporan Klasifikasi:\n", report)
```

Kode ini mengevaluasi performa model yang telah dilatih dengan memprediksi data uji (X_test). Hasil prediksi kemudian dibandingkan dengan label actual (Y_test) untuk menghitung nilai akurasi menggunakan *accuracy_score*. Selanjutnya, fungsi classification_report digunakan untuk menghasilkan laporan evaluasi yang mencakup metrik-metrik penting seperti *precision*, *recall*, dan *f1-score* pada masing-masing kelas sentimen: positif, negatif, dan netral.