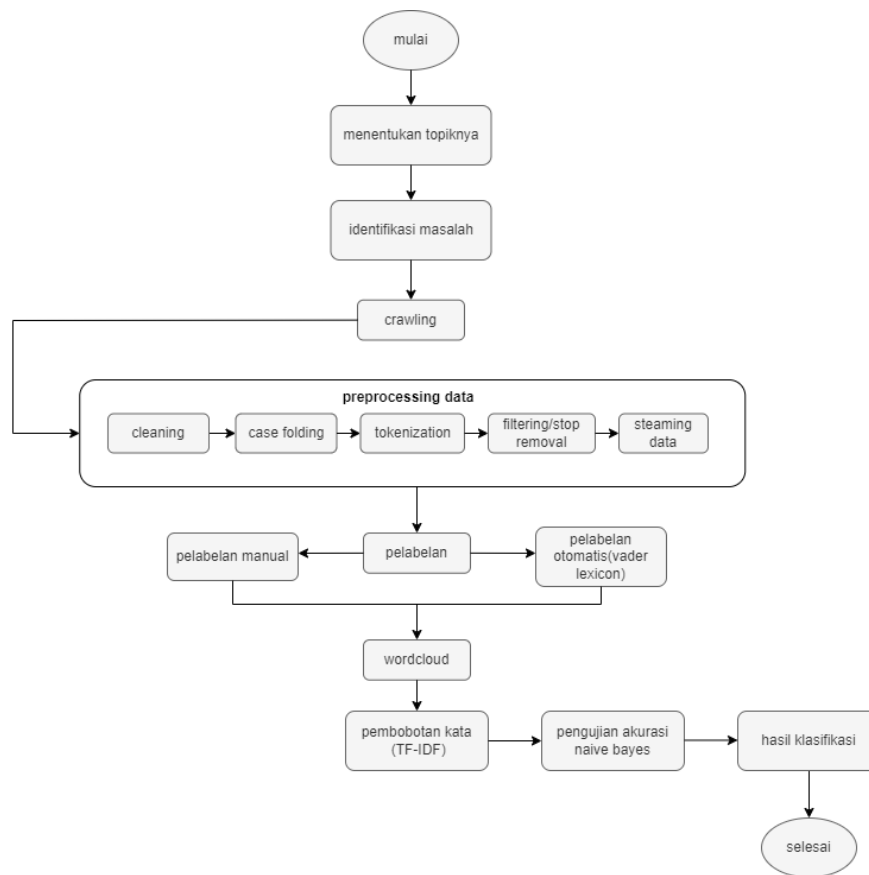


## BAB III METODE PENELITIAN

### 3.1 Alur Penelitian

Alur penelitian adalah konsep atau gambar dari penelitian yang akan dilakukan. Penjabaran alur penelitian tersebut dapat dilihat pada gambar di bawah ini :



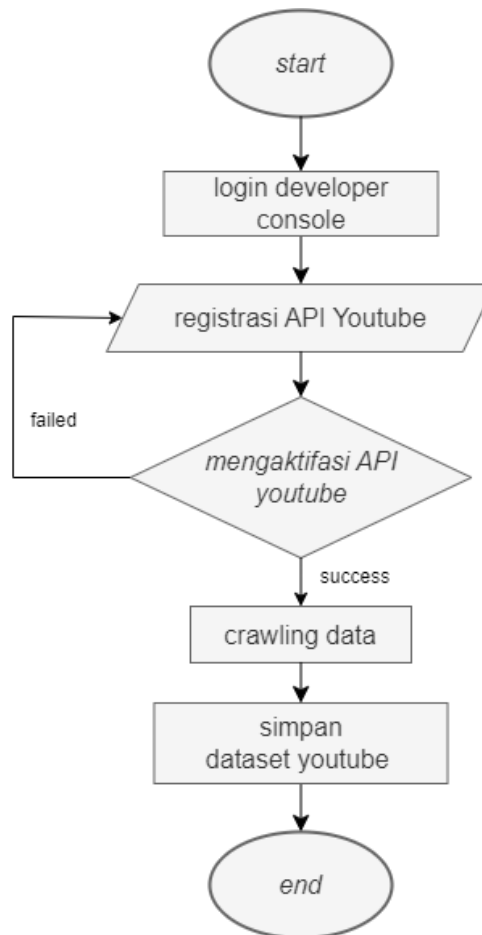
**Gambar 3. 1 Alur Penelitian**

Tahap penelitian ini melibatkan beberapa langkah utama :

- a. Mulai
- b. Tahap pertama adalah menentukan sebuah topik penelitian yang akan di bahas.

- c. Tahap kedua identifikasi masalah dilakukan dengan mengamati komentar youtube pada *food vlogger* Mgdalenaf untuk memahami persepsi pengguna terhadap *akun* Youtube.
- d. Tahap ketiga pengumpulan data, di mana data komentar dari *platform* Youtube di *crawling* menggunakan API Google *Developer Console* dengan akun Google.
- e. Tahap keempat adalah *preprocessing text*, di mana data mentah yang diambil dari *platform* Youtube diubah menjadi data yang terstruktur untuk memudahkan analisis.
- f. Tahap kelima adalah pelabelan sentiment merupakan proses menandai atau memberikan label pada data dengan kategori atau class seperti “*cyberbullying*” dan “*non\_cyberbullying*”.
- g. Tahap keenam yaitu Wordcloud adalah kata-kata yang digambarkan kedalam suatu visualisasi dimana besar kecilnya huruf bergantung pada frekuensi munculnya kata tersebut didalam teks. singkatnya adalah kata yang paling sering muncul dalam suatu data.
- h. Tahap ketujuh adalah pembobotan kata untuk menghitung bobot dari setiap kata yang ada pada dokumen.
- i. Selanjutnya tahap kedelapan adalah pengujian akurasi *naïve bayes*, di mana data dari *platform* YouTube yang telah dibersihkan diubah menjadi nilai numerik dan kemudian dianalisis dengan metode *Naïve Bayes* pada Google Colab *Phyton* untuk mengklasifikasikannya.
- j. Hasil klasifikasi akan ditampilkan menggunakan diagram garis untuk memvisualisasikan perbandingan performa model dalam mengidentifikasi sentimen *cyberbullying* dan *non-cyberbullying*.
- k. Selesai

### 3.2 Teknik Pengumpulan Data



Gambar 3. 2 Alur Teknik Pengumpulan Data

Langkah awal dalam proses pengumpulan data adalah login menggunakan akun Google dan melakukan registrasi pada Google Developer Console. Setelah berhasil registrasi, tahap selanjutnya adalah mengaktifasi API Youtube pada Google *Developer Console* dengan akun Google. Youtube Data API v3 ini diperlukan agar kita dapat mengakses data dari *platform* Youtube.

Setelah mendapatkan API Youtube, langkah berikutnya adalah membuat sebuah *script* atau program komputer khusus. *Script* ini bertindak sebagai media atau alat yang akan digunakan untuk melakukan *crawling* atau

pengambilan data dari *platform* Youtube dengan perangkat lunak google colab *research python*. Berikut ini adalah *link* Youtube dari sumber data yang diperoleh : <https://www.youtube.com/shorts/C8NvLMfJagE>.

Selanjutnya kode ini akan menggunakan API yang telah diperoleh untuk mengakses *platform* Youtube dan mengambil data yang diperlukan. *Script* ini akan menjelajahi *platform* Youtube, melakukan pencarian, dan mengambil data komentar yang sesuai dengan parameter yang telah ditentukan. Berikut adalah kode yang digunakan untuk melakukan proses *crawling* :

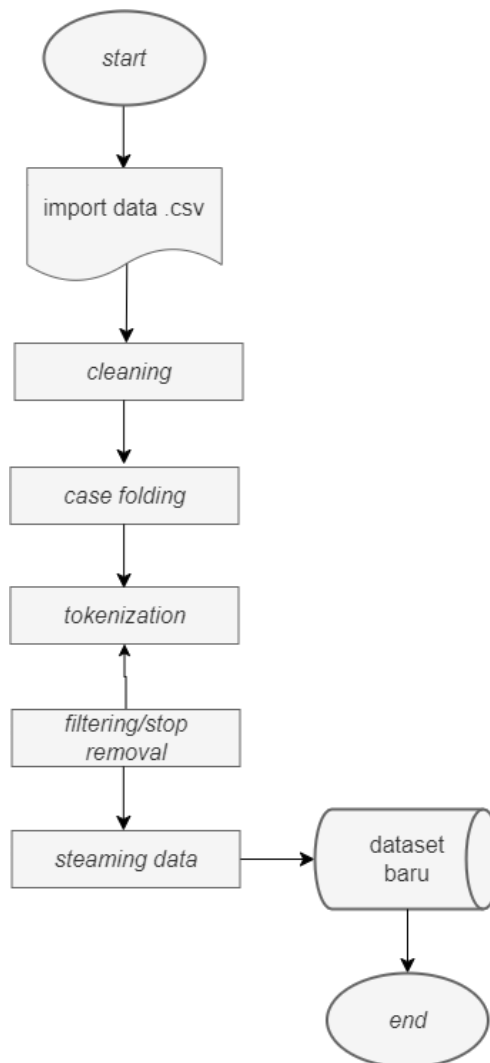
CODE
<pre>def video_comments(video_id):     replies = []     # membuat objek YouTube API     youtube = build('youtube', 'v3', developerKey=api_key)     # Mengambil komentar dan balasan     video_response = youtube.commentThreads().list(part='snippet,replies', videoId=video_id).execute()     while video_response:</pre>

Dengan menggunakan kode tersebut, dapat mengakses komentar dan balasan dari video YouTube tertentu dan mendapatkan informasi seperti waktu publikasi, penulis komentar, teks komentar, jumlah suka, dan balasan atas komentar tersebut, yang nantinya digunakan untuk analisis atau keperluan lainnya. Data yang diambil akan diteruskan ke langkah selanjutnya dalam proses penelitian, seperti *preprocessing* dan analisis data.

### 3.3 Preprocessing Text

Dalam penelitian ini analisis teks atau *text preprocessing*, *flowchart* dapat digunakan untuk menggambarkan alur langkah-langkah dalam mempersiapkan

data teks sebelum analisis lebih lanjut. Penjabaran tahapan *preprocessing text* dapat dilihat pada gambar di bawah ini :



**Gambar 3. 3 Text Preprocessing**

Data yang telah didapatkan dari hasil *crawling* pada google colab *research python*, selanjutnya disimpan dalam bentuk folder *.csv* dan dilakukan pembersihan atau *preprocessing*. *Text preprocessing* adalah langkah pertama

dalam proses *text mining*. Tujuannya adalah membersihkan data dari (*noise*) sehingga data menjadi lebih ringkas dan tersusun dengan baik (Ismail, Rahmah, and Sulistiyowati 2022).

Tahap *preprocessing* merupakan langkah kunci yang penting dalam persiapan untuk tahap berikutnya, yaitu menghapus atribut yang tidak berguna selama proses klasifikasi. Data yang dimasukkan ke dalam tahap ini masih dalam bentuk mentah dan belum melalui proses *cleansing*, sehingga ketika tahap *preprocessing* selesai, ini akan memudahkan tahap klasifikasi berikutnya. Sebelum teks dianalisis, berikut penjelasan langkah-langkah dalam tahap *preprocessing* data dilakukan (Sitepu, Wanayumini, and Situmorang 2021) :

Peneliti melakukan serangkaian langkah *preprocessing text* pada kolom "textDisplay" menggunakan beberapa tahapan proses yaitu :

### 1. Cleaning

Cleaning adalah proses untuk menghilangkan tanda baca, angka, simbol, URL, dan username dari teks. Penghapusan stopwords akan menghilangkan kata-kata yang dianggap tidak penting dalam teks (Khairunnisa, Adiwijaya, and Faraby 2021).

Berikut adalah kode yang digunakan untuk melakukan proses *Cleaning*:

<b>CODE</b>
<pre>import re import string import nltk  def remove_URL(youtube): url = re.compile(r'https?://\S+ www\.\S+') return url.sub(r'', youtube)</pre>

```

def remove_html(youtube):
    html = re.compile(r'<.*?>')
    return html.sub(r'', youtube)

def remove_emoji(youtube):
    emoji_pattern = re.compile("[
    u"\U0001F600-\U0001F64F"
    u"\U0001F300-\U0001F5FF"
    u"\U0001F680-\U0001F6FF"
    u"\U0001F1E0-\U0001F1FF"
    "]" + "", flags=re.UNICODE)
    return emoji_pattern.sub(r'', youtube)

def remove_numbers(youtube):
    youtube = re.sub(r'\d+', '', youtube)
    return youtube

def remove_symbols(youtube):
    youtube = re.sub(r'^[a-zA-Z0-9\s]', '', youtube) #menghapus semua
    simbol
    return youtube

```

Berdasarkan kode di atas, diketahui bahwa :

- a. **remove\_URL(youtube)**: Menghapus URL atau tautan web dari teks. Regex **r'https?://\S+|www\.\S+'** digunakan untuk mencocokkan dan menghapus tautan web dari teks input.
- b. **remove\_html(youtube)**: Menghapus tag HTML dari teks. Regex **<.\*?>** akan mencocokkan dan menghapus segala sesuatu yang ada di dalam tag HTML.
- c. **remove\_emoji(youtube)**: Menghapus emoji dari teks. Emoji adalah karakter non-standar yang sering digunakan dalam pesan teks.

- d. **remove\_numbers(youtube)**: Menghapus angka dari teks. Fungsi **re.sub(r'\d+', '', youtube)** akan menghapus segala digit atau angka.
- e. **remove\_symbols(youtube)**: Menghapus simbol atau karakter khusus dari teks. Regex **r'^a-zA-Z0-9\s'** akan mencocokkan dan menghapus segala karakter kecuali huruf (baik huruf besar atau kecil), angka, dan spasi.

## 2. Case Folding

*Case folding* adalah langkah yang digunakan untuk mengubah semua huruf dalam teks menjadi satu format tertentu, biasanya menjadi huruf kecil (*lowercase*) (Albab, Karuniawati, and Fawaiq 2023). Karakter selain huruf dihilangkan dan dianggap sebagai pembatas.

Berikut adalah kode yang digunakan untuk melakukan proses *Case Folding*:

<b>CODE</b>
<pre>def case_folding(text):     if isinstance(text, str):         lowercase_text = text.lower()         return lowercase_text     else:         return text</pre>

Potongan kode di atas adalah fungsi yang bertujuan untuk mengubah semua karakter dalam teks menjadi huruf kecil atau *lowercase*. Fungsi ini menggunakan metode '**lower()**'

## 3. Tokenization

Ini adalah proses membagi teks menjadi unit-unit yang lebih kecil, yang biasanya disebut sebagai "token". Token ini bisa berupa kata-kata individu atau bagian-bagian yang lebih kecil dari teks, seperti frasa atau



karakter. Tujuan utama tokenisasi adalah untuk memecah teks menjadi bagian-bagian yang lebih mudah dikelola dan dianalisis (Jurnal et al. 2022).

Berikut adalah kode yang digunakan untuk melakukan proses *Tokenization*:

<b>CODE</b>
<pre>def tokenize(text):     tokens = text.split()     return tokens</pre>

Fungsi kode diatas '**text.split( )**', merupakan metode untuk membagi string **text** menjadi daftar kata atau token. Metode '**split( )**' memisahkan string berdasarkan spasi atau karakter

#### **4. Filtering atau Stop Removal**

Tahapan ini ialah proses memilih kata yang penting dari hasil token, yaitu penghapusan kata-kata umum yang tidak memberikan kontribusi penting dalam analisis teks (Rifaldi, Abdul Fadlil, and Herman 2023). Berikut adalah kode yang digunakan untuk melakukan proses *Filtering* atau *Stop Removal*:

<b>CODE</b>
<pre>#instal library from nltk.corpus import stopwords nltk.download('stopwords') stop_words = stopwords.words('indonesian') def remove_stopwords(text):     return [word for word in text if word not in stop_words]</pre>

Kode `remove_stopwords( )` adalah fungsi yang didefinisikan untuk menghapus stopwords dari teks yang diterima sebagai parameter `text`. Fungsi ini memeriksa setiap kata dalam `text` dan memasukkannya ke dalam list hanya jika kata tersebut tidak termasuk dalam daftar stopwords yang telah diunduh sebelumnya dari NLTK untuk bahasa Indonesia.

## 5. Steaming Data

Tahapan stemming pada *preprocessing text* bertujuan untuk mengubah kata-kata menjadi bentuk dasarnya dengan cara menghapus imbuhan seperti awalan, akhiran, atau sisipan pada kata-kata (Ulgasesa, Negara, and Tursina 2022). Hal ini dilakukan untuk memastikan bahwa hanya kata dasar yang disimpan dalam indeks atau basis data dokumen. Proses pada penelitian ini menggunakan *library* Sastrawi (untuk bahasa Indonesia) yang menyediakan fungsi-fungsi untuk melakukan stemming pada teks.

Berikut adalah kode yang digunakan untuk melakukan proses *Steaming Data*:

<b>CODE</b>
<pre>!pip install Sastrawi  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory from nltk.stem import PorterStemmer from nltk.stem.snowball import SnowballStemmer  factory = StemmerFactory() stemmer = factory.create_stemmer() def stem_text(text):     return[stemmer.stem(word) for word in text]</pre>

Fungsi `stem_text(text)` adalah untuk melakukan proses stemming pada setiap kata dalam teks yang diberikan.

### 3.4 Klasifikasi Naïve Bayes

Tahapan berikutnya dari penelitian ini adalah melakukan klasifikasi data komentar YouTube yang telah melewati tahap pra-pemrosesan teks, *Labeling* dan *term weighting* atau pembobotan kata TF-IDF akan digunakan sebagai input dalam proses pelatihan dan pengujian (test) model klasifikasi. Metode yang digunakan untuk klasifikasi ini adalah *Naïve Bayes*. Penelitian ini bertujuan untuk mengidentifikasi komentar yang mengandung unsur *cyberbullying* atau *non-cyberbullying*.

Dalam penerapannya menggunakan *Naïve Bayes*, terdapat dua proses utama: data latih dan data uji. Langkah pertama adalah melatih sistem menggunakan data latih. Selanjutnya, proses uji dilakukan untuk mengevaluasi performa model dengan mengacu pada probabilitas yang didapatkan dari data latih.

Klasifikasi Naïve Bayes memanfaatkan pustaka (*library*) dari NLTK (*Natural Language Toolkit*), sebuah *library Python* yang digunakan dalam proses *machine learning* teks. Dengan menggunakan NLTK, proses pelatihan dan pengujian *Naïve Bayes* pada data komentar YouTube dapat dilakukan untuk mengklasifikasikan komentar sebagai *cyberbullying* atau *non-cyberbullying*.

### 3.5 Analisis Kebutuhan Perangkat

#### 3.5.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan dalam pembuatan program adalah sebagai berikut :

- a. ACER Laptop-38BRP51;
- b. RAM 4GB;

- c. Processor Intel® Celeron® Core™ i3-6006U CPU @ 2.00GHz (4CPUs), 2.0GHz;
- d. SSD 250GB;

### 3.5.2 Perangkat Lunak (Software)

Perangkat lunak yang digunakan dalam pembuatan program adalah sebagai berikut :

- a. Sistem Operasi (64bit);
- b. Platform Youtube berfungsi sebagai sumber data untuk mengumpulkan komentar yang terkait *cyberbullying* ;
- c. Google Colab *Research Python* digunakan sebagai lingkungan pengembangan (IDE) untuk melakukan analisis, pemrosesan, dan pemodelan data.
- d. Excel atau spreadsheet memungkinkan kami menggabungkan dan menganalisis data dari berbagai sumber.