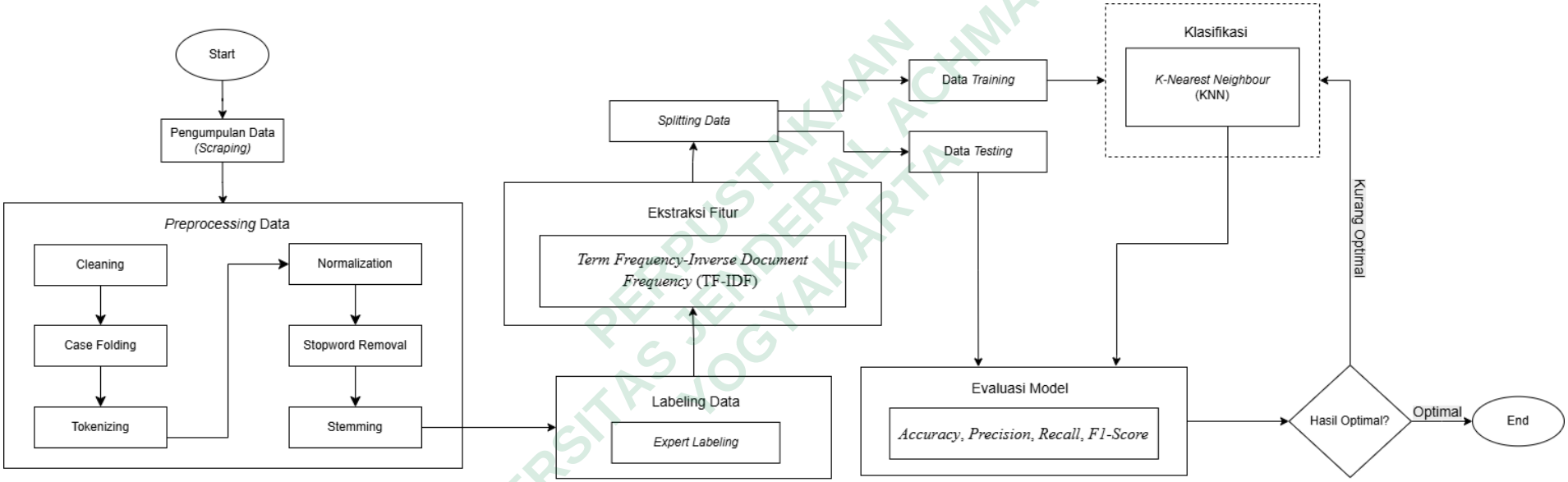
BAB 3
METODE PENELITIAN

Penelitian ini memanfaatkan algoritma *K-Nearest Neighbor* (KNN) sebagai algoritma klasifikasi untuk mendeteksi *cyberbullying*. *K-Nearest Neighbor* (KNN) dipilih karena kemampuannya dalam mengelompokkan data berdasarkan karakteristik antara satu komentar dengan komentar lainnya, yang diukur dari seberapa mirip kata-kata atau pola teks di dalamnya, sehingga cocok digunakan dalam tugas klasifikasi teks seperti komentar media sosial.

Tahapan proses penelitian dimulai dari pengumpulan data *primer* melalui *scraping* komentar dari platform TikTok. Data yang diperoleh lalu diproses melalui tahapan *pre-processing* data. Setelah itu, setiap komentar diberi label (*bullying* atau non-*bullying*) dan direpresentasikan dalam bentuk numerik dengan penerapan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Dataset yang telah diproses akan dibagi menjadi data latih (*training*) dan data uji (*testing*), yang selanjutnya digunakan dalam proses klasifikasi menggunakan algoritma *K-Nearest Neighbor* (KNN). Hasil evaluasi model yang mencakup metrik *accuracy*, *precision*, *recall*, dan *f1-Score* diperoleh dari analisis *confusion matrix*. Untuk memudahkan pemahaman, hasil penelitian disajikan dalam bentuk tabel dan visualisasi. Alur lengkap proses penelitian ditampilkan pada Gambar 3.1 berikut.



Gambar 3. 1 Alur Penelitian

Penjelasan mengenai tahap-tahap yang dilakukan berdasarkan gambar 3.1 adalah sebagai berikut:

1. **Pengumpulan Data (*Scraping*)**

   Data dikumpulkan melalui komentar pengguna di platform TikTok dengan fokus pada interaksi yang berpotensi mengandung *cyberbullying*. Komentar ini merepresentasikan ekspresi pengguna, termasuk ujaran negatif (Junifer Pangaribuan & Putra Barus, 2023).

2. ***Pre-processing* Data**

   Karena data yang digunakan berupa teks, diperlukan serangkaian proses *pre-processing* untuk memastikan data dalam kondisi bersih dan siap diolah. Beberapa langkah utama dalam proses ini meliputi (Salim & Sutabri, 2023).

   a. *Cleaning*: Menghapus elemen yang tidak diperlukan dalam teks, seperti tanda baca, angka yang tidak relevan, simbol khusus, serta spasi ganda.

   b. *Case Folding*: Mengubah seluruh karakter dalam teks menjadi huruf kecil (*lowercase*), agar variasi penulisan tetap dikenali sebagai satu kat untuk menyeragamkan format teks dengan mengonversi seluruh karakter menjadi huruf kecil, guna memastikan konsistensi pengenalan kata.

   c. *Tokenizing*: Memecah teks menjadi kata-kata agar lebih mudah dianalisis. Dalam konteks penelitian *cyberbullying*, tahap ini membantu algoritma mengenali pola kata yang sering muncul dalam komentar negatif.

   d. *Normalization:* Normalisasi teks berfungsi untuk menyeragamkan kata-kata tidak baku, singkatan, bahasa gaul, atau kesalahan pengetikan (*typo*) yang sering muncul di komentar TikTok ke dalam bentuk standar.

   e. *Stopword Removal*: Menghapus kata-kata yang bersifat umum dan tidak memberikan kontribusi signifikan terhadap makna, seperti

'dan', 'di', 'ke', serta 'yang', agar fokus analisis tetap pada kata-kata penting.

f. *Stemming*: Mengonversi kata ke dalam bentuk dasar untuk menghindari duplikasi kata dengan arti yang sama. Pada penelitian ini, proses *stemming* menggunakan library *sastrawi* untuk mengonversi kata-kata ke bentuk dasar, guna memastikan konsistensi teks dalam proses klasifikasi.

3. Pelabelan Data

Proses pelabelan data dilakukan terhadap komentar-komentar yang diperoleh dari platform TikTok. Label yang digunakan yaitu:

a. 1 → jika komentar tersebut mengandung unsur *cyberbullying* (misalnya mengandung hinaan, ejekan, kata kasar, yang merendahkan orang lain).
b. 0 → jika komentar tersebut tidak mengandung unsur *cyberbullying* (komentar normal atau netral tanpa unsur negatif).

Tahapan pelabelan ini bertujuan untuk validitas data yang akan digunakan dalam pelatihan model klasifikasi, sehingga hasil analisis optimal (Salim & Sutabri, 2023).

4. Pembobotan Kata (TF-IDF)

Proses pembobotan kata dilakukan dengan mengubah teks menjadi representasi numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF), yang berfungsi untuk mengukur frekuensi kemunculan suatu kata dalam satu dokumen dibandingkan dengan keseluruhan jumlah dokumen (Wijaya & Suwandhi, 2024).

Formula perhitungannya dapat dilihat pada persamaan berikut.

1. $$TF\ (t,d)\ =\ \frac{Jumlah\ kemunculan\ term\ t\ dalam\ dokumen\ d}{Total\ Jumlah\ kata\ dalam\ dokumen\ d}$$ (1)

Dimana:

a. $TF\ (t,d)$: Nilai Term Frequency untuk term *t* dalam dokumen *d*.

2. $IDF\ (t)\ = log\ \frac{N}{df\ (t)}$ (2)

Dimana:

a. $IDF\ (t)$: Inverse Document Frequency untuk term *t*.
b. N: Total jumlah dokumen.
c. $df\ (t)$: jumlah dokumen yang mengandung term.

3. $TF\ -\ IDF\ (t,d) = TF\ (t,d) \times IDF(t)$ (3)

Dimana:

a. $TF\ -\ IDF\ (t,d)$: Bobot TF-IDF untuk term *t* pada dokumen *d*.
b. $TF\ (t,d)$: kemunculan term *t* terhadap total jumlah kata dalam dokumen *d*.
c. $IDF(t)$: Total jumlah dokumen terhadap jumlah dokumen yang mengandung term *t*.

5. *Split* Data

Setelah tahap *pre-processing*, pelabelan, dan pembobotan menggunakan TF-IDF selesai dilakukan, dataset selanjutnya dibagi menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*).

a. Data latih bagian dari data yang digunakan untuk melatih algoritma *K-Nearest Neighbor* (KNN) agar bisa belajar mengenali pola komentar yang mengandung atau tidak mengandung *cyberbullying*.
b. Data Uji bagian dari data yang digunakan untuk menilai seberapa baik model *K-Nearest Neighbor* (KNN) yang telah dilatih bekerja pada data yang belum pernah diproses sebelumnya.

6. Klasifikasi *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* (KNN) merupakan metode klasifikasi yang menentukan label suatu data baru berdasarkan kemiripannya dengan sejumlah data terdekat yang telah diketahui kategorinya. Tahap awal dalam proses ini melibatkan perhitungan jarak antara data yang ingin diklasifikasikan dengan seluruh data pelatihan.

Setelah itu, dipilih k data dengan jarak terdekat. Penentuan label dilakukan dengan melihat label yang paling dominan di antara k tetangga tersebut. Untuk mengukur kedekatan antar data, algoritma ini umumnya menggunakan jarak *Euclidean*, yang dihitung menggunakan rumus berikut.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (4)$$

Dimana:

a. $d(x, y)$; Jarak antara dua data x, y.
b. $x_i\ y_i$: Nilai fitur ke-i dari masing-masing data.
c. n: Jumlah total fitur pada setiap data.

Dalam konteks penelitian ini, setiap data komentar TikTok telah direpresentasikan dalam bentuk vektor menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF). Oleh karena itu, nilai x_i dan y_i pada formula di atas merupakan bobot dari masing-masing *term* dalam ruang fitur hasil transformasi TF-IDF.

Untuk memperoleh nilai k yang paling optimal dalam proses klasifikasi dengan algoritma *K-Nearest Neighbor* (KNN), penelitian ini menggunakan metode *k-fold cross-validation*. Metode ini membagi dataset menjadi beberapa subset (*fold*), di mana pada setiap putaran, sebagian data digunakan sebagai data latih, sementara sisanya digunakan sebagai data uji, dilakukan secara bergantian hingga seluruh data terlibat dalam proses pelatihan dan pengujian. Proses ini diulang untuk berbagai nilai k, dan hasil akurasi rata-rata dari setiap nilai tersebut digunakan sebagai dasar untuk memilih nilai k terbaik.

7. Evaluasi Model (*Confusion Matrix*)

Evaluasi model dilakukan menggunakan matriks *Confusion Matrix*. Berikut terdapat tabel dari c*onfusion matrix* dapat dilihat pada tabel 3.1.

Tabel 3. 1 *Confusion Matrix*

	Predicted Class	
Actual Class	**Positive**	**Negative**
Positive	*(TP) True Positive*	*(FN) False Negative*
Negative	*(FP) False Positive*	*(TN) True Negative*

Berdasarkan tabel 3.1, maka:

1. True Positive terjadi ketika data yang seharusnya termasuk dalam kategori positif berhasil diklasifikasikan dengan benar sebagai positif.
2. False Positive terjadi ketika data yang sebenarnya negatif, tetapi salah diklasifikasikan sebagai positif.
3. False Negative terjadi ketika data yang seharusnya positif, tetapi salah diklasifikasikan sebagai negatif.
4. True Negative terjadi ketika data yang memang negatif dan diklasifikasikan dengan benar sebagai negatif.

Berikut cara menghitung beberapa metrik evaluasi model:

1. *Accuracy*

   Mengukur tingkat ketepatan model dengan menghitung rasio antara jumlah prediksi benar dan total jumlah data.

   Menghitung *Accuracy* dapat dilihat pada persamaan berikut.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN} \quad (5)$$

2. *Precision*

   Mengukur seberapa akurat model dalam memprediksi kelas positif, dengan melihat rasio data yang benar-benar positif dibandingkan semua data yang diprediksi positif.

   Menghitung *precision* dapat dilihat pada persamaan berikut.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

3. *Recall*

   Menilai sejauh mana model mampu mendeteksi seluruh data yang termasuk dalam kategori positif, termasuk yang sulit terdeteksi.

   Menghitung *recall* dapat dilihat pada persamaan berikut.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

4. *F1-Score*

   Menunjukkan keseimbangan *presisi* dan *recall*, sehingga memberikan evaluasi model yang lebih stabil dalam kasus data tidak seimbang.

   Menghitung *f-1 score* dapat dilihat pada persamaan berikut.

$$F1\text{-}Score = \frac{2\ x\ PrecisionxRecall}{Precision+Recall} \quad (8)$$

3.1 Bahan dan Alat Penelitian

Penelitian ini memanfaatkan data *primer* yang akan dikumpulkan melalui proses *scraping* komentar dari platform TikTok, terutama komentar yang mengindikasikan adanya *cyberbullying*. Dataset ini dikumpulkan secara mandiri untuk keperluan analisis dan klasifikasi, guna mengidentifikasi apakah suatu komentar termasuk kategori *bullying* atau non-*bullying*. Data yang sudah terkumpul akan digunakan sebagai bahan pelatihan dan pengujian dalam pengembangan model klasifikasi pada penelitian ini.

Alat yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi yang mendukung analisis dan pemrosesan data penelitian. Laptop ini dilengkapi

dengan sistem operasi dan software yang sesuai untuk kebutuhan penelitian serta memiliki konektivitas internet. Penelitian ini menggunakan perangkat dengan spesifikasi sebagai berikut.

1. Sistem Operasi: Windows 11.
2. Prosesor: AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz.
3. Memori RAM: 8 GB.
4. Penyimpanan Internal: 120 GB.
5. Tipe Sistem: 64-bit operating system.
6. *Software*: *Tools* Apify dan Google Colab.

3.2 Jalan Penelitian

Penelitian ini menerapkan algoritma *K-Nearest Neighbor* (KNN) sebagai metode klasifikasi untuk mendeteksi komentar yang mengandung unsur *cyberbullying*. Tahapan penelitian dimulai dari pengumpulan data melalui *scraping*, dilanjutkan dengan *pre-processing* data. Setelah itu, data akan diberi label dan dilakukan proses pembobotan menggunakan *Term Frequency-Inverse Document Frequency*. Data kemudian dibagi menjadi dua, data latih dan data uji sebelum diklasifikasikan menggunakan *K-Nearest Neighbor* (KNN), hingga evaluasi model dilakukan dengan *confusion matrix*, dan hasil akhir akan disajikan dalam bentuk tabel dan grafik. Tahapan tersebut dilakukan secara berurutan untuk mendukung proses klasifikasi.

1. Pengumpulan dan Persiapan Data

Dalam tahap ini data dikumpulkan melalui proses *scraping* dari platform TikTok, dengan fokus pada komentar yang berpotensi mengandung unsur *cyberbullying*. Setelah itu dilakukan *pre-processing* data yang mencakup: *cleaning*, *case folding*, *tokenizing*, *normalisasi*, *stopword removal*, dan *stemming*.

2. Pelabelan dan Pembobotan Data

Setelah komentar-komentar berhasil dikumpulkan dan dibersihkan melalui proses *pre-processing*, setiap komentar perlu diberi label untuk melatih model klasifikasi. Data yang telah diproses kemudian diberi label 1 jika komentar mengandung *cyberbullying* dan label 0 jika tidak mengandung *cyberbullying*. Kemudian dilakukan pembobotan *Term Frequency-Inverse Document Frequency* untuk mengubah teks menjadi vektor numerik yang merepresentasikan pentingnya setiap kata dalam dokumen.

3. Pembagian Data dan Klasifikasi *K-Nearest Neighbor*

Setelah proses *pre-processing*, pelabelan, dan pembobotan TF-IDF selesai, dataset dibagi menjadi data latih (*training*) dan data uji (*testing*). Setalah itu, dilakukan klasifikasi dengan algoritma *K-Nearest Neighbor* (KNN), yang yang mengidentifikasi k tetangga terdekat untuk menentukan kelas suatu data. Pemilihan nilai k yang optimal akan dilakukan menggunakan teknik *k-fold cross-validation*.

4. Evaluasi Model (*Confusion Matrix*)

Pada tahap evaluasi, kinerja model diukur menggunakan *confusion matrix* seperti, *accuracy*, *precision*, *recall*, *f1-Score*. Hasil akhir akan ditampilkan berupa tabel dan visualisasi guna mempermudah analisis dan pemahaman.