

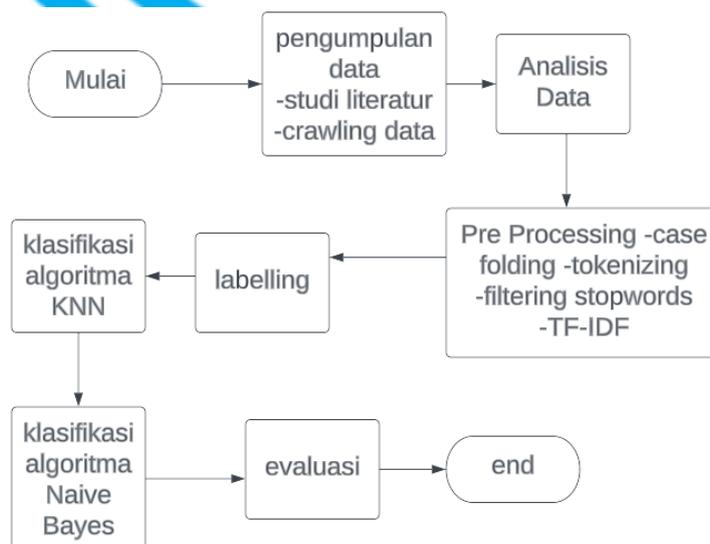
BAB III METODE PENELITIAN

3.1 Objek Penelitian

Pada penelitian ini objek yang digunakan ialah opini masyarakat pada sosial media Twitter mengenai *Bullying*. Data diambil dengan menggunakan proses crawling pada python yaitu sebanyak 1.075 data. Data dibagi menjadi 2 kelas, yaitu kelas positif dan negatif.

3.2 Prosedur Penelitian

Prosedur penelitian yang digambarkan dengan menggunakan bentuk *Flowchart* dalam Gambar 3.1 Penelitian ini dimulai dari Analisis data lalu pengumpulan data yaitu pengumpulan studi literatur untuk mencari referensi dan *crawling data* menggunakan API Twitter di *Google colab* menggunakan Bahasa *python*. Kemudian data yang sudah terkumpul akan di proses pada bagian *Pre-processing* yang terdiri dari proses *Case Folding*, *Tokenizing*, *Filtering stopwords* dan *TF-IDF* untuk membersihkan data. Data yang sudah melewati tahap *pre-processing* selanjutnya akan diberi label dan dikategorikan dalam kelas yaitu kelas positif dan negatif, data yang sudah diklasifikasi akan diimplementasikan dengan algoritma *Naive Bayes* dan *K-Nearest Neighbor*, kemudian dilanjutkan ke tahap evaluasi yaitu pengujian model klasifikasi dengan menggunakan *Confusion Matrix*.



3.3 Pengumpulan

Gambar 3.1 Prosedur Penelitian

Pengumpulan data pada penelitian ini dibagi menjadi dua bagian, diantaranya:

3.3.1 Studi Literatur

Studi literatur bertujuan untuk mencari dasar pemikiran dan memperoleh referensi dari berbagai jurnal, buku, dan internet dengan mengumpulkan teori-teori dasar yang berhubungan dengan penelitian.

3.3.2 *Crawling Data*

Proses mengumpulkan data dengan mengambil data tweet yang dikumpulkan menggunakan python, pencarian dengan kata kunci “Kasus Audrey”. Data diambil dengan total 1.075 *tweet*, sejak awal mula pemberitaan tentang kasus tersebut yaitu pada 10 April 2019 hingga 19 April 2022. Data tersebut kemudian akan di klasifikasikan dengan Algoritma *K-nearest neighbor* dan *naïve bayes*. Atribut yang didapat dari hasil *crawling* data ialah *date time*, *tweet id*, *text*, *username*, label.

Untuk *crawling* data menggunakan skrip:

```
for i,tweet in enumerate(sntwitter.TwitterSearchScrapper('kasus
audrey since:2019-04-10 until:2022-04-19').get_items()):
if i>2000:
break
tweets_list.append([tweet.date, tweet.id, tweet.content, tweet.username,
analyze_sentiment(tweet.content)])
```

3.4 Analisis Data

Analisis data bertujuan untuk mengetahui permasalahan yang terdapat pada media sosial twitter yaitu opini masyarakat tentang kasus *bullying*. Analisis data dilakukan untuk mengetahui sentimen positif dan negatif. Contoh *tweet* sentimen positif dan negatif terdapat pada Gambar 3.2 dan Gambar 3.3.



Gambar 3. 2 Contoh *Tweet* Positif.



Gambar 3. 3 Contoh *Tweet* Negatif

3.5 *Pre Processing*

Pre-processing merupakan tahap sebelum klasifikasi yaitu untuk mengganti kata tidak terstruktur menjadi kata terstruktur sehingga dapat mempermudah pemrosesan dataset kata tersebut. Tahap ini melakukan *Cleaning*, *Case Folding*, *Tokenizing*, *Filtering stopwords* dan TF-IDF.

3.5.1 *Cleaning*

Cleaning yaitu proses untuk menghapus karakter non-abjad dan mengurangi *noise* yang terdiri dari tanda baca, simbol-simbol seperti tanda '@' untuk nama pengguna, hashtag (#), emotikon dan referensi alamat situs.

Tabel 3.1 *Cleaning*

Input	Output
@AkanSehat @gloryakuwin @luckyy_st @Kunyitttttt @jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita, tapi gak ada salahnya percaya sama korban yang bohong. Tapi aku sih bagian orang ² yang takut jadi #justiceforaudrey part ke sekian dengan kejadian macam ² ðŸ˜, . Lebih kasian ke Jeje kalau bener gitu	AkanSehat gloryakuwin luckyyst Kunyitttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong Tapi aku sih bagian orang ² yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam ² ðŸ˜, Lebih kasian ke Jeje kalau bener gitu

Cleaning menggunakan skrip:

```
import string
kalimat = "@AkanSehat @gloryakuwin @luckyy_st @Kunyitttttt @jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita, tapi gak ada salahnya percaya sama korban yang bohong. Tapi aku sih bagian orang2 yang takut jadi #justiceforaudrey part ke sekian dengan kejadian macam2 ðŸ˜, . Lebih kasian ke Jeje kalau bener gitu"
hasil = kalimat.translate(str.maketrans("", "", string.punctuation))
print(hasil)
```

3.5.2 Case Folding

Tahapan ini mengubah huruf menjadi *lowercase* atau huruf kecil.

Tabel 3.2 *Case Folding*

Input	Output
AkanSehat glorykuwin luckyyst Kunyitttttt jastairyine Ini baiknya kalua netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong. Tapi akusih bagian orang ² yng takut jadi	akansehat gloryakuwin luckyyst kunyitttttt jastairvine ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang

justiceforaudrey part ke sekian dengan bohong tapi aku sih bagian orang²
 kejaian macam² ðÿ~, Lebih kasian ke Jeje yang takut jadi justiceforaudrey
 sih kalo bener gitu part ke sekian dengan kejadian
 macam² ðÿ~, lebih kasian ke jeje
 kalau bener gitu

Case folding menggunakan skrip:

```

kalimat = "AkanSehat gloryakuwin luckyyst Kunyitttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemuan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong Tapi aku sih bagian orang2 yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam2 ðÿ~, Lebih kasian ke Jeje kalau bener gitu"
lower_case = kalimat.lower()
print(lower_case)

```

3.5.3 Tokenizing

Proses *tokenizing* melibatkan pemisahan *string* atau kata terhadap suatu teks berdasarkan tiap kata pada kalimat kalimat yang terdapat dari *tweet*.

Tabel 3.3 *Tokenizing*

Input	Output
AkanSehat gloryakuwin luckyyst Kunyitttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemuan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korbanyang bohong Tapi aku sih bagian orang ² yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam ² ðÿ~, Lebih kasian ke Jeje kalau bener gitu	'AkanSehat', 'gloryakuwin', 'luckyyst', 'Kunyitttttt', 'jastairvine', 'Ini', 'baiknya', 'kalau', 'netral', 'aja', 'dulu', 'sampai', 'dipertemuan', 'dan', 'ada', 'beberapa', 'versi', 'cerita', 'tapi', 'gak', 'ada', 'salahnya', 'percaya', 'sama', 'korban', 'yang', 'bohong', 'Tapi', 'aku', 'sih', 'bagian', 'orang ² ', 'yang', 'takut', 'jadi', 'justiceforaudrey', 'part', 'ke', 'sekian', 'dengan', 'kejadian', 'macam ² ', 'ðÿ~', 'Lebih', 'kasian', 'ke', 'Jeje', 'kalau', 'bener', 'gitu'

Tokenizing menggunakan skrip:

```
kalimat = "AkanSehat gloryakuwin luckyyst Kunyittttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong Tapi aku sih bagian orang2 yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam2 ðŸ˜, Lebih kasian ke Jeje kalau bener gitu"
```

```
pisah = kalimat.split()
print(pisah)
```

3.5.4 Filtering Stopwords

Proses menghapus kata-kata yang tidak berguna dan tidak ada kaitannya dengan analisis sentimen, biasanya menghilangkan kata-kata yang sering muncul dan kurang signifikan pada tahap ini.

Tabel 3.4 *Filtering Stopword*

Input	Output
AkanSehat gloryakuwin luckyyst Kunyittttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong Tapi aku sih bagian orang ² yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam ² ðŸ˜, Lebih kasian ke Jeje kalau bener gitu	['akansehat', 'gloryakuwin', 'tjastairvine', 'Ini', 'baiknya', 'kalau', 'netral', 'aja', 'dulu', 'sampai', 'dipertemukan', 'dan', 'ada', 'beberapa', 'versi', 'cerita', 'percaya', 'sama', 'korban', 'yang', 'bohong', 'Tapi', 'aku', 'sih', 'bagian', 'orang ² ', 'yang', 'takut', 'jadi', 'justiceforaudrey', 'part', 'ke', 'sekian', 'dengan', 'kejadian', 'macam ² ', 'ðŸ˜', 'Lebih', 'kasian', 'ke', 'Jeje', 'kalau', 'bener', 'gitu']

Filtering stopwords menggunakan skrip:

```
import string
kalimat = "AkanSehat gloryakuwin luckyyst Kunyittttttt jastairvine Ini baiknya kalau netral aja dulu sampai dipertemukan dan ada beberapa versi cerita tapi gak ada salahnya percaya sama korban yang bohong Tapi aku sih bagian orang2 yang takut jadi justiceforaudrey part ke sekian dengan kejadian macam2 ðŸ˜, Lebih kasian ke Jeje kalau bener gitu"
kalimat = kalimat.translate(str.maketrans('', '', string.punctuation)).lower()
tokens = word_tokenize(kalimat)
```

```
listStopword = set(stopwords.words('indonesian'))
removed = []
for t in tokens:
    if t not in listStopword:
        removed.append(t)
print(removed)
```

3.6 Ekstraksi Fitur

Sebuah prosedur yang disebut pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) mengubah data tekstual menjadi data numerik sedemikian rupa sehingga setiap kata atau karakteristik diberi bobot tertentu. (Septian et al., 2019). TF-IDF ini adalah alat statistik untuk menghitung jumlah kata dalam kumpulan data.

Rumus pembobotan kata TF-IDF adalah sebagai berikut:

$$W_t = tf \times idf$$

$$DF = \log \frac{n}{df}$$

Keterangan:

TF = Jumlah kata dalam setiap dokumen

DF = Jumlah total kata dari seluruh dokumen

N = Jumlah total dataset

TF-IDF = Jumlah dokumen dalam koleksi dokumen yang mengandung kosakata.

Dengan mengalikan nilai TF dan IDF, maka pembobotan kata dengan TF-IDF dihitung. Jika sebuah kata sering muncul disetiap *dataset*, maka bobotnya akan lebih rendah, tetapi jika jarang, maka bobotnya akan lebih tinggi.

3.7 Labelling Data

Labelling atau Pelabelan dengan mengklasifikasikan setiap data menggunakan label yang berbeda, seperti kelas positif dan negatif. Label positif ditandai dengan angka 1 sedangkan negatif ditandai dengan -1 (Tineges et al., 2020).

Tabel 3.5 *Labelling Data*

Date Time	Text	Label
2022-02-12	main hakim didunia maya menyebabkan kerugian (image, pekerjaan, mental, dll). Keduanya merugikan. Sanksi untuk pelaku main hakim sendiri apa? Audrey, Ratna Sarumpaet, yg psikolog tuh siapa namanya, segelintir kasus gimana netizen ngeduluin jempol dibanding struktur berpikir	1
2021-12-24	Kaya kasus audrey aja dulu, gampang bener kebawa netijen indo. Giliran kegocek alesannya "aware lebih dulu terhadap pembullying itu ga salah"	1
2021-11-14	Terlepas ini prank, w salut sih sama org yg saat itu gembor ngebela korban. Setidaknya w tau banyak org yg aware sama kasus si Audrey.	1
2019-07-16	Intinya jangan percaya 100% dengan apa yang dikatakan oleh media. Masih anget juga rasanya kasus Audrey dan wow fucked up banget setelah tau faktanya kaya gimana.	-1
2019-04-21	VICTIM BLAMING TEROSS Mana disangkutpautkan sama kasusnya Audrey. Wong kasus yang itu aja sampai sekarang masih diproses, kok. Karena kasus Audrey, kalian jadi victim blaming? Ckckck	-1

3.8 Implementasi Algoritma *K-Nearest Neighbor* dan *Naïve Bayes*

Algoritma *Naive Bayes* dan algoritma *K-Nearest Neighbor* dapat digunakan untuk memproses dan mengklasifikasikan data setelah melalui *pre-processing* dan pelabelan data. Klasifikasi adalah metode pengelompokan dan pemecarian *item set* berdasarkan nilai bobot yang terdapat pada setiap *item set*. Data dari penelitian ini akan di klasifikasi menjadi dua bagian, yaitu positif (1) dan negatif (-1). Metode perhitungan *Naïve Bayes* dan *K-Nearest Neighbor* bisa dilihat pada persamaan [1] dan [2].

3.9 Evaluasi

Tahap selanjutnya adalah menggunakan *confusion matrix* untuk menghitung akurasi setelah melakukan proses klasifikasi dengan *naive bayes* dan *K-nearest neighbor*. Langkah-langkah pengujian *Confusion Matrix* untuk menentukan akurasi, presisi dan *recall* adalah sebagai berikut:

1. Akurasi

Akurasi, yakni untuk mengetahui jumlah data yang di klasifikasikan secara benar. Berikut rumus perhitungan Akurasi:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100$$

2. Presisi

Presisi, digunakan untuk mengetahui tingkat ketepatan prediksi dari suatu sistem, dengan menghitung berdasarkan jumlah prediksi positif dari total data yang diprediksi sistem. Berikut rumus perhitungan Presisi:

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100$$

3. Recall

Recall, digunakan untuk mengenali tingkat keberhasilan dari suatu kelas. Berikut rumus perhitungan *Recall*:

$$\text{Recall} = \frac{TP}{TP+FN} \times 100$$

Keterangan

- TP adalah *True* Positif, yaitu jumlah data positif yang diklasifikasikan dengan benar oleh sistem.
- FN adalah *False* Negatif, yaitu jumlah data negatif namun diklasifikasikan salah oleh sistem.
- TN adalah *True* Negatif, yaitu jumlah data negatif yang diklasifikasi dengan benar oleh sistem.
- FP adalah *False* Positif, yaitu jumlah data positif yang di klasifikasikan dengan salah oleh sistem.

3.9.1 Evaluasi klasifikasi naïve Bayes

Tabel 3.6 contoh hasil pengolahan *naïve bayes*

No	Data	Klasifikasi <i>Naïve Bayes</i>	Nilai
1	1	1	True Positif
2	1	1	True Positif
3	1	-1	False Positif
4	-1	-1	True Negatif
5	-1	-1	True Negatif

Contoh hasil akurasi dengan metode *Naïve Bayes* dengan menggunakan 5 *record* data *testing*.

Accuracy : 80.00%

Precision : 66.67%

Recall : 66.67%

Tabel 3.7 Contoh Hasil Akurasi *Naive Bayes*

	<i>True positive</i>	<i>True negative</i>
<i>Pred. Positive</i>	2	1
<i>Pred. Negative</i>	0	2

3.9.2 Evaluasi Klasifikasi K-Nearest Neighbor

Tabel 3.8 contoh hasil pengolahan *K-nearest neighbor*

No	Data	Klasifikasi K- <i>Nearest Neighbor</i>	Nilai
1	1	1	True Positif
2	1	1	True Positif
3	1	1	True Positif
4	-1	1	False Negatif
5	-1	-1	True Negatif

Contoh hasil akurasi dengan metode *K-Nearest Neighbor* dengan menggunakan 5 *record* data uji.

Accuracy : 80.00%

Precision : 100.00%

Recall : 100.00%

Tabel 3.9 Contoh Hasil Akurasi *K-Nearest Neighbor*

	<i>True positive</i>	<i>True negative</i>
<i>Pred. Positive</i>	3	1
<i>Pred. Negative</i>	0	1

Berdasarkan contoh evaluasi di atas, analisis sentimen pada kasus *bulying* ini akan menentukan hasil klasifikasi dari kedua algoritma tersebut. Serta untuk menentukan akurasi, presisi dan *recall* yang tertinggi dari kedua algoritma tersebut, pada contoh evaluasi klasifikasi *Naïve Bayes* dan *K-Nearest Neighbor* memberikan nilai akurasi, presisi dan *recall* tertinggi yaitu dengan menggunakan algoritma *K-Nearest Neighbor* dengan Akurasi 80.00%, Presisi 100.00% dan *recall* 100.00%.

