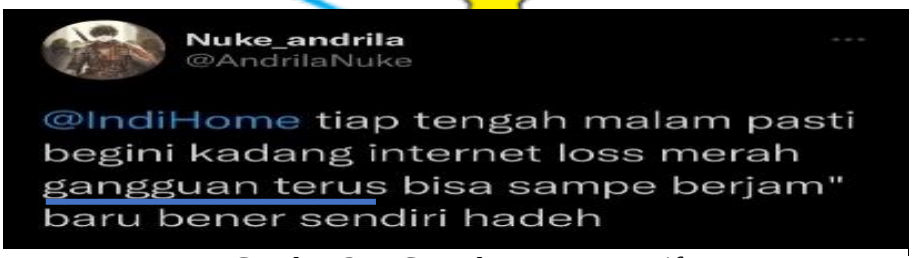
BAB III
METODE PENELITIAN

3.1. Objek Penelitian

Pada penelitian ini menggunakan objek penelitian yang didapatkan berdasarkan pada opini masyarakat pada sosial media Twitter mengenai pelayanan Indihome. Data diambil dengan menggunakan proses *crawling* pada *python* yaitu sebanyak 1.008 data *tweet* dari tanggal 01 Maret 2020-11 Desember 2022. Data yang *crawling* dipecah menjadi dua kategori yaitu positif dan negatif.



Gambar 3. 1 Contoh *Tweet Negatif*

Gambar 3. 2 Contoh *Tweet Positif*

3.2. Alat dan Bahan Penelitian

3.2.1. Alat Penelitian

Berikut merupakan perangkat keras dan perangkat lunak lunak yang digunakan pada penelitian ini :

1. Perangkat Keras *(Hardware)*
   a. Laptop Asus X441M (*Processor pentium*, 4GB RAM, 210 GB SSD).
2. Perangkat Lunak (Software)
   a. Twitter (Sebagai media untuk mencari data).
   b. *Google Chrome* (Untuk mencari referensi jurnal maupun artikel).
   c. *Microsoft Office* (Untuk membuat laporan).
   d. *Google Colab* (Sebagai text editor dalam penelitian ini).

3.2.2. Bahan Penelitian

Penelitian ini menggunakan data berupa *tweet*/komentar masyarakat terhadap pelayanan indihome dengan total data sebanyak 1.008 *tweet* yang berbahasa Indonesia, hasil pengumpulan data ini disimpan ke dalam format excel (.xlsx).

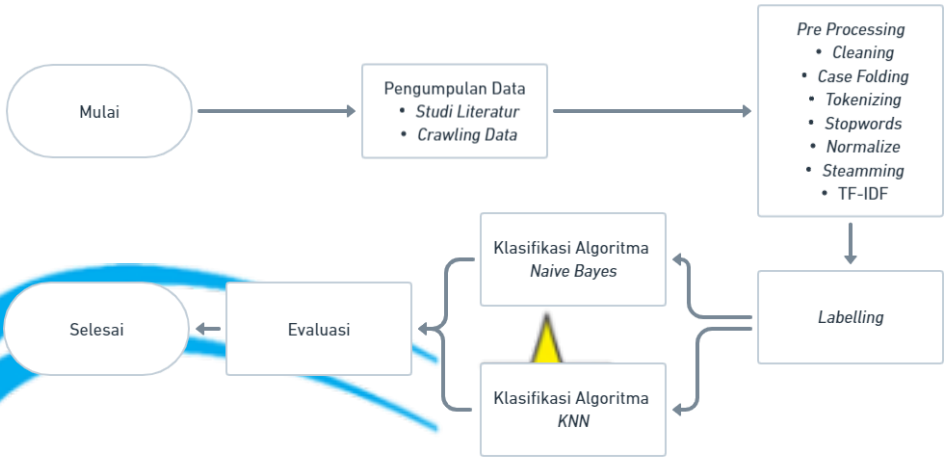
3.3. Lokasi dan Waktu Penelitian

Selama enam bulan, penelitian dilakukan di Universitas Buana Perjuangan Karawang.

Tabel 3. 1 Waktu Penelitian

No	Keterangan	Okt 2022				Nop 2022				Des 2022				Jan 2023				Feb 2023	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
1.	Pengumpulan Data																		
2.	*Preprocessing*																		
3.	*Word Cloud*																		
4.	*Labelling*																		
5.	Klasifikasi *Naïve Bayes*																		
6.	Klasifikasi KNN																		
7.	Evaluasi																		

3.4. Prosedur Penelitian



Gambar 3. 3 *Flowchart* Prosedur Penelitian

Prosedur penelitian yang digambarkan dengan menggunakan bentuk *Flowchart* dalam Gambar 3.1. Penelitian ini diawali dengan melakukan pengumpulan data studi literatur pada jurnal-jurnal terdahulu dan tahap *crawling* data dengan menggunakan *library textblob* pada google colab menggunakan Bahasa python. Kemudian data yang sudah terkumpul akan di proses pada bagian Proses *Cleaning, Casefolding, Tokenizing, Filtering, Stemming,* dan *TF-IDF* adalah bagian dari *Pre-processing* untuk proses membersihkan data. Setelah data dibersihkan maka data akan diklasifikasikan dalam dua kategori kelas yaitu kelas positif dan negatif. data yang sudah di klasifikasi akan diimplementasikan dengan algoritma *Naive Bayes* dan *K-Nearest Neighbor*, kemudian dilanjutkan ke tahap evaluasi yaitu pengujian model klasifikasi dengan menggunakan *Confusion Matrix.*

3.5 Pengumpulan Data

Pengumpulan data dari penelitian ini dibagi menjadi dua bagian antara lain :

3.5.1 *Studi Literatur*

Studi literatur dilakukan untuk mencari landasan teori dan mendapatkan referensi dari buku, jurnal, dan internet. Tahapan juga mencari pemahaman tentang tinjauan literatur, penelitian sebelumnya, dan metode yang digunakan..

3.5.2 *Crawling Data*

Proses *crawling data* dapat dilakukan dengan melakukan pengambilan data pada twitter lalu dikumpulkan menggunakan *python*, pencarian dengan kata kunci "Indihome". Data diambil dengan total 1.008 *tweet* yaitu pada 01 Maret 2020 - 11 Desember 2022. Data tersebut kemudian akan di klasifikasikan dengan Algoritma *Naive Bayes* dan *K-Nearest Neighbor*.

	Datetime	Tweet Id	Text	Username	Label
0	2020-03-29 23:56:28+00:00	1244413146749472769	@putranto1125 @TelkomCare @IndiHome Gangerti l...	bchtirfi	1
1	2020-03-29 23:44:26+00:00	1244410115127570432	@IndiHome perimis qaqa, kenapa akhir2 ini jari...	orinkrisna	1
2	2020-03-29 22:53:24+00:00	1244397273506656256	@IndiHome Mohon bantuannya, internet saya LOS....	mikhaahkim	1
3	2020-03-29 21:10:59+00:00	1244371499550556162	Tolong dong Indihome, wifi dirumah ku aktifin ...	brightfat	1
4	2020-03-29 20:02:34+00:00	1244354281982980105	@jordi154 Ok Kak Jordi, sudah di respon DM. -...	IndiHome	1
...	...	...	...	...	...
1004	2022-11-29 12:51:06+00:00	1597573869690200064	@moanamenfess Poor signal cenah....indihome ru...	rejistancetxt	-1
1005	2022-11-29 12:26:06+00:00	1597567576980455425	hari ini namatin reparasi indihome, bookings s...	notsannyforreal	-1
1006	2022-12-11 10:48:34+00:00	1601891687592128515	Yg hobby maen game tapi saat maen internt nya ...	IrfanJa44129301	-1
1007	2022-12-11 10:45:12+00:00	1601890840145805316	Yg hobby maen game tapi saat maen internet nya...	UZulviana	-1
1008	2022-12-11 06:42:09+00:00	1601829674287448064	Mumpung di kantor sudah berlangganan wifi Indi...	ArdouAjawaila	-1

Gambar 3. 4 Contoh Kata Dengan Kata Kunci Indihome

3.6 *PreProcessing*

Sebelum tahap klasifikasi, kata-kata yang tidak terstruktur diubah menjadi kata-kata yang lebih terstruktur. Tahap *pre-processing* dapat membuat kumpulan kata pada dataset lebih mudah diproses. *Pre-processing* ini dilakukan menggunakan google colab dengan library *nltk* dan *sastrawi*. Pada tahap ini akan dilakukan proses *Cleaning, Case Folding, Tokenizing, Filtering, Stemming* dan TF-IDF

3.6.1 *Cleaning*

Proses *Cleaning* melibatkan penghapusan karakter non-abjad, seperti tanda baca, simbol seperti tanda "@" di nama pengguna, tagar (#), *emoticon*, dan url situs web, serta mengurangi kesalahan. Berikut merupakan contoh tahapan *cleaning* :

Tabel 3. 2 Cleaning

INPUT	*OUTPUT*
@Indihome tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh	tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh

3.6.2 *Case Folding*

Proses mengubah huruf menjadi huruf kecil atau *lowercase* dikenal dengan istilah *Case Folding*. Berikut adalah contoh dari proses *Case Folding* :

Tabel 3. 3 Case Folding

INPUT	*OUTPUT*
tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh	tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh

3.6.3 *Tokenizing*

Tokenizing adalah proses memisahkan teks menjadi *string* atau kata-kata berdasarkan setiap kata dalam kalimat yang diperoleh dari tweet.

Tabel 3. 4 Tokenizing

INPUT	*OUTPUT*
tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh	'tiap' 'tengah' 'malam' 'pasti' 'begini' 'kadang' 'internet' 'loss' 'merah' 'gangguan' 'terus' 'bisa' 'sampe' 'berjam"' 'baru' 'bener' 'sendiri' 'hadeh'

3.6.4 Stopwords

Stopwords digunakan untuk menghilangkan kata-kata yang tidak berarti dan tidak berpengaruh pada analisis sentimen serta kata-kata yang sering muncul tetapi kurang *relevan*.

Tabel 3. 5 Stopwords

INPUT	*OUTPUT*
tiap tengah malam pasti begini kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri hadeh	tiap tengah malam kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri

3.6.5 *Normalize*

Proses *Normalize* digunakan untuk mengubah kata, kesalahan dalam pengetikan (typo) menjadi kata yang mudah di pahami atau sesuai KBBI.

Tabel 3. 6 Normalize

INPUT	*OUTPUT*
tiap tengah malam kadang internet loss merah gangguan terus bisa sampe berjam" baru bener sendiri	tiap tengah malam kadang internet loss merah gangguan terus bisa sampe berjam-jam baru bener sendiri

3.6.6 *Stemming*

Stemming adalah fase normalisasi dimana kata diubah menjadi kata dasar atau imbuhan kata dihilangkan. Bobby Nazief dan Mirna Adriani memelopori pembuatan Algoritma Stemming Nazief & Adriani. Penggunaan bahasa Indonesia menjadi dasar dari algoritma ini. Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.

Tahapan-tahapan algoritma Nazief & Adriani adalah :

1. Pada kamus dicari kata yang belum di-*stemming* jika ditemukan, kata tersebut dianggap sebagai kata dasar yang benar, dan algoritma dihentikan.
2. Hilangkan *inflectional suffixes*, khususnya partikel (seperti -lah, -kah, -tah, atau -pun), diikuti sufiks kata ganti posesif infleksi (seperti -ku, -mu, atau -his). Algoritma kemudian akan berhenti bekerja jika kata tersebut ditemukan di kamus dasar, jika tidak ditemukan lanjutkan ke langkah 3.
    a. Hapus partikel ("-i" atau "-an"), juga dikenal sebagai *Derivational suffix.* Algoritme akan berhenti bekerja jika kata tersebut ditemukan di kamus dasar setelah diperiksa; jika tidak, lanjutkan ke langkah 3a.

b. Huruf "k" juga harus dihilangkan jika akhiran "-an" telah dihilangkan dan huruf terakhir kata tersebut adalah "-k". Algoritme berhenti bekerja jika kata ditemukan. Jika tidak, lanjutkan ke langkah 3b.
c. Akhiran kata yang memiliki *particle*(“-i”, “- an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

3. Menghilangkan *Derivational Prefix* dari kata yang mengandung partikel seperti "be", "di", "ke", "me", "pe", "se", dan "te", Algoritma dihentikan jika kata tersebut ditemukan dalam kamus . Jika tidak, tahap perekaman harus diselesaikan. Jika salah satu kondisi berikut terpenuhi, tahap ini dihentikan.:
   a. Terdapat kombinasi awalan dan akhiran yang tidak diijinkan.
   b. Awalan yang terdeteksi sama dengan awalan yang dihilangkan sebelumnya.
   c. Tiga awalan telah dihilangkan.
4. Algoritma ini mengembalikan kepada kata asal sebelum stemming jika semua langkah telah selesai tetapi kata dasar tidak ditemukan dalam kamus.

Tabel 3. 7 Stemming

INPUT	*OUTPUT*
tiap tengah malam kadang internet loss merah gangguan terus bisa sampe berjam-jam baru bener sendiri	tiap tengah malam kadang internet loss merah gangguan terus bisa sampe jam-jam baru bener sendiri

3.6.7 Pembobotan Kata

Pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) digunakan untuk mengubah data tekstual menjadi data numerik sehingga setiap kata atau fitur kalimat dapat diberi bobot (Septian et al., 2019). *TF-IDF* ini digunakan untuk mengetahui seberapa banyak kata yang terdapat dalam sebuah dataset dalam sebuah ukuran statistik.

Rumus pembobotan kata TF-IDF adalah sebagai berikut :

$$Wt = tf \text{ x } idf$$

$$\text{IDF} = \log \frac{\text{N}}{\text{df}}$$

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 di setiap dataset, maka bobotnya akan lebih rendah, tetapi jika jarang, maka bobotnya akan lebih tinggi. Berikut merupakan contoh perhitungan pembobotan kata menggunakan *TF-IDF* :

D1 = indihome keren promonya *good* akses internet cepat dan stabil wow.

D2 = indihome *trouble*.

TF = Menghitung "*Term*" atau "kata" pada dokumen. Contoh : *Term* "indihome" pada dokumen 1 berjumlah 1 term.

DF = Menghitung banyaknya dokumen dalam suatu term. Contoh : *Term* "indihome" pada dokumen 1 berjumlah 1 dan pada dokumen 2 berjumlah 1 sehingga nilai DF nya adalah 2.

IDF = IDF : $\log \frac{N}{df}$ = IDF : $\log \frac{2}{2}$ = 0

TF-IDF = $Wt : tf$ x idf = Wt : 1 x 0 = 0

Tabel 3. 8 Contoh Pembobotan Kata Dengan TF-IDF

Term	D1	D2	DF	IDF	TF-IDF	
					D1	D2
Indihome	1	1	2	0	0	0
Trouble	1		1	0.301	0.301	
Keren		1	1	0.301		0.301
Promo		1	1	0.301		0.301
Good		1	1	0.301		0.301
Akses		1	1	0.301		0.301
Internet		1	1	0.301		0.301
Cepat		1	1	0.301		0.301
Stabil		1	1	0.301		0.301
wow		1	1	0.301		0.301

3.7 *Labelling* Data

Setelah *Pre-processing* dan Pembobotan TF-IDF tahap selanjutnya *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. 9 *Labelling Data*

Datetime	Text	Label
2022-10-29	nonton netflix paling seru di barengi jaringan internet indihome yang super super luar biasa	1
2022-10-29	sekarang kalau mau tanya terkait layanan indihome gampang banget. Bisa telepon 147 sampai dm di sosmed	1
2022-10-29	indihome keren promonya good akses internet cepat dan stabil wow	1
2022-10-29	indihome trouble	-1
2022-10-29	kualitas dan harga kalian yang gak masuk akal	-1

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

Algoritma *Naive Bayes* dan algoritma *K-Nearest Neighbor* dapat digunakan untuk memproses dan mengklasifikasikan data setelah melalui *pre-processing* dan pelabelan data. Klasifikasi merupakan metode untuk mengelompokan dan menemukan *itemset* sesuai pada nilai bobot yang terdapat pada setiap *itemset*. Data dalam penelitian ini akan klasifikasi menjadi dua bagian, yaitu positif (1) dan negatif (-1). Metode perhitungan algoritma *Naïve Bayes* ditunjukkan pada persamaan [1].

$$P(H|X) = \frac{P(X).P(H)}{P(X)}$$

X : Data yang kelasnya tidak diketahui
H : Kelas spesifik adalah data hipotesis X
P(H)|(X) : Probabilitas hipotesis H berdasar kondisi
P(H) : Probabilitas hipotesis H
P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis
P(X) : Probabilitas X

[1]

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

Tabel 3. 10 Contoh Hasil Pengolahan *Naive Bayes*

No	Data	Klasifikasi Naïve Bayes
1	Data Positif	Data Positif
2	Data Positif	Data Positif
3	Data Positif	Data Negatif
4	Data Negatif	Data Negatif
5	Data Negatif	Data Negatif

Metode perhitungan algoritma *K-Nearest Neighbor* dengan *Euclidean distance* ditunjukkan pada persamaan [2].

$$d = \sum_{i=1}^{n} (a_i - b_1)^2$$

d : Jarak
a : Data uji/testing
b : Sampel pada data
i : Variabel pada data
n : Dimensi pada data

[2]

Contoh hasil pengolahan dengan metode *K-Nearest Neighbor* dengan menggunakan 5 record data *testing*.

Tabel 3. 11 Contoh Hasil Pengolahan *K-Nearest Neighbor*

No	Data	Klasifikasi K-Nearest Neighbor
1	Data Positif	Data Positif

2	Data Positif	Data Positif
3	Data Positif	Data Positif
4	Data Negatif	Data Positif
5	Data Negatif	Data Negatif

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 *accuracy, precision*, dan *recall* adalah sebagai berikut.

1. *Accuracy*

   Accuracy, yakni untuk mengetahui jumlah data yang di klasifikasikan secara benar. Berikut rumus perhitungan *Accuracy*

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100$$

2. *Precision*

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

$$Precision = \frac{TP}{TP+FP}$$

3. *Recall*

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

$$Recall\ = \frac{TP}{TP+FN}$$

Keterangan :

A. TP adalah *True Positif*, yaitu jumlah data positif yang diklasifikasikan

dengan benar oleh sistem.

B. FP adalah *False Positif*, yaitu jumlah data positif yang diklasifikasikan dengan salah oleh sistem

C. FN adalah *False Negatif*, yaitu jumlah data negatif namun diklasifikasikan salah oleh sistem.

D. TN adalah *True Negatif*, yaitu jumlah data negatif yang diklasifikasikan dengan benar oleh sistem.

3.9.1. Evaluasi Klasifikasi *Naïve Bayes*

No	Data	Klasifikasi *Naïve Bayes*	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. 12 Contoh Hasil Akurasi *Naive Bayes*

	True positive	*True negative*
Pred. positive	2	1
Pred. negative	0	2

3.9.2. Evaluasi Klasifikasi *K-Nearest Neighbor*

No	Data	Klasifikasi *K-Nearest Neighbor*	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 *testing*.

Accuracy **: 80.00%**

Precision **: 100.00%**

Recall **: 100.00%**

Tabel 3. 13 Contoh Hasil Akurasi *K-Nearest Neighbor*

	True positive	*True negative*
Pred. positive	3	1
Pred. negative	0	1

Berdasarkan contoh evaluasi diatas, analisis sentimen pada pelayanan indihome ini akan menentukan hasil klasifikasi dari kedua algoritma tersebut. Serta untuk menentukan *accuracy, precision, recall* yang tertinggi dari kedua algoritma tersebut, dari contoh evaluasi klasifikasi *Naïve Bayes* dan *K-Nearest Neighbor* menghasilkan nilai *accuracy, precision,* dan *recall* tertinggi yaitu dengan menggunakan algoritma *K-Nearest Neighbor* dengan *Accuracy* 80.00%, *Precision* 100.00% dan *recall* 100.00%.