

BAB III METODE PENELITIAN

3.1. Objek Penelitian

Objek yang digunakan pada penelitian ini adalah *tweet* dari para pengguna *Twitter* yang berkaitan dengan kata BBM naik yang diambil pada saat terjadinya kenaikan harga BBM, yaitu pada tanggal 3 September 2022 hingga 30 November 2022 sebanyak 14.520 *Tweet*. Data yang di *crawling* terbagi menjadi 2 kelas yaitu: kelas positif, kelas negatif. Setelah itu data mengalami tahap *filtering* menjadi 2 kelas yaitu: kelas positif dan negatif dengan jumlah data 1.334.



Gambar 3. 1 *Tweet* Positif



Gambar 3. 2 *Tweet* Negatif

Berdasarkan yang ditunjukkan oleh Gambar 3.1 terdapat kalimat “Aku sih setuju kalo BBM naik, ternyata kalo BBM naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo BBM nggak naik” mengandung makna positif pada kalimat “setuju”. Gambar 3.2 terdapat “Bkn cm ngutang. Bahkan menindas rakyatnya. Contoh kenaikan bbm. Sy korbannya” mengandung makna negatif pada kalimat “menindas rakyatnya”.

3.2. Prosedur Penelitian

Prosedur penelitian ini terdiri dari dua bagian antara lain:

3.2.1. Alat Penelitian

Dalam penelitian ini, digunakan peralatan *hardware* dan *software* sebagai berikut:

1. Perangkat Keras (*Hardware*)
 - a. Laptop (Core i3, 4GB RAM, 210 GB SSD)
2. Perangkat Lunak (*Software*)
 - a. Twitter (Sebagai media untuk mencari data)
 - b. Google Chrome (Mencari referensi jurnal maupun artikel)
 - c. Microsoft Office (Untuk membuat laporan dan proposal)
 - d. Google Colab (Sebagai *text editor* dalam penelitian ini)

3.2.2. Bahan Penelitian

Data yang digunakan yaitu *tweet* atau komentar masyarakat terkait kenaikan harga Bahan Bakar Minyak (BBM) dengan total data sebanyak 14.520 *tweet* yang berbahasa Indonesia, hasil pengumpulan data ini disimpan ke dalam format *excel* (.xlsx).

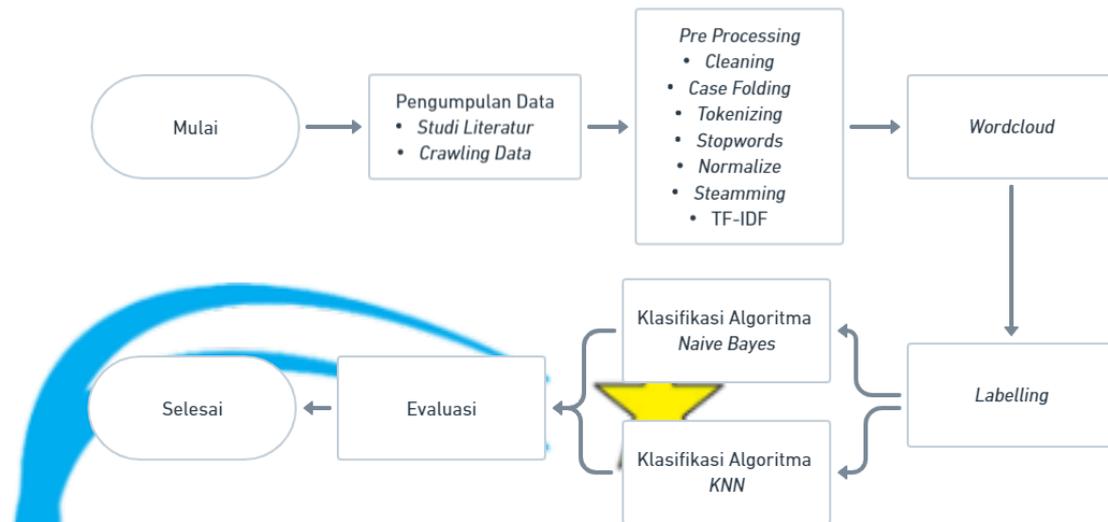
3.3. Lokasi dan Waktu Penelitian

Penelitian dilakukan Universitas Buana Perjuangan Karawang. Kemudian, pelaksanaan penelitian berlangsung selama 6 Bulan.

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	3	4
1.	Pengumpulan Data																				
2.	<i>Preprocessing</i>																				
3.	<i>Word Cloud</i>																				
4.	<i>Labelling</i>																				
5.	Klasifikasi <i>Naïve Bayes</i>																				
6.	Klasifikasi SVM																				
7.	Evaluasi																				

3.4. Prosedur Penelitian



Gambar 3. 3 *Flowchart* Prosedur Penelitian

Prosedur penelitian yang digambarkan dengan menggunakan bentuk *Flowchart* dalam Gambar 3.3. Penelitian ini dimulai dari Analisis data kemudian *crawling* data dengan menggunakan *library textblob* pada Google Colab menggunakan bahasa Python. Kemudian data yang sudah terkumpul akan diproses pada bagian *Preprocessing* yang terdiri dari proses *Cleaning*, *Case Folding*, *Tokenizing*, *Filtering*, *Stemming* dan *TF-IDF* untuk membersihkan data, lalu memunculkan *Word Cloud*. Setelah data bersih yaitu pemberian label data dalam kategori kelas yang berbeda yaitu kelas positif, negatif. data yang sudah diklasifikasi akan implementasikan kepada algoritma *Naive Bayes* dan *Support Vector Machine*, kemudian dilanjutkan ke tahap evaluasi yaitu pengujian model klasifikasi dengan menggunakan *Confusion Matrix*.

3.5. Pengumpulan Data

Dalam proses pengumpulan data dalam penelitian ini, terdapat dua bagian antara lain:

3.5.1. *Studi Literatur*

Studi literatur dilakukan sebagai pencarian landasan teori dan untuk mendapatkan referensi dari beraneka macam jurnal, buku, dan juga internet seperti mencari pengertian pengertian terhadap untuk tinjauan pustaka, penelitian terdahulu, metode yang dipakai.

3.5.2. Crawling Data

Proses mengumpulkan data dilakukan dengan mengambil data *tweet* yang dikumpulkan menggunakan Python, pencarian dengan kata kunci “Harga BBM naik”. Data diambil dengan total 14.520 *tweet*, sejak 3 September 2022 hingga 30 November 2022. Data kemudian akan diklasifikasi dengan Algoritma *Naive Bayes* dan SVM.

	Datetime	Tweet Id	Text	Username	Label
0	2022-09-29 23:52:39+00:00	1575634693693001728	@KompasTV Iya dong!! Cuma kasdrone yang gak te...	emyr_z	0
1	2022-09-29 23:41:25+00:00	1575631864957526016	@CNNIndonesia Justru ini yg mengawatirkan,har...	sabenilah	0
2	2022-09-29 23:05:49+00:00	1575622907489030144	malu di ledekin si bogel ?mau tinggi ?pake cal...	gifarihwi	1
3	2022-09-29 23:02:38+00:00	1575622108515737604	@OposisiCerdas Lanjutkan para mahasiswa suarak...	juanto_mm	0
4	2022-09-29 22:35:41+00:00	1575615322559651840	malu di ledekin si bogel ?mau tinggi ?pake cal...	gifarihwi	1
...
14515	2022-12-01 00:16:02+00:00	1598108625498558464	Harga BBM Pertamina Turbo, Dexlite, dan Pertami...	KompasEkonomi	0
14516	2022-12-01 00:12:25+00:00	1598107714928119809	@panca66 harusnya kalau harga BBM naik tuh dib...	aagunawan_	0
14517	2022-12-01 00:11:32+00:00	1598107495540531201	Kompak! Ikut Pertamina, Harga BBM BP-AKR Resmi...	cnbcindonesia	0
14518	2022-12-01 00:05:32+00:00	1598105985087700992	Resmi Naik! Ini Daftar Harga BBM Pertamina di ...	cnbcindonesia	0
14519	2022-12-01 00:01:02+00:00	1598104852630228994	Harga BBM Pertamina Naik 1 Desember 2022, Cek ...	liputan6dotcom	0

14520 rows x 5 columns

Gambar 3. 4 Contoh data dengan kata kunci Harga BBM naik

3.6. Preprocessing

Preprocessing merupakan bagian awal sebelum proses klasifikasi dan mempunyai tujuan untuk mengubah kata-kata yang tidak teratur menjadi kata-kata yang teratur dan lebih mudah diproses. Dalam tahap ini dilakukan beberapa proses seperti membersihkan (*cleaning*), membuat huruf menjadi huruf kecil (*case folding*), membagi kata (*tokenizing*), menghapus kata tidak penting (*stopwords*), *normalize*, dan melakukan *stemming* serta menggunakan teknik TF-IDF.

3.6.1. Cleaning

Cleaning adalah suatu tahapan dalam *preprocessing* data yang melibatkan penghapusan unsur-unsur yang tidak relevan seperti tanda baca, simbol "@" yang digunakan sebagai nama pengguna, *hashtag* (#), *emoticon*, dan URL situs web.

Tabel 3. 2 Contoh *Cleaning*

INPUT	OUTPUT
-------	--------

Aku sih setuju kalo BBM naik, ternyata kalo BBM naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo BBM nggak naik.

Aku sih setuju kalo BBM naik, ternyata kalo BBM naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo BBM nggak naik.

3.6.2. Case Folding

Case Folding merupakan proses pengubahan huruf besar menjadi *lowercase*. Proses ini dilakukan untuk memastikan bahwa setiap kata dalam data terstruktur dengan huruf kecil sehingga mempermudah dalam pemrosesan data. Langkah-langkah dalam melakukan *Case Folding* meliputi:

Tabel 3. 3 Contoh *Case Folding*

<i>INPUT</i>	<i>OUTPUT</i>
Aku sih setuju kalo BBM naik, ternyata kalo BBM naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo BBM nggak naik.	aku sih setuju kalo bbm naik, ternyata kalo bbm naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo bbm nggak naik.

3.6.3. Tokenizing

Tokenizing merupakan proses pemisahan *string* atau kata-kata yang terkandung dalam sebuah teks berdasarkan pada kata-kata pada setiap kalimat yang ada dalam *tweet*.

Tabel 3. 4 Contoh *Tokenizing*

<i>INPUT</i>	<i>OUTPUT</i>
--------------	---------------

aku sih setuju kalo BBM naik, ternyata 'aku' 'sih' 'setuju' 'kalo' 'BBM' kalo BBM naik pengeluaran kita itu jadi 'naik' 'ternyata' 'kalo' 'BBM' 'naik' lebih sedikit dibandingkan kalo BBM 'pengeluaran' 'kita' 'itu' 'jadi' nggak naik. 'lebih' 'sedikit' 'dibandingkan' 'kalo' 'BBM' 'nggak' 'naik'

3.6.4. Stopword

Menghapus kata-kata tidak signifikan merupakan salah satu proses penting dalam memfilter informasi yang tidak relevan atau tidak berkaitan dengan analisis sentimen. Tujuannya adalah untuk memastikan bahwa data yang akan dianalisis hanya mengandung informasi yang relevan dan memiliki pengaruh yang nyata pada analisis sentimen.

Tabel 3. 5 Contoh *stopword*

<i>INPUT</i>	<i>OUTPUT</i>
aku sih setuju kalo BBM naik ternyata	aku setuju kalo BBM naik ternyata
kalo BBM naik pengeluaran kita itu jadi	kalo BBM naik pengeluaran kita jadi
lebih sedikit dibandingkan kalo BBM	lebih sedikit dibandingkan kalo BBM
nggak naik	nggak naik

3.6.5. Normalize

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

Tabel 3. 6 Contoh *Normalize*

<i>INPUT</i>	<i>OUTPUT</i>
aku setuju kalo BBM naik ternyata kalo	aku setuju kalau BBM naik ternyata
BBM naik pengeluaran kita jadi lebih	kalau BBM naik pengeluaran kita
sedikit dibandingkan kalo BBM nggak	menjadi lebih sedikit dibandingkan
naik	kalau BBM engga naik

3.6.6. Stemming

Stemming merupakan proses untuk memetakan kata ke bentuk dasar-nya dengan menghapus imbuhan pada akhiran, awalan, atau kedua-duanya. Ini dilakukan untuk mengkonsolidasikan bentuk varian dari kata yang sama menjadi satu kata dasar yang lebih sederhana, sehingga mempermudah identifikasi kata yang serupa dan

meminimalisir dampak dari variasi bahasa pada hasil analisis. Penelitian ini menggunakan Algoritma Nazief & Adriani, yang ditemukan oleh Bobby Nazief dan Mirna Adriani dan dirancang khusus untuk Bahasa Indonesia. Algoritma ini memanfaatkan sebuah kamus kata dasar dan memiliki kemampuan untuk melakukan proses *reconstructing*, yaitu membentuk kembali kata-kata yang sudah melalui proses *stemming* yang berlebihan. Proses yang dilakukan oleh Algoritma Nazief & Adriani meliputi beberapa langkah-langkah tertentu:

1. Melakukan pencarian pada kamus untuk setiap kata yang belum melalui proses *stemming*. Bila kata tersebut terdapat dalam kamus, maka dikategorikan sebagai kata dasar yang valid dan proses dalam algoritma akan berakhir.
2. Proses pembuangan imbuhan Infleksi. Pertama, imbuhan partikel seperti "-lah", "-kah", "-tah", atau "-pun" akan dihapus. Selanjutnya, imbuhan possessive pronoun seperti "-ku", "-mu", atau "-nya" juga akan dihapus. Setelah itu, kata tersebut akan dicari dalam kamus kata dasar. Bila kata tersebut ditemukan, proses algoritma akan berakhir, namun bila tidak, proses akan dilanjutkan ke langkah berikutnya.
3. *Derivational Suffix* akan dihapus (seperti "-i" atau "-an"). Kemudian, kata tersebut akan dicari pada kamus kata dasar. Jika ditemukan, proses algoritma akan berhenti. Namun jika tidak, maka akan melanjutkan ke langkah 3a.
 - a. Setelah menghapus akhiran derivasional "-an", jika huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga akan dikeluarkan. Kemudian, kata tersebut akan dicari dalam kamus. Jika ditemukan, proses algoritma akan dihentikan, namun jika tidak, akan melanjutkan ke tahap 3b.
 - b. Setelah menghapus akhiran, seperti "i", "an" atau "kan", akhiran tersebut akan dikembalikan dan proses algoritma akan melanjutkan ke langkah 4.
4. Selanjutnya, Algoritma Nazief & Adriani akan melakukan penghapusan terhadap awalan derivasional, seperti "be-", "di-", "ke-", "me-", "pe-", "se-" dan "te-". Apabila kata yang diperoleh sudah terdapat dalam kamus kata dasar, maka proses algoritma akan berakhir. Namun, jika kata tersebut tidak ditemukan, algoritma akan melakukan proses *recoding*. Pemrosesan ini akan dihentikan bila memenuhi kondisi-kondisi tertentu:

- a. Ada kombinasi *prefiks* dan *sufiks* yang tidak diterima dalam proses.
 - b. Awalan yang teridentifikasi seiring dengan awalan yang sudah dihapus sebelumnya.
 - c. Tiga awalan telah dihapuskan.
5. Jika setelah menjalani semua langkah dalam algoritma Nazief & Adriani, tidak ditemukan kata dasar yang dicari dalam kamus, maka algoritma ini akan mengembalikan kata asli tanpa adanya perubahan yang terjadi selama proses *stemming*. Ini berarti bahwa kata tidak berubah dan masih sama seperti sebelum dilakukan proses *stemming*.

Tabel 3. 7 *Stemming*

<i>INPUT</i>	<i>OUTPUT</i>
aku setuju kalau bbm naik ternyata kalo bbm naik pengeluaran kita menjadi lebih sedikit dibandingkan kalau bbm engga naik	aku setuju kalau bbm naik ternyata kalo bbm naik pengeluaran kita menjadi lebih sedikit banding kalau bbm engga naik

3.6.7. Pembobotan Kata

Metode pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah cara untuk mengubah data teks menjadi bentuk numerik dengan memperhitungkan bobot pada setiap kata atau fitur yang ada. Hal ini dilakukan untuk mempermudah pemrosesan data dan memberikan representasi yang lebih kuat pada setiap fitur (Septian et al., 2019).

Adapun rumus pembobotan kata TF-IDF adalah:

$$Wt = tf \times idf$$

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

Keterangan :

TF = Jumlah kata dalam setiap dokumen

DF = Jumlah total kata dari seluruh dokumen

N = Jumlah total dataset

(5)

Metode pembobotan TF-IDF memperhitungkan bobot pada setiap kata atau fitur yang ada dalam data teks dan mengubahnya menjadi data numerik. Hal ini dilakukan untuk mempermudah proses pemrosesan data dan memperkuat representasi setiap fitur. Penerapan pembobotan memanfaatkan hasil dari perkalian antara *Term*

Frequency (TF) dan *Inverse Document Frequency* (IDF). Kata yang sering terdapat dalam seluruh data akan memiliki bobot yang lebih rendah, sementara kata yang jarang muncul akan memperoleh bobot yang lebih tinggi. Dengan demikian, proses pembobotan ini bertujuan untuk memberikan bobot yang tepat pada setiap kata sesuai dengan frekuensinya dalam dataset.

Tabel 3. 8 TF-IDF

<i>Term</i>	D1	D2	DF	IDF	TF_IDF	
					D1	D2
sesuai	1	0	1	0,301	0,301	0
harga	2	0	2	0	0	0
bbm	1	1	2	0,301	0,301	0,301
langkah	1	0	1	0,301	0,301	0
ideal	1	0	1	0,301	0,301	0
tengah	1	0	1	0,301	0,301	0
gejolak	1	0	1	0,301	0,301	0
global	1	0	1	0,301	0,301	0
naik	1	1	2	0	0	0
bukan	0	1	1	0,301	0	0,301
hutang	0	1	1	0,301	0	0,301
bahkan	0	1	1	0,301	0	0,301
tindas	0	1	1	0,301	0	0,301
rakyat	0	1	1	0,301	0	0,301
contoh	0	1	1	0,301	0	0,301
saya	0	1	1	0,301	0	0,301
korban	0	1	1	0,301	0	0,301

Keterangan:

D1 = penyesuaian harga bbm langkah ideal ditengah gejolak bbm global naik.

D2 = bukan cuma ngutang bahkan menindas rakyatnya contoh kenaikan harga bbm saya korbannya.

TF = Menghitung “*Term*” atau “Kata” pada dokumen. Contoh *Term* “bbm” pada dokumen 1 berjumlah 1 dan pada dokumen 2 berjumlah 1 *trem*.

DF = Menghitung banyaknya dokumen dalam satu term. Contoh *Term* “bbm” pada dokumen 1 berjumlah 1 dan pada dokumen 2 berjumlah 1 sehingga nilai DF nya adalah 2.

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

$$TF-IDF = wt : ft \times idf = Wt : 1 \times 0 = 0$$

3.7. Word Cloud

Word Cloud suatu proses visualisasi yang terdiri dari kumpulan kata yang sering muncul atau kata yang paling banyak timbul pada dataset. *Word Cloud* sebuah representasi data yang berbentuk teks dan juga dapat berguna dalam berbagai konteks untuk memvisualisasikan data teks secara menarik. *Word Cloud* biasanya digunakan untuk menampilkan data teks atau kata yang sering muncul.

3.8. Labelling Data

Setelah *Preprocessing* dan Pembobotan TF-IDF tahap selanjutnya memberikan label atau tanda pada setiap data, memasukkannya ke dalam kategori yang berbeda, seperti kelas positif ditandai angka 1 dan kelas negatif ditandai angka -1. Proses ini disebut dengan pelabelan atau *labelling*.

Tabel 3. 9 *Labelling*

Datetime	Text	Label
2022-10-29	Penyesuaian harga BBM langkah ideal di tengah gejolak harga BBM global naik	1
2022-10-29	Aku sih setuju kalo BBM naik, ternyata kalo BBM naik pengeluaran kita itu jadi lebih sedikit dibandingkan kalo BBM nggak naik	1
2022-10-29	Wajar karena BBM naik, tapi masa hitungan harga tarif kapasitas PPKM masih dipake juga sampe skrg? Padahal kapasitas bus udah normal	1
2022-10-29	brow awal bbm naik harga gue isi full tank pertalite 45k buat 2 minggu bisa, sekarang cepet bener anjay abisnya masa baru seminggu abis	-1

2022-10-29	Bkn cm ngutang. Bahkan menindas rakyatnya. Contoh kenaikan bbm. Sy korbannya	-1
------------	--	----

3.9. Implementasi *Naive Bayes* dan *Support Vector Machine*

Setelah data telah melalui *preprocessing* dan *labelling* data maka dapat diolah dan diklasifikasikan dengan menggunakan algoritma *Naive Bayes* dan *Support Vector Machine*. Klasifikasi merupakan metode untuk mengelompokkan dan menemukan *itemset*. Dalam penelitian ini akan dikelompokkan menjadi dua bagian yaitu positif, negatif. Metode perhitungan algoritma *Naive Bayes* ditunjukkan pada persamaan (1).

$$P(H|X) = \frac{P(X|K).P(K)}{P(X)} \quad (1)$$

Keterangan :

- X : Data dengan class yang belum diketahui
H : Hipotesis data X merupakan suatu class spesifik
P(H|X) : Probabilitas hipotesis H berdasar kondisi X (posteriori probability)
P(H) : Probabilitas hipotesis H (prior probability)
P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H
P(X) : Probabilitas X

Langkah-langkah dari implementasi metode *Naive Bayes* adalah sebagai berikut:

1. Baca data *training*.
2. Hitung Jumlah dan probabilitas, namun apabila data numerik maka:
 - a. Menentukan nilai rata-rata dan deviasi standar dari setiap parameter yang bersifat numerik.
 - b. Proses menentukan nilai probabilistik dilakukan dengan menghitung rasio antara jumlah data yang sesuai dengan suatu kategori tertentu dan jumlah data keseluruhan pada kategori yang sama.
3. Mendapatkan nilai dalam tabel *mean*, standar deviasi dan probabilitas.

Hasil pengolahan dengan metode *Naive Bayes* dengan menggunakan 5 record data *testing*.

Tabel 3. 10 Hasil Pengolahan *Naive Bayes*

No	Data	Klasifikasi <i>Naive Bayes</i>
----	------	--------------------------------

1	Positif	Positif
2	Positif	Positif
3	Positif	Negatif
4	Negatif	Negatif
5	Negatif	Negatif

Hasil pengolahan dengan metode SVM dengan menggunakan 5 record data *testing*.

Tabel 3. 11 Hasil Pengolahan SVM

No	Data	Klasifikasi SVM
1	Positif	Positif
2	Positif	Positif
3	Positif	Positif
4	Negatif	Positif
5	Negatif	Negatif

3.10. Evaluasi

Pada tahap evaluasi akhir, penelitian ini menggunakan *Confusion Matrix* sebagai alat untuk melakukan pengujian dan evaluasi performa. *Confusion Matrix* merupakan teknik evaluasi yang digunakan untuk mengevaluasi kinerja dan efektivitas suatu metode klasifikasi. Ini memberikan visualisasi bagaimana metode klasifikasi memprediksi label dan membandingkan dengan label sebenarnya (Karsito, 2019). Pengujian dilakukan dengan menghitung *accuracy*, *recall*, *precision* ditampilkan dalam bentuk persentase, terdapat 4 istilah yaitu *True Positif (TP)*, *True Negatif (TN)*, *False Positif (FP)*, *False Negatif (FN)*. Tabel *Confusion Matrix* :

Tabel 3. 12 *Confusion Matrix*

Kelas	Positif	Negatif
Positif	<i>True Positif (TP)</i>	<i>False Negatif (FN)</i>
Negatif	<i>False Positif (FP)</i>	<i>True Negatif (TN)</i>

Keterangan untuk tabel diatas yaitu:

1. TP adalah *True Positif*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
2. FN adalah *False Negatif*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
3. FP adalah *False Positif*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem
4. TN adalah *True Negatif*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

Pengujian dilakukan dengan menghitung nilai *accuracy*, *precision*, dan *recall* lalu ditampilkan dalam bentuk *persentase*.

1. Accuracy

Accuracy merupakan mengetahui jumlah data yang diklasifikasikan secara benar. Berikut rumus perhitungan *Accuracy*.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (2)$$

2. Precision

Precision adalah ukuran yang mengukur kualitas dari hasil pencarian. *Precision* dihitung sebagai perbandingan jumlah data relevan yang ditemukan dibandingkan dengan jumlah data yang sebenarnya ditemukan

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

3. Recall

Jumlah data yang skala pada jumlah teks yang relevan dikalangan dokumen teks pada koleksi. Berikut rumus perhitungan *Recall*

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

3.10.1. Evaluasi Klasifikasi *Naïve Bayes*

Tabel 3. 13 Klasifikasi *Naïve Bayes*

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

Tabel 3. 14 Hasil Akurasi *Naïve Bayes*

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

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

Accuracy : 80.00%

Precision : 66.67%

Recall : 66.67%

3.10.2. Evaluasi Klasifikasi *Support Vector Machine*

Tabel 3. 15 Klasifikasi *Support Vector Machine*

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

Tabel 3. 16 Hasil Akurasi *Support Vector Machine*

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

Hasil akurasi dengan metode *Support Vector Machine* dengan menggunakan 5 *record data testing*.

Accuracy : 80.00%

Precision : 100.00%

Recall : 100.00%

Berdasarkan contoh evaluasi diatas, analysis sentiment 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 *Support Vector Machine* menghasilkan nilai *accuracy*, *precision*, dan *recall* tertinggi yaitu dengan menggunakan algoritma *Support Vector Machine* dengan *accuracy* 80.00%, *precision* 100.00% dan *recall* 100.00%.

