

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Objek dan Subjek Penelitian**

Objek penelitian merupakan suatu yang dijadikan fokus dalam melakukan suatu penelitian. Objek yang dijadikan fokus dalam penelitian ini adalah mengenai analisis sentimen vaksin Astrazeneca. Subjek dari penelitian ini adalah pengguna twitter yang menulis tweet terkait dengan vaksin Astrazeneca dalam rentang waktu Mei – Juni 2021.

#### **3.2 Alat dan Bahan**

Bahan penelitian yang digunakan dalam penelitian ini adalah data opini berbahasa Indonesia yang telah didapatkan dari sosial media twitter. Sedangkan alat yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

- a. Perangkat Keras (*Hardware*)

**Tabel 3.1 Spesifikasi Hardware**

<i>Hardware</i>	<b>Spesifikasi</b>
Laptop	ASUS X441UV
RAM	4 GB, SSD 128 GB
Grafik	NVIDIA GeForce 920MX
Processor	Intel Core i3

- b. Perangkat Lunak (*Software*)

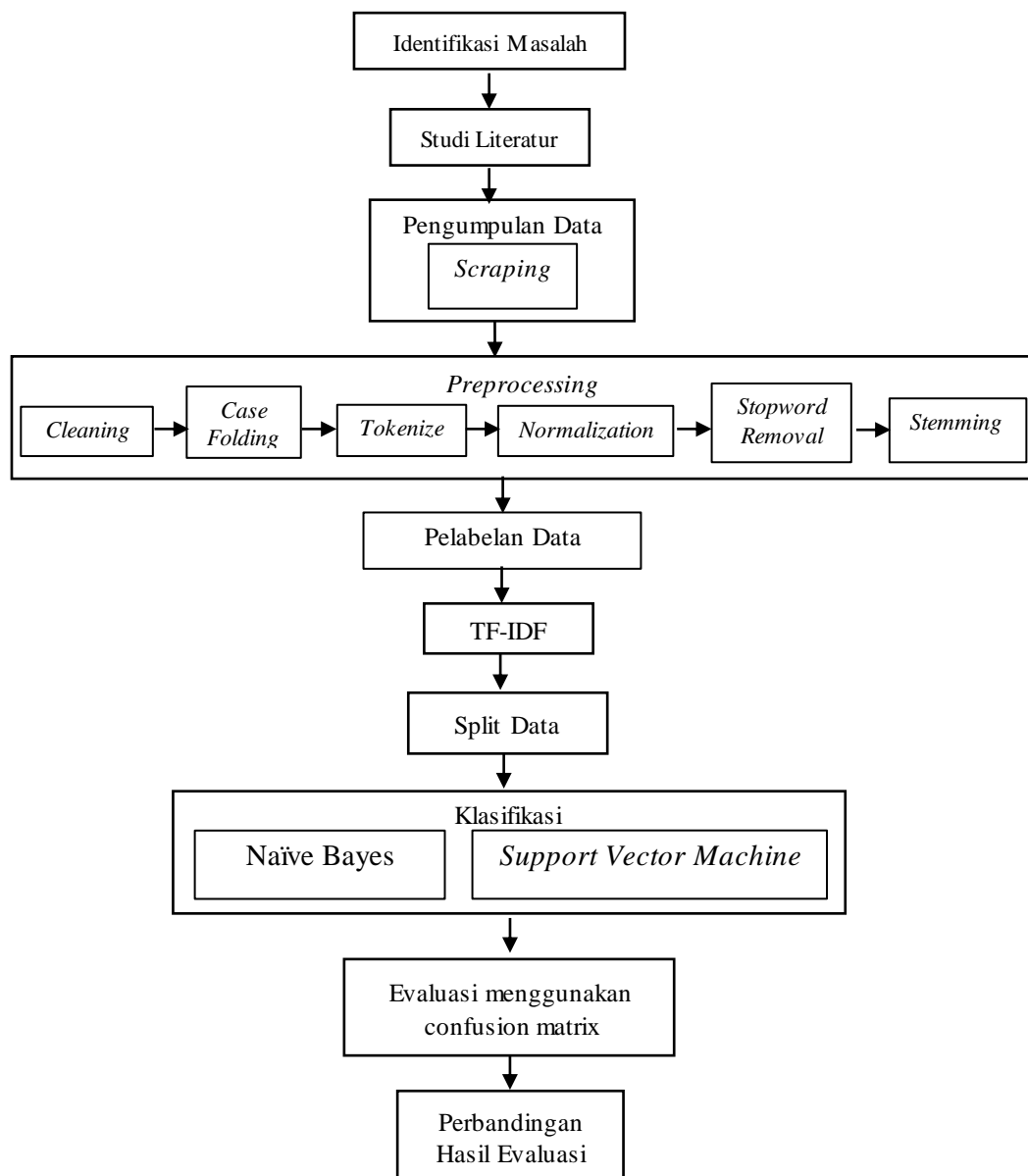
**Tabel 3.2 Software yang Digunakan**

<i>Software</i>
Sistem Operasi Windows 10 64 Bit
Bahasa Pemrograman Python 3.8
Google Colab
Snsrape
Matplotlib
Command Prompt
Pandas

<i>Software</i>
Nltk
Numpy
Sklearn

### 3.3 Diagram Alir Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini digambarkan melalui diagram yang ditunjukkan pada Gambar 3.1



**Gambar 3.1 Diagram Alir penelitian**

Dibawah ini adalah langkah-langkah dari diagram alir penelitian pada Gambar 3.1. Berikut penjelasannya :

1. Tahap pertama dari penelitian ini adalah dengan mengidentifikasi masalah dan merumuskan masalah penelitian. Setelah mengidentifikasi masalah kemudian dilakukan studi literatur yang berkaitan dengan penelitian yang akan dilakukan.
2. Mengambil data dari twitter dengan teknik *scraping* data.
3. Melakukan *preprocessing* yang berguna untuk menyeleksi data dan mengubahnya menjadi data yang terstruktur. Terdapat 4 tahap *preprocessing* yaitu *Cleaning*, *Case Folding*, *Tokenization*, *Normalization*, *Stopword Removal*, dan *Stemming*.
4. Setelah semua dataset melewati proses *preprocessing* selanjutnya data akan diberi label secara otomatis menggunakan kamus *lexicon* kedalam dua kategori berbeda yaitu positif atau negatif pada setiap dataset untuk memberi label dari setiap tweet.
5. Langkah kelima adalah ekstraksi fitur, dimana pada tahap ini dilakukan pembuatan fitur untuk mempermudah proses pembelajaran mesin Naïve Bayes dan *Support Vector Machine*.
6. Langkah selanjutnya yaitu klasifikasi menggunakan Naïve Bayes dan *Support Vector Machine*. Pada tahap ini dimulai pembuatan model untuk pengklasifikasian berdasarkan sentimen yang ada dalam dokumen.
7. Tahap selanjutnya adalah evaluasi dan validasi model, yaitu untuk mengukur nilai performa klasifikasi yang telah dilakukan dengan menghitung nilai *accuracy*, *precision*, *recall* dan *f1-score*.
8. Setelah mendapatkan hasil klasifikasi dari kedua metode, dilakukan perbandingan akurasi antara dua metode tersebut untuk melakukan perbandingan performa klasifikasi antara metode Naïve Bayes dan *Support Vector Machine*.

### 3.4 Uraian Tahapan Penelitian

#### 3.4.1 Identifikasi Masalah

Langkah pertama yang dilakukan pertama kali yaitu dengan mengidentifikasi dan merumuskan masalah yang akan diteliti. Dengan cara menentukan bidang, topik, masalah penelitian, serta metode yang akan digunakan dalam penelitian. Pada penelitian ini, ide atau masalah berawal dari bagaimana cara menganalisis sentimen terhadap opini tentang vaksin Astrazeneca di media sosial twitter.

#### 3.4.2 Studi Literatur

Melalui metode ini, peneliti mengumpulkan dan melakukan pembelajaran dari jurnal-jurnal, paper, artikel, *literature review*, serta buku yang berkaitan dengan analisis sentimen dengan metode *naïve bayes* dan *support vector machine* serta materi materi penunjang lainnya seperti tentang vaksin Astrazeneca, *scraping data*, *k-fold cross validation*, dan sebagainya

#### 3.4.3 Pengumpulan Data

Data yang diambil dalam penelitian ini adalah data tweet yang didapatkan dari pencarian tweet pada twitter. Data yang dikumpulkan berupa teks yang diambil dengan teknik *scraping* menggunakan *library* Snsrape. *Scraping* menggunakan *snsrape* ini dilakukan tanpa menggunakan akses API twitter sehingga lebih mudah digunakan. Dipilihnya Snsrape untuk teknik *scraping* karena *library* ini tidak memiliki limit jumlah tweet ataupun rentang waktu ketika melakukan pengambilan data [1].

Pengumpulan dataset yang digunakan pada penelitian ini adalah kumpulan opini berupa tweet terhadap topik vaksin astrazeneca berdasarkan pencarian kata kunci ‘vaksin astrazeneca’ pada periode bulan Mei sampai Juni 2021. Rentang tersebut dipilih karena pada bulan Mei sampai Juni vaksin Astrazeneca Indonesia menerima hibah vaksin Astrazeneca dari beberapa Negara. Selain itu, pada bulan Juni 2021 vaksin Astrazeneca juga sempat menjadi *trending topic* di twitter karena banyak

masyarakat yang beropini tentang vaksin Astrazeneca. Seluruh data tweet yang berhasil dikumpulkan kemudian dimuat dalam bentuk file JSON (*JavaScript Object Notation*) karena lebih mudah dalam mengelola data, format JSON juga kecil dalam ukuran.

#### 3.4.4 *Preprocessing Data*

Preprocessing data merupakan tahap dimana teks akan diseragamkan bentuk dan format agar dipersiapkan menjadi data yang dapat diolah pada tahap selanjutnya. Adapun tahap *preprocessing* meliputi *cleaning*, *case folding*, *tokenization*, *normalization*, *stopword removal*, dan *stemming*.

##### 1. *Cleaning*

Tahapan *cleaning* adalah tahap pembersihan kata dan atribut yang tidak berpengaruh sama sekali terhadap klasifikasi sentimen. Komponen tweet pada dasarnya memiliki berbagai atribut yang tidak berpengaruh terhadap penilaian sentimen. Beberapa contoh atribut pada tweet yang tidak diperlukan adalah URL, *mention username* yang diawali dengan atribut ('@'), tanda *hashtag* (#), angka, serta simbol atau karakter spesial selain abjad seperti tanda baca [39].

##### 2. *Case Folding*

Proses kedua dalam preprocessing adalah *case folding*. Pada tahap ini semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Perubahan dilakukan dengan memeriksa ukuran setiap karakter dari awal hingga akhir. Jika ditemukan karakter yang menggunakan huruf kapital, maka huruf tersebut akan diubah menjadi huruf kecil [40].

##### 3. *Tokenization*

Tahap *tokenization* ini bertujuan untuk memecah setiap kalimat menjadi kata per kata dengan menggunakan karakter spasi sebagai pemisah setiap kata. Langkah pertama, program akan membaca data tweet, kemudian data tersebut dipisah menjadi per kata yang

dihimpun pada *array*. Himpunan *array* ini selanjutnya akan digunakan pada proses berikutnya.

#### 4. *Normalization*

*Normalization* atau Normalisasi merupakan perubahan kata yang tidak baku seperti kata yang disingkat, kata gaul atau *slang*, serta kata yang berbahasa inggris yang kemudian diubah ke kata baku yang sesuai dalam KBBI. Database normalisasi berisi pasangan kata yang terdiri dari kata tidak baku beserta padanan kata bakunya. Program berjalan dengan cara mencari setiap kata yang tidak baku di dalam data kemudian menggantinya dengan padanan kata baku yang sudah ada [18]. Berikut ini adalah contoh kata yang dimuat dalam kamus normalisasi ditampilkan pada Tabel 3.3.

**Tabel 3.3 Daftar Kata Dalam Kamus Normalisasi**

Daftar Kata	Persamaan
cm	cuma
ga	tidak
yg	yang

Kemudian setiap kata akan dicocokkan dengan kata yang sudah tersimpan di dalam kamus yang selanjutnya akan disesuaikan dengan persamaan kata yang sesuai dengan KBBI.

#### 5. *Stopword Removal*

Tahap *stopword removal* yaitu tahap menghilangkan kata yang tidak sesuai dengan topik dokumen, jika terdapat kata tersebut maka kemungkinan akan mempengaruhi hasil akurasi klasifikasi. Pada tahap *stopword removal* digunakan library *nlk* (*natural language tool kit*) untuk mendapatkan daftar *stopword* dalam bahasa Indonesia [41]. Dibawah ini adalah contoh daftar *stopword* dalam bahasa Indonesia.

**Tabel 3. 4 Contoh *Stopword* Bahasa Indonesia**

<i>Stopword</i> Indonesia
ada
anu
dari
dong

#### 6. *Stemming*

*Stemming* adalah tahap mencari *root* (dasar) kata dari tiap kata dengan menghapus imbuhan yang ada di depan (*prefix*) misalnya imbuhan *me-* pada kata ‘melempar’, maupun di belakang kata (*suffix*), contohnya imbuhan *-kan* pada kata ‘buatkan’. [41].

#### 3.4.5 *Labeling Data*

Setelah data disimpan kemudian dilakukan pelabelan untuk menentukan pendapat dari berbagai tweet yang diambil. Pada proses ini, label dibagi menjadi dua kategori yaitu kelas positif atau kelas negatif. Proses pelabelan pada penelitian ini yang dilakukan secara otomatis menggunakan pendekatan *lexicon*. Pendekatan dengan *Lexicon-based* bekerja dengan menggunakan kamus *lexicon* yang dilengkapi dengan bobot pada setiap katanya sebagai sumber leksikal. Hasil dari metode ini berupa klasifikasi sentimen positif dan negatif.

Kamus *lexicon* yang digunakan dalam penelitian ini adalah InSet *Lexicon* berdasarkan penelitian Fajri di tahun 2017 [32] yang berjudul “Evaluasi daftar kata untuk sentimen analisis berbahasa Indonesia”. Kamus Inset ini digunakan karena sudah teruji cukup baik untuk analisis sentimen data berbahasa Indonesia. Hasil dari penelitian tersebut juga menyatakan bahwa dengan menggunakan kamus *lexicon* bahasa Indonesia Inset menunjukkan performa yang lebih baik daripada metode *lexicon* yang lainya seperti SentiWordNet, Liu Lex ataupun AFINN. InSet *Lexicon* terdiri dari dua kamus *lexicon* yaitu *lexicon* negatif yang berisi 6.609 kata

negatif dan *lexicon* positif yang berisi 3.609 kata positif. Masing masing kata memiliki bobot nilai atau *polarity score* dengan kisaran bobot antara -5 sampai +5. Nilai dari *Polarity score* ini yang nantinya akan digunakan untuk mengklasifikasikan jenis label pada setiap data apakah tweet tersebut bersifat negatif atau positif [42]. Contoh dari kata negatif dan positif beserta bobotnya yang termuat dalam kamus InSet *lexicon* dapat dilihat pada tabel berikut:

**Tabel 3.5 Contoh Daftar Kata Negatif pada InSet *Lexicon***

<b>Kata</b>	<b>Bobot</b>
buram	-3
kabur	-4
dilema	-3
nyesal	-5
azab	-5
kejam	-4

**Tabel 3.6 Contoh Daftar Kata Positif pada InSet *Lexicon***

<b>Kata</b>	<b>Bobot</b>
cinta	3
detail	2
mari	4
gemar	5
ceria	4
efisien	4

Hasil dari pelabelan menggunakan kamus *lexicon* akan berupa skor yang diperoleh dari total bobot dari setiap kata dalam daftar kamus *lexicon*.



### 3.4.6 Ekstraksi Fitur

Setelah semua data melewati tahap *preprocessing* dan pelabelan, maka langkah selanjutnya adalah pembuatan fitur untuk mempermudah proses klasifikasi menggunakan TF-IDF. TF-IDF adalah metode pembobotan dari hubungan suatu kata *term* terhadap istilah di setiap kalimat yang dianggap sebagai dokumen [43]. Contoh pemberian bobot setiap kata dalam suatu tweet dapat dilihat sebagai berikut:

Dokumen 1 : Indonesia, datang, vaksin, astrazeneca, dosis

Dokumen 2 : vaksin, pakai, astrazeneca, Alhamdulillah, tidak, ada, demam

Dari contoh dokumen 1 dan dokumen 2, berikut ini adalah langkah-langkah untuk menghitung TF-IDF secara manual adalah sebagai berikut:

1. Setiap *term* atau kata pada suatu dokumen akan dihitung frekuensi kemunculan nya dalam sebuah dokumen. Karena dokumen telah melewati tahap normalisasi sebelumnya maka persamaan untuk menghitung TF (*Term Frequency*) ditulis sebagai berikut :

$$tf_{i,d} = \frac{\text{frekuensi kemunculan term } t \text{ pada dokumen } d}{\text{Total term pada dokumen } d} \quad (3.1)$$

Tabel 3.7 dibawah ini menunjukkan hasil perhitungan TF pada tiap dokumen:

**Tabel 3.7 Kemunculan *term* pada suatu dokumen**

Term	TF		Skor TF Normalisasi	
	D1	D2	D1	D2
indonesia	1	0	0,2	0
datang	1	0	0,2	0
vaksin	1	1	0,2	0,142
astrazeneca	1	1	0,2	0,142
dosis	1	0	0,2	0
pakai	0	1	0	0,142
alhamdulillah	0	1	0	0,142

Term	TF		Skor TF Normalisasi	
	D1	D2	D1	D2
tidak	0	1	0	0,142
ada	0	1	0	0,142
demam	0	1	0	0,142

2. Setelah mendapatkan skor TF kemudian dilanjutkan dengan menghitung skor DF (*Document Frequency*) yaitu banyaknya dokumen dimana suatu term muncul, nilai DF dapat dilihat pada Tabel 3.8:

**Tabel 3.8 Tabel Nilai DF**

Term	DF
Indonesia	1
Datang	1
vaksin	2
Astrazeneca	2
Dosis	1
Pakai	1
Alhamdulillah	1
Tidak	1
Ada	1
Demam	1

3. Tahap selanjutnya dilanjutkan dengan menghitung skor IDF (*Inverse Document Frequency*) menggunakan persamaan (2.1) yang dapat dilihat pada Tabel 3.9

**Tabel 3.9 Tabel Skor IDF**

<b>Term</b>	<b>DF</b>	<b>Nilai Idf</b>
Indonesia	1	$Log\left(\frac{2}{1}\right) = 0.301$
Datang	1	$Log\left(\frac{2}{1}\right) = 0.301$
vaksin	2	$Log\left(\frac{2}{2}\right) = 0$
Astrazeneca	2	$Log\left(\frac{2}{2}\right) = 0$
Dosis	1	$Log\left(\frac{2}{1}\right) = 0.301$
Pakai	1	$Log\left(\frac{2}{1}\right) = 0.301$
Alhamdulillah	1	$Log\left(\frac{2}{1}\right) = 0.301$
Tidak	1	$Log\left(\frac{2}{1}\right) = 0.301$
Ada	1	$Log\left(\frac{2}{1}\right) = 0.301$
Demam	1	$Log\left(\frac{2}{1}\right) = 0.301$

4. Setelah nilai TF dan IDF diperoleh, maka dapat dihitung nilai TF-IDF dengan mengalikan skor TF dengan skor IDF, skor TF-IDF dari contoh dokumen dapat dilihat pada Tabel 3.10

**Tabel 3.10 Skor TF-IDF**

<b>Term</b>	<b>TF</b>		<b>IDF</b>	<b>TF*IDF</b>	
	<b>D1</b>	<b>D2</b>		<b>D1</b>	<b>D2</b>
Indonesia	0.2	0	0.301	0.060	0
Datang	0.2	0	0.301	0.060	0
vaksin	0,2	0.142	0	0	0
Astrazeneca	0.2	0.142	0	0	0
Dosis	0.2	0	0.301	0.060	0
Pakai	0	0.142	0.301	0	0.430
Alhamdulillah	0	0.142	0.301	0	0.430

Term	TF		IDF	TF*IDF	
	D1	D2		D1	D2
Tidak	0	0.142	0.301	0	0.430
Ada	0	0.142	0.301	0	0.430
Demam	0	0.142	0.301	0	0.430

### 3.4.7 Membangun Model Klasifikasi

Untuk melakukan klasifikasi sentimen, digunakan data hasil proses preprocessing sampai dengan TF-IDF. Sebelum membuat model klasifikasi, dataset dibagi menjadi dua bagian yaitu data latih dan data uji. Perbandingan antara data latih dan data uji adalah sebanyak 80 dibanding 20. Pembagian data dengan rasio 80:20 dikategorikan sebagai *starting point* yang bagus karena pembagian yang adil. Semakin banyak data latih yang dipakai biasanya akan menghasilkan akurasi yang lebih besar, namun jika data uji semakin sedikit maka varian sampel data uji random yang digunakan semakin rendah yang artinya tidak dapat merepresentasikan data secara keseluruhan. Rasio 80:20 merupakan rasio yang tepat karena sesuai dengan hukum Pareto (hukum 80:20) yang menyatakan bahwa untuk banyak kejadian, sekitar 80% dari efeknya disebabkan oleh 20% dari penyebabnya [44]. Data latih ini selanjutnya akan digunakan pada tahap klasifikasi. Sedangkan data uji digunakan untuk evaluasi metode klasifikasi.

#### a. Klasifikasi Naïve Bayes

Metode Klasifikasi Naïve Bayes merupakan salah satu metode *supervised learning* untuk mengklasifikasikan data dengan cara pendekatan peluang atau probabilitas [45]. Berikut ini langkah – langkah dalam perhitungan yang menggunakan Naïve Bayes adalah :

##### 1. Membentuk fitur

Fitur yang dimaksud dalam penelitian ini adalah yang menjadi parameter dalam data latih, yaitu data tweet yang diklasifikasikan ke

dalam dua kelas label yaitu positif yang bernilai 0 dan negatif yang bernilai 1.

**Tabel 3.11 Contoh Dokumen Untuk Perhitungan Naïve Bayes**

Dokumen	Fitur (kemunculan)	Label
D1	alhamdulillah(1), tenang(1), mbak(1), vaksin(1), pakai(1), astrazeneca(1)	0
D2	sesal(1), suntik(1), vaksin(1), astrazeneca(1)	1

## 2. Menghitung Probabilitas Kelas/Label

Setelah fitur dari setiap term telah terbentuk, kemudian dihitung probabilitas dari setiap kelas dengan rumus dari Persamaan (2.3).

**Tabel 3.12 Contoh Perhitungan Probabilitas Kelas**

Kelas/Label	Data		fd(ci)	p(ci)
	D1	D2		
Kelas 0 (Positif)	1	0	1	½
Kelas 1 (Negatif)	0	1	1	½

## 3. Menentukan Probabilitas setiap Fitur

Setelah mendapat probabilitas dari tiap kelas, maka selanjutnya dihitung juga probabilitas setiap kata pada setiap kelas, dengan menggunakan persamaan (2.4)

**Tabel 3.13 Contoh Perhitungan Probabilitas Kata**

Data $f(w_{ki}, c_i)$	Kelas	
	Kelas 0 (Positif)	Kelas 1 (Negatif)
alhamdulillah	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{0+1}{4+8} = \frac{1}{12}$
tenang	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{0+1}{4+8} = \frac{1}{12}$
mbak	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{0+1}{4+8} = \frac{1}{12}$
vaksin	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{1+1}{4+8} = \frac{1}{12}$

Data $f(w_{ki}, c_i)$	Kelas	
	Kelas 0 (Positif)	Kelas 1 (Negatif)
pakai	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{0+1}{4+8} = \frac{1}{12}$
astrazeneca	$\frac{1+1}{6+8} = \frac{2}{14}$	$\frac{1+1}{4+8} = \frac{2}{12}$
sesal	$\frac{0+1}{6+8} = \frac{1}{14}$	$\frac{1+1}{4+8} = \frac{2}{12}$
suntik	$\frac{0+1}{6+8} = \frac{1}{14}$	$\frac{1+1}{4+8} = \frac{2}{12}$

4. Tahapan setelah melakukan uji data adalah sebagai berikut:

Langkah terakhir yaitu dengan menghitung  $V_{map}$  menggunakan persamaan (3.2)

$$V_{map} = \underset{\{kelas1, kelas\ 0, kelas\ 2\}}{argmax} \prod_{i=1}^n P(w_k | c_i) x P(c_i) \quad (3.2)$$

Sampel data : sesal suntik vaksin astrazeneca

$$1) V_{map} (\text{"kelas 0"}) = P(\text{"sesal"} | \text{kelas 0}) P(\text{"suntik"} | \text{kelas 0})$$

$$P(\text{"vaksin"} | \text{kelas 0}) P(\text{"astrazeneca"} | \text{kelas 0})$$

$$= \frac{1}{14} \times \frac{1}{14} \times \frac{2}{14} \times \frac{2}{14} = 0.000104$$

$$2) V_{map} (\text{"kelas 1"}) = P(\text{"sesal"} | \text{kelas 1}) P(\text{"suntik"} | \text{kelas 1})$$

$$P(\text{"vaksin"} | \text{kelas 0}) P(\text{"astrazeneca"} | \text{kelas 1})$$

$$= \frac{2}{12} \times \frac{2}{12} \times \frac{1}{12} \times \frac{2}{12} = 0.000385$$

Setelah menghitung  $V_{map}$ , dilakukan perbandingan dari  $V_{map}$  kelas 0 (positif) dan  $V_{map}$  1 (negatif) dan ditentukan mana yang lebih besar. Pada perhitungan diatas menyatakan bahwa sampel data termasuk kelas 1, sehingga dapat disimpulkan bahwa tweet tersebut diklasifikasikan dalam sentimen negatif [46].

## b. Klasifikasi Support Vector Machine

Cara kerja klasifikasi menggunakan Algoritma *Support Vector Machine* (SVM) adalah sebagai berikut:

1. Pembagian data menjadi data latih dan data uji. Data latih digunakan untuk proses pelatihan dalam mencari nilai  $w$  dan  $b$  untuk digunakan pada fungsi keputusan (*classifier*). Sedangkan data uji digunakan untuk proses pengujian yang nantinya akan menghasilkan klasifikasi data apakah positif atau negatif [47].
2. Menentukan fungsi kernel yang akan digunakan dalam model hyperplane SVM. Kernel yang akan digunakan dalam penelitian ini adalah kernel linear. Pemilihan kernel linear dilakukan karena kernel linear adalah kernel terbaik untuk data teks [41]. Selain itu, dari penelitian yang dilakukan oleh Rachmad [18] juga menunjukkan bahwa dengan menggunakan kernel linear hasil akurasi klasifikasi yang didapatkan lebih tinggi daripada kernel polynomial. Fungsi kernel linear dinyatakan dalam persamaan berikut:

$$K(x_i, x_j) = (x_i, x_j) \quad (3.3)$$

Fungsi kernel diatas akan menghasilkan matriks kernel. Setiap elemen matriks kernel  $K(x_i, x_j)$  digunakan untuk menggantikan *dot-product*  $(x_i \cdot x_j)$  dalam persamaan dualitas Lagrange:

$$Ld = \sum_{i=1}^N a_i - \sum_{i=1}^N \sum_{j=1}^N a_i a_j y_i y_j K(x_i, x_j) \quad (3.4)$$

Dimana:

$Ld$  = dualitas langrange *multiplier*

$a_i$  = nilai bobot setiap titik data

$C$  = nilai konstanta

3. Hitung nilai  $w$  dan  $b$  melalui persamaan (3.4) dan (3.5).

$$w = \sum_{i=1}^N a_i y_i x_i \quad (3.5)$$

$$b = -\frac{1}{2}(w \cdot x + w \cdot x) \quad (3.6)$$

4. Hitung fungsi keputusan klasifikasi  $\text{sign}(f(x))$  berdasarkan persamaan (3.6)

$$f(x) = w \cdot x + b \quad (3.7)$$

Atau bisa menggunakan persamaan (3.7)

$$F(x) = \sum_{i=1}^m a_i y_i K(x_i, x_j) + b \quad (3.8)$$

Dimana:

$m$  = adalah jumlah titik data yang memiliki  $a_i > 0$

$K(x_i, x_j)$  = adalah fungsi kernel yang digunakan

$x_i$  = Data yang ke- $i$

$x_j$  = Data yang ke- $j$

$y_i$  = Kelas pada data ke- $i$

### 3.4.8 Evaluasi dan Validasi Model Klasifikasi

Evaluasi dilakukan untuk menguji hasil klasifikasi dengan mengukur nilai kinerja yang dilakukan oleh sistem. Pengukuran kinerja klasifikasi diperlukan untuk menggambarkan seberapa baik metode dalam klasifikasi data. Parameter pengujian yang digunakan untuk mengevaluasi adalah nilai akurasi atau kebenaran tingkat proses klasifikasi yang perhitungannya diperoleh dari tabel matriks. Pada tahap evaluasi, digunakan metode *confusion matrix*. Dari *confusion matrix* akan diketahui nilai performa dari metode klasifikasi yang terdiri dari *accuracy*, *precision*, *recall* serta *f1-score* [48].

Untuk pembuatan validasi model, dilakukan dengan metode *10-fold cross validation*, yaitu metode yang membagi seluruh data menjadi 10 potongan dimana 20% dataset dijadikan sebagai data uji dan sisanya menjadi data latih. Sehingga dilakukan 10 kali pembuatan model dengan proporsi 80% data latih dan 20% data uji secara acak bergiliran [20].