BAB IV

ANALISIS

4.1 ANALISIS MASALAH

Twitter adalah salah satu *microblogging* yang sangat popular di tengah masyarakat. Biasanya Twitter digunakan sebagai sarana untuk menyampaikan suatu informasi. Informasi yang terkandung pada *tweets* ini sangat berharga sebagai alat penentu kebijakan. Salah satunya adalah untuk menilai kebijakan yang dikeluarkan oleh pemerintah atau yang sering disebut dengan analisis sentimen atau *opinion mining*.

Pemerintah dapat memanfaatkan salah satu *microblogging* ini sebagai media untuk melihat tanggapan masyarakat pada setiap kebijakannya. Tanggapan dari masyarakat biasanya berupa opini-opini untuk menanggapi setiap kebijakan yang dibuat. Hal ini sangat berguna bagi pemerintah dalam meninjau kembali kebijakannya. Opini dari masyarakat itu nantinya akan dilakukan tahapan analisis sentimen apakah termasuk opini positif atau opini negatif. Namun tahapan untuk analisis sentimen ini ada tantangan berupa bentuk bahasa tidak formal yang digunakan para pengguna Twitter. Maka dari itu, sebelum melakukan analisis sentimen, harus dilakukan *text processing* pada setiap data *tweets* yang akan digunakan. Hal ini berguna untuk mengatasi bentuk bahasa yang tidak formal yang sering digunakan pengguna Twitter. Selain itu, pengklasifikasian sentimen saat ini masih dilakukan dengan cara manual oleh manusia. Permasalahan ini berdampak pada kualitas dan kecepatan dalam menganalisis sentimen dengan data yang sangat

banyak. Maka dari itu, penerapan *text mining* untuk melakukan analisis sentimen secara otomatis merupakan salah satu solusi untuk mengatasi masalah ini.

4.2 ANALISIS SUMBER DATA

Data yang digunakan pada penelitian ini diambil dari kumpulan *tweets* bahasa Indonesia yang diambil dari *hashtag* #IbuKotaPindah. Data *tweets* ini diperoleh dengan cara *crawling* yang menggunakan API Twitter. Dalam proses *crawling*, secara otomatis akan mengambil data *tweets* yang mengandung kata "#IbuKotaPindah", "Ibu Kota Pindah", dan "Pemindahan Ibu Kota". Data *tweet*s yang terkumpul nantinya akan dilakukan tahap *preprocessing* teks dan selanjutnya akan diklasifikasikan. Dalam analisis sentimen ini, *tweets* akan diklasifikasikan ke dalam dua bentuk sentimen, yait sentimen positif dan sentimen negatif. Contoh data *tweets* yang dipakai pada penelitian ini dapat dilihat pada gambar 4.1 berikut.

Gambar 4.1 Contoh *Tweets*

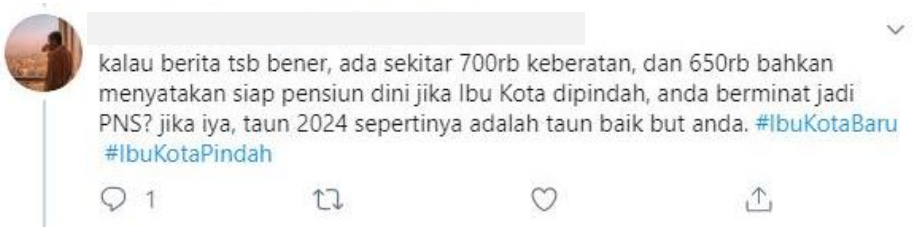
Data yang dibutuhkan pada penelitian ini terdiri dari dua jenis, yaitu data *training* dan data *testing*. Data *training* diambil dari kumpulan *tweets* yang telah dilabeli dengan kelas sentimennya secara manual. Data inilah yang digunakan sebagai data *training* untuk model analisis sentimen. Model ini nantinya akan digunakan untuk mengklasifikasikan *tweets* pada kelas sentimennya. Pada

penelitian ini, metode untuk pengklasifikasian menggunakan algoritma *Naïve Bayes Classifier*. Sebagian data *tweets* dari hasil *crawling*, nantinya akan digunakan sebagai data *testing*. Data *testing* yang digunakan berupa kumpulan *tweets* yang belum memiliki label.

Setiap pengguna memiliki cara dalam penulisan *tweets*. Dari hasil pencarian yang dilakukan pada *hashtag* #IbuKotaPindah, terdapat berbagai macam karakteristik dalam penulisan *tweets*, seperti:

1. Penyingkatan kata.

   Adanya keterbatasan karakter yang dapat ditulis dalam suatu *tweets* membuat pengguna Twitter mempersingkat kata dalam penulisan *tweets*. Contohnya dapat dilihat pada Gambar 4.2



Gambar 4.2 Penyingkatan Kata

2. Penggunaan titik (.) atau koma (,) pada akhiran *tweets*

   Ada beberapa pengguna yang biasa menggunakan lebih dari satu titik (.) ataupun koma (,) pada akhiran *tweets*. Contohnya seperti pada Gambar 4.3

Gambar 4.3 Penggunaan Titik Koma Pada *tweets*

3. Penggunaan *emoticon*

   Dalam pembuatan *tweets*, ada beberapa pengguna yang menggunakan *emoticon* untuk menyatakan sentimennya. Contohnya dapat dilihat pada gambar 4.4 berikut.

Gambar 4.4 Penggunaan *Emoticon*

4.3 KLASIFIKASI SENTIMEN MANUAL

Pembentukan nilai sentimen pada sebuah *tweets* membutuhkan ilmu dari seorang ahli bahasa dan komunikasi, hal ini berguna untuk melakukan *comparasion* antara analisa mesin dan ahli bahasa. Namun pada proses penelitian ini klasifikasi teks secara manual dilakukan oleh pegawai kantor bahasa dengan latar belakang ilmu bahasa dan komunikasi. Kelas sentimen yang di hasilkan berupa data sebanyak 400 *tweets* dengan dua label sentimen, yaitu positif dan negatif. Contoh *tweets* yang telah dibeli label sentimen dapat dilihat pada tabel 4.1 berikut

Tabel 4.1 Klasifikasi Manual

Tweets	**Label**
buat indonesia yang lebih baik	positif
presiden lagi ngibulin rakyat	negatif
menambah suram kalau ibu kota pindah	negatif
wacana ibu kota pindah sangat menggembirakan sekali	positif

4.4 ANALISIS *TEXT PREPROCESSING*

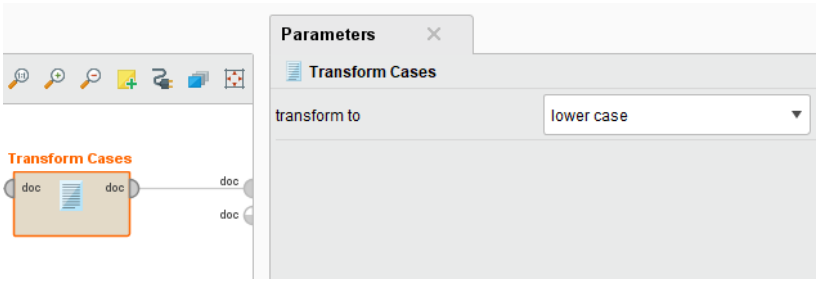
Text processing merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk lebih efektif dalam proses dilakukan langkah transformasi data ke dalam suatu format yang memudahkan untuk kebutuhan pemakai. *Preprocessing* merupakan salah satu langkah yang penting dalam analisis sentimen. Sama halnya *preprocessing* pada *Information Retrieval* (IR), tahapannya terdiri dari *tokenizing*, normalisasi fitur, *case folding*, *stopword removal* dan *stemming*. Namun pada tahap *preprocessing* analisis sentimen, ada beberapa tambahan seperti *convert emoticon* dan *convert negation*. Tahapan dari preprocessing adalah sebagai berikut

1. *Case Folding*

   Pada tahap ini, semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Berikut merupakan tahapan *case folding* dalam contoh salah satu contoh *tweets*: "Kalimantan sudah begini ,berangkat kerja baju bau asap sesak what happened ?? Habittatku :(Kalimantan :(#SaveKalimantan #IbuKotaPindah".

   a. Memeriksa ukuran setiap karakter dari awal sampai akhir karakter.
   b. Jika ditemukan karakter yang menggunakan huruf kapital atau *uppercase*, maka huruf tersebut akan diubah menjadi huruf kecil atau *lowercase*. Tahapan dan hasil *case folding* dapat dilihat pada gambar 4.5 dan gambar 4.6

Pada tahapan ini *case folding* dilakukan dengan memanfaatkan operator *transform case* pada Rapidminer, berikut contoh proses *case folding*.



Gambar 4.5 Tahapan *Case Folding*

Input :

Wacana Ibu Kota pindah sangat menggembirakan sekali #IbuKotaPindah

⇩

Output :

wacana ibu kota pindah sangat menggembirakan sekali #ibukotapindah

Gambar 4.6 Hasil *Case Folding*

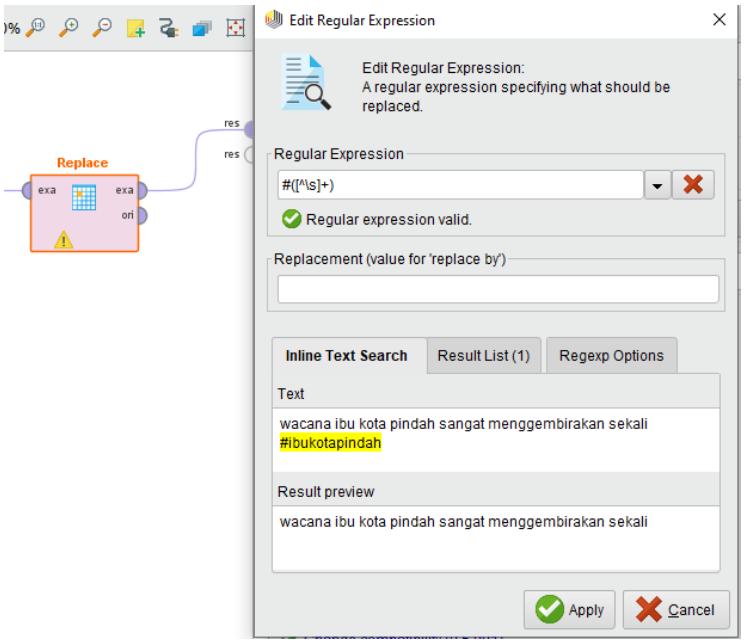
2. Normalisasi Fitur

Di setiap *tweet*s ada beragam macam ciri khas seperti *username*, URL (*Uniform Resource Locator*), "RT" (tanda *retweet*) dan *hashtag* atau tagar (#) . Komponen seperti *username*, URL, dan "RT" tidak mempunyai pengaruh apapun terhadap nilai sentimen, maka ketiga komponen tersebut bisa dihilangkan. Untuk *username* ditandai dengan kemunculan karakter '@'. Selain *username*, karakter '@' biasa juga digunakan untuk pemanggilan suatu tempat seperti @MonasJakarta. Namun nama tempat juga tidak akan berpengaruh pada analisis sentimen sehingga nama tempat

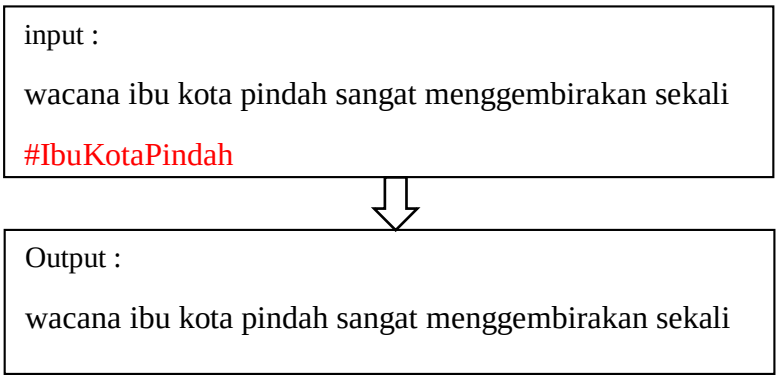
pun harus dihilangkan. Untuk komponen seperti *hashtag* dapat dikenali dengan kemunculan karakter #. Pada komponen URL dikenali dengan ciri *regular expression* (seperti http, www). Berikut langkah-langkah pada tahap normalisasi fitur:

a. Kata yang digunakan hasil dari *case folding*.
b. Hasil dari *case folding* akan diperiksa apakah terdapat *username*, URL, *Hashtag* dan RT.
c. Jika terdapat username, URL, *Hashtag* dan RT maka akan dihilangkan.

Pada proses ini, normalisasi dilakukan dengan cara memanfaatkan operator *replace* dan mengatur *attribute filter* ke *regular_expression*. Untuk tahapan dan hasil normalisasi dapat dilihat pada gambar 4.7 dan 4.8 berikut :



Gambar 4.7 Tahapan Normalisasi Fitur



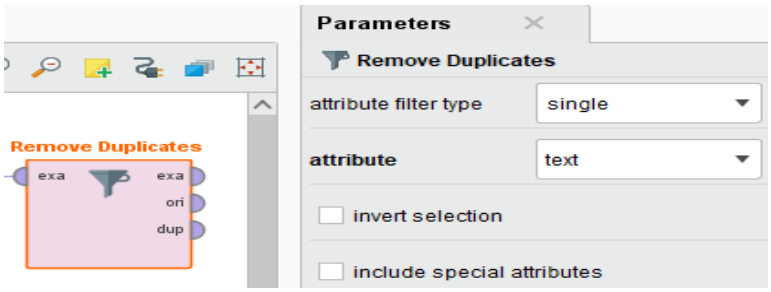
Gambar 4.8 Hasil Normalisasi Fitur

3. *Remove Duplicate*

Dengan adanya fitur *retweet* di Twitter tentu memberikan dampak besar untuk perulangan *tweets* dengan isi yang sama tetapi di posting dengan pengguna yang berbeda. Hal ini menyebabkan terjadinya redudansi data. Penelitian ini memanfaatkan operator *remove duplicates* untuk menangani kasus duplikasi *tweets* . Contoh tahapan dan hasil proses *remove duplicate* dapat dilihat pada gambar 4.9 dan tabel 4.3

Tabel 4.2 Contoh *Tweets Duplicates*

ID	*Text*
1123405485552520000	wacana ibu kota pindah sangat menggembirakan sekali
1133404261238783000	wacana ibu kota pindah sangat menggembirakan sekali



Gambar 4.9 Tahapan *Remove Duplicates*

Tabel 4.3 Hasil Tahapan *Tweets Duplicates*

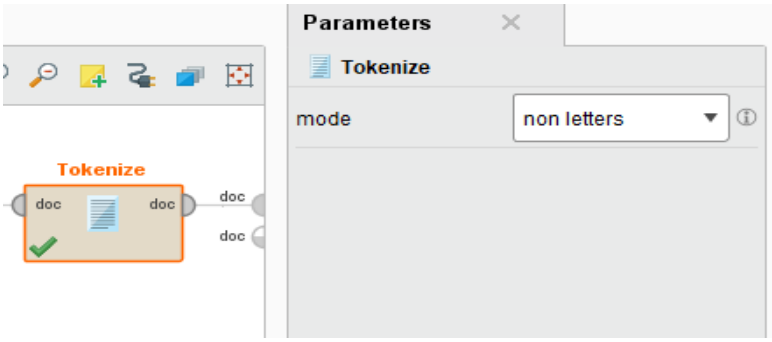
ID	***Text***
1133404261238783000	wacana ibu kota pindah sangat menggembirakan sekali

4. Tokenizing

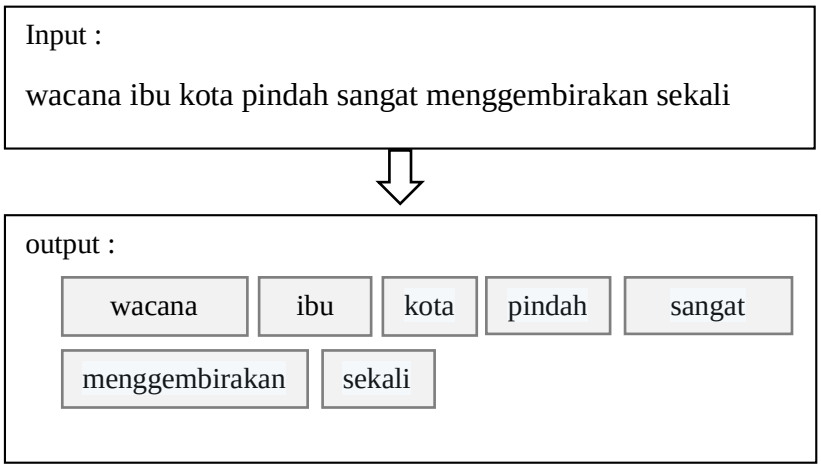
Tahapan *tokeninzing* merupakan tahap yang berfungsi untuk memisahkan antara kata yang terdapat pada *tweets*. Proses *tokenizing* menggunakan spasi sebagai karakter untuk tanda pemisah setiap katanya. Setiap *tweets* terdiri dari beberapa kata yang saling terhubung dan dipisahkan dengan spasi. Untuk mempermudah dalam pemerosesan sebuah teks maka masing-masing kata dalam sebuah kalimat tersebut harus dipisahkan. Apabila karakter ke-n bukan tanda pemisah kata seperti titik(.), koma(,), spasi dan yang lainnya, maka akan digabungkan dengan karakter selanjutnya. Berikut langkah pada tahap *tokenizing* :

a. Kata yang digunakan adalah hasil dari proses *convert negation*.
b. Memotong setiap kata dalam teks berdasarkan pemisah kata seperti titik(.), koma(,), dan spasi.
c. Bagian yang hanya memiliki satu karakter non alfabet dan angka akan dibuang.

Proses *tokenizing* dilakukan dengan cara memanfaatkan operator *tokenize* pada Rapidminer, contoh tahapan dan hasil *tokenizing* dapat dilihat pada gambar 4.10 dan gambar 4.11 berikut :



Gambar 4.10 Tahapan *Tokenizing*



Gambar 4.11 Hasil *Tokenizing*

5. *Stopwords Removal*

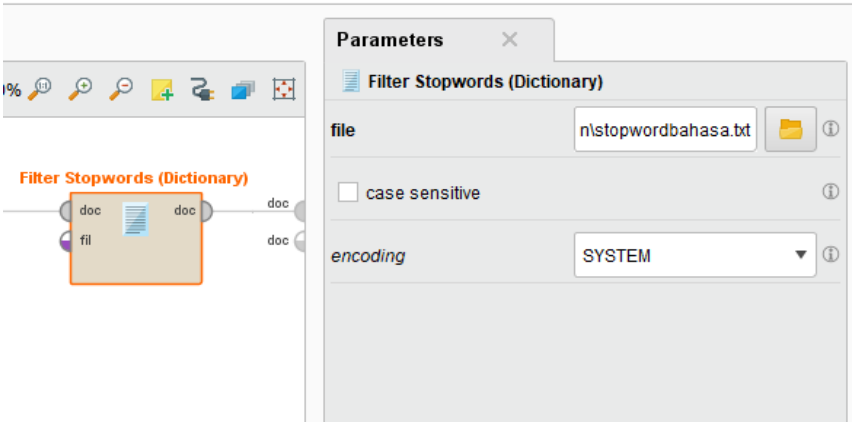
Pada tahapan *stopword removal*, kumpulan *tweets* yang telah melalui proses *tokenzing* akan dilanjutkan ke proses *stopword removal*. Setiap kata yang terdapat pada *tweets* akan diperiksa. Jika *tweets* tersebut mengandung kata sambung, kata depan, kata ganti atau kata yang tidak ada hubungannya dengan analisis sentimen, maka kata tersebut akan dihapus. Berikut tahapan pada proses *stopword removal* :

a. Kata dari hasil proses *tokenizing* akan dibandingkan dengan daftar *stopwords*.

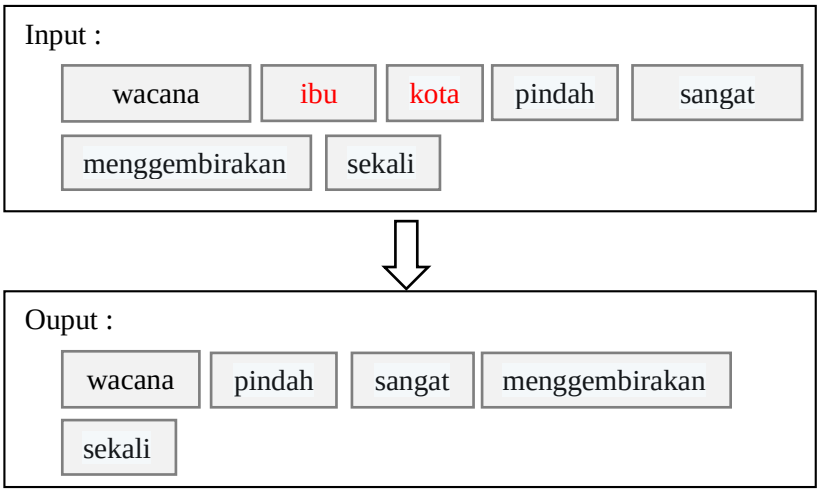
b. Dilakukannya pengecekan apakah ada kata yang sama dengan daftar *stopword* atau tidak.

c. Jika ada kata yang sama dengan yang ada pada daftar *stopword*, maka akan dihilangkan.

Proses *stopword removal* dilakukan dengan cara memanfaatkan operator *Filter Stopwords (dictionary)* dan memanfaatkan *file stopwords* yang telah di *download* pada akun Github riochr17, contoh proses tahapan dan hasil *stopwords* dapat dilihat pada gambar 4.12 dan gambar 4.13 berikut :



Gambar 4.12 Tahapan *Stopwords Removal*



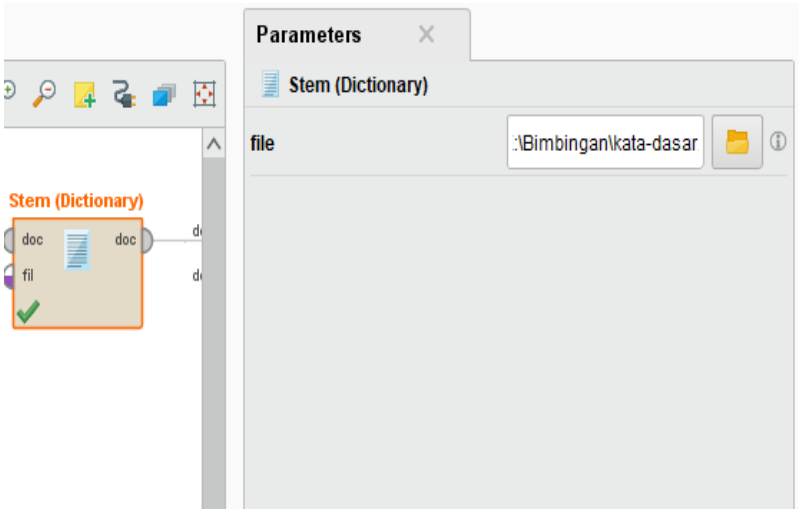
Gambar 4.13 Hasil *Stopwords Removal*

6. *Stemming*

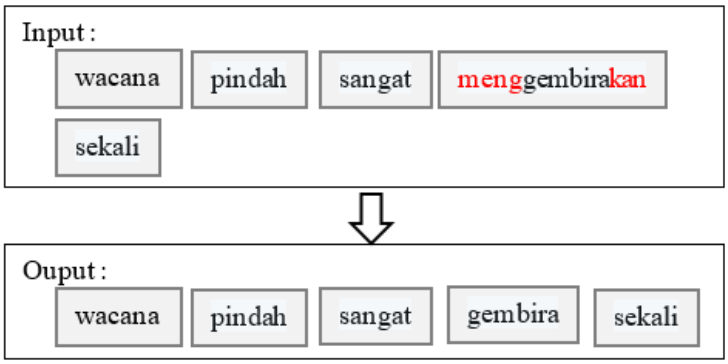
Setiap kata yang muncul di dalam *tweets* sering mempunyai banyak varian morfologik. Oleh karena itu, setiap kata-kata direduksi ke bentuk *stemmed word* (*term*) yang cocok. Kata-kata tersebut akan diambil bentuk kata dasarnya dengan cara menghilangkan awalan atau akhiran. Langkah-langkah pada tahap *stemming* adalah sebagai berikut:

a. Kata yang digunakan pada proses *stemming* adalah kata dari hasil *stopword removal*.
b. Setiap kata dalam *tweets* akan diperiksa dari awal hingga akhir kata.
c. Jika terdapat kata yang mengandung imbuhan, maka imbuhan pada kata tersebut akan dihapus.

Proses *stemming* dilakukan memanfaatkan operator *stem (dictionary)* dan memilih *file* kata dasar yang telah di *download* pada akun Github emaknyus, contoh proses tahapan dan hasil *stemming* dapat dilihat pada gambar 4.14 dan gambar 4.15 berikut :



Gambar 4.14 Tahapan *Stemming*



Gambar 4.15 Hasil *Stemming*

4.5 ANALISIS ALGORITMA *NAÏVE BAYES CLASSIFIER*

Pada proses klasifikasi dengan algoritma *Naïve Bayes Classifier* dibagi ke dalam dua proses, yaitu *training* dan *testing*. Pada tahap *training* digunakan untuk menghasilkan model analisis sentimen yang nantinya akan digunakan sebagai acuan untuk mengklasifikasi sentimen dengan data *testing*. Berikut adalah algoritma klasifikasi sentimen menggunakan *Naïve Bayes Classifier* :

1. Tahapan *training*
   a. Hitung p (c_i)
   b. Hitung p ($w_k \mid c_i$) untuk setiap kata w_k pada model
2. Tahapan *testing*
   a. Hitung Πk p ($w_k \mid c$) × p (c) untuk setiap kategori
   b. Tentukan kategori dengan nilai Πk p ($w_k \mid c$) × p (c) maksimal .

Berikut contoh klasifikasi *tweets* menggunakan *Naïve Bayes Classifier* :

1. Proses *training*

   Sebuah *tweets training* yang diklasifikasikan secara manual dan dilakukan proses *text processing* sebelumnya adalah sebagai berikut:

a. *Tweets* yang termasuk ke dalam sentimen positif :

*Tweets*1 : buat lebih baik

*Tweets*2 : wacana sangat gembira sekali

b. *Tweets* yang termasuk ke dalam sentimen negatif :

Tweets3 : ngibul rakyat

*Tweets*4 : tambah suram

Gambaran dari kasus diatas dapat dilihat pada tabel 4.4 berikut:

Tabel 4.4 Data *Training*

Tweets	**Label**	***Tweets***
1	Positif	buat lebih baik
2	Positif	wacana sangat gembira sekali
3	Negatif	ngibul rakyat
4	Negatif	tambah suram

Pada tabel diatas, dapat dibentuk model probabilitas sebagai berikut :

$P(w_{gembira} \mid c_{positif}) = \frac{\#Words\ (W_{gembira}, C\ positif) + 1}{\#Words\ (C_{positif,}) + |V|} = \frac{1+1}{7+11} = \frac{2}{18}$

$P(w_{gembira} \mid c_{negatif}) = \frac{\#Words\ (W_{gembira}, C\ negatif) + 1}{\#Words\ (C_{positif,}) + |V|} = \frac{0+1}{4+11} = \frac{1}{15}$

Jika dimasukkan ke sebuah tabel, maka hasil dari perhitungan probabilistik setiap kata pada data *training* terdapat pada tabel 4.5 berikut

Tabel 4.5 Perhitungan Probabilitas Data *Training*

Label		Positif	Negatif
P(c)		$\frac{1}{2}$	$\frac{1}{2}$
$P(W_{kj} \mid C)$	buat	$\frac{2}{18}$	$\frac{1}{15}$

	lebih	$\frac{2}{18}$	$\frac{1}{15}$
	baik	$\frac{2}{18}$	$\frac{1}{15}$
	wacana	$\frac{2}{18}$	$\frac{1}{15}$
	sangat	$\frac{2}{18}$	$\frac{1}{15}$
	gembira	$\frac{2}{18}$	$\frac{1}{15}$
	sekali	$\frac{2}{18}$	$\frac{1}{15}$
	ngibul	$\frac{1}{18}$	$\frac{2}{15}$
	rakyat	$\frac{1}{18}$	$\frac{2}{15}$
	tambah	$\frac{1}{18}$	$\frac{2}{15}$
	suram	$\frac{1}{18}$	$\frac{2}{15}$

Hasil dari perhitungan probabilitas tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan label data *testing*. Contoh kasus untuk data *testing* yang sudah melalui tahap *text processing* sebelumnya terdapat pada tabel 4.6 berikut :

Tabel 4.6 Data *Testing*

Tweets	**Label**	***Tweets***
5	?	wacana tambah suram
6	?	rakyat sangat gembira

Untuk penentuan label pada *tweets* 5 sebagai data *testing* 1 menggunakan perhitungan sebagai berikut :

P(*tweets*_5 | $c_{positif}$)

= $(w_{wacana}|\ C_{positif}) \times (w_{tambah}|\ C_{positif}) \times (w_{suram}|\ C_{positif}) \times P(C_{positif})$

= $\frac{2}{18} \times \frac{1}{18} \times \frac{1}{18} \times \frac{1}{2}$

= **0,0001714677**

P(*tweets*_5 | $c_{negatif}$)

= $(w_{wacana}|\ C_{negatif}) \times (w_{tambah}|\ C_{negatif}) \times (w_{suram}|\ C_{negatif}) \times P(C_{negatif})$

= $\frac{1}{15} \times \frac{2}{15} \times \frac{2}{15} \times \frac{1}{2}$

= **0,0005925925**

Sedangkan perhitungan untuk *tweets 6* sebagai data *testing* 2 sebagai berikut :

P(*tweets*_6 | $c_{positif}$)

= $(w_{rakyat}|\ C_{positif}) \times (w_{sangat}|\ C_{positif}) \times (w_{gembira}|\ C_{positif}) \times P(C_{positif})$

= $\frac{1}{18} \times \frac{2}{18} \times \frac{2}{18} \times \frac{1}{2}$

= **0,000342936**

P(*tweets*_6 | $c_{negatif}$)

= $(w_{rakyat}|\ C_{negatif}) \times (w_{sangat}|\ C_{negatif}) \times (w_{gembira}|\ C_{negatif}) \times P(C_{negatif})$

= $\frac{2}{15} \times \frac{1}{15} \times \frac{1}{15} \times \frac{1}{2}$

= **0,000296296**

Setelah probabilitas terhadap masing-masing *tweets* di hitung, hasil dari perhitungan tersebut selanjutnya dilihat nilai terbesarnya untuk dijadikan

sebagai label dari *tweets testing* tersebut. Hasil perhitungan kedua data *testing* dapat dilihat pada tabel 4.7 berikut :

Tabel 4.7 Nilai Probabilitas Data *Testing*

Tweets	**Positif**	**Negatif**
5	0,000171467	**0,000592592**
6	**0,000342936**	0,000296296

Untuk *tweets 5* sebagai data *testing* pertama dapat disimpulkan bahwa termasuk ke dalam sentimen negatif, karena dilihat dari nilai perbandingan terbesar pada setiap label. Sedangkan untuk *tweets* 6 sebagai data *testing* kedua termasuk ke dalam sentimen positif karena nilai pada label positif lebih besar dari label negatif.

4.6 ANALISIS PEMBOBOTAN KATA (TF-IDF)

Pada penelitian analisis sentimen, pembobotan kata digunakan untuk mendapatkan suatu topik atau kata kunci dari banyaknya kumpulan sentimen. Salah satu metode pembobotan pada analisis sentimen adalah TF-IDF (*Term Frequency – Inverse Document Frequency*). Nilai bobot pada suatu kata *(term)* menyatakan kepentingan bobot tersebut dalam merepresentasikan *tweets*. Pada pembobotan TF-IDF, bobot akan semakin besar jika frekuensi kemunculan kata semakin tinggi, tetapi bobot akan berkurang jika kata tersebut semakin sering muncul pada *tweets* lainnya .

Contoh : Terdapat 3 *tweets* yang sudah melewati tahapan *text processing* seperti berikut :

Tweets1 : polusi indonesia tambah

Tweets2 : warga jakarta jadi orang kampung

Tweets3 : datang dari kampung

Dari 3 *tweets* diatas dapat diketahui nilai frekuensinya sebagai berikut :

$$\text{Idf} = \log\left(\frac{N}{df}\right)$$

N = jumlah *tweets,* yaitu 3 *tweets*

df = banyaknya suatu kata (*term*) yang muncul pada *tweets*

Tabel 4.8 Pembobotan Kata Pada *Tweets*

Kata	**tf T1**	**tf T2**	**tf T3**	**df**	$\frac{N}{df}$	**Idf**	**W T1**	**W T2**	**W T3**
polusi	1	0	0	1	$\frac{3}{1} = 3$	0,477	0,477	0	0
indonesia	1	0	0	1	$\frac{3}{1} = 3$	0,477	0,477	0	0
tambah	1	0	0	1	$\frac{3}{1} = 3$	0,477	0,477	0	0
warga	0	1	0	1	$\frac{3}{1} = 3$	0,477	0	0,477	0
jakarta	0	1	0	1	$\frac{3}{1} = 3$	0,477	0	0,477	0
jadi	0	1	0	1	$\frac{3}{1} = 3$	0,477	0	0,477	0
orang	0	1	0	1	$\frac{3}{1} = 3$	0,477	0	0,477	0

kampung	0	1	1	2	$\frac{3}{2} = 1{,}5$	0,176	0	0,176	0,176
datang	0	0	1	1	$\frac{3}{1} = 3$	0,477	0	0	0,477
dari	0	0	1	1	$\frac{3}{1} = 3$	0,477	0	0	0,477

Keterangan:

tf D1 = Banyaknya muncul kata yang muncul di *tweets* 1 (D1)

tf D2 = Banyaknya muncul kata yang muncul di *tweets* 2 (D2)

tf D3 = Banyaknya muncul kata yang muncul di *tweets* 3 (D3)

W D1 = Bobot kata di *tweets* 1

W D2 = Bobot kata di *tweets* 2

W D3 = Bobot kata di *tweets* 3

Berdasarkan pada Tabel 4.8, dapat disimpulkan bahwa kata yang merepresentasikan ketiga *tweets* diatas adalah polusi, indonesia, tambah, warga, jakarta, jadi, orang, datang dan dari.