

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Subjek Dan Objek Penelitian**

Subjek dalam penelitian ini adalah para pengguna Twitter yang memberikan opini terhadap sirkuit mandalika. Kemudian untuk objek penelitian ini adalah pendapat atau respon masyarakat terhadap sirkuit mandalika di Twitter.

#### **3.2 Alat Dan Bahan Penelitian**

##### **3.2.1 Alat**

Dalam penelitian ini menggunakan beberapa alat berupa perangkat keras maupun lunak :

##### **1. Perangkat Keras**

Laptop : Asus X515

Processor : Intel Core i3

VGA : Intel UHD Graphics

Memory : 12GB

Hard Disk : 500GB

##### **2. Perangkat Lunak**

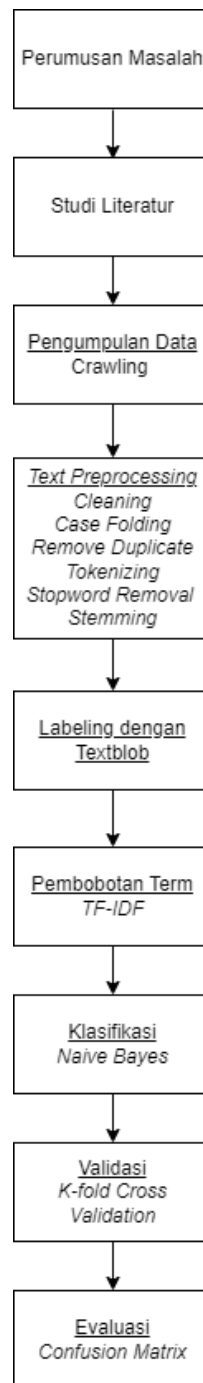
Windows 10, Microsoft Excel , Google Chrome dan Google Colab.

##### **3.2.2 Bahan**

Bahan yang digunakan pada penelitian ini berupa data dari media sosial Twitter terkait opini atau pendapat mengenai sirkuit mandalika dengan jumlah data 1148 Data.

#### **3.3 Diagram Alir Penelitian**

Dalam melakukan penelitian terdapat beberapa langkah-langkah sebelum akhirnya mendapatkan hasil. Berikut adalah diagram alir dari penelitian ini.



Gambar 3.1 Diagram Alir Penelitian

### 3.4 Teknik Pengumpulan Data

Pengumpulan data dilakukan dengan cara *crawling* di media sosial Twitter menggunakan Twitter API. Data yang diambil berupa teks cuitan yang menggunakan hashtag mandalika. Data yang diambil berbahasa Indonesia dan

diambil dalam rentang waktu 21 Maret 2022 hingga 29 Maret 2022, yang menghasilkan 1148 data dalam format CSV.

### 3.5 Analisis Data

Dalam penelitian ini, terdapat beberapa tahapan yang harus dilalui dalam upaya untuk mencapai hasil akurasi dari algoritma yang digunakan. Setelah melakukan pengambilan data dengan melakukan *crawling* di media sosial Twitter.

Pada langkah selanjutnya melakukan teks *Preprocessing*, pada langkah ini terdapat lima proses: proses pertama *cleaning*, *cleaning* adalah proses pembersihan kalimat dari tanda baca, angka, url, dan username serta *emoticon*. Berikut contoh dari *Cleaning* dapat dilihat di Tabel 3.1.

Tabel 3.1 Contoh *Cleaning*

Teks	Hasil
Sirkuit Mandalika itu dianggap sebagai sirkuit dengan pemandangan tercantik di dunia.\n\n#Sandiagauno #motogp2022\xe2\x80\xa6 https://t.co/dDKbyP3OFd'	Sirkuit Mandalika itu dianggap sebagai sirkuit dengan pemandangan tercantik di dunia

Selanjutnya dilakukan proses *case folding* yang mengubah huruf besar menjadi huruf kecil. Berikut contoh dari *case folding* dapat dilihat di Tabel 3.2.

Tabel 3.2 Contoh *Case Folding*

Teks	Hasil
Sirkuit Mandalika itu dianggap sebagai sirkuit dengan pemandangan tercantik di dunia	sirkuit mandalika itu dianggap sebagai sirkuit dengan pemandangan tercantik di dunia

Pada tahap berikutnya dilakukan *tokenizing* yaitu melakukan pemisahan kata dari yang semula kalimat atau menjadikan urutan berupa *token*. Berikut contoh dari *tokenizing* dapat dilihat di Tabel 3.3.

Tabel 3.3 Contoh *Tokenizing*

Teks	Hasil
sirkuit mandalika itu dianggap sebagai sirkuit dengan pemandangan tercantik di dunia	“sirkuit”, ”mandalika”, ”itu”, “dianggap”, “sebagai”, “sirkuit”, “dengan”, “ pemandangan”, “tercantik”, “di”, “dunia”

Lalu dilakukan tahapan *Stopword removal* yaitu dilakukan memilih kata yang penting. Berikut contoh dari *Stopword removal* dapat dilihat di Tabel 3.4.

Tabel 3.4 Contoh *Stopword Removal*

Teks	Hasil
“sirkuit”, ”mandalika”, ”itu”, “dianggap”, “sebagai”, “sirkuit”, “dengan”, “ pemandangan”, “tercantik”, “di”, “dunia”	“sirkuit”, ”mandalika”, “dianggap”, “sirkuit”, “ pemandangan”, “tercantik”, “dunia”

Dan tahapan terakhir dilakukan *stemming* yaitu mengubah kata menjadi bentuk dasarnya. Berikut contoh dari *stemming* dapat dilihat di Tabel 3.5.

Tabel 3.5 Contoh *Stemming*

Teks	Hasil
“sirkuit”, ”mandalika”, “dianggap”, “sirkuit”, “ pemandangan”, “tercantik”, “dunia”	sirkuit mandalika anggap sirkuit pandang cantik dunia

Setelah melakukan proses teks *Preprocessing* selanjutnya data dilakukan *labeling* dengan Textblob pada data yang sudah diproses, dengan tiga sentimen yaitu Positif, Negatif maupun Netral serta dua sentimen yaitu positif dan negatif. Setelah itu dilakukan pembobotan *Term* dengan menggunakan metode Tf-IDF digunakan untuk menghitung *Term Frequency* dan *Inverse Document Frequency*. TF dihitung dengan cara membandingkan antara *frequency* sebuah term dengan nilai maksimal dari *frequency term* pada dokumen. IDF atau *Inverse Document Frequency* adalah perhitungan dari bagaimana term di distribusikan pada dokumen.

Tabel 3.6 Contoh teks Perhitungan TF-IDF dan Naive Bayes

Dok	Teks	Label
1	Sirkuit Mandalika Sangat Indah	Positif
2	Fasilitas Transportasi mandalika belum tertata rapi	Negatif
3	Sirkuit Internasional	Netral

Adapun Ilustrasi pembobotan menggunakan TF-IDF dapat dilihat di Tabel 3.7 dan Tabel 3.8.

Tabel 3.7 Ilustrasi TF

Dokumen <i>Term</i>	TF			DF
	D1	D2	D3	
sirkuit	1	0	1	2
mandalika	1	1	0	2
Sangat	1	0	0	1
indah	1	0	0	1
Fasilitas	0	1	0	1
Transportasi	0	1	0	1
Belum	0	1	0	1
Tertata	0	1	0	1
rapi	0	1	0	1
internasional	0	0	1	1

Tabel 3.8 Ilustrasi TF-IDF

IDF	TF.IDF		
	D1	D2	D3
$\text{Log}(3/2) = 0,176$	0,176	0	0,176
$\text{Log}(3/2) = 0,176$	0,176	0,176	0
$\text{Log}(3/1) = 0,477$	0,477	0	0
$\text{Log}(3/1) = 0,477$	0,477	0	0
$\text{Log}(3/1) = 0,477$	0	0,477	0
$\text{Log}(3/1) = 0,477$	0	0,477	0
$\text{Log}(3/1) = 0,477$	0	0,477	0
$\text{Log}(3/1) = 0,477$	0	0,477	0
$\text{Log}(3/1) = 0,477$	0	0,477	0
$\text{Log}(3/1) = 0,477$	0	0	0,477

Pada tahap berikutnya dilakukan klasifikasi dengan menerapkan algoritma Naive Bayes. Naive Bayes merupakan pengklasifikasi yang mengakar pada Teorema Bayes dengan menggunakan metode statistik dan probabilitas. Adapun Ilustrasi perhitungan Naive Bayes dapat dilihat di Tabel 3.9.

Tabel 3.9 Perhitungan Naive Bayes

No.	Kata	TF(Pos)	TF(Neg)	TF(Net)	P(pos)	P(Neg)	P(Net)
1	Sirkuit	1	0	1	0,1428	0,0625	0,1666
2	Mandalika	1	1	0	0,1428	0,125	0,0833
3	Sangat	1	0	0	0,1428	0,0625	0,0833
4	Indah	1	0	0	0,1428	0,0625	0,0833
5	Fasilitas	0	1	0	0,0714	0,125	0,0833
6	Transportasi	0	1	0	0,0714	0,125	0,0833
7	Belum	0	1	0	0,0714	0,125	0,0833
8	Tertata	0	1	0	0,0714	0,125	0,0833
9	Rapi	0	1	0	0,0714	0,125	0,0833
10	Internasional	0	0	1	0,0714	0,0625	0,1666
Jumlah Term		4	6	2			

1. Hitung Probabilitas Prior Setiap kategori

$$P(\text{Positif}) = \frac{1}{3} = 0,3333$$

$$P(\text{Negatif}) = \frac{1}{3} = 0,3333$$

$$P(\text{Netral}) = \frac{1}{3} = 0,3333$$

2. Menghitung probabilitas likelihood setiap term dari semua dokumen.

1) Probabilitas kata sirkuit

$$P(\text{sirkuit}|\text{Positif}) = \frac{1 + 1}{4 + 10} = \frac{2}{14} = 0,1428$$

$$P(\text{sirkuit}|\text{negatif}) = \frac{0 + 1}{6 + 10} = \frac{1}{16} = 0,0625$$

$$P(\text{sirkuit}|\text{netral}) = \frac{1 + 1}{2 + 10} = \frac{2}{12} = 0,1666$$

2) Probabilitas kata mandalika

$$P(\text{mandalika}|\text{Positif}) = \frac{1 + 1}{4 + 10} = \frac{2}{14} = 0,1428$$

$$P(\text{mandalika}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$

$$P(\text{mandalika}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

3) Probabilitas kata sangat

$$P(\text{sangat}|\text{Positif}) = \frac{1 + 1}{4 + 10} = \frac{2}{14} = 0,1428$$

$$P(\text{sangat}|\text{negatif}) = \frac{0 + 1}{6 + 10} = \frac{1}{16} = 0,0625$$

$$P(\text{sangat}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

4) Probabilitas kata indah

$$P(\text{indah}|\text{Positif}) = \frac{1 + 1}{4 + 10} = \frac{2}{14} = 0,1428$$

$$P(\text{indah}|\text{negatif}) = \frac{0 + 1}{6 + 10} = \frac{1}{16} = 0,0625$$

$$P(\text{indah}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

5) Probabilitas kata fasilitas

$$P(\text{fasilitas}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{fasilitas}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$

$$P(\text{fasilitas}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

6) Probabilitas kata transportasi

$$P(\text{transportasi}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{transportasi}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$

$$P(\text{transportasi}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

7) Probabilitas kata belum

$$P(\text{belum}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{belum}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$

$$P(\text{belum}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

8) Probabilitas kata Tertata

$$P(\text{Tertata}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{Tertata}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$

$$P(\text{Tertata}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

9) Probabilitas kata rapi

$$P(\text{rapi}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{rapi}|\text{negatif}) = \frac{1 + 1}{6 + 10} = \frac{2}{16} = 0,125$$



$$P(\text{rapi}|\text{netral}) = \frac{0 + 1}{2 + 10} = \frac{1}{12} = 0,0833$$

10) Probabilitas kata internasional

$$P(\text{internasional}|\text{Positif}) = \frac{0 + 1}{4 + 10} = \frac{1}{14} = 0,0714$$

$$P(\text{internasional}|\text{negatif}) = \frac{0 + 1}{6 + 10} = \frac{1}{16} = 0,0625$$

$$P(\text{internasional}|\text{netral}) = \frac{1 + 1}{2 + 10} = \frac{2}{12} = 0,1666$$

Contoh terdapat teks testing yaitu “mandalika sangat rapi”

Positif :

$P(\text{Positif}) \cdot p(\text{mandalika} | \text{Positif}) \cdot p(\text{sangat} | \text{Positif}) \cdot p(\text{rapi} | \text{Positif})$

$$0,3333 \cdot 0,1428 \cdot 0,1428 \cdot 0,0714 = 0,00048$$

Negatif :

$P(\text{Negatif}) \cdot p(\text{mandalika} | \text{Negatif}) \cdot p(\text{sangat} | \text{Negatif}) \cdot p(\text{rapi} | \text{Negatif})$

$$0,3333 \cdot 0,125 \cdot 0,0625 \cdot 0,125 = 0,00032$$

Netral :

$P(\text{netral}) \cdot p(\text{mandalika} | \text{Netral}) \cdot p(\text{sangat} | \text{Netral}) \cdot p(\text{rapi} | \text{Netral})$

$$0,3333 \cdot 0,0833 \cdot 0,0833 \cdot 0,0833 = 0,00019$$

Jadi kesimpulan dari kata testing tersebut di klasifikasikan sebagai positif karena memiliki nilai paling besar 0,00048

Setelah pengimplementasian algoritma Naive Bayes dilakukan validasi yaitu dengan *K-Fold Cross Validation*. *K-Fold Cross Validation* digunakan untuk menilai kinerja proses dari sebuah algoritma.

Tahapan terakhir yaitu evaluasi dengan *confusion matrix* digunakan untuk melihat hasil akurasi atau performa dari klasifikasi. *Confusion matrix* menghasilkan nilai *accuracy*, *precision*, *recall* dan *f1 score*.