

## BAB III METODE PENELITIAN

### 3.1 Objek Penelitian

Objek penelitian dilakukan pada media sosial *microblogging x* dengan mengambil sampel dari pendapat masyarakat ditunjukkan pada Tabel 3.1.

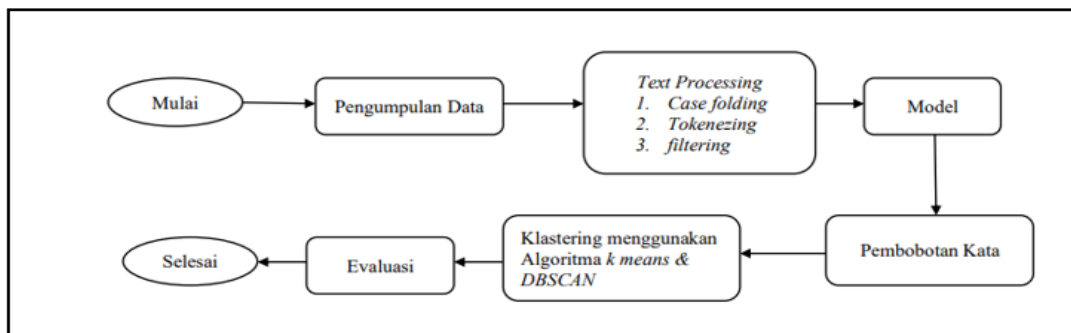
Tabel 3.1 Sampel Postingan

Postingan Positif	Postingan Negatif
Ide E-tilang sangat <u>membantu</u> baik korlantas maupun pengendara.	Sumpah Birokrasi pemerintahan yang jelas itu apasi semuanya <u>nyusahin</u> rakyat mulu nisekarang E-tilang ya allah ngelus dada ngurusnya di urus online malah jadi susah poll.

Berdasarkan yang ditunjukkan Tabel 3.1 pada kolom postingan positif terdapat kalimat “ide Etilang sangat membantu baik korlantas maupun pengendara” mengandung kalimat positif pada kalimat “membantu” pada Tabel 3.1 pada kolom postingan negatif terdapat kalimat “sumpah birokrasi pemerintah yang jelas itu apasi semuanya nyusahin rakyat mulu sekarang E-tilang ya allah ngelusdada ngurus nya di urus online malah susah pol” mengandung kalimat negatif pada kalimat “nyusahin”.

### 3.2. Prosedur Penelitian

Tahapan prosedur penelitian dimulai dengan mengumpulkan data yang di ambil dari *microblogging X*, *Text Preprocessing*, pembobotan kata, model dan algoritma *k means* dan *DBSCAN* evaluasi menggunakan metode *sillhouette score*. Tahapan prosedur penelitian di gambarkan dengan *Flowchart* ditunjukkan Gambar 3.1.



Gambar 3.1 Prosedur Penelitian

### 3.3.1. Pengumpulan Data

Proses pengumpulan data dilakukan dengan mengumpulkan data di media sosial *Microblogging x*. Data diambil dengan total data 1855 data *tweet*, sejak awal mula penerapan sistem e-tilang yaitu pada 25 November 2018 hingga januari 2024. Berikut Tabel 3.3 yang menjelaskan alur dari algoritma pengumpulan data.

Tabel 3.2 Alur Pengumpulan Data

Algoritma Pengumpul Data
Input : apikey,apisecret,acctoken dan tokensecret
Proses : membaca token developer
Output :kalimat

### 3.3.2. Text Preprocessing

*Text Preprocessing* merupakan tahapan sebelum melakukan perhitungan algoritma pada proses ini dilakukan pemilihan data dengan tujuan data yang tidak dimaksudkan untuk membuat data terstruktur menjadi tidak terstruktur. Terdapat beberapa tahapan pada *Text Preprocessing* yaitu ada *case folding*, *Tokenizing*, dan *filtering* sebagai berikut.

#### a. Case folding

Pada tahapan *case folding*, semua karakter diubah menjadi huruf kecil, ukuran setiap karakter diperiksa dari awal hingga akhir, dan jika ditemukan karakter yang menggunakan huruf besar, karakter diubah akan menjadi huruf kecil.

Tabel 3.3 Algoritma *case folding*

Algoritma <i>Case Folding</i>
Input : Kalimat
Proses :
1. Baca kalimat
2. Merubah kalimat kapital pada kalimat menjadi huruf nonkapital
Output : kalimat menjadi nonkapital

#### b. Tokenizing

*Tokenizing* pada tahapan ini merupakan tahapan dalam memecahkan string atau input terhadap suatu teks berdasarkan tiap kata yang Menyusun nya dan menghilangkan *mention* yang ada pada dokumen *tweet*. Beserta menghilangkan tanda titik (.), koma (,), tanda seru (!), tanda tanya (?).

Tabel 3.4 *Tokenizing*

Algoritma <i>Tokenizing</i>
Input : kalimat
Proses :
1. Baca Kalimat
2. Menghilangkan Tanda Baca
Output : Kalimat menjadi tidak ada tanda baca dan karakter lain nya

### c. *Filtering*

Kata-kata yang sering muncul secara umum dan tidak relevan dengan teks dihilangkan. Tahap ini disebut juga dengan stop word removal, yaitu menghilangkan kata-kata yang tidak berarti dan tidak berdampak pada analisis sentimen

Tabel 3.5 Algoritma *Filtering*

Algoritma <i>Filtering</i>
Input : kalimat
Proses :
1. Baca Kalimat
2. Menghilangkan kalimat yang tida penting
Output : kealimat yang tidak penting menjadi menghilang

### 3.3.3. Model

Model yang di hasilkan berbentuk *Wordcloud* dan histogram kata yang sering muncul. *Wordcloud* sebuah representasi data yang berbentuk teks dan juga dapat berguna dalam berbagai konteks untuk memvisualisasikan data teks secara menarik. Berikut Tabel 3.10 dan gambar 3.2 yang menjelaskan alur algoritma model dan contoh hasilnya.

Tabel 3.6 Algoritma Model

Algoritma Model
Input : kalimat
Proses :
1. Baca Kalimat
2. Memisahka kalimat menjadi kata
Output : Memunculkan kata yang sering muncul

### 3.3.4. Pembobotan Kata

Langkah selanjutnya adalah menggunakan metode TF-IDF untuk menghitung bobot setiap kata atau kata berdasarkan seberapa sering kata tersebut muncul dalam dokumen. TF-IDF adalah statistik numerik yang menampilkan kata kunci dengan kata-kata tertentu. Selain itu, TF-IDF dapat menemukan kata-kata yang sering muncul dalam dokumen.

Tabel 3.7 Algoritma TF-IDF

Algoritma TF-IDF
Input : k: jumlah kluster, inisiasi jumlah cluster awal k =2 nilai <i>centroid</i> awal
Menggunakan metode mean, data set n objek.
Proses : Membobotkan kata
Output : Memunculkan Bobot Nilai pada kata

### 3.3.5. Klastering Menggunakan Algoritma *K means*

*K-means Clustering* termasuk salah satu algoritma pengelompokan yang melakukan partisi set data kedalam sejumlah k *cluster* yang sudah di tetapkan dengan mengambil nilai minimal dari jarak titik data terhadap k *cluster*. Berikut proses tahapan algoritma *k means* :

1. Menentukan jumlah *cluster*.
2. Mengelompokan data sehingga terbentuk K *cluster* dengan titik *centroid* dari setiap *cluster* merupakan titik *centroid* yang telah dipilih sebelumnya.
3. Hitung pusat *centroid* dari data yang ada dimasing-masing kelompok tempat *centroid* setiap kelompok diambil dari rata-rata (*mean*) semua nilai data pada setiap fituranya.
4. Alokasikan masing-masing data pada *centroid* terdekat. Untuk mengukur jarak data pusat *centroid* dapat dilakukan dengan rumus *Eucilidien distance*.

### 3.3.6. Klastering Menggunakan Algoritma *DBSCAN*

*DBSCAN* merupakan metode *clustering* sebuah proses yang mengelompokan data dalam satu cluster atau kelompok sehingga data dalam satu cluster memiliki Tingkat kemiripan maksimum dan data antar cluster memiliki kemiripan yang minimum. berikut langkah proses pengelompokan menggunakan metode *DBSCAN* :

1. Inialisasi parameter *Eps* ( $\epsilon$ ) dan *minPts*.
2. Menentukan titik awal atau P yang belum dikunjungi secara acak.
3. Hitung semua jarak titik yang kepadatannya dapat dicapai pada titik P.
4. Jika titik yang memenuhi nilai *Eps* ( $\epsilon$ ) lebih dari sama *minPts* maka titik P adalah *core points* dan kluster terbentuk.

5. Jika titik yang memenuhi nilai  $Eps$  ( $\epsilon$ ) kurang dari  $minPts$  maka titik P akan ditandai sebagai *noise* dan proses dilanjutkan ke titik yang lain.

### 3.3.7. Evaluasi

Pada tahapan evaluasi akan dilakukan pengujian menggunakan metode *sillhouette score* untuk tujuan dilakukanya pengujian ini untuk mengetahui tingkat akurasi melalui metode *sillhouette score*. Terdapat tiga langkah perhitungan *sillouete*.

1. Hitung rata-rata jarak objek dengan semua objek lain yang berada di dalam suatu *cluster* dengan persamaan.
2. Hitung semua jarak objek dengan objek lain yang berada pada *cluster* lain kemudian ambil nilai minimum dengan persamaan.
3. Hitung nilai dengan *sillhouette coefficient* dengan persamaan (2).

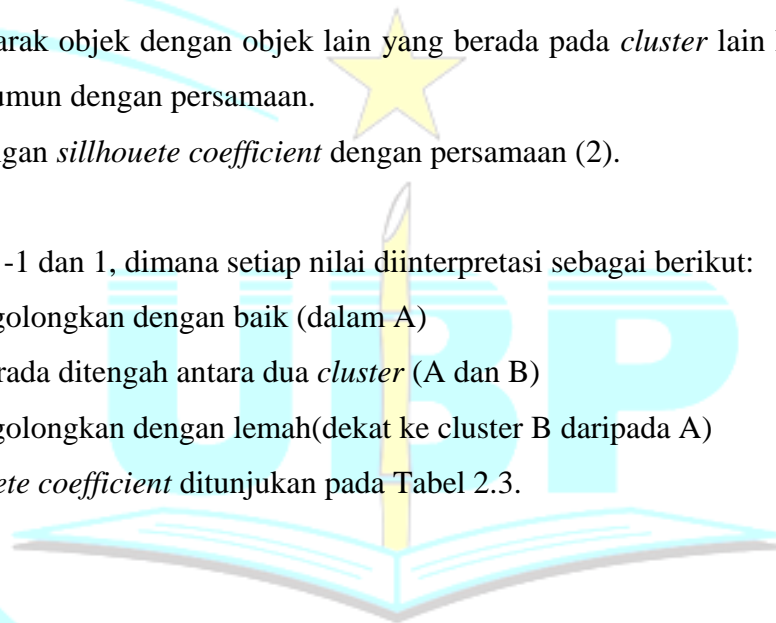
Nilai  $S(i)$  berada antara -1 dan 1, dimana setiap nilai diinterpretasi sebagai berikut:

$S(i) \approx 1 \Rightarrow$  data ke- $i$  digolongkan dengan baik (dalam A)

$S(i) \approx 0 \Rightarrow$  data ke- $i$  berada ditengah antara dua *cluster* (A dan B)

$S(i) \approx -1 \Rightarrow$  data ke- $i$  digolongkan dengan lemah(dekat ke cluster B daripada A)

Penafsiran nilai *sillhouette coefficient* ditunjukkan pada Tabel 2.3.



**KARAWANG**