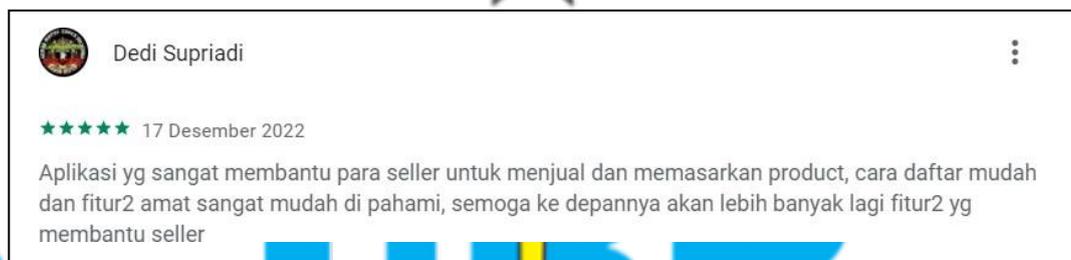


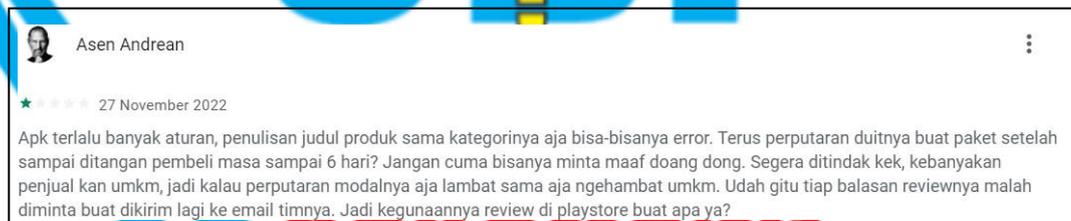
## BAB III METODE PENELITIAN

### 3.1 Objek Penelitian

Objek penelitian yang digunakan pada penelitian ini adalah data *review* dari para pengguna aplikasi *Tiktok Shop Seller Center* berdasarkan *Google Play Store*, sebanyak 1250 *review*. Kemudian data tersebut mengalami *filtering* menjadi 2 kategori yaitu kategori positif dan kategori negatif dengan jumlah data *review* sebanyak 611.



Gambar 3. 1 *Review* Positif



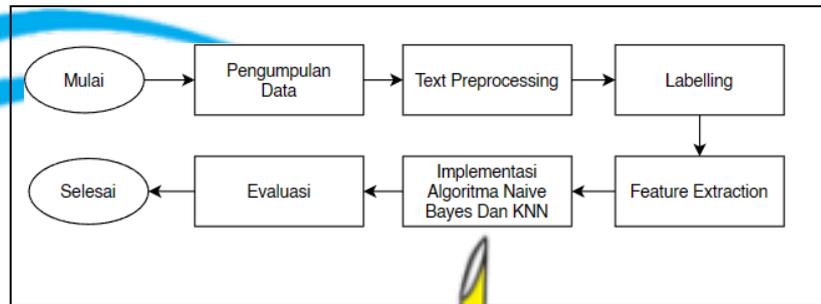
Gambar 3. 2 *Review* Negatif

Berdasarkan yang ditunjukkan oleh Gambar 3.1 terdapat kalimat “Aplikasi yg sangat membantu para seller untuk menjual dan memasarkan product, cara daftar mudah dan fitur2 amat sangat mudah dipahami, semoga kedepannya akan lebih banyak lagi fitur2 yang membantu seller” mengandung makna positif pada kalimat “Aplikasi yg sangat membantu para seller”. Selanjutnya pada Gambar 3.2 terdapat kalimat “Apk terlalu banyak aturan, penulisan judul produk sama kategorinya aja bisa-bisanya error. Terus perputaran duitnya buat paket setelah sampai ditangan pembeli masa sampai 6 hari? Jangan cuma bisanya minta maaf doang dong. Segera ditindak kek, kebanyakan penjual kan UMKM, jadi kalau perputaran modalnya aja lambat sama aja ngehambat usaha umkm. Udah gitu tiap balasan reviewnya malah diminta buat dikirim lagi ke email timnya. Jadi kegunaannya review di playstore

buat apa ya?”, mengandung makna negatif pada kalimat “Apk terlalu banyak aturan, penulisan judul produk sama kategorinya aja bisa-bisanya error”.

### 3.2 Prosedur Penelitian

Dalam penelitian ini terdapat tahapan-tahapan yang akan dilakukan. Seperti pada Gambar 3.3 Alur Prosedur Penelitian.



Gambar 3. 3 Alur Prosedur Penelitian

Penelitian ini dimulai dengan pengumpulan data *review* aplikasi *tiktok shop seller center* pada *playstore* menggunakan bahasa pemrograman *python*, dilanjutkan dengan proses *text preprocessing* yang terdiri dari proses *cleaning*, *case folding*, *tokenize*, *normalize*, *filtering*, dan *stemming*. Dilanjutkan dengan proses *labelling* yaitu memberikan label positif dan negatif pada setiap data *review*, proses selanjutnya adalah *feature extraction* yaitu proses untuk mengubah data *text* menjadi data *numeric* agar dapat dipahami oleh komputer, selanjutnya proses implementasi algoritma *Naive Bayes* dan *K-Nearest Neighbor*, setelah kedua algoritma tersebut selesai di implementasikan langkah selanjutnya adalah melakukan evaluasi model dengan menggunakan *confusion matrix*.

#### 3.2.1 Pengumpulan Data

Pengumpulan data dari penelitian ini dibagi menjadi dua bagian antara lain:

1. Studi Literatur

Studi literatur dilakukan sebagai pencarian landasan teori dan untuk mendapatkan referensi dari berbagai sumber informasi seperti jurnal, buku, dan juga informasi internet.

## 2. *Scraping Data*

Proses pengumpulan data-data *review* dari *Google Playstore* dengan menggunakan library pemrograman python. Hasil dari *scraping* ini mendapatkan 1250 data *review* pada aplikasi *Tiktok Shop Seller Center* di *Google Playstore*. Berikut ini adalah *pseudocode* dari *library* yang digunakan untuk melakukan *scraping* data dari *google playstore* :

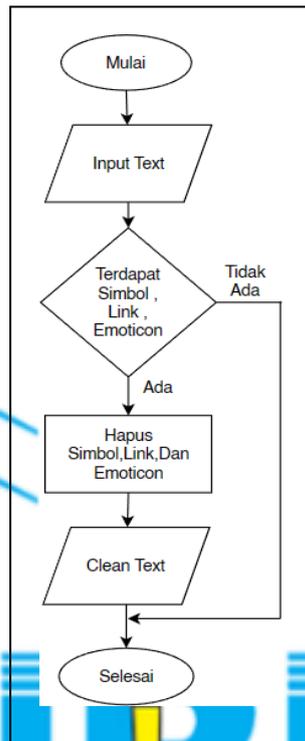
- a. Mulai
- b. *Import* *Sort* dan *reviews* module dari *google\_play\_scraper*
- c. *Define* variabel *result* dan variabel *continuation\_token* yang akan menyimpan nilai yang dikembalikan dari *function reviews*
- d. *Call function reviews* dengan parameter berikut :
  - 1) *App\_id* (str)
  - 2) *Lang* (str)
  - 3) *Country* (str)
  - 4) *Sort* (enum)
  - 5) *Count*(int)
  - 6) *Filter\_score\_with* (int)
- e. Selesai

### 3.2.2 *Text Preprocessing*

*Text preprocessing* merupakan sebuah proses yang dilakukan untuk mempersiapkan data sebelum data tersebut dimodelkan oleh *machine learning*. Tahapan dalam *text preprocessing* yang digunakan pada penelitian ini yaitu :

#### 1. *Cleaning*

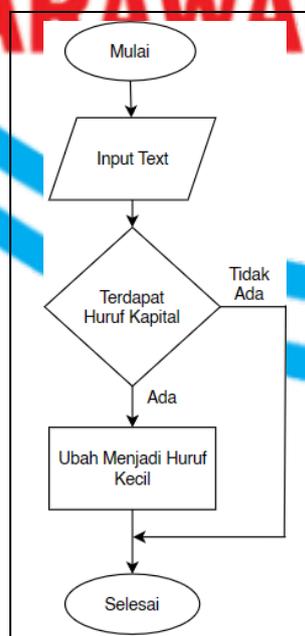
*Cleaning* adalah proses untuk membersihkan *text* dari karakter yang tidak diperlukan seperti simbol, *emoticon*, dan suatu *link* dari situs *website*. Berikut ini adalah alur dari proses *cleaning* :



Gambar 3. 4 Alur Proses *Cleaning*

## 2. *Case Folding*

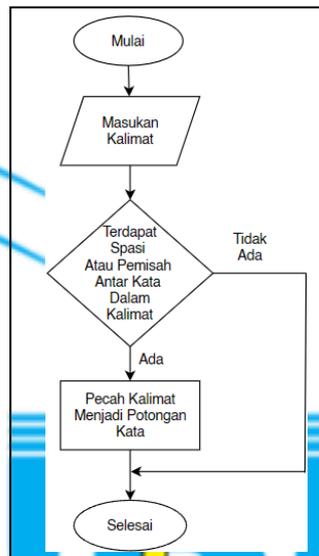
*Case folding* merupakan suatu proses untuk mengubah semua huruf didalam sebuah *text* menjadi huruf kecil. Berikut ini adalah alur dari proses *case folding* :



Gambar 3. 5 Alur Proses *Case Folding*

### 3. Tokenize

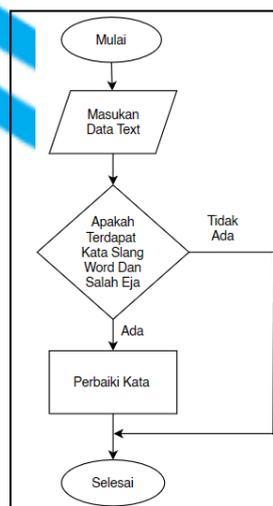
*Tokenize* merupakan proses yang dilakukan untuk memecah suatu kalimat menjadi potongan kata. Berikut ini adalah alur dari proses *tokenize* dapat dilihat pada Gambar 3. 6 :



Gambar 3. 6 Alur Proses *Tokenize*

### 4. Normalize

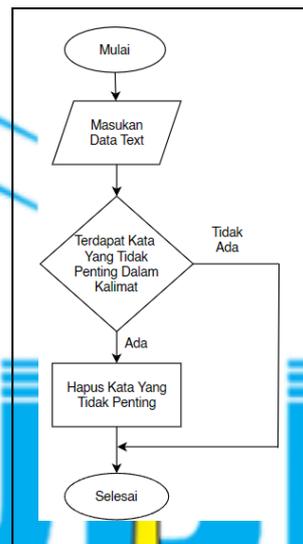
*Normalize* atau normalisasi merupakan proses untuk memperbaiki kesalahan eja suatu kata, mengubah *slang words* menjadi kata yang lebih umum, dan mengubah kata singkatan menjadi kata yang lebih umum. Berikut ini adalah alur dari proses *normalize* :



Gambar 3. 7 Alur Proses *Normalize*

### 5. Filtering

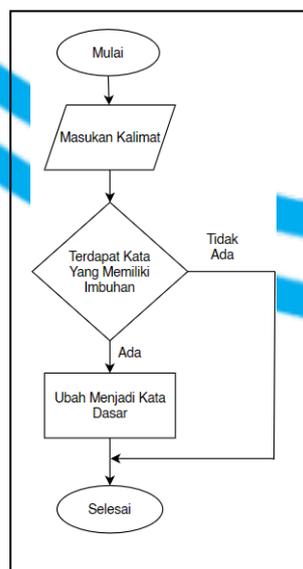
*Filtering* yaitu suatu proses yang dilakukan untuk menghapus kata yang tidak penting atau kata yang tidak bermakna, *filtering* bertujuan untuk menyisakan kata-kata yang penting dan memiliki makna. Berikut ini adalah alur dari proses *filtering* :



Gambar 3. 8 Alur Proses *Filtering*

### 6. Stemming

*Stemming* yaitu proses untuk mengubah suatu kata yang berimbuhan menjadi kata dasar. Berikut ini adalah alur dari proses *stemming* :



Gambar 3. 9 Alur Proses *Stemming*

### 3.2.3 Labelling

*Labelling* merupakan proses pemberian label pada setiap data *review* ke dalam dua kategori kelas yang berbeda yaitu kelas positif dan kelas negatif. Data *review* dapat diklasifikasi sebagai kelas positif jika berisi ungkapan rasa senang dan rasa kepuasan, sedangkan data *review* yang berisi ungkapan kekecewaan dan keluhan dapat diklasifikasi sebagai kelas negatif. Biasanya proses *labelling* ini melibatkan SME (*subject matter expert*) yaitu seorang profesional yang memiliki pengetahuan lebih dalam bidang tertentu.

### 3.2.4 Feature Extraction

Setelah dilakukan proses *labelling* langkah selanjutnya melakukan *feature extraction* yaitu mengubah data *text* menjadi data numerik agar data tersebut dapat diolah oleh komputer, salah satu teknik *feature extraction* adalah *Term Frequency (TF) Inverse Document Frequency (IDF)* yaitu suatu metode algoritma yang berguna untuk menghitung bobot setiap kata yang umum digunakan. Metode ini terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini akan menghitung nilai *Term Frequency (TF)* dan *Inverse Document Frequency (IDF)* pada setiap kata di setiap dokumen dalam korpus.

TF-IDF dapat dihitung menggunakan rumus berikut :

$$TF(i, j) = \frac{\text{Term } i \text{ frequency in document}}{\text{Total words in document}} \quad (6)$$

$$IDF(i) = \log_2 \left( \frac{\text{Total documents}}{\text{documents with term } i} \right) \quad (7)$$

$$TF - IDF(i) = TF(i, j) \times IDF(i) \quad (8)$$

### 3.2.5 Implementasi Algoritma *Naïve Bayes* dan *K-Nearest Neighbor*

Hal yang perlu dilakukan sebelum melakukan implementasi algoritma *Naïve Bayes* dan *K-Nearest Neighbor* yaitu harus dilakukan *splitting* data terlebih dahulu dengan menentukan jumlah pembagian data menjadi data latih, validasi, dan data uji. Berikut ini adalah implementasi algoritma *Naïve Bayes* dan *K-Nearest Neighbor* dalam mengklasifikasi kelas suatu sentimen:

### 1. Implementasi algoritma *Naïve Bayes*

Algoritma *Naïve Bayes* akan menghitung suatu probabilitas atau kemiripan tertinggi data uji terhadap model *machine learning* yang dibuat menggunakan data latih. Dibawah ini adalah rumus *naïve bayes* untuk mengklasifikasi sebuah *class* dari sebuah sentimen :

$$P(y|x_i, \dots, x_n) = \frac{P(y) \cdot P(x_i, \dots, x_n|y)}{P(x_i, \dots, x_n)} \quad (9)$$

Keterangan :

$P(y|x_i, \dots, x_n)$  = Posterior

$P(y)$  = Prior

$P(x_i|y)$  = Likelihood

$P(x_i, \dots, x_n)$  = Evidence

$y$  = Class

$x$  = Vektor dari nilai atribut n

### 2. Implementasi algoritma *K-Nearest Neighbor*

Algoritma K-NN akan mencari tetangga terdekat dari data uji lalu menghitung jaraknya sesuai nilai K yang ditentukan. Jika proses pencarian tetangga terdekat sudah dilakukan maka model *machine learning* akan menghasilkan *ouput* dari data uji tersebut apakah data tersebut merupakan sentimen positif atau sentimen negatif. Dibawah adalah tahapan algoritma K-NN dalam mengklasifikasi sebuah *class* dari sebuah sentimen :

- a. Menentukan nilai K
- b. Hitung jarak *euclidean* dari jumlah tetangga K menggunakan rumus berikut :

$$d = \sqrt{\sum_{i=1}^p (x_{2_i} - x_{1_i})^2} \quad (10)$$

Keterangan :

$d$  = Jarak pada *euclidean*

$p$  = banyaknya atribut

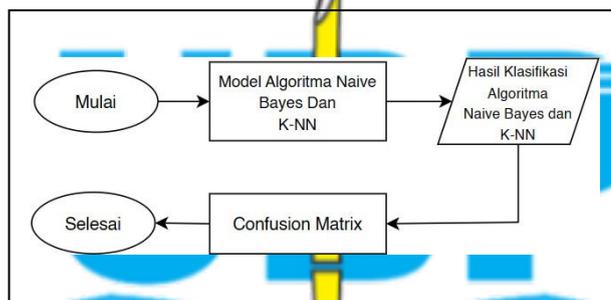
$x_{2_i}$  = Nilai pada baru atau data uji ke -i

$x_{1_i}$  = Nilai pada baru atau data latih ke -i

- c. Urutkan hasil perhitungan jarak mulai dari yang terkecil sesuai nilai  $K$  yang ditentukan
- d. Tetapkan *class* data baru sesuai jumlah tetangga yang lebih banyak

### 3.2.6 Evaluasi

Tahap terakhir yaitu evaluasi, tahap ini merupakan proses untuk mengetahui informasi ketika sudah dilakukan implementasi algoritma *Naive Bayes* dan *K-Nearest Neighbor*. Evaluasi model yang digunakan adalah confusion matrix karena confusion matrix akan memberikan 4 nilai *output* yaitu : TP (*True Positive*) , FN (*False Negative*), TN (*True Negative*), dan FP (*False Positive*). Sehingga dapat dihitung nilai *accuracy*, *recall*, *precision*, dan *f1 score* dari model yang sudah dibuat. Alur dari evaluasi dapat dilihat pada Gambar 3. 10 :



Gambar 3. 10 Alur Evaluasi

Berdasarkan Gambar 3.10 proses evaluasi dimulai dengan tersedianya model *machine learning* dari kedua algoritma kemudian kedua model tersebut akan menghasilkan hasil klasifikasi, selanjutnya hasil klasifikasi tersebut akan dihitung menggunakan confusion matrix agar dapat diketahui nilai akurasi dari setiap model algoritma.