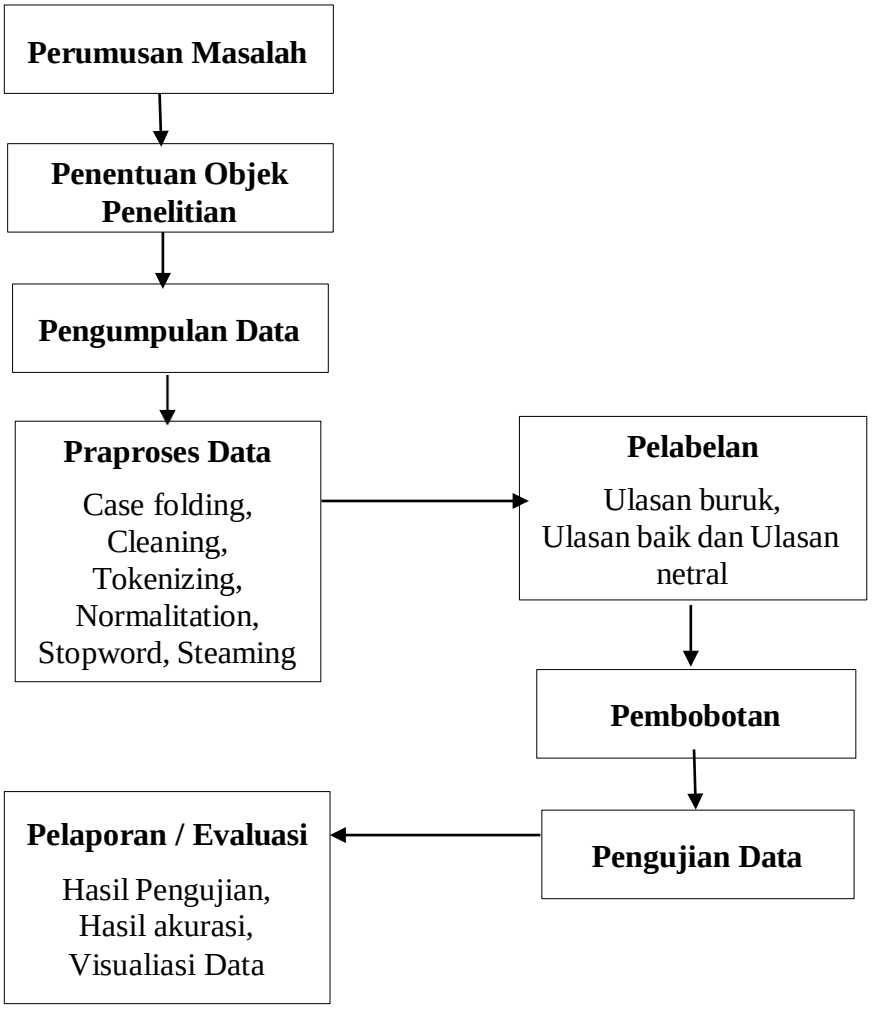
BAB III

METODE PENELITIAN

3.1 Metode penelitan

Proses sentiment analisis pada pengguna Bukalapak di mulai berdasarkan tahap penelitian. Tahap pertama peneliti melakukan pengumpulan data kemudian tahap praproses, tahap pemodelan klasifikasi, tahap pengujian, dan tahap evaluasi hasil. Kemudian peneliti melakukan prediksi sentiment terhadap data yang telah di kumpulkan dan memberikan rekomendasi pada setiap sentiment.



Gambar 3.1. Diagram Alir Prosess Penelitian

1. **Perumasan Masalah**

Peneliti melakukan observasi dan pemahaman dari keseluruhan ulasan pada aplikasi bukalapak, permasalahan yang terjadi pada Bukalapak adalah ulasan pengguna yang terlalu banyak sehingga sulit dalam mengkategorikan dan menganalisis ulasan para pengguna, untuk mengetahui kecenderungan komentar serta informasi yang terdapat dalam ulasan bukan hal yang mudah, karena jumlah data ulasan yang terlalu banyak dan memakan waktu yang lama.

2. **Penentuan Objek Penelitian**

Peneliti menggunakan aplikasi bukalapak kedalam perumusan definisi masalah naïve bayes classfier, menyiapkan sterategi untuk mencapai tujuan tersebut

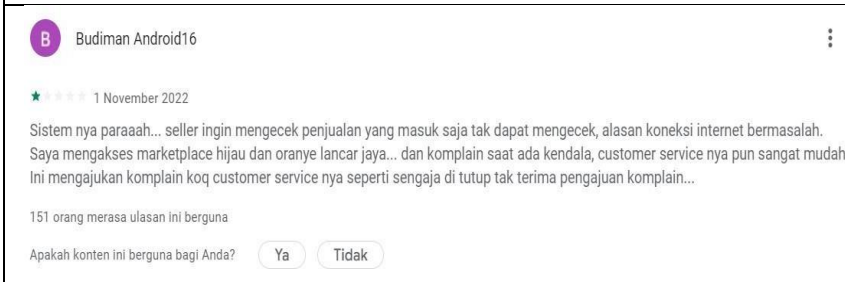
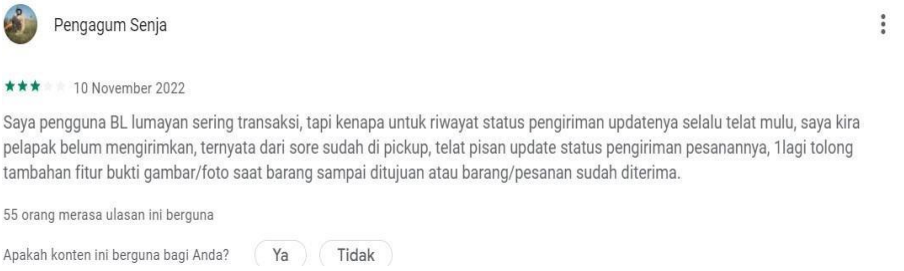
3. **Pengumpulan**

Proses ini di mulai dengan proses Penggunaan *library scraping* pada *python* yaitu pengambilan data dari *Play Store official aplikasi* Bukalapak dengan cara mencetak data dari ulasan pengguna Bukalapak, *y*ang akan di input ke dalam file csv Pengumpulan data pengguna *aplikasi Bukalapak* dilakukan dengan cara mengambilan sampel melalui akses dari com.bukalapak.android.

4. Input

Input yang di maksudkan adalah berupa ulasan dari akun pengguna yang berupa opini. Data ulasan tersebut di dapat dengan memanfaatkan fitur Komentar yang di sediakan oleh *Play Store.*

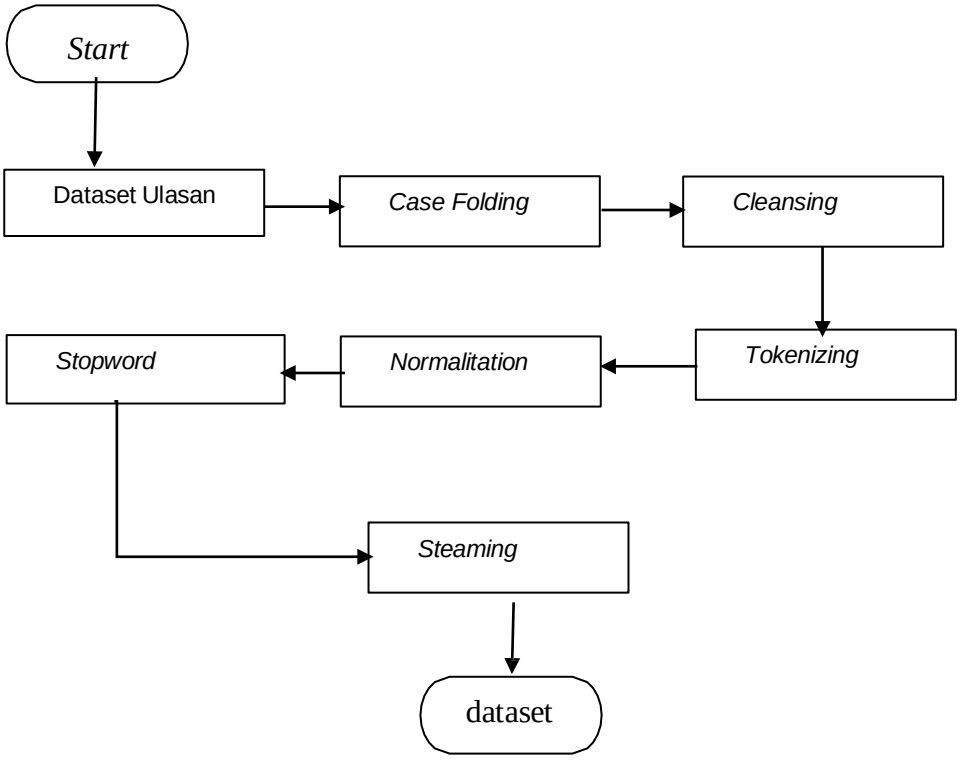
Tabel 3.1. kolom kometar Ulasan Bukalapak

Ulasan	***Rate***
	***1***
	***3***

Dataset berupa teks berbahasa Indonesia yang diambil dari *website com.bukalapak.android.* Data yang diambil untuk penelitian ini data yang diambil menggunakan *query* ‘Pengguna’, ‘*Rating*’, ‘Ulasan. *Query* tersebut merupakan akun resmi dari *e- commerce* Bukalapak. Ulasan yang diambil merupakan postingan dari pelanggan tersebut.

3.2 *Preprocessing*

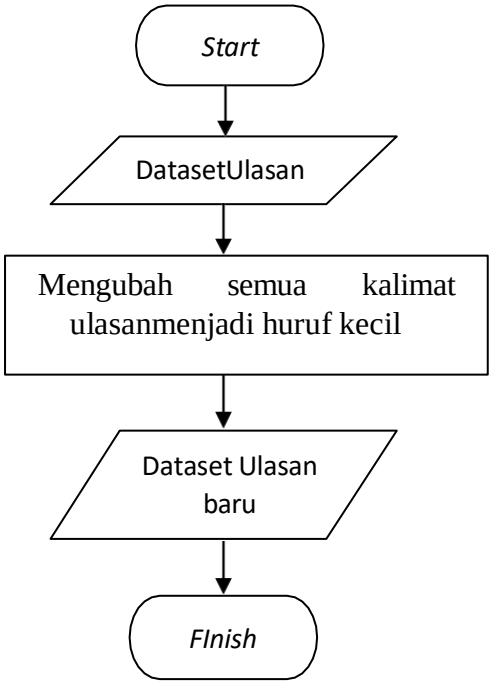
Proses *preprocessing* merupakan hal yang penting untuk tahap selanjutnya, yaitu mengurangi atribut yang kurang berpengaruh terhadap proses klasifikasi. Data yang dimasukkan pada tahap ini masih berupa data mentah yang masih kotor, sehingga hasil dari proses ini adalah dokumen yang berkualitas yang harapannya mempermudah dalam proses klasifikasi. Proses *preprocessing* terdiri dari beberapa tahapan yang dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 Diagram Alir *Preprocessing*

1. ***Case Folding***

Pada tahap case folding huruf kapital pada semua dokumen tweet diubah menjadi huruf kecil. Tujuannya untuk menghilangkan redudansi data yang hanya berbeda pada hurufnya saja. Berikut diagram alir *case folding* terdapat pada

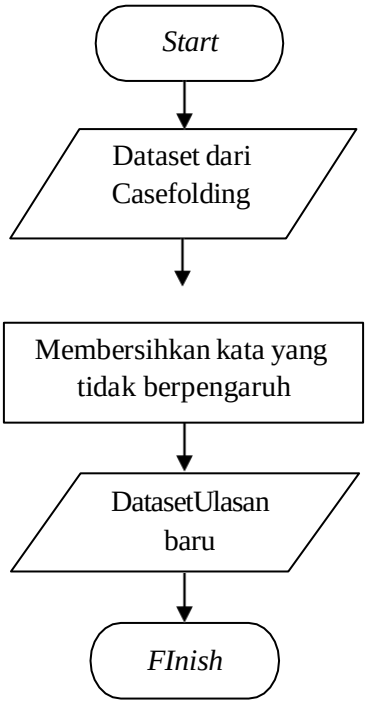


Gambar 3.3 Diagram Alir *Folding*

Sebagai gambaran dari proses *case folding* berikut contoh *ulasan* yang di hasilkan terdapat pada tabel 3.3.

2. *Cleansing*

Tahapan *cleansing* merupakan tahap pembersihan kata yang tidak berpengaruh sama sekali terhadap hasil klasifikasi sentimen. Komponen dokumen ulasan memiliki berbagai atribut yang tidak berpengaruh terhadap sentimen, karena setiap ulasan hampir semua memiliki atribut tersebut. Contoh dari atribut yang tidak penting tersebut adalah yaitu mention yang diawali dengan atribut (‘@’), hastag yang diawali dengan atribut (‘#’), link yang diawali dengan atribut (‘http’,’bit.ly’) dan karakter simbol (~!@#$%^&*()_+?<>,.?:{}[]|). Atribut yang tidak berpengaruh tersebut akan dihilangkan dari dokumen kemudian akan digantikan dengan karakter spasi. Berikut diagram alir cleansing terdapat pada gambar 3.5 .

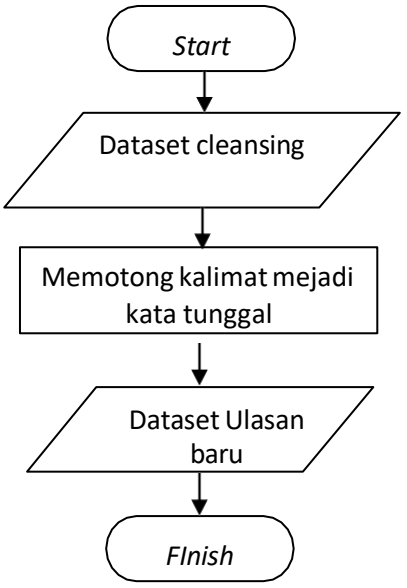


Gambar 3.4 Diagram Alur *Cleansing*

Sebagai gambaran dari proses *cleansing* berikut penulis memberikan contoh pada table 3.4

3. *Tokenizing*

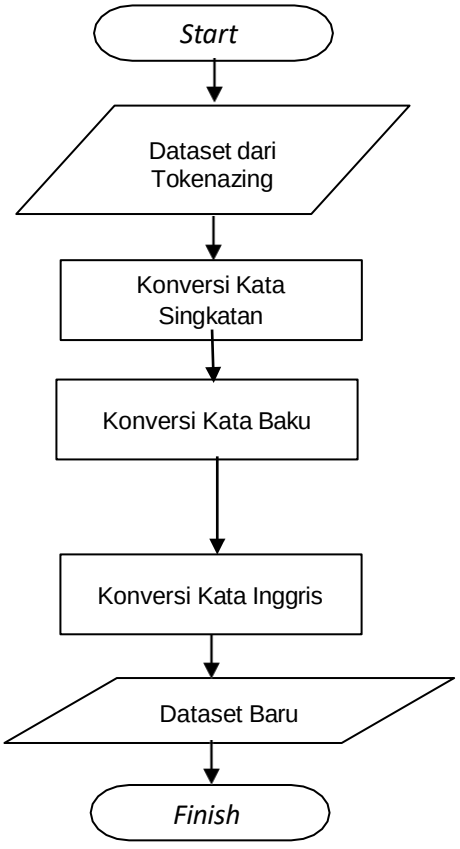
Tahap *tokenizing* merupakan pemotongan kata berdasarkan tiap kata yang menyusunnya menjadi potongan tunggal. Kata dalam dokumen yang dimaksud adalah kata yang dipisah oleh spasi. Sehingga hasil dari proses ini merupakan kata tunggal yang dimasukkan ke dalam *database* untuk keperluan pembobotan. Berikut diagram alir *tokenizing* terdapat pada Gambar 3.5.



Gambar 3.5 Diagram Alur *Tokenizing*

4. *Normalization*

Pada tahap *normalization* ini dilakukan pengubahan kata yang tidak sesuai dengan EYD, sehingga dapat mengurangi hasil sentimen dokumen. Tahap ini dibagi menjadi tiga langkah, yaitu konversi kata singkatan, konversi kata baku, dan konversi kata inggris. Berikut diagram alir *normalization* pada Gambar 3.6.

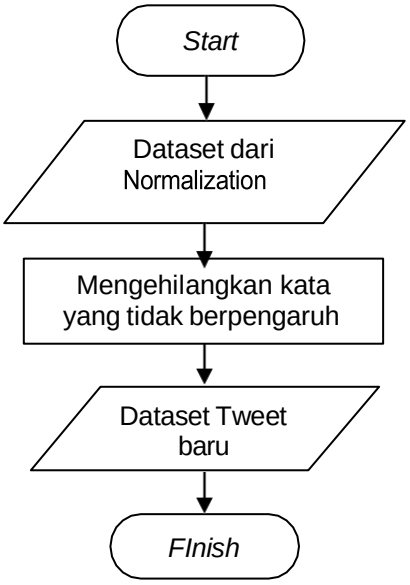


Gambar 3.6 Diagram Alir *Normalization*

Ulasan pengguna banyak yang menulis dengan kata singkatan agar apa yang ditulisnya dapat terekspresikan. Hal tersebut mebuat masalah terhadap performasi sentimen dokumen. Pada tahap ini dilakukan proses untuk mengubah kata singkatan menjadi kata normal, yang dicontohkan pada tabel 3.6.

5. *Stopword Removal*

Tahap *stopword removal* merupakan tahap menghilangkan kata yang tidak sesuai dengan topik dokumen, jika ada kata tersebut tidak mempengaruhi akurasi dalam klasifikasi sentimen dokumen. Kata yang akan dihilangkan dihimpun dalam *databse* kata *stopword*. Jika dalam dokumen ulasan ada yang sesuai dengan kata dalam *stopword* maka kata tersebut akan dihilangkan dan diganti dengan karakter spasi. Berikut diagram alir *stopword removal* pada Gambar 3.7.



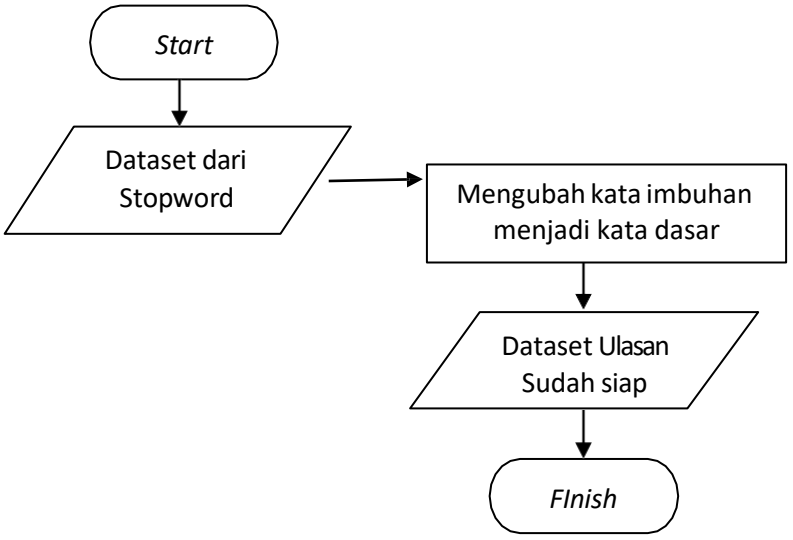
Gambar 3.7 Diagram Alir *Stopword Removal*

6. *Stemming*

Pada tahap *stemming* merupakan suatu proses untuk mengubah kata – kata yang terdapat dalam suatu dokumen ke dalam kata – kata akarnya dengan menggunakan aturan – aturan tertentu. Proses *stemming* bahasa Indonesia dilakukan dengan menghilangkan *sufiks, prefix, dan konfiks* pada dokumen. Pada proses *stemming* ini penulis menggunakan metode yang dibuat oleh Bobby Naziefdan Mirna Adriani, dengan tahapan sebagai berikut: (Agusta, 2009) Cari kata yang akan di*stem* dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah *root word*. Maka metode berhenti. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa *particles* ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* ("-ku", "-mu", atau "- nya"), jika ada. Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka metode berhenti. Jika tidak maka ke langkah 3a Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "- k", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka metode berhenti. Jika tidak ditemukan maka lakukan langkah 3b. Akhiran yang dihapus ("-i", "-an" atau "-kan") dikembalikan, lanjut ke langkah 4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b. Periksa tabel kombinasi awalan – akhiranyang tidak diijinkan.Jikaditemukan maka metode berhenti, jika tidak pergi ke langkah 4b.*For* i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka metode berhenti. Catatan: jika awalan kedua sama dengan awalan pertama metode berhenti.Melakukan *recording*.

Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

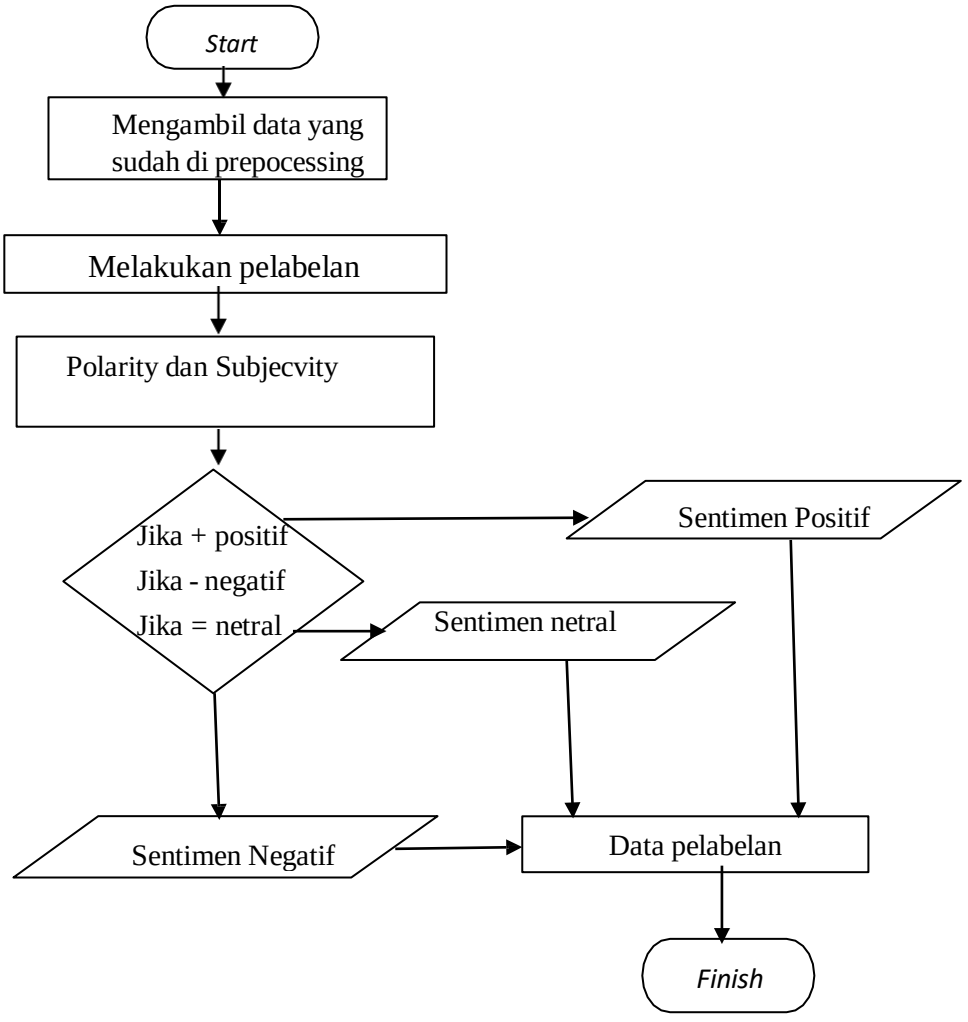
Untuk membuat proses *stemming* secara sempurna harus melaluli beberapa tahapan poroses yang telah disebutkan pada poin diatas.Berikut diagram alir *stemming* pada Gambar 3.8.



Gambar 3.8. Diagram *Alur Steamming*

3.3 Pelabelan (Subjecvity Dan Polarity)

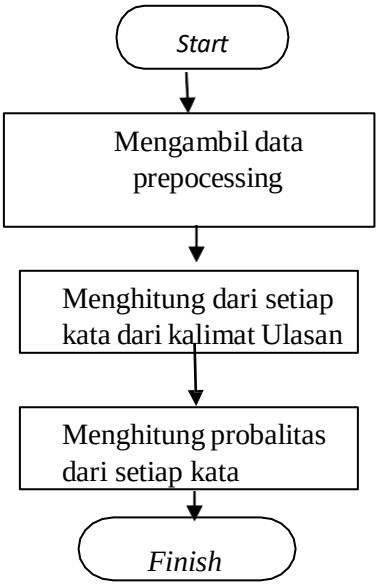
Pada tahap pelabelan ini menggunakan metode pembagian dibagi menjadi dua proses, yaitu proses *positif*, proses *negatif dan netral*. Pada tahap ini dilakukan proses pelabelan terlebih dahulu untuk pelatihan mengacu teks *probabilitas* dari *dataset* Diagram alur tahapan pelabelan ini ditunjukkan pada Gambar 3.9 berikut.



Gambar 3.9. Alur proses Pelabelan *Subjecvity* Dan *Polarity*

3.4 *TF- IDF (Term Frequency-Inverse Document Frequency)*

Pada tahap ini akan melakukan pembobotan menghitung setiap kata dan mencari nilai probalitas yang banyak keluar dari dataset ulasan, Diagram alur tahapan di tujukan pada Gambar 3.10.



Gambar 3.10. Alur Proses Menghitung Pembobotan *TF-IDF*

Perhitungan rumus pembobotan dari setiap kata yang kaluar dari masing masing dokumen akan di beri nilai dan akan di jumlahkan. (Komang et al. 2018).

$$tf_i = \frac{freq_i(d_j)}{\sum_{i=1}^{k} freq_i(d_j)} \quad \textbf{(1)}$$

Term frekuensi di hitung menggunakan persaman (1)

$$idf_i = log \frac{|D|}{|\{d:t_i \in d\}|} \quad \textbf{(2)}$$

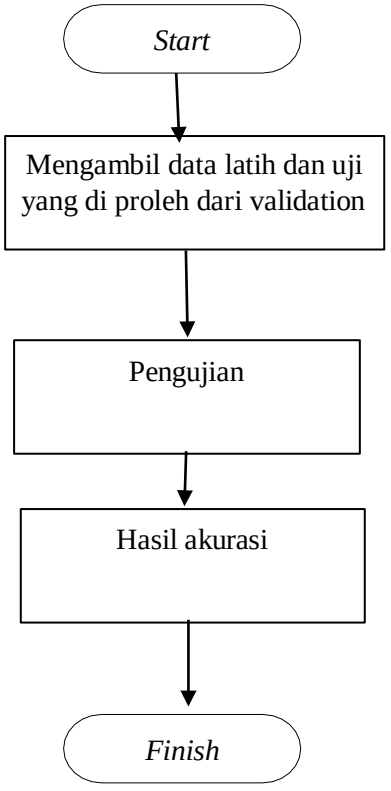
Inverse Document Frequency *(IDF)* adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki term yang dimaksud seperti yang dituliskan secara matematis pada Persamaan(2)

$$(tf - idf)_{ij} = tf_i(d_j) * idf_i \quad \textbf{(3)}$$

Nilai didapatkan dengan mengalikan keduanya yang diformulasikan pada Persamaan(3) tf sebagai kata yang keluar pada dokumen (Dj) kata yang di dokumen * bobot dari hasil log positif.

3.5 Proses Menggunakan Metode Naïve *bayes classifier*

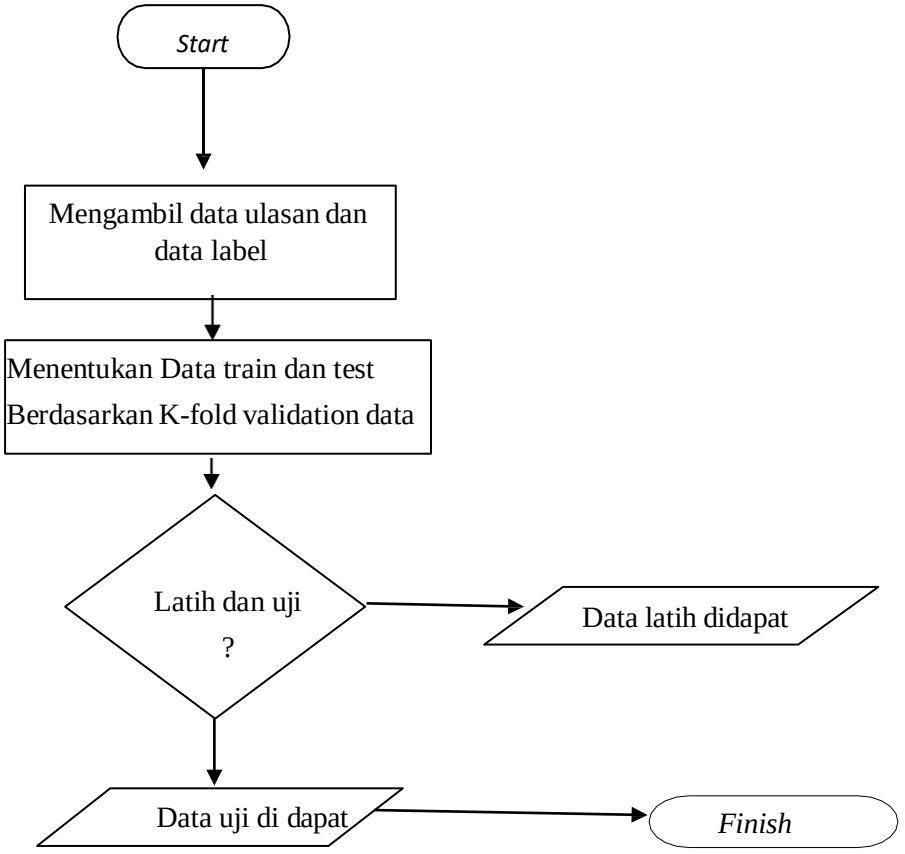
Pada tahapan ini akan mengukur akurasi menggukana gaussion yang mana dalam pelabelan bersifat numerik atau kontinu berdasar kan *train* dan *test* yang di dapat yang telah di kumupulkan alur proses di tujukan seperti Gambar 3.11.



Gambar 3.11 Alur Proses *Naïve bayes classfier*

1. ***Cross validation naïve bayes***

Pada tahap ini akan melakukan memisahkan data antara baik dan buruk ke train and test berdasarkan data dan proses naïve bayes alur proses di tunjukan seperti Gambar 3.11



Gambar 3.12 Alur Proses *Cross Validation Naïve bayes*

Berdasar kan tahap gambar 3.12 memisahkan data berdasarkan cross validation naïve bayes bagaimana menentukan data berdasarkan ukuran data menjadi seimbang dalam pengukuran Naïve bayes

2. Pengujian

Pengujian model menggunakan data uji yang telah di validasi data *negatif* dan *positif*. Model pengujian naïve bayes classifier memprediksikan setiap bobot kata tf idf ke dalam suatu kelas. Penentuan kelas ini berdasarkan model yang terbentuk dari data latih.

Menghitung data uji berisikan nilai dari bobot setiap kata dan di jumlahkan, test berisikan label ulasan negative dan positif. (Rasyadi 2017).

$$P(label|fitur)= \frac{P(label) * P(fitur|label)}{P(fitur)}$$

$$P(label|fitur)= \frac{fitur|label)}{P(fitur}$$

Library ini membuat asumsi bahwa semua fitur bersifat independen ,sehingga:

$$P(label|fitur)= \frac{P(label) * P(f_1|label)*P(f_2|label)*P(f_3|label)*\ldots*P(f_n|label)}{P(fitur)}$$

Algoritme ini hanya menghitung pembilangan untuk setiap label, dan menormalkannya menjadi satu dari pada menghitung peluang dari fitur, seperti berikut:

$$(\quad|\quad)= \frac{P(label)* P(k\ |labe\ell)*P(k\ |labe\ell)*P(f\ |labe\ell)*\ldots*P(f\ |lab_n}{\text{SUM[label]}(\ P(f_1|fitur)*P(f_2|fitur)*P(f_3|fitur)*\ldots*P(f_n|fitur))} P\ label\ fi$$

3.6 Visualisasi

Visualisasi menggunkan word cloud sering digunakan untuk menyoroti istilah populer atau trend berdasarkan frekuensi pengguna dari hasil perhitungan pembobotan dari Tf-idf yang sudah di hitung menampilakan kata mendasarkan nilai banyak yang paling sering keluar semakin banyak kata yang keluar akan semakin dominasi hasil

3.7 Evaluasi Hasil

Proses evaluasi menggunakan confusion matrix dan classifikasi akurasi rata-rata yang di dapat dalam proses pengujian, serta menampilkakn visualisasi data hasil pengujian dan akurasi.

Confusion matrix adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah. Contoh *confusion matrix multi class* untuk klasifikasi biner ditunjukkan pada Tabel. (Normawati and Prayogi 2021).

$$Akurasi = \frac{TP+TN+TN}{TP+TN+TP+FN+FN+FN} \times 100\ \% \quad (1)$$

Mengukur kinerja dari setiap class akurasi tergantung sebesar precisi dan recall pada setiap class (1).

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

Presisi yang rendah menunjukkan bahwa model Anda memiliki banyak kesalahan dalam mengklasifikasikan data sebagai positif ketika seharusnya tidak (2).

$$Recall = \frac{TP}{TP+FN} \times 100\ \% \quad (3)$$

Recall yang rendah menunjukkan bahwa model anda memiliki banyak kesalahan dalam mengklasifikasikan data sebagai negatif ketika seharusnya positif (3)