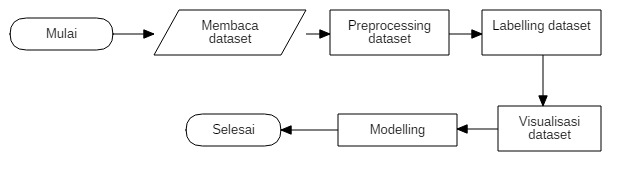
**BAB 3**

**METODE PENELITIAN**

**3.1 Rancangan Alur Proses**

Berikut merupakan rancangan alur dari analisis sentimen dengan menggunakan metode Multinomial Naive Bayes dapat dilihat pada gambar 3.1.



**Gambar 3.1 *Flowchart* Analisis Sentimen**

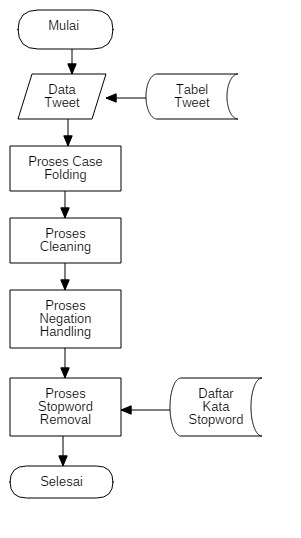
Pada gambar 3.1 analisis sentimen dimulai dengan membaca dataset berisi komentar yang sudah disimpan pada file dengan format csv. Setelah membaca dataset, maka tahap selanjutnya adalah tahap *preprocessing*. Pada tahap *preprocessing* akan dilakukan *case folding, cleaning,* *negation handling*, dan *stopword removal*. Setelah dataset sudah melalui tahap *preprocessing* maka dataset akan diberi *label*. Pada tahap *labelling* dataset akan diklasifikasi menjadi sentimen positif, negatif, dan netral menggunakan kamus kata positif dan negatif. Kemudian akan dilakukan visualisasi data dengan menampilkan diagram dan *word cloud*. Selanjutnya akan dilakukan *modelling* untuk membuat model klasifikasi dengan metode Multinomial Naive Bayes. Setelah modeltelah dibuat, maka tahap terakhir adalah evaluasi model dengan melakukan klasifikasi data uji guna mengetahui nilai akurasi, *recall*, dan presisi.

**3.2 Pengumpulan Data *Tweet***

Data *tweet* mengenai MRT Jakarta diambil dari sosial media Twitter. Data *tweet* dikumpulkan menggunakan *keyword* “@mrtjakarta”, “naik mrt jakarta”, dan “#mrtjakarta”. Data *tweet* disimpan dalam format file csv.

**3.3 *Preprocessing***

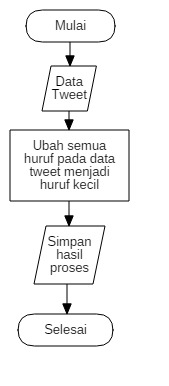
Proses ini berfungsi untuk mengurangi atribut yang dianggap tidak berpengaruh dan mengurangi dimensi kata. Proses *preprocessing* meliputi proses *case folding, cleaning*, *negation handling*, dan *stopword removal*. Berikut ini merupakan *flowchart* dari proses *preprocessing* dapat dilihat pada gambar 3.2.



**Gambar 3.2 *Flowchart* Proses *Preprocessing***

**3.3.1 *Case Folding***

*Case Folding* merupakan tahap awal yang dilakukan dalam proses *preprocessing*. Pada tahap ini sistem akan mengubah semua huruf pada kalimat *tweet* menjadi huruf kecil. Hal ini dilakukan agar data *tweet* yang ada konsisten menggunakan huruf kecil. *Flowchart* dari proses *case folding* dapat dilihat pada gambar 3.3.



**Gambar 3.3 *Flowchart* Proses *Case Folding***

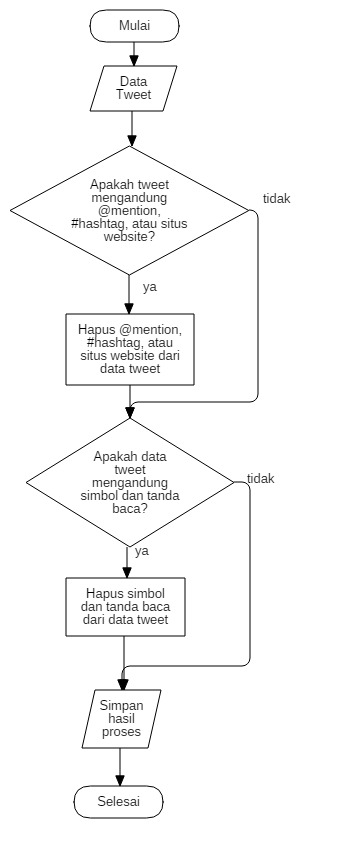
Pada proses *case folding* mula-mula akan memasukkan data *tweet*. Setelah data *tweet* dimasukkan maka seluruh huruf yang ada pada data *tweet* akan diubah menjadi huruf kecil. Berikut ini merupakan contoh *input* dan *output* dari proses *case folding* dapat dilihat pada tabel 3.1.

**Tabel 3.1 Tahap *Case Folding***

|  |  |
| --- | --- |
| *Input* (Data Awal) | *Output* (Hasil *Case Folding*) |
| **M**encoba naik railink dan **MRT** hari ini, takjub ternyata jakarta udah keren transportasi umumnya. **N**yaman dan cepet yah. **S**alut | mencoba naik railink dan mrt hari ini, takjub ternyata jakarta udah keren transportasi umumnya. nyaman dan cepet yah. salut |
| **A**ntrian panjang untuk membeli tiket, petugas tidak ramah. @ stasiun bundaran **HI** @mrtjakarta | antrian panjang untuk membeli tiket, petugas tidak ramah. @ stasiun bundaran hi @mrtjakarta |

**3.3.2 *Cleaning***

*Cleaning* merupakan tahapan dimana menghapus karakter-karakter yang tidak dibutuhkan dalam proses klasifikasi data *tweet*. Hal yang akan dihilangkan pada proses ini adalah angka, simbol, *emoticon*, *hashtag*, *username* (@*username*), dan url suatu *website.* Berikut ini merupakan *flowchart* dari proses *cleaning* dapat dilihat pada gambar 3.4.

**

**Gambar 3.4 *Flowchart* Proses *Cleaning***

Contoh *input* dan *output* dari proses *cleaning* dapat dilihat pada tabel 3.2.

**Tabel 3.2 Tahap *Cleaning***

|  |  |
| --- | --- |
| *Input* (Hasil *Case Folding*) | *Output* (Hasil *Cleaning*) |
| mencoba naik railink dan mrt hari ini**,** takjub ternyata jakarta udah keren transportasi umumnya**.** nyaman dan cepet yah. salut | mencoba naik railink dan mrt hari ini takjub ternyata jakarta udah keren transportasi umumnya nyaman dan cepet yah salut |
| antrian panjang untuk membeli tiket, petugas tidak ramah.**@** stasiun bundaran hi **@mrtjakarta** | antrian panjang untuk membeli tiket petugas tidak ramah stasiun bundaran hi |

**3.3.3 *Negation Handling***

*Negation Handling* dilakukan setelah proses *cleaning*. Jika pada *tweet* terdapat kata negasi “tidak” dan “belum”, maka kata negasi akan digabungkan dengan kata setelahnya sehingga menjadi satu kesatuan. Hal ini dilakukan karena kata negasi dapat mengubah makna dari suatu kata. Contoh *input* dan *output* dari proses *cleaning* dapat dilihat pada tabel 3.3.

**Tabel 3.3 Tahap *Negation Handling***

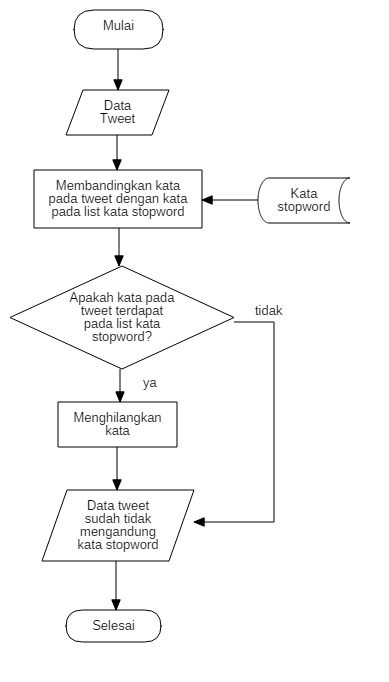
|  |  |
| --- | --- |
| *Input* (Hasil *Cleaning*) | *Output* (Hasil *Negation Handling*) |
| mencoba naik railink dan mrt hari ini takjub ternyata jakarta udah keren transportasi umumnya nyaman dan cepet yah salut | mencoba naik railink dan mrt hari ini takjub ternyata jakarta udah keren transportasi umumnya nyaman dan cepet yah salut |
| antrian panjang untuk membeli tiket petugas **tidak ramah** stasiun bundaran hi | antrian panjang untuk membeli tiket petugas tidak\_ramah stasiun bundaran hi |

**3.3.4 *Stopword Removal***

Tahap *stopword removal* dilakukan setelah proses *negation handling*. Kata-kata yang termasuk ke dalam *stopword* akan dihilangkan guna mengurangi term atau kata yang terdapat pada data *tweet*. *Flowchart* dari proses *stopword removal* dapat dilihat pada gambar 3.5. Daftar kata *stopword* didapatkan dari penelitian [22]. Adapun isi dari sebagian dari daftar kata *stopword* akan disajikan pada tabel 3.4.

**Tabel 3.4 Daftar Kata *Stopword***

|  |  |  |  |
| --- | --- | --- | --- |
| Daftar Kata *Stopword* | | | |
| ada | bagai | cara | dan |
| adalah | bagi | cukup | dapat |
| agak | bahwa | cuma | dari |
| akhir | baru | dalam | datang |



**Gambar 3.5 *Flowchart* Proses *Stopword Removal***

Pada proses *stopword removal* mula-mula akan memasukkan data *tweet*. Setelah data *tweet* dimasukkan maka seluruh kata yang ada pada data *tweet* akan dicocokkan dengan kamus kata *stopword*. Apabila kata pada *tweet* terdapat pada kamus kata *stopword*, maka kata tersebut akan dihapus dari data *tweet*. Berikut ini merupakan contoh *input* dan *output* dari proses *stopword removal* dapat dilihat pada tabel 3.5.

**Tabel 3.5 Tahap *Stopword Removal***

|  |  |
| --- | --- |
| *Input* (Hasil *Negation Handling*) | *Output* (Hasil *Stopword Removal*) |
| mencoba naik railink **dan** mrt **hari ini** takjub **ternyata** jakarta udah keren transportasi **umumnya** nyaman **dan** cepet yah salut | mencoba railink mrt takjub jakarta udah keren transportasi nyaman cepet yah salut |
| antrian **panjang** **untuk** membeli tiket petugas tidak\_ramah stasiun bundaran hi | antrian membeli tiket petugas tidak\_ramah stasiun bundaran hi |

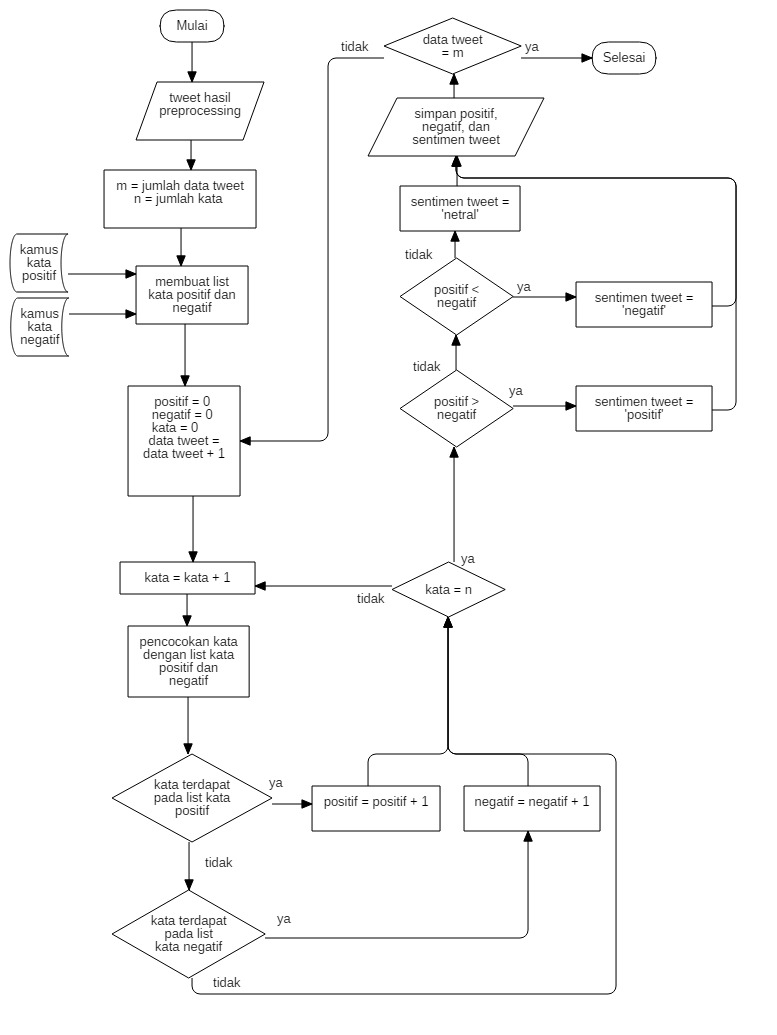
**3.4 *Labelling* Dataset**

Tahap selanjutnya adalah *labelling* dataset yang telah melalui proses *preprocessing*. *Labelling* dilakukan menggunakan pendekatan *lexicon* dengan menggunakan kamus kata positif dan negatif yang sebagian diperoleh dari penelitian [25] dengan penyesuaian terhadap tema MRT Jakarta. Pendekatan *lexicon* digunakan untuk menentukan apakah data *tweet* bersentimen positif atau negatif sesuai dengan kamus data yang ada. *Flowchart* dari proses *labelling* dapat dilihat pada gambar 3.6. Pada proses *labelling* kata pada data *tweet* akan dicocokan dengan kata yang berada pada kamus kata positif dan negatif. Jika terdapat kata pada *tweet* yang sesuai dengan kamus kata positif maka kemungkinan bahwa *tweet* bersentimen positif akan meningkat dan jika terdapat kata pada *tweet* yang sesuai dengan kamus kata negatif maka kemungkinan bahwa *tweet* bersentimen negatif akan meningkat. Apabila jumlah kata pada *tweet* yang terdapat pada kamus kata positif lebih besar dari jumlah kata pada *tweet* yang terdapat pada kamus kata negatif maka *tweet* bersentimen positif, apabila jumlah kata pada *tweet* yang terdapat pada kamus kata negatif lebih besar dari jumlah kata pada *tweet* yang terdapat pada kamus kata positif maka *tweet* bersentimen negatif, dan *tweet* bersentimen netral apabila jumlah kata pada *tweet* yang terdapat pada kamus kata positif sama dengan jumlah kata pada *tweet* yang terdapat pada kamus kata negatif. Berikut ini merupakan contoh kata positif dan negatif yang terdapat pada kamus dapat dilihat pada tabel 3.6. Untuk kamus lengkapnya dapat dilihat pada halaman lampiran.

**Tabel 3.6 Contoh Kamus Kata Positif dan Negatif**

|  |  |
| --- | --- |
| Kamus Kata Positif | Kamus Kata Negatif |
| kredibel | mafia |
| peduli | omdo |
| mampu | takut |
| alhamdulillah | nipu |
| kuat | rampok |

Setelah proses *labelling* berhasil dilakukan, data *tweet* dengan hasil klasifikasi positif dan negatif akan dipisahkan untuk digunakan dalam proses selanjutnya. Hal ini dilakukan karena model Multinomial Naive Bayes Classifier akan dilatih menggunakan data *tweet* bersentimen positif dan negatif.



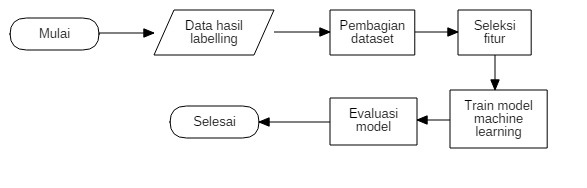
**Gambar 3.6 *Flowchart* Proses *Labelling***

**3.5 Visualisasi**

Visualisasi dilakukan dengan tujuan untuk mengkomunikasikan informasi mengenai hasil dari proses sebelumnya. Pada tahap ini dilakukan visualisasi hasil dari proses *preprocessing* dan *labelling* dengan menampilkan diagram dan *word cloud*.

**3.6 *Modelling***

Dalam membuat model klasifikasi dengan metode Multinomial Naive Bayes terdapat beberapa tahapan yang dilakukan. Berikut ini merupakan *flowchart* pembuatan model klasifikasi dapat dilihat pada gambar 3.7.



**Gambar 3.7 *Flowchart* Proses *Modelling***

Proses *modelling* dimulai dengan pengambilan data hasil *labelling* dengan hasil klasifikasi positif dan negatif. Kemudian data akan dibagi menjadi data *training* dan data *testing*. Data *training* akan digunakan untuk seleksi fitur, ekstraksi fitur, dan *training* model klasifikasi. Sedangkan data *testing* akan digunakan untuk melakukan evaluasi terhadap model klasifikasiyang dibuat. Setelah data dibagi, maka proses selanjutnya adalah menentukan parameter yang akan digunakan dalam proses *training* model klasifikasi, melakukan *training* model klasifikasi, dan menyimpan model hasil *training*. Setelah model telah dilatih maka akan dilakukan evaluasi untuk mengetahui performa dari model yang telah dibuat.

**3.6.1 Pembagian Dataset**

Dataset telah diklasifikasi pada proses *labelling*. Selanjutnya dataset akan dibagi menjadi data *training* dan data *testing* dengan perbandingan 80% data *training* dan 20% data *testing*. Data *training* akan digunakan dalam proses seleksi fitur dan *training* model klasifikasi. Sedangkan data *testing* akan digunakan untuk melakukan evaluasi terhadap model Multinomial Naive Bayes Classifier yang telah dibuat.

**3.6.2 Seleksi dan Ekstraksi Fitur**

Seleksi dan ekstraksi fitur merupakan tahap yang penting sebelum membuat suatu model klasifikasi. Tujuan dilakukannya seleksi fitur adalah untuk menentukan fitur yang digunakan dalam membuat model dengan tujuan agar model memiliki performa terbaik. Kemudian ekstraksi fitur dilakukan untuk mengubah fitur-fitur yang didapat menjadi vektor atau angka. Hal ini dilakukan karena mesin tidak dapat membaca teks, sehingga harus mengubah teks menjadi vektor atau angka.

**3.6.3 *Training* Model**

Model klasifikasi dilatih menggunakan fitur yang telah terpilih pada proses seleksi fitur dan ekstraksi fitur. Perbedaan fitur yang digunakan dapat meningkatkan atau menurunkan keakuratan model dalam melakukan prediksi. Metode Multinomial Naive Bayes digunakan karena proses yang sederhana dan mudah diaplikasikan pada berbagai keadaan sehingga tidak akan mengalami kegagalan secara keseluruhan pada hasilnya [13]. Pada metode Multinomial Naive Bayes kemunculan kata atau term bersifat independen antara satu term dengan term lainnya. Pada pembuatan model Multinomial Naive Bayes Classifier terdapat parameter alpha yang digunakan untuk melakukan proses *smoothing* [8]. Proses *smoothing* dilakukan untuk menangani probabilitas nol pada data *test* akibat terdapat suatu kata pada data *test* yang tidak terdapat pada fitur yang didapat dari data *training* [8].

**3.6.4 Evaluasi Model**

Langkah selanjutnya adalah melakukan evaluasi model. Evaluasi dilakukan guna mengetahui apakah model yang dibuat dapat bekerja dengan baik. Data *testing* akan digunakan untuk pengujian terhadap model Multinomial Naive Bayes Classifier. Hasil dari pengujian berupa kelas prediksi yang akan dibandingkan dengan label sebenarnya dari data *testing* dengan menggunakan *confusion matrix* seperti pada tabel 3.7.

**Tabel 3.7 *Confusion Matrix***

|  |  |  |
| --- | --- | --- |
| Label Sebenarnya | Prediksi Model | |
| Negatif | Positif |
| Negatif | True Negative (TN) | False Positive (FP) |
| Positif | False Negative (FN) | True Positive (TP) |

Evaluasi dilakukan untuk mengukur kemampuan dari model yang dibuat dalam melakukan prediksi pada data baru atau data yang belum digunakan pada saat melakukan *training* model klasifikasi *Naive Bayes Classifier*. Berdasarkan dari nilai yang terdapat pada *confusion matrix* dapat dihitung akurasi dari model. Penghitungan akurasi dilakukan untuk mengetahui sejauh mana ketepatan model dalam melakukan prediksi pada data yang belum pernah digunakan sebelumnya.