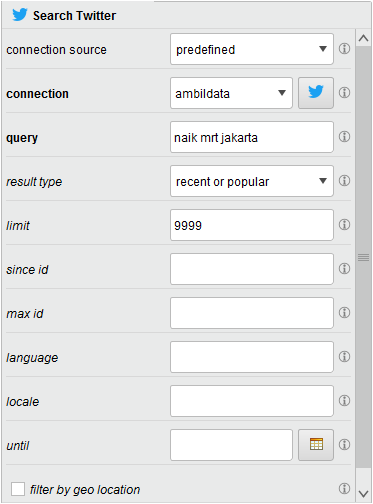
**BAB 4**

**HASIL DAN PEMBAHASAN**

**4.1 Pengumpulan Data *Tweet***

Data yang dikumpulkan bersumber dari media sosial Twitter. Data *tweet* yang digunakan merupakan data *tweet* yang berhubungan dengan MRT Jakarta. Pengumpulan data tweet dilakukan menggunakan *software* rapidminer dengan menggunakan operator *search* Twitter.





**Gambar 4.1 Rapidminer Operator *Search* Twitter**

Gambar 4.1 menggambarkan operator *search* Twitter dari *software* rapidminer yang digunakan untuk melakukan *crawling* data Twitter dengan menggunakan Twitter API. Dalam melakukan *crawling* dengan menggunakan operator *search* Twitter terdapat parameter yang dapat diatur agar data yang didapatkan sesuai dengan kebutuhan. Parameter *query* berfungsi untuk mengatur *keyword* dari *tweet* yang ingin didapatkan dan terdapat parameter *limit* untuk menentukan batas maksimal jumlah *tweet* yang didapatkan.

Berikut ini merupakan contoh data *tweet* yang dikumpulkan menggunakan *software* rapidminer dapat dilihat pada tabel 4.1.

**Tabel 4.1 Contoh Data *Tweet***

|  |  |
| --- | --- |
| No | *Tweet* |
| 1 | Naik @mrtjakarta moda transportasi modern keren deh pokok nya ???? https://t.co/MS9Ws4pJDo |
| 2 | @mrtjakarta tolong disediakan atm dong di stasiun....susah bgt niy mau narik uang cash. ?? |
| 3 | @upena @mrtjakarta Semenjak ada dirimu. Dunia terasa indahnya. |
| 4 | @dondiindrayana @mrtjakarta @DKIJakarta Memang baunya sangat menyengat dan membuat sesak napas |
| 5 | @detikcom anak kecil juga tau. kami warga jakarta tidak butuh analisis beginian. Kami butuh solusinya. Misalnya percepat itu MRT tahap Kedua, Tahap Ketiga; LRT dan seterusnya. |

**Sumber: www.twitter.com**

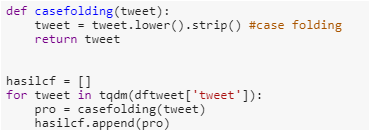
* 1. ***Preprocessing* Data**

Sebelum data *tweet* digunakan untuk melakukan analisis, langkah yang perlu dilakukan sebelumnya adalah mengurangi atau menghapus beberapa kata sehingga sistem dapat melakukan klasifikasi data dengan lebih cepat dan juga lebih akurat [7]. Tahapan *preprocessing* data terdiri dari *case folding*, *cleaning*, *negation handling*, dan *stopword removal.*

**4.2.1 *Case Folding***

Pada tahap ini program akan mengubah semua huruf pada data *tweet* menjadi huruf kecil. Hal ini dilakukan agar data *tweet* yang ada konsisten menggunakan huruf kecil.

Berikut ini merupakan *listing* programyang digunakan dalam melakukan proses *case folding* :



Contoh hasil dari proses *case folding* dapat dilihat pada tabel 4.2.

**Tabel 4.2 Hasil Proses *Case Folding***

|  |  |
| --- | --- |
| No | Hasil *Case Folding* |
| 1 | naik @mrtjakarta moda transportasi modern keren deh pokok nya ???? https://t.co/ms9ws4pjdo |
| 2 | @mrtjakarta tolong disediakan atm dong di stasiun....susah bgt niy mau narik uang cash. ?? |
| 3 | @upena @mrtjakarta semenjak ada dirimu. dunia terasa indahnya. |
| 4 | @dondiindrayana @mrtjakarta @dkijakarta memang baunya sangat menyengat dan membuat sesak napas |
| 5 | @detikcom anak kecil juga tau. kami warga jakarta tidak butuh analisis beginian. kami butuh solusinya. misalnya percepat itu mrt tahap kedua, tahap ketiga; lrt dan seterusnya. |

**Sumber: Data yang telah diolah**

**4.2.2 *Cleaning***

Pada tahap ini program akan menghapus simbol, angka, dan url yang terdapat pada data *tweet*. Hal ini dilakukan untuk menghapus karakter-karakter yang tidak digunakan dalam analisis sentimen.

Berikut ini merupakan *listing* programyang digunakan dalam melakukan proses *cleaning*:



Contoh hasil dari proses *cleaning* dapat dilihat pada tabel 4.3.

**Tabel 4.3 Hasil Proses *Cleaning***

|  |  |
| --- | --- |
| No | Hasil *Cleaning* |
| 1 | naik moda transportasi modern keren deh pokok nya |
| 2 | tolong disediakan atm dong di stasiun susah bgt niy mau narik uang cash |
| 3 | semenjak ada dirimu dunia terasa indahnya |
| 4 | memang baunya sangat menyengat dan membuat sesak napas |
| 5 | anak kecil juga tau kami warga jakarta tidak butuh analisis beginian kami butuh solusinya misalnya percepat itu mrt tahap kedua tahap ketiga lrt dan seterusnya |

**Sumber: Data yang telah diolah**

**4.2.3 *Negation Handling***

Pada tahap ini program akan menggabungkan kata negasi “tidak” dan “belum” dengan kata yang muncul setelah kata negasi. Kata negasi akan dihubungkan dengan kata selanjutnya menjadi “tidak\_” + kata dan “belum\_”+ kata.

Berikut ini merupakan *listing* programyang digunakan dalam melakukan proses *negation handling*:



Contoh hasil dari proses *stemming* dapat dilihat pada tabel 4.4.

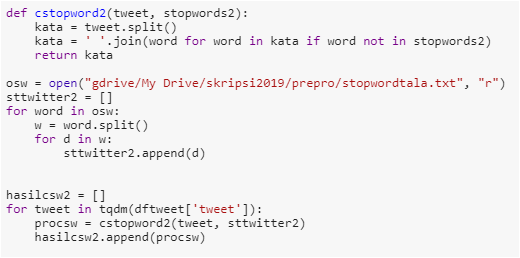
**Tabel 4.4 Hasil Proses *Negation Handling***

|  |  |
| --- | --- |
| No | Hasil *Negation Handling* |
| 1 | moda transportasi modern keren deh pokok nya |
| 2 | tolong sedia atm stasiun susah bgt niy narik uang cash |
| 3 | semenjak ada diri dunia indah |
| 4 | bau sengat sesak napas |
| 5 | anak kecil juga tau kami warga jakarta tidak\_butuh analisis beginian kami butuh solusinya misalnya percepat itu mrt tahap kedua tahap ketiga lrt dan seterusnya |

**Sumber: Data yang telah diolah**

**4.2.4 *Stopword Removal***

Pada tahap ini program akan menghapus kata pada *tweet* yang terdapat pada kamus kata *stopword*. Kamus *stopword* disimpan dalam file .txt yang berisi 758 kata. Berikut ini merupakan *listing* programyang digunakan dalam melakukan proses *stopword removal* :



Contoh hasil dari proses *stopword removal* dapat dilihat pada tabel 4.5.

**Tabel 4.5 Hasil Proses *Stopword Removal***

|  |  |
| --- | --- |
| No | Hasil *Stopword Removal* |
| 1 | moda transportasi modern keren deh pokok nya |
| 2 | tolong disediakan atm stasiun susah bgt niy narik uang cash |
| 3 | semenjak ada dirimu dunia indahnya |
| 4 | baunya menyengat sesak napas |
| 5 | anak tau warga jakarta tidak\_butuh analisis butuh solusinya percepat mrt tahap tahap ketiga lrt |

**Sumber: Data yang telah diolah**

**4.3 *Labelling* Dataset**

Setelah proses *preprocessing* selesai dilakukan, maka tahap selanjutnya adalah proses *labelling*. Proses *labelling* dilakukan karena pada tahap selanjutnya akan dilakukan pembuatan model klasifikasi menggunakan pendekatan *machine learning* yang bersifat *supervised learning*. Sehingga diperlukan data yang sudah diklasifikasi untuk melatih model klasifikasi. Pendekatan yang digunakan pada proses *labelling* ini adalah pendekatan *lexicon* dengan menggunakan kamus kata positif dan negatif. Berikut ini merupakan *listing* programyang dilakukan dalam proses *labelling*:



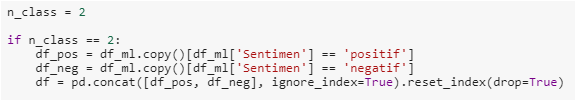
Berikut ini merupakan contoh hasil *labelling* dari data tweet dapat dilihat pada tabel 4.6.

**Tabel 4.6 Hasil Proses *Labelling***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tweet | Hasil Preprocessing | Kata Positif | Kata Negatif | Hasil Labelling |
| Naik @mrtjakarta moda transportasi modern keren deh pokok nya ???? https://t.co/MS9Ws4pJDo | moda transportasi modern keren deh pokok nya | 2 | 0 | positif |
| @mrtjakarta tolong disediakan atm dong di stasiun....susah bgt niy mau narik uang cash. ?? | tolong disediakan atm stasiun susah bgt niy narik uang cash | 0 | 2 | negatif |
| @upena @mrtjakarta Semenjak ada dirimu. Dunia terasa indahnya. | semenjak ada dirimu dunia indahnya | 1 | 0 | positif |
| @dondiindrayana @mrtjakarta @DKIJakarta Memang baunya sangat menyengat dan membuat sesak napas | baunya menyengat sesak napas | 0 | 1 | negatif |
| @detikcom anak kecil juga tau. kami warga jakarta tidak butuh analisis beginian. kami butuh solusinya. misalnya percepat itu mrt tahap kedua, tahap ketiga; lrt dan seterusnya. | anak tau warga jakarta tidak\_butuh analisis butuh solusinya percepat mrt tahap tahap ketiga lrt | 0 | 1 | negatif |

**Sumber: Data yang telah diolah**

Pada tabel 4.6 terdapat kata yang diberi warna biru dan merah. Kata yang berwarna biru menandakan bahwa kata tersebut terdapat pada kamus kata positif, sedangkan kata yang diberi warna merah menandakan bahwa kata tersebut terdapat pada kamus kata negatif. Berdasarkan dari proses *labelling* diperoleh hasil klasifikasi sejumlah 549 untuk klasifikasi positif, 323 untuk klasifikasi netral, dan 268 untuk klasifikasi negatif. Setelah proses *labelling* berhasil dilakukan, data *tweet* dengan hasil *labelling* positif dan negatif akan dipisahkan untuk digunakan dalam proses selanjutnya. Berikut ini merupakan *listing* program yang digunakan untuk memisahkan data *tweet* dengan hasil *labelling* positif dan negatif:



**4.4 Visualisasi**

Visualisasi dilakukan dengan tujuan untuk mendapatkan informasi berdasarkan gambar yang dihasilkan menggunakan dataset. Visualisasi dilakukan dengan menggunakan *library* matplotlib dan nltk dalam menampilkan diagram dan *library* WordCloud untuk menampilkan *word cloud*.

**4.4.1 Visualisasi *Word Cloud***

Visualisasi *word cloud* bertujuan untuk menampilkan gambar berdasarkan kata yang terdapat pada dataset yang telah melalui proses *preprocessing*. Pada visualisasi *word cloud* frekuensi kemunculan kata pada dataset menjadi acuan. Semangkin sering kata muncul maka ukuran kata pada visualisasi *word cloud* akan semangkin besar. Berikut ini merupakan *listing* program yang digunakan untuk visualisasi data menggunakan *word cloud*:



Visualisasi *word cloud* dapat dilihat pada gambar 4.2.

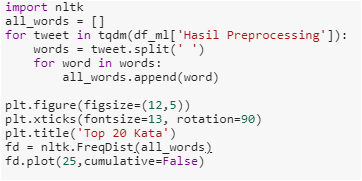


**Gambar 4.2 Visualisasi *Word Cloud***

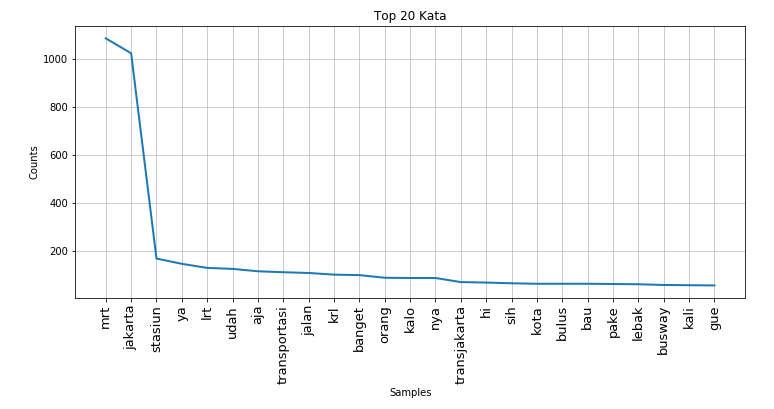
**Sumber: Data yang telah diolah**

**4.4.2 Visualisasi Frekuensi Kata**

Visualisasi frekuensi kata bertujuan untuk mendapatkan informasi mengenai kata yang memiliki frekuensi paling banyak pada dataset yang sudah melalui proses *preprocessing*. Berikut ini merupakan *listing* program yang digunakan untuk melakukan visualisasi frekuensi kata:



Visualisasi 20 kata dengan frekuensi paling banyak setelah proses *preprocessing* dilakukandapat dilihat pada gambar 4.3.

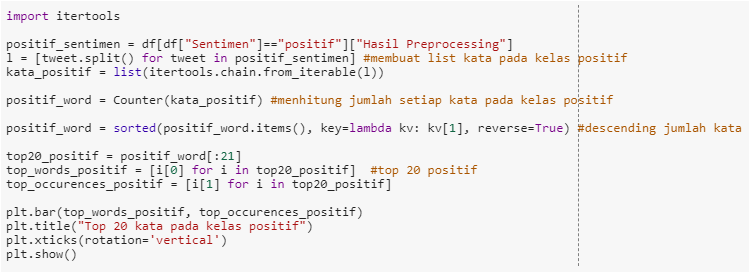


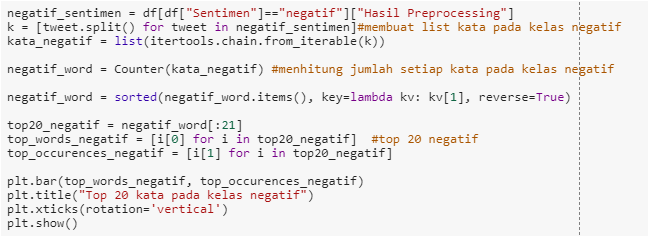
**Gambar 4.3 Visualisasi Frekuensi Kata**

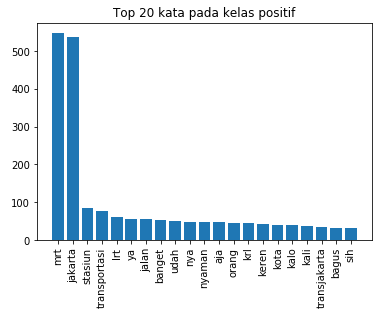
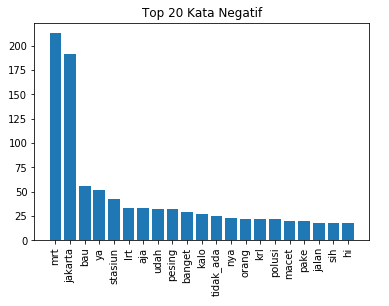
**Sumber: Data yang telah diolah**

**4.4.3 Visualisasi Dataset Positif dan Negatif**

Visualisasi dataset positif dan negatif bertujuan untuk mendapatkan informasi mengenai kata yang memiliki frekuensi paling banyak pada dataset bersentimen positif dan negatif. Berikut ini merupakan listing program yang digunakan untuk melakukan visualisasi dataset bersentimen positif dan negatif:





Berikut ini merupakan visualisasi 20 kata dengan frekuensi paling banyak pada dataset bersentimen positif dan negatif dapat dilihat pada gambar 4.4.

**Gambar 4.4 Visualisasi Kata Positif dan Negatif**

**Sumber: Data yang telah diolah**

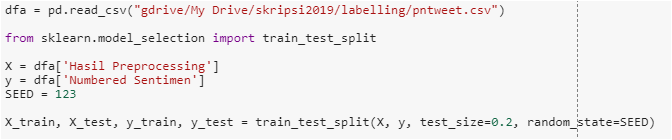
Kata dengan frekuensi tertinggi ini akan memiliki pengaruh yang kuat dalam pelatihan model klasifikasi Multinomial Naive Bayes*.* Hal ini diakibatkan karena model melakukan pembelajaran berdasarkan frekuensi kemunculan kata dan probabilitas kata tersebut muncul pada kelas tertentu.

**4.5 *Modelling***

*Modelling* dilakukan untuk membuat model klasifikasi baru dengan metode Multinomial Naive Bayes terdapat beberapa tahapan yang dilakukan. Setiap tahap yang dilakukan saling berkaitan dan dilakukan untuk menghasilkan model Multinomial Naive Bayes Classifier.

**4.5.1 Pembagian Dataset**

Pembagian dataset dilakukan menggunakan 817 data dengan perbandingan 80% data *training* dan 20% data *testing*. Pembagian data dilakukan secara acak oleh program. Berikut ini merupakan *listing* program dari proses pembagian dataset:



Data *training* (X\_train, y\_train) dan data *testing* (X\_test, y\_test). Contoh dari data *training* dan data *testing* dapat dilihat pada tabel 4.7 dan 4.8.

**Tabel 4.7 Contoh Data *Training***

|  |  |  |
| --- | --- | --- |
| Index | X\_train | y\_train |
| 274 | pantes kaya bau pesing salut petugas sigap langsung bersihkan | 1 |
| 676 | gw jg udah x pas ngetap error disuruh ngetap lg saldo kesedot double saksi | 0 |
| 150 | tidak\_ke pusat jakarta tidak\_naik transportasi tidak\_jalan jalan jakarta bolak kantor kosan menit jalan kaki kangen krl busway eh mrt | 1 |

**Sumber: Data yang telah diolah**

**Tabel 4.8 Contoh Data Testing**

|  |  |  |
| --- | --- | --- |
| Index | X\_test | y\_test |
| 185 | nyobain mrt jakarta suka deh tidak\_di kopo mrt | 1 |
| 210 | berkat org teroris masup jakarta eo ketemunya sexy killer mrt wkwkkwkwk | 1 |
| 587 | udh neng jakarta blm nyobain mrt hiks sedi | 0 |

**Sumber: Data yang telah diolah**

X\_train dan X\_test merupakan hasil dari *preprocessing* data *tweet. S*edangkan y\_train dan y\_test merupakan representasi hasil *labelling*, yang dimana nilai 0 merepresentasikan hasil *labelling* negatif dan 1 merepresentasikan positif.

**4.5.2 Seleksi dan Ekstraksi Fitur**

Untuk dapat menggunakan *machine learning* seperti Multinomial Naive Bayes Classifier, data yang digunakan harus dikonversi terlebih dahulu menjadi bentuk numerik. Hal ini disebabkan karena mesin tidak dapat membaca teks, sehingga teks harus diubah menjadi bentuk numerik atau vektor. Seleksi dan ekstraksi fitur dapat dilakukan dengan menggunakan CountVectorizer yang disediakan oleh *library* scikit-learn. Berikut merupakan *listing* program yang digunakan untuk melakukan seleksi dan ekstraksi fitur:



Pada CountVectorizer terdapat beberapa parameter yang dapat diatur untuk melakukan seleksi fitur. Berikut ini merupakan parameter yang dapat diatur pada CountVectorizer untuk melakukan seleksi fitur dapat dilihat pada tabel 4.9.

**Tabel 4.9 Parameter pada CountVectorizer**

|  |  |
| --- | --- |
| Parameter | Kegunaan |
| max\_df | Untuk memilih fitur dengan jumlah kemunculan tidak boleh lebih dari nilai yang ada pada parameter max\_df dengan nilai max\_df adalah bilangan bulat. |
| min\_df | Untuk memilih fitur dengan jumlah kemunculan tidak boleh kurang dari nilai yang ada pada parameter min\_df dengan nilai min\_df adalah bilangan bulat. |
| max\_features | Untuk memilih fitur dengan jumlah kemunculan terbanyak sejumlah nilai yang ada pada parameter max\_features dengan nilai max\_df adalah bilangan bulat. |

**Sumber: www.scikit-learn.org**

**4.5.2.1 Pengaruh Fitur Terhadap Model Klasifikasi**

Berdasarkan hasil dari uji coba model Multinomial Naive Bayes Classifier menggunakan 653 data *training* dan 164 data *testing* dengan nilai dari parameter min\_df yang berbeda diperoleh hasil seperti pada tabel 4.10.

**Tabel 4.10 Pengaruh Fitur Terhadap Model Klasifikasi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Min\_df | Jumlah Fitur | Test Accuracy | Average Precision | Average Recall |
| 1 | 3201 | 80% | 84% | 74% |
| 2 | 1112 | 82% | 85% | 77% |
| 3 | 670 | 83% | 85% | 77% |
| 4 | 454 | 80% | 81% | 75% |
| 5 | 340 | 81% | 82% | 76% |

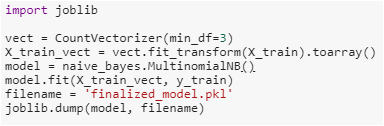
**Sumber: Data yang telah diolah**

Berdasarkan pada tabel 4.10 dapat disimpulkan sebagai berikut:

1. Semangkin besar nilai dari parameter min\_df, maka jumlah fitur yang digunakan pada ekstraksi fitur semangkin berkurang.
2. Jumlah fitur yang digunakan dalam *training* model klasifikasi mempengaruhi kemampuan model dalam melakukan prediksi terhadap data *testing*.
3. *Test accuracy* atau akurasi uji coba terhadap data *testing* terbaik terjadi pada saat fitur yang digunakan berjumlah 670 dengan frekuensi kemunculan fitur atau kata minimal 3 kali pada data *training* atau min\_df = 3*.* Nilai *test accuracy* sebesar 83%, nilai *average precision* sebesar 85%, dan nilai *average* *recall* sebesar 77%.

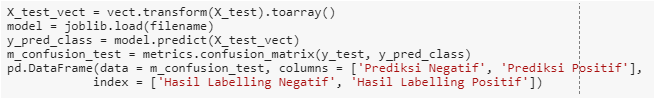
**4.5.3 *Training* Model**

Setelah melakukan ekstraksi fitur pada dataset, selanjutnya dilakukan pelatihan model atau *training* model. Hal ini dilakukan untuk menghasilkan model klasifikasi Multinomial Naive Bayes. *Training* dapat dilakukan dengan menggunakan MultinomialNB yang disediakan oleh *library* scikit-learn. Pada MultinomialNB terdapat parameter yang alpha yang digunakan untuk melakukan *smoothing*, alpha bernilai satu apabila tidak ditentukan nilainya. Model yang telah dilatih dapat disimpan dalam format pkl dengan menggunakan *library* joblib. Berikut merupakan *listing* program yang digunakan untuk melakukan *training* model dan menyimpan model:



**4.5.4 Evaluasi Model**

Evaluasi model dilakukan setelah model klasifikasi telah melewati proses *training* dan telah disimpan dalam format .pkl. Evaluasi model dilakukan dengan melakukan uji coba melakukan prediksi terhadap data *testing.* Berikut ini merupakan *listing* program yang digunakan untuk melakukan evaluasi model Multinomial Naive Bayes Classifier dengan parameter yang telah ditentukan:



Berdasarkan hasil dari uji coba model Multinomial Naive Bayes Classifierterhadap data *testing* diperoleh *confusion matrix* seperti pada tabel 4.11.

**Tabel 4.11 *Confusion Matrix* Hasil *Testing***

|  |  |  |
| --- | --- | --- |
| Hasil Labelling | Prediksi Model | |
| Negatif | Positif |
| Negatif | TN = 34 | FP = 24 |
| Positif | FN = 4 | TP = 102 |

**Sumber: Data yang telah diolah**

Berdasarkan *confusion matrix* pada tabel 4.13 dapat diperoleh hasil perhitungan *accuracy*, *recall*, dan *precision* dengan perhitungan sebagai berikut:

1. Menghitung *Accuracy*





1. Menghitung *Recall*







1. Menghitung *Precision*



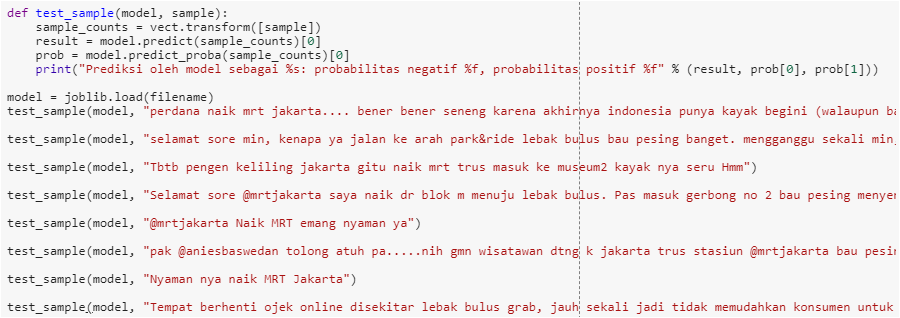




Berdasarkan hasil perhitungan di atas dapat diketahui bahwa hasil uji coba menggunakan data *testing* menghasilkan tingkat *accuracy* 82,9%, *average recall* sebesar 77,3%, dan *average precision* sebesar 85,18%. Model klasifikasi mampu melakukan prediksi data *testing* dengan jumlah benar sejumlah 136 dari total data *testing* sejumlah 164 data.

**4.6 Uji Coba Menggunakan Data Baru**

Model Multinomial Naive Bayes Classifier telah dilatih dan dievaluasi pada proses sebelumnya. Pada tahap ini model klasifikasi akan melakukan prediksi terhadap data baru yang belum pernah digunakan pada proses sebelumnya. Berikut ini merupakan *listing* program yang digunakan untuk melakukan pengujian:



Data baru yang digunakan untuk uji coba dapat dilihat pada tabel 4.12.

**Tabel 4.12 Data *Tweet* Baru**

|  |  |
| --- | --- |
| Data | Data Tweet Baru |
| 1 | perdana naik mrt jakarta.... bener bener seneng karena akhirnya indonesia punya kayak begini (walaupun baru di ibu kota, it’s okay progress takes time) |
| 2 | selamat sore min, kenapa ya jalan ke arah park&ride lebak bulus bau pesing banget. mengganggu sekali min, harap diperhatikan dan cari solusi @mrtjakarta" |
| 3 | Tbtb pengen keliling jakarta gitu naik mrt trus masuk ke museum2 kayak nya seru Hmm |
| 4 | Selamat sore @mrtjakarta saya naik dr blok m menuju lebak bulus. Pas masuk gerbong no 2 bau pesing menyengat sekali |
| 5 | @mrtjakarta Naik MRT emang nyaman ya |
| 6 | pak @aniesbaswedan tolong atuh pa.....nih gmn wisatawan dtng k jakarta trus stasiun @mrtjakarta bau pesing...trus mereka review k sosmed...😩 kelar dh jakarta hadeeehhh |
| 7 | Nyaman nya naik MRT Jakarta |
| 8 | Tempat berhenti ojek online disekitar lebak bulus grab, jauh sekali jadi tidak memudahkan konsumen untuk naik @mrtjakarta |

**Sumber: www.twitter.com**

Hasil prediksi sentimen menggunakan model Multinomial Naive Bayes Classifier dapat dilihat pada tabel 4.13.

**Tabel 4.13 Hasil Prediksi Data Baru**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data | Probabilitas Positif | Probabilitas Negatif | Hasil Prediksi Model | Pendapat Subjektif |
| 1 | 0.999864 | 0.000136 | Positif | Positif |
| 2 | 0.001090 | 0.998910 | Negatif | Negatif |
| 3 | 0.918014 | 0.081986 | Positif | Positif |
| 4 | 0.028456 | 0.971544 | Negatif | Negatif |
| 5 | 0.922671 | 0.077329 | Positif | Positif |
| 6 | 0.000452 | 0.999548 | Negatif | Negatif |
| 7 | 0.984281 | 0.015719 | Positif | Positif |
| 8 | 0.129002 | 0.870998 | Negatif | Negatif |

**Sumber: Data yang telah diolah**

Berdasarkan tabel 4.13 dapat disimpulkan bahwa model Multinomial Naive Bayes Classifier melakukan prediksi dengan menghitung nilai probabilitas dan membandingkan nilai probabilitas dari hasil perhitungan. Jika nilai probabilitas positif lebih besar dari nilai probabilitas negatif maka *tweet* diklasifikasi bersentimen positif. Jika nilai probabilitas negatif lebih besar dari nilai probabilitas positif maka *tweet* diklasifikasi bersentimen negatif. Hasil dari klasifikasi yang dilakukan oleh model klasifikasi dibandingkan dengan pendapat subjektif dari penulis. Hasil uji coba pada tabel 4.13 menunjukkan bahwa Model Multinomial Naive Bayes Classifier mampu mengklasifikasikan data baru yang belum pernah digunakan dalam proses sebelumnya.