

menggunakan dataset serta rasio pembagian data set yang sama dengan algoritma Multinomial Naive Bayes yaitu 20% data *test* dan 80% data *training* pada data set yang sama seperti algoritma *Multinomial Naive Bayes*. Pengujian ini menggunakan *library* dari scikit-learn RandomForest pada python dengan *source code* sebagai berikut:

1. *From sklear.ensemble import RandomForestClassifier*

Library python Scikit learn digunakan untuk tahapan pemodelan algoritma dengan modul yang digunakan *RandomForestClassifier*. Hasil dari digunakan *library python* tersebut untuk membuat pemodelan berdasarkan algoritma *RandomForestClassifier* yang dapat mengklasifikasikan data berdasarkan jumlah kata yang muncul pada setiap kalimat.

2. *Classifier = RandomForestClassifier(n_estimators=100)*

Source code yang bertujuan untuk pendeklarasian rumus algoritma yang digunakan untuk pemodelan.

3. *Classifier.fit (X_train, y_train)*

Source code yang bertujuan untuk pendeklarasian bentuk pemodelan yang dibuat untuk melakukan uji algoritma pada data *training*.

4. *Random_forest_pred = classifier.predict(X_test)*

Source code yang bertujuan untuk pendeklarasian bentuk prediksi dari *source code* rumus pemodelan yang sudah dibuat menggunakan data *test* yang dimana akan menghasilkan hasil prediksi algoritma untuk mengklasifikasikan data *test* berupa data hasil prediksi algoritma dengan data *actual* yang sudah ada.

Pemodelan ini dapat mendefinisikan data sentimen positif diprediksi secara benar sebanyak 261 data dan sentimen negatif sebanyak 325 data diprediksi secara benar dari 733 data *test* yang digunakan.

5. `Print(Classification_report(y_test, random_forest_pred))`

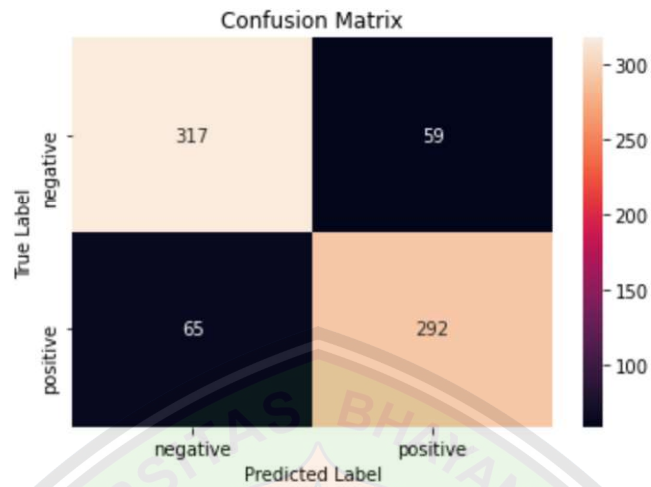
Fungsi print yang digunakan untuk menampilkan hasil dari *accuracy* pengujian dari pemodelan dengan algoritma *RandomForestClassifier*. Berdasarkan hasil dari prediksi pada pemodelan ini menghasilkan nilai *accuracy* 0.81 atau 81%.

4.1.6 Evaluasi Model

Tahapan selanjutnya setelah dilakukannya pemodelan yaitu evaluasi model. Evaluasi model yang digunakan yaitu *confusion matrix*. Pada *confusion matrix* terdapat beberapa informasi yang digunakan untuk dilakukannya evaluasi yaitu *Accuracy*, *Presisi*, *Recall*, dan *F1-Score*. Tujuan dari dilakukannya evaluasi model yaitu untuk mengetahui performa algoritma mana yang lebih baik digunakan untuk pemodelan machine learning pada data set yang dimiliki. Hasil dari performa model ini akan ditampilkan dengan heatmap *confusion matrix* dan dilengkapi dengan perhitungan matematika dengan persamaan matematika.

1. Evaluasi Model Algoritma Multinomial Naive Bayes

Evaluasi model untuk Algoritma Multinomial Naive Bayes berdasarkan hasil uji pemodelan yang telah dilakukan pada sub bab sebelumnya, maka data evaluasi yang didapatkan dalam bentuk *confusion matrix* seperti diperlihatkan pada Gambar 4.3.



Gambar 4.1 Heatmap Confusion Matrix Algoritma Multinomial Naïve Bayes

Gambar 4.1 diperoleh beberapa informasi data sebagai berikut:

- 1) *True Negatif* (TN): Data dengan label negatif yang diprediksi benar sebagai label negatif sebanyak 317
- 2) *True Positif* (TP): Data dengan label positif yang diprediksi benar sebagai label positif sebanyak 292
- 3) *False Negatif* (FN): Data dengan label positif yang diprediksi salah sebagai label negatif sebanyak 65
- 4) *False Positif* (FP): Data dengan label negatif yang diprediksi salah sebagai label positif sebanyak 59

Data yang terdapat pada heatmap confusion matrix selanjutnya dilakukan perhitungan secara manual dengan persamaan matematika sebagai berikut:

a. *Accuracy* (Akurasi)

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} = \frac{(292+317)}{(292+59+65+317)} = \frac{609}{733} = 0,83082$$