In Search

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

Analisis Sentimen Dengan Metode Klasifikasi Naïve Bayes dan Seleksi Fitur *Chi-Square*

Albert Raja Harungguan, Herlina Napitupulu, Firdaniza Firdaniza
Departemen Matematika, Fakultas MIPA, Universitas Padjadjaran
Email: albert19002@mail.unpad.ac.id; herlina.napitupulu@mail.unpad.ac.id; firdaniza@mail.unpad.ac.id.

Abstrak

Analisis sentimen merupakan metode komputasi untuk mengevaluasi teks dengan tujuan menentukan emosi yang terkandung di dalamnya. Analisis sentimen dapat dilakukan dengan pendekatan pembelajaran mesin. Salah satu metode pembelajaran mesin yang sering digunakan untuk analisis sentimen adalah Klasifikasi Naïve Bayes, yaitu metode yang menggunakan teorema Bayes sebagai dasar untuk melakukan klasifikasi. Penggunaan Klasifikasi Naïve Bayes juga dapat dikombinasikan dengan seleksi fitur. Studi sebelumnya menunjukkan bahwa seleksi fitur *Chi-Square* merupakan satu metode seleksi fitur yang efektif, yang mana merupakan proses untuk memilih kata yang paling merepresentasikan data berdasarkan nilai *Chi-Square* untuk mempercepat proses komputasi. Penelitian ini bertujuan untuk memperoleh model Klasifikasi Naïve Bayes dari data teks yang telah melalui tahap seleksi fitur *Chi-Square*.

Kata Kunci: analisis sentimen, Naïve Bayes, *Chi-Square*.

Abstract

Sentiment analysis is a computational method for evaluating text with the aim of determining the emotions contained in it. Sentiment analysis can be done with a machine learning approach. One machine learning method that is often used for sentiment analysis is Naïve Bayes Classification, which uses Bayes' theorem as the basis for classification. The use of Naïve Bayes Classification can also be combined with feature selection. Previous studies have shown that Chi-Square feature selection is an effective feature selection method, which is the process of selecting words that best represent the data based on the Chi-Square value to speed up the computational process. This research aims to obtain a Naïve Bayes Classification model from text data that has gone through the Chi-Square feature selection stage.

Keywords: *sentiment analysis, Naïve Bayes, Chi-Square.*

1 PENDAHULUAN

Analisis sentimen merupakan metode komputasi yang bertujuan untuk menentukan penilaian dari teks yang kemudian diklasifikasikan ke dalam kelas tertentu (Liu, 2012). Analisis sentimen dapat dilakukan dengan menggunakan *machine learning*. Klasifikasi Naïve Bayes merupakan salah satu metode *machine learning* yang banyak digunakan untuk pengklasifikasian sentimen karena memiliki tingkat akurasi dan kecepatan yang tinggi (Liu dkk., 2013).

Terdapat beberapa penelitian sebelumnya yang menggunakan Klasifikasi Naïve Bayes, seperti Permadi (2020) melakukan analisis sentimen terhadap 1000 ulasan restoran yang berada di Singapura dengan dua kelas klasifikasi sentimen yaitu positif dan negatif. Penelitian ini memberikan

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

kesimpulan bahwa ulasan terhadap restoran cenderung positif dengan nilai *performance metrics*: *precision* sebesar 73,02%, *recall* sebesar 74%, dan *accuracy* sebesar 73,33%. Selain itu, Kustanto dkk. (2021) menganalisis ulasan pengguna terhadap aplikasi *Mobile* JKN dengan kelas klasifikasi yang sama, hasilnya dari 13417 ulasan menunjukkan bahwa ulasan pengguna aplikasi cenderung negatif dengan nilai *performance metrics*: *precision* sebesar 94%, *recall* sebesar 93%, *accuracy* sebesar 93%, dan F1-*Score* sebesar 93%.

Penggunaan *machine learning* untuk masalah klasifikasi dapat dipadukan dengan metode seleksi fitur untuk meningkatkan efisiensi model. Terdapat beberapa seleksi fitur, yaitu: *Chi-Square*, *Information Gain*, frekuensi dokumen, dan *Improved Gini Index*. Iqbal dkk. (2020) membandingkan seleksi fitur untuk klasifikasi teks dimana dari segi *precision* seleksi fitur terbaik adalah Chi-Square.

Penelitian terdahulu yang menggunakan metode Klasifikasi Naïve Bayes dipadukan dengan seleksi fitur *Chi-Square* dilakukan oleh Nurhayati dkk. (2019) dalam menganalisis ulasan terhadap film berbahasa Indonesia. Hasil *accuracy* yang diperoleh menggunakan perpaduan Klasifikasi Naïve Bayes dengan seleksi fitur *Chi-Square* yaitu sebesar 90%. Hasil tersebut lebih baik dibandingkan dengan *accuracy* yang hanya menggunakan Klasifikasi Naïve Bayes yaitu sebesar 73,33%.

Berdasarkan hal tersebut, pada penelitian ini penulis menggunakan metode Klasifikasi Naïve Bayes dan seleksi fitur *Chi-Square* untuk melakukan analisis sentimen.

2 KAJIAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen atau *opinion mining* adalah bidang *natural language processing* (NLP) yang mempelajari dan menganalisis opini, evaluasi, sikap, penilaian, dan emosi seseorang terhadap suatu produk, organisasi, individu, isu, kegiatan, atau topik tertentu yang kemudian diklasifikasikan berdasarkan sentimennya. Analisis sentimen bermanfaat untuk berbagai aplikasi, termasuk pemasaran, pengambilan keputusan, dan pemantauan opini publik (Liu, 2012).

Menurut Medhat dkk. (2014), analisis sentimen untuk klasifikasi teks dapat dilakukan dengan pendekatan *machine learning*. Pendekatan *machine learning* secara garis besar dibagi menjadi dua yaitu *supervised* dan *unsupervised learning*. *Supervised learning* digunakan untuk kasus klasifikasi yang memiliki data latih dalam jumlah besar sedangkan *unsupervised learning* digunakan kasus klasifikasi yang memiliki data latih yang sedikit.

Supervised learning memiliki beberapa metode klasifikasi, yaitu:

(i) metode klasifikasi dengan probabilitas: Klasifikasi Naïve Bayes, Bayesian *Network* (BN), dan *Maximum Entropy* (ME);
(ii) metode klasifikasi linear: Klasifikasi *Support Vector Machines* (SVM), *Neural Network* (NN);
(iii) metode *Decision Tree*; dan metode *Rule-Based*.

2.2 Seleksi Fitur Chi-Square

Thaseen dan Kumar (2016) menjelaskan bahwa dalam seleksi fitur dengan pendekatan *Chi-Square*, kata pada dokumen dievaluasi dengan menghitung nilai statistik *Chi-Square* relatif terhadap kelas. Perhitungan seleksi fitur Chi-Square dinyatakan dalam persamaan (2.1) (Liu dan Setiono, 1995).

$$\chi^2(x_i) = \sum_{j=1}^{m} \frac{\left(N_{(x_i,c_j)} - E_{(x_i,c_j)}\right)^2}{E_{(x_i,c_j)}}, \qquad (1)$$

dengan

$$E_{(x_i,c_j)} = \frac{\left(\sum_{j=1}^{m} N_{(x_i,c_j)}\right)\left(\sum_{i=1}^{k} N_{(x_i,c_j)}\right)}{N}, \qquad (2)$$

dimana

m : jumlah kelas
k : jumlah kata
N : jumlah dokumen

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

x_i : kata ke-i yang muncul pada suatu dokumen, dimana $x_i \in X, X = \{x_1, x_2, \dots, x_k\}$

C_j : kelas klasifikasi ke-j, dimana $C_j \in C, C = \{C_1, C_2, \dots, C_m\}$

$N_{(x_i,C_j)}$: frekuensi munculnya x_i pada kelas C_j

$E_{(x_i,C_j)}$: frekuensi yang diharapkan munculnya x_i pada kelas C_j.

Semakin besar nilai $\chi^2(x_i)$, maka kata tersebut semakin tepat untuk representatif isi dokumen.

2.3 Klasifikasi Naïve Bayes

Menurut Liu dkk. (2013), Klasifikasi Naïve Bayes adalah salah satu pendekatan sederhana yang berdasar pada teorema Bayes dan bertujuan untuk melakukan klasifikasi dengan akurasi yang baik. Klasifikasi Naïve Bayes mengasumsikan setiap fitur pada suatu kelas bersifat independen (tidak bergantung) satu sama lain.

Proses Klasifikasi Naïve Bayes adalah sebagai berikut:

1. Probabilitas prior untuk masing-masing kelas C_j dapat dihitung menggunakan persamaan (3),
$$P(C_j) = \frac{N_{C_j}}{N} \quad (3)$$
dengan N_{C_j} adalah jumlah dokumen pada kelas C_j.
2. Probabilitas munculnya setiap kata pada masing-masing kelas klasifikasi dapat dihitung menggunakan persamaan (4),
$$P(x_i|C_j) = \frac{N_{(x_i,C_j)}}{N_{C_j}}. \quad (4)$$
Nilai $P(x_i|C_j) = 0$ ketika x_i tidak muncul pada kelas C_j, sehingga dibutuhkan metode pemulusan atau *smoothing* agar nilai probabilitas yang dihasilkan bukan nol. Menurut Kilimci dan Ganiz (2015), *laplace smoothing* adalah metode pemulusan yang umum digunakan dengan menambahkan parameter (α) ke dalam perhitungan. Jadi, probabilitas munculnya setiap kata pada masing-masing kelas klasifikasi adalah
$$P_\alpha(x_i|C_j) = \frac{\alpha + N_{(x_i,C_j)}}{V + N_{C_j}} \quad (5)$$
dengan asumsi $\alpha = 1$ dan V adalah banyaknya kata yang muncul setidaknya sekali pada keseluruhan dokumen.
3. Misalkan D adalah himpunan semua ulasan, dinotasikan dengan $D = \{d_1, d_2, \dots, d_N\}$. Probabilitas dokumen d_k termasuk kedalam kelas klasifikasi C_j dapat dihitung dengan teorema Bayes pada persamaan (6). Probabilitas ini disebut juga probabilitas posterior.
$$P(C_j|d_k) = \frac{P(C_j)P(d_k|C_j)}{P(d_k)} \quad (6)$$
$P(d_k)$ tidak bergantung pada C berdasarkan asumsi Naïve Bayes sehingga $P(d_k)$ sama untuk $k = 1,2, \dots, N$. Hal ini menyebabkan dalam penentuan probabilitas posterior maksimal atau *maximum a posteriori* (MAP) hanya pembilang dari persamaan (6) saja yang dibandingkan. Oleh karena itu, persamaan (6) dapat ditulis menjadi
$$P(C_j|d_k) \propto P(C_j) \prod_{i=1}^{N_{d_k}} [P(x_i|C_j)]^{t_{x_i}} \quad (7)$$
dengan N_{d_k} adalah banyaknya kata yang muncul setidaknya sekali pada dokumen d_k dan t_{x_i} adalah banyaknya kemunculan x_i pada dokumen d_k. Untuk menghindari nilai yang kecil, digunakan transformasi logaritma sehingga dapat ditulis menjadi persamaan (8)
$$\log P(C_j|d_k) \propto \log P(C_j) + \sum_{i=1}^{N_{d_k}} [t_{x_i} \log P(x_i|C_j)]. \quad (8)$$
4. Penentuan kelas untuk suatu ulasan diperoleh dengan mencari MAP menggunakan persamaan (9)

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

$$C_{MAP} = \max_{j=1,2,\dots,m} \left\{ \log P(C_j) + \sum_{i=1}^{N_{d_k}} \left[t_{x_i} \log P_\alpha (x_i | C_j) \right] \right\}. \quad (9)$$

2.4 Confusion Matrix

Menurut Géron (2022), *confusion matrix* adalah matriks yang digunakan untuk mengevaluasi kinerja model dalam melakukan klasifikasi. Elemen *confusion matrix* dapat dilihat pada Tabel 1.

Tabel 1. *Confusion matrix*

Nilai		Aktual	
		Positif	Negatif
Prediksi	Positif	*True Positive* (TP)	*False Positive* (FP)
	Negatif	*False Negative* (FN)	*True Negative* (TN)

Confussion matrix digunakan untuk menghitung *performance metrics*. Menurut Vakili dkk. (2020), *performance metrics* yang digunakan untuk masalah klasifikasi adalah sebagai berikut:

a. *Accuracy*
Accuracy adalah rasio antara jumlah data yang diprediksi dengan benar dengan jumlah keseluruhan elemen matriks. *Accuracy* dapat dihitung dengan persamaan (10),

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \quad (10)$$

b. *Precision*
Precision adalah rasio jumlah data yang diprediksi positif dengan benar dan jumlah data yang diprediksi positif. *Precision* dapat dihitung dengan persamaan (11),

$$Precision = \frac{TP}{TP + FP}. \quad (11)$$

c. *Recall*
Recall adalah rasio jumlah data yang diprediksi positif dengan benar dan jumlah data positif yang benar. *Recall* dapat dihitung dengan persamaan (12),

$$Recall = \frac{TP}{TP + FN}. \quad (12)$$

d. F1-*Score*
F1-*Score* adalah rata-rata harmonik antara *precision* dan *recall* yang hasilnya berkisar dari nol hingga satu. F1-*Score* dapat dihitung dengan persamaan (13),

$$\text{F1-}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \quad (13)$$

2.5 K-*Fold Cross Validation*

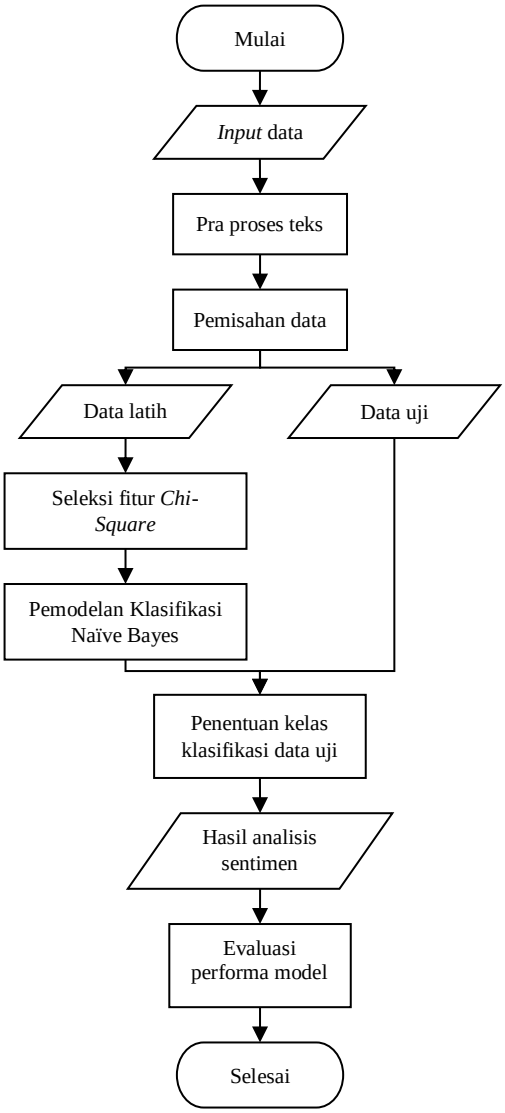
Menurut Pitria (2014), K-*Fold Cross Validation* adalah metode yang digunakan untuk mengetahui rata-rata keberhasilan suatu algoritma dengan melakukan pengacakan atribut *input* secara berulang sehingga algoritma tersebut teruji. Kohavi (1995) menyatakan bahwa jumlah *fold* yang dianjurkan menggunakan 10-*fold cross validation* untuk kasus klasifikasi.

Menurut Kohavi (1995), K-*Fold Cross Validation* diawali dengan membagi data (Z) secara acak menjadi k buah partisi, $Z_1, Z_2, \dots, Z_k$, dengan ukuran yang sama. Pelatihan dan pengujian model dilakukan sebanyak k iterasi, dimana untuk setiap $i \in \{1,2, \dots, k\}$, model dilatih dengan data $Z \backslash Z_i$ (Z tanpa Z_i) dan diuji dengan Z_i, dilanjutkan dengan perhitungan *performance metrics*. Kemudian, rata-rata dari *performance metrics* dari seluruh iterasi dihitung.

3 METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah Klasifikasi Naïve Bayes dan seleksi fitur *Chi-Square* untuk melakukan analisis sentimen dengan menggunakan bahasa pemrogrman Python. Diagram alir dapat dilihat pada Gambar *1*.

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023



Gambar 1. Diagram alir penelitian

4 HASIL DAN PEMBAHASAN

4.1 Import Library

Penggunaan Python untuk melakukan analisis sentimen dengan metode Klasifikasi Naïve Bayes dan seleksi fitur *Chi-Square* memerlukan beberapa *library* pendukung. Berikut *script* Python untuk pemanggilan *library*.

```
import pandas as pd
import numpy as np
import re
import string
from scipy.stats import chi2
import matplotlib.pyplot as plt
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
from
Sastrawi.StopWordRemover.StopWordRemoverFacto
ry import StopWordRemoverFactory
from sklearn.preprocessing import
LabelBinarizer
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn.model_selection import
train_test_split
from sklearn.model_selection import KFold
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix,
ConfusionMatrixDisplay,
classification_report, accuracy_score,
precision_score, recall_score, f1_score
```

4.2 Input Data

Data yang dikumpulkan adalah data sekunder berbentuk teks yang sudah memiliki label. Setelah data terkumpul dan disimpan pada file dengan format *comma-separated values*, dilakukan *input* data ke dalam *interpreter* Python dengan *script* sebagai berikut.

```
df = pd.read_csv('data.csv')
```

4.3 Pra proses teks

Langkah pra proses teks yang dilakukan adalah *case folding* yaitu mengubah semua huruf menjadi huruf kecil, *punctuation removal* yaitu untuk menghilangkan semua karakter selain huruf, *tokenizing* yaitu mengubah teks menjadi token atau bagian-bagian kecil, *normalization* yaitu mengubah kata yang tidak baku menjadi baku dengan menggunakan *set* data yang diperoleh dari internet, *stemming* yaitu menghilangkan imbuhan, dan *stopwords removal* yaitu menghilangkan kata-kata yang kurang bermakna.

Proses tersebut pada Python dilakukan dengan membuat *user defined function* sebagai berikut.

```
kamus = pd.read_csv('link')

factory = StemmerFactory()
stemmer = factory.create_stemmer()
stop_factory = StopWordRemoverFactory()
stopwords = stop_factory.get_stop_words()
```

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

```
def pra_proses_teks(text):
  text = text.lower()
  text = re.sub('\d',' ',text)
  text = text.replace('\w',' ')
  text = re.sub('[^a-zA-Z]',' ',text)

  text = word_tokenize(text)
  text = [stemmer.stem(w) for w in text]

  for i in range(len(text)):
    if text[i] in kamus['takbaku'].values:
      text[i] = kamus[kamus['takbaku'] ==
      text[i]]['baku'].values[0]

  temp_text = []
  for i in range(len(text)):
    if (text[i] not in stopwords):
      temp_text.append(text[i])

  text = ' '.join(temp_text)
  return text

df['cleaned_content'] =
df['content'].apply(lambda x:
cleaning(str(x)))
df.drop(df[df['cleaned_content']==''].index,
inplace = True)
df = df.reset_index(drop = True)
```

4.4 Pemisahan Data

Hasil dari pra proses teks dilakukan pemisahan data menjadi data latih dan data uji. Data dipartisi dengan rasio 90% sebagai data latih dan 10% sebagai data uji menggunakan *script* Python sebagai berikut.

```
X_train, X_test, y_train, y_test =
train_test_split(df['cleaned_content'],
df['class'], test_size = 0.1, shuffle =
False)
```

4.5 Seleksi Fitur Chi-Square

Data latih hasil pemisahan data dilakukan seleksi fitur *Chi-Square*. Seleksi fitur *Chi-Square* dilakukan untuk memilih fitur atau kata yang dapat mewakili data secara akurat. Pada penelitian ini, pemilihan kata pada proses pemodelan klasifikasi adalah dengan menentukan nilai *Chi-Square* terendah yang diterima dengan taraf signifikansi sebesar 0,05. Proses ini dilakukan dengan script Python sebagai berikut.

```
def feature_selection(X_train, y_train):
  vect = CountVectorizer()
  X_dtm = vect.fit_transform(X_train)
  df_X_dtm =
pd.DataFrame.sparse.from_spmatrix(X_dtm,
columns = vect.get_feature_names_out())
  df_X_dtm['class'] = list(y_train.values)
  df_X_dtm_pos =
df_X_dtm[df_X_dtm['class']=='Positif']
  df_X_dtm_neg =
df_X_dtm[df_X_dtm['class']=='Negatif']

  df_freq = pd.DataFrame({'kata':
list(df_X_dtm_pos.columns)[:-1], 'n_pos':
list(df_X_dtm_pos[list(df_X_dtm_pos.columns)[
:-1]].sum(axis=0)), 'n_neg':
list(df_X_dtm_neg[list(df_X_dtm_neg.columns)[
:-1]].sum(axis=0))})
  df_freq['total_n'] =
list(df_freq[['n_pos','n_neg']].sum(axis =
1))
  df_freq['exp_pos'] = [x *
df_freq['n_pos'].sum(axis=0)/len(X_train) for
x in df_freq['total_n']]
  df_freq['exp_neg'] = [x *
df_freq['n_neg'].sum(axis=0)/len(X_train) for
x in df_freq['total_n']]

  y_binarized =
LabelBinarizer().fit_transform(y_train)
  y_binarized = np.hstack((y_binarized, 1 -
y_binarized))

  observed = np.array([i for i in
y_binarized.T*X_dtm])
  expected = [df_freq['exp_pos'].values,
df_freq['exp_neg'].values]

  chisq = np.square(observed - expected) /
expected
  chisq_score = chisq.sum(axis = 0)

  df_chisq = pd.DataFrame(chisq_score.T,
index = vect.get_feature_names_out())
  df_chisq =
df_chisq.sort_values(df_chisq.columns[0],
ascending=False)
  df_chisq = df_chisq.reset_index()
  df_chisq.columns = ['word',
'chi_square_score']

  df_chisq_selected =
df_chisq[df_chisq['chi_square_score'] >=
chi2.ppf(1-.05, df=1)].reset_index(drop=True)
  df_chisq_rejected =
df_chisq[df_chisq['chi_square_score'] <
chi2.ppf(1-.05, df=1)].reset_index(drop=True)

  return df_X_dtm, df_freq, df_chisq,
df_chisq_selected, df_chisq_rejected

df_X_dtm, df_freq, df_chisq,
df_chisq_selected, df_chisq_rejected =
feature_selection(X_train, y_train)
```

4.6 Pemodelan Klasifikasi Naïve Bayes

Pemodelan Klasifiaksi Naïve Bayes dilakukan pada data latih yang telah melalui

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

proses seleksi fitur *Chi-Square*. *Script* Python yang digunakan adalah sebagai berikut.

```
def naive_bayes_classifier(X_train, X_test,
y_train, words_rejected):
  vect =
CountVectorizer(stop_words=words_rejected)
  vect.fit(X_train)

  X_train_dtm = vect.transform(X_train)
  X_test_dtm = vect.transform(X_test)

  nb = MultinomialNB()
  nb.fit(X_train_dtm, y_train)
  y_pred = nb.predict(X_test_dtm)

  return y_pred

y_pred = naive_bayes_classifier(X_train,
X_test, y_train,
list(df_chisq_rejected['word'].values))
```

4.7 Evaluasi Model Klasifikasi

Evaluasi performa model klasifikasi dilakukan dengan menghitung *confusion matrix* untuk menghitung nilai *performance metrics* menggunakan *script* Python berikut.

```
print('\nperformance metrics')
print('accuracy\t:',accuracy_score(y_test,
y_pred))
print('precision\t:',precision_score(y_test,
y_pred, pos_label='Positif'))
print('recall\t\t:',recall_score(y_test,
y_pred,  pos_label='Positif'))
print('f1_score\t:',f1_score(y_test,  y_pred,
pos_label='Positif'))
```

Selanjutnya menghitung rata-rata *performance metrics* dengan 10-*fold cross validation* untuk mendukung keajegan performa dengan *script* Python berikut.

```
X = df['cleaned_content']
y = df['class']
fold = KFold(n_splits=10, shuffle=True,
random_state=42)
fold_10 = fold.split(X, y)

score_10fold = {}
score_10fold['accuracy'] = []
score_10fold['precision'] = []
score_10fold['recall'] = []
score_10fold['f1_score'] = []

for k, (train, test) in enumerate(fold_10):
  df_X_dtm_fold, df_freq_fold, df_chisq_fold,
df_chisq_selected_fold,
df_chisq_rejected_fold =
feature_selection(X.iloc[train],
y.iloc[train])
  y_pred_fold =
naive_bayes_classifier(X.iloc[train],
X.iloc[test], y.iloc[train],
list(df_chisq_rejected_fold['word'].values))

score_10fold['accuracy'].append(accuracy_scor
e(y.iloc[test], y_pred_fold))
score_10fold['precision'].append(precision_sc
ore(y.iloc[test], y_pred_fold,
pos_label='Positif'))
score_10fold['recall'].append(recall_score(y.
iloc[test], y_pred_fold,
pos_label='Positif'))
score_10fold['f1_score'].append(f1_score(y.il
oc[test], y_pred_fold, pos_label='Positif'))

df_10fold =
pd.DataFrame.from_dict(score_10fold)

print('rata-rata performance metrics')
print('accuracy\t:',list(df_10fold.sum(axis=0
)/10)[0])
print('precision\t:',list(df_10fold.sum(axis=
0)/10)[1])
print('recall\t\t:',list(df_10fold.sum(axis=0
)/10)[2])
print('f1_score\t:',list(df_10fold.sum(axis=0
)/10)[3])
```

5 SIMPULAN

Berdasarkan pembahasan yang telah dilakukan, dapat disimpulkan bahwa analisis sentimen dapat dilakukan menggunakan metode Klasifikasi Naïve Bayes yang dipadukan dengan seleksi fitur *Chi-Square* dan menggunakan bahasa pemrograman Python.

DAFTAR PUSTAKA

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. *Sebastopol: O'Reilly Media, Inc.*

Iqbal, M., Abid, M.M., Khalid, M.N., dan Manzoor, A. (2020). Review of feature selection methods for text classification. *International Journal of Advanced Computer Research*, 10(49), pp: 138–152.

Kilimci, Z.H. dan Ganiz, M.C. (2015). Evaluation of classification models for language processing. *2015 International Symposium on Innovations in Intelligent*

p-ISSN: 2085-7993
e-ISSN: 2580-3239
Volume 22 No. 02, November 2023

SysTems and Applications (INISTA), pp. 1–8.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence* (*IJCAI*), 14(2), pp. 1137–1145.

Kustanto, N.S., Yulita, I.N., dan Sarathan, I. (2021). Sentiment analysis of Indonesia's national health insurance mobile application using Naïve Bayes algorithm. *2021 International Conference on Artificial Intelligence and Big Data Analytics*, pp. 38–42.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *San Rafael: Morgan & Claypool Publishers*.

Liu, B., Blasch, E., Chen, Y., Shen, D., dan Chen, G. (2013). Scalable sentiment classification for Big Data analysis using Naive Bayes Classifier. *2013 IEEE International Conference on Big Data*, pp. 99–104.

Liu, H. dan Setiono, R. (1995). Chi2: Feature Selection and Discretization of Numeric Attributes. *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 388–391.

Mubarok, A., Riana, D., Sanjaya, R., Prasetio, R.T., Ramdhani, Y., Rismayadi, A.A., Hidayatulloh, S., Shobary, M.N., Fitriyani, F., Arifin, T., dan Herliana, A. (2018). Sistem informasi pelayanan online di Mapolresta Bandung. *Jurnal Abdimas BSI: Jurnal Pengabdian Kepada Masyarakat*, 1(1), pp. 1–6.

Nurhayati, Putra, A. E., Wardhani, L. K., dan Busman (2019). Chi-Square feature selection effect on Naive Bayes Classifier algorithm performance for sentiment analysis document. *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–7.

Permadi, V.A. (2020). Analisis sentimen menggunakan algoritma Naïve Bayes terhadap review restoran di Singapura. *Jurnal Buana Informatika*. 11(2), pp. 141–151.

Pitria, P. (2014). *Analisis sentimen pengguna Twitter pada akun resmi Samsung Indonesia dengan menggunakan Naïve Bayes*. Tesis. Bandung: Universitas Komputer Indonesia.

Thaseen, I. S. dan Kumar, C.A. (2016). Intrusion detection model using Chi Square feature selection and modified Naïve Bayes Classifier. *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16')*, 49, pp. 81–91.

Vakili, M., Ghamsari, M., dan Rezaei, M. (2020). Performance analysis and comparison of Machine and Deep Learning algorithms for IoT data classification. [Preprint].