

ANALISIS SENTIMEN TERHADAP TEMPAT WISATA DI KABUPATEN REMBANG MENGGUNAKAN METODE NAIVE BAYES CLASSIFIER

Ali Imron

Fakultas Teknologi Industri, Teknik Informatika
Univaersitas Islam Indonesia, UII
Yogyakarta, Indonesia
14523239@students.uui.ac.id

Abstrak—Kabupaten Rembang merupakan Kabupaten yang memiliki banyak pilihan pariwisata. Wisata dan budaya merupakan salah satu sektor yang sangat berpengaruh untuk berlangsungnya pembangunan suatu daerah. Dengan adanya dorongan dari masyarakat yang sadar wisata, Pemerintah Kabupaten Rembang berupaya untuk mengembangkan dan memajukan wilayahnya dengan memanfaatkan wisata yang ada di wilayahnya. Tanggapan dan komentar dari pengunjung wisata sangat diperlukan untuk pengembangan objek wisata. Dengan adanya feedback dari pengunjung diharapkan mempermudah Pemerintah Kabupaten Rembang menentukan langkah apa yang harus dilakukan dalam pengembangan objek wisata yang ada.

Penelitian ini dilakukan untuk mengetahui performa algoritman Naive Bayes Classifier dalam melakukan proses klasifikasi berdasarkan komentar pengunjung wisata yang ada di Kabupaten Rembang. Sumber data pada penelitian ini menggunakan komentar pengunjung wisata yang terdapat pada situs Tripadvisor dan Facebook. **Kata kunci**—*analisis sentimen, performa, klasifikasi, naive bayes.*

I. PENDAHULUAN

Travelling atau berwisata merupakan kegiatan yang tidak bisa dipisahkan dari kehidupan masyarakat. Tujuan dari berwisata sendiri tak lain untuk bersenang-senang dan mencari situasi baru untuk menenangkan pikiran dari aktivitas sehari-hari. Rembang merupakan salah satu Kabupaten di Provinsi Jawa Tengah yang memiliki banyak tempat wisata dan jenis wisata yang layak dikunjungi. Dengan adanya dorongan dari masyarakat yang sadar wisata, Pemerintah Kabupaten Rembang berupaya untuk mengembangkan dan memajukan wilayahnya dengan memanfaatkan wisata yang ada di wilayahnya. Wisata dan budaya merupakan salah satu sektor yang sangat berpengaruh untuk berlangsungnya pembangunan suatu daerah.

Problematika yang dihadapi Pemerintah Kabupaten Rembang dalam bidang kepariwisataan masih banyak pekerjaan yang wajib dipecahkan dengan tepat, di antaranya yaitu implementasi kebijakan pengembangan objek wisata yang masih belum optimal. Banyaknya wisata menjadi salah satu faktor menyulitkan Dinas Pariwisata dan Kebudayaan Kabupaten Rembang untuk memantau perkembangan di setiap wisata yang ada di wilayah Kabupaten Rembang. Tanggapan

dan komentar dari pengunjung wisata sangat diperlukan untuk pengembangan objek wisata. Dengan adanya *feedback* dari pengunjung diharapkan mempermudah Pemerintah Kabupaten Rembang menentukan langkah apa yang harus dilakukan dalam pengembangan objek wisata yang ada.

Dengan majunya perkembangan teknologi informasi saat ini membawa banyak manfaat dan keuntungan bagi orang lain. Perkembangan teknologi informasi juga berbanding lurus dengan berkembangnya pariwisata. Pengunjung wisata dapat memberikan komentar terhadap wisata yang dikunjungi. Salah satu aplikasi dan media sosial yang sering digunakan untuk melihat dan memberikan komentar wisata yaitu *Tripadvisor* dan *Facebook*. Dalam aplikasi *Tripadvisor* dan *Facebook* terdapat banyak informasi wisata dan komentar pengunjung yang sangat membantu untuk mengetahui kualitas dari wisata yang akan dikunjungi. Dengan demikian pengguna lain dapat mudah menemukan informasi apa yang sedang diinginkan. Analisis Sentimen adalah jenis *natural language* yang dapat dilakukan untuk mendapatkan *Opinion Public*. Dengan adanya analisis sentimen dapat dilakukan pengolahan kata untuk melacak mood pengunjung wisata dari komentar pengunjung yang didapatkan dalam aplikasi *Tripadvisor* dan *Facebook*.

Oleh karena itu, penelitian ini akan menawarkan solusi dalam melakukan analisis sentimen terhadap komentar pengunjung wisata di Kabupaten Rembang. Analisis sentimen ini digunakan untuk mengetahui performa algoritma Naive Bayes Classifier dalam melakukan klasifikasi berdasarkan komentar pengunjung pariwisata yang ada di Kabupaten Rembang, sehingga mempermudah Dinas Pariwisata dan Kebudayaan Kabupaten Rembang dalam memantau perkembangan wisata dan membantu dalam pengambilan keputusan untuk langkah apa yang tepat dalam mengembangkan wisata yang ada. Untuk membuat sebuah analisis sentimen perlu banyak hal dilakukan, diantaranya memilih *classifier* dan sumber data yang akan digunakan untuk analisis.

II. DASAR TEORI

A. Machine Learning

Machine learning merupakan suatu ilmu yang membuat sistem dapat secara otomatis belajar sendiri tanpa

harus berulang kali diprogram oleh manusia. *Machine Learning* sendiri merupakan salah satu disiplin ilmu dalam kecerdasan buatan atau yang sering biasa dikenal dengan *Artificial Intelligent (AI)*. *Machine Learning* juga sering disebut dengan *Artificial Intelligent (AI)* konvensional karena merupakan kumpulan metode-metode yang digunakan dalam penerapannya. *Machine Learning* berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar sendiri. Sebelum *Machine Learning* bisa bekerja, maka ia membutuhkan data untuk *training* (latihan) kemudian hasil dari *training* tersebut akan diuji atau di test dengan data yang sama atau bertolak belakang.

B. Pengolahan Bahasa Alami

Pengolahan Bahasa Alami (PBA) atau *Natural Language Processing (NLP)* merupakan bagian penting dari *text mining* dan juga sub bidang dari kecerdasan buatan atau yang sering disebut dengan *Artificial Intelligence* atau *AI*. PBA mempelajari tentang bagaimana membuat komputer yang mampu mengerti dan memahami makna bahasa manusia, dengan cara mengubah bahasa manusia ke dalam dokumen atau teks dan menjadikannya lebih formal supaya lebih mudah untuk dimanipulasi oleh program komputer dan memberikan respon yang sesuai. Tujuan PBA adalah untuk melangkah melebihi manipulasi teks berbasis sintaks (yang sering kali disebut dengan 'wordcounting') ke pemahaman yang benar dan memproses bahasa alami dengan mempertimbangkan batasan semantik, gramatikal, dan konteks (Kumar, 2011).

Komputer dapat memahami bahasa alami manusia dengan cara membuat gambaran bahasa manusia dan mengubahnya menjadi angka-angka yang nantinya akan diproses dan dimasukkan ke dalam perhitungan dengan metode tertentu, yang kemudian akan menghasilkan respon kepada pengguna berupa tanggapan dari masukan yang telah diproses komputer sehingga membuat komputer terkesan dapat berinteraksi dengan pengguna menggunakan bahasa alami manusia.

C. Text Mining

Text mining adalah proses menambang data berupa informasi dan pengetahuan yang berguna dimana sumber data didapatkan dari dokumen atau teks, seperti dokumen Word, PDF, kutipan teks, atau sebagainya. *Text mining* sendiri memiliki tujuan untuk mencari kata-kata dan mendapatkan informasi yang berguna dimana informasi tersebut dapat mewakili isi dari dokumen yang berkaitan sehingga dapat dilakukan analisa yang berhubungan antar dokumen. Sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau kurang terstruktur. Pada dasarnya, *text mining* bisa dipikir sebagai suatu proses (dengan dua langkah utama) yang mulai dengan memaksakan struktur ke berbagai sumber data berbasis teks yang diikuti dengan mengekstrak informasi dan *knowledge* yang relevan dari data berbasis teks yang sudah terstruktur tersebut dengan menggunakan berbagai *tools* dan teknik data mining.

Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna mendapatkan informasi yang

bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses *text mining* memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.

D. Sentiment Analysis

Sentiment Analysis (SA) merupakan proses memahami dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat atau teks yang berupa opini. Tujuan dilakukan *sentiment analysis* untuk melihat pandangan atau pendapat teks yang berkaitan terhadap sebuah masalah atau objek, apakah cenderung berpandangan *positif* atau *negatif*. *Sentiment analysis* terdiri dari pemrosesan bahasa alami, analisis teks dan komputasi linguistik untuk mengidentifikasi sentimen dari suatu dokumen (Vinodhini dan Chandrasekaran, 2016).

E. Classification

Menurut Prasetyo (2012), klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

F. Naive Bayes Classification

Naive Bayes Classification merupakan teknik klasifikasi berdasarkan Teorema Bayes dengan asumsi independensi di antara para prediktor. Naive Bayes Classifier memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Dalam istilah sederhana, penggolongan Naive Bayes menganggap bahwa kehadiran fitur tertentu di kelas tidak terkait dengan kehadiran fitur lainnya (Hidayatullah, 2014). Keuntungan penggunaan adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*training data*) yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

G. Cross Validation

Cross Validation merupakan salah satu teknik untuk menilai atau memvalidasi keakuratan sebuah model yang dibangun berdasarkan dataset tertentu. Pembuatan model biasanya bertujuan untuk melakukan prediksi maupun klasifikasi terhadap suatu data baru yang boleh jadi belum pernah muncul di dalam dataset. Data yang digunakan dalam proses pembangunan model disebut data latih atau data *training*, sedangkan data yang akan digunakan untuk memvalidasi model disebut sebagai data *test*. Salah satu metode *Cross-Validation* yang populer adalah *K-Fold Cross Validation*. *K-fold* bekerja dengan melipat data sebanyak K dan mengulangi (men-iterasi) eksperimennya sebanyak K juga.

H. Performance Evaluation Measure

Performance Evaluation Measure (PEM) atau dalam Bahasa Indonesia bisa disebut pengukuran evaluasi performa adalah satu bundel tahapan yang digunakan untuk mengukur performa suatu sistem. PEM dalam banyak kasus digunakan dalam training data, tujuannya untuk mengevaluasi model yang sudah dibuat. Ada banyak perhitungan untuk mendapatkan nilai PEM, biasanya diterapkan sebagai kombinasi atau juga secara parsial. Beberapa perhitungan dalam PEM antara lain :

- Precision.

Precision adalah tingkat ketepatan antara request pengguna dengan jawaban sistem;

- Accuration.

Accuration adalah perbandingan antara informasi yang dijawab oleh sistem dengan benar dengan keseluruhan informasi; dan

- Recall.

Recall adalah ukuran ketepatan antara informasi yang sama dengan informasi yang sudah pernah dipanggil sebelumnya.

Rumus *precision* (pre) :

$$pre = \frac{TP}{FP + TP}$$

Rumus *accuracy* (acc) :

$$acc = \frac{TN + TP}{FN + FP + TN + TP}$$

Rumus *recall* (rec) :

$$rec = \frac{TP}{FN + TP}$$

PEM biasanya digambarkan dalam *confusion matrix*, yaitu berupa tabel yang berisi hasil pengujian model yang telah dibandingkan dengan *dataset*, terdiri dari kelas *true* dan *false*.

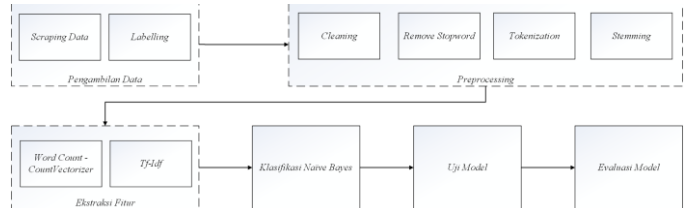
Tabel 2. 1 Confusion Matrix

	<i>Predicted Class</i>	
<i>True Class</i>	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP	FN

<i>Negative</i>	FP	TN
-----------------	----	----

III. METODE PENELITIAN

Langkah-langkah pengerjaan penelitian dari awal hingga akhir ditujukan pada Gambar 1.



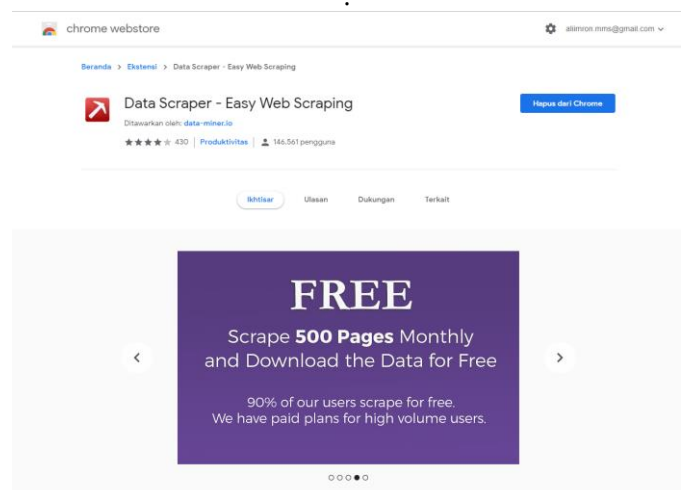
Gambar 1 Metode penelitian

IV. HASIL DAN ANALISIS

Dalam proses analisis yang dilakukan terdiri dari beberapa tahap, diantaranya pengambilan data, *preprocessing*, ekstraksi fitur, klasifikasi *naive bayes*, uji model, dan evaluasi model.

A. Pengambilan Data

Pengambilan data dilakukan dengan menggunakan bantuan ekstensi Google Chrome, yaitu *Data Scraper*. Ekstensi *Data Scraper* merupakan salah satu ekstensi pada Google Chrome yang berfungsi sebagai *web scraper* untuk mengambil data atau konten yang terdapat pada situs *online* secara gratis. *Data Scraper* mengekstraksi data dari halaman web HTML dan mengimpornya ke dalam *spreadsheet Microsoft Excel*.



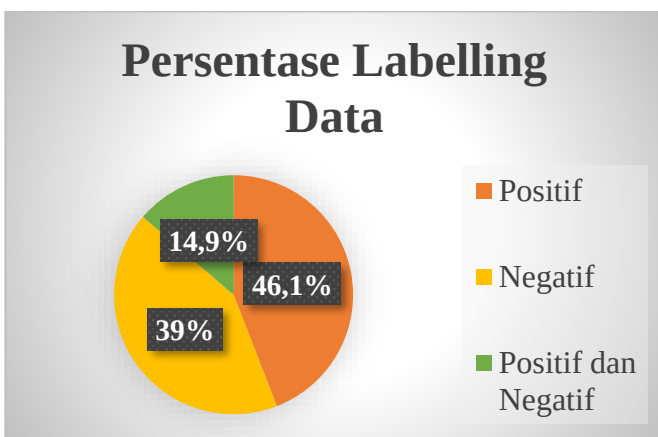
Gambar 4. 1 Ekstensi Data Scraper

Sebelum menggunakan ekstensi *Data Scraper*, langkah pertama yang harus dilakukan yaitu menginstall dan mengaktifkannya ke dalam Google Chrome. Setelah *Data Scraper* terinstall, selanjutnya membuat daftar wisata dari situs Tripadvisor yang telah ditentukan untuk diambil komentar pengunjunnya.

Setelah seluruh komentar berhasil disimpan selanjutnya dilakukan proses pelabelan. Dalam proses pelabelan yang

dilakukan seharusnya untuk menentukan komentar yang didapatkan apakah bernilai *positif*, *negatif*, ataupun keduanya harus dilakukan oleh pakar di bidangnya dan setidaknya membutuhkan tiga orang ataupun lebih dengan ketentuan jumlahnya ganjil untuk menghindari perbedaan pendapat dalam menentukan sentimen komentar. Namun pada proses pelabelan pada penelitian ini hanya dilakukan oleh dua mahasiswa yang bukan merupakan pakar bahasa di bidangnya. Konsekuensi yang didapatkan dalam proses pelabelan ini yaitu terdapat beberapa komentar yang tidak sesuai dengan sentimen komentar yang sebenarnya. Karena dilakukan secara manual, dalam proses pelabelan ini membutuhkan waktu yang tidak cepat, hal itu menjadi kelemahan apabila data yang harus dilabeli sangatlah banyak.

Dan hasil akhir dari proses pelabelan data menghasilkan data sebanyak 688 komentar bernilai *positif*, 581 komentar bernilai *negatif* dan 222 komentar bernilai *positif negatif*. *Persentase* data bernilai *positif*, *negatif* dan *positif negatif* dan dapat dilihat pada gambar 4.2.



Gambar 4. 2 Persentase Data Hasil Labelling

B. Preprocessing

Tahapan ini terdiri dari beberapa proses karena data komentar tidak sepenuhnya menggunakan kata baku. Tahap preprocessing dilakukan dengan menggunakan bantuan library pada bahasa pemrograman Python3. Penerapan tahap preprocessing data pada penelitian ini dilakukan dengan melakukan 4 proses secara urut, diantaranya:

1. Cleaning

Proses *cleaning* pada tahap ini bertujuan untuk membersihkan data komentar dari hal yang tidak diperlukan seperti tanda baca, normalisasi unicode, dan sebagainya. Dalam melakukan proses *cleaning* tersebut dilakukan 4 tahapan untuk mendapatkan hasil yang maksimal, diantaranya ialah:

- Menghapus tanda baca
- Menghapus angka
- Menyeragamkan huruf menjadi huruf kecil semua
- Menghapus kelebihan spasi.

2. Remove Stopword

Tahap *remove stopwords* merupakan tahapan yang dilakukan untuk menghapus *stopword*. Proses penghapusan *stopword* dari data dilakukan dengan mendefinisikan kata-kata

yang terdaftar sebagai *stopword* terlebih dahulu. Dalam hal ini, seluruh daftar *stopword* disimpan ke dalam sebuah file yang diberi nama *stopword_id*. File tersebut disimpan pada sebuah folder di dalam *corpora* yang terdapat pada *nltk_data*. Tahap *remove stopwords* dilakukan dengan menggunakan bantuan *library* pada bahasa pemrograman Python3 yang bernama *nltk*.

3. Tokenization

Tahap *tokenization* merupakan tahap dimana memisahkan kata, simbol, frase, dan entitas penting lainnya (yang disebut sebagai token) dari sebuah teks. Tahap *tokenization* dilakukan dengan menggunakan bantuan *library* pada bahasa pemrograman Python3 yang bernama *nltk*.

4. Stemming

Tahap *Stemming* adalah tahap mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata.

Tahap *stemming* dilakukan dengan menggunakan bantuan *library* pada bahasa pemrograman Python3 yang bernama *Sastrawi*.

C. Ekstraksi Fitur

Setelah terbentuknya *file* yang akan dijadikan *dataset*, maka selanjutnya data tersebut akan dibentuk menjadi sebuah model klasifikasi. Namun sebelum membentuk model, ada beberapa tahapan yang harus dilakukan agar terbentuknya suatu model yang baik. Yang pertama dilakukan adalah membaca *file* *xlsx* dan kemudian dilakukan tokenisasi terhadap seluruh dokumen dalam *file* tersebut. Berdasarkan hasil tokenisasi yang dilakukan, maka penulis juga ingin mengetahui frekuensi kata yang banyak diperbincangkan oleh pengunjung wisata, untuk itu penulis memvisualisasikannya dalam bentuk *wordcloud* pada gambar 4.3.



Gambar 4. 3 Visualisasi Kata Terpopuler dengan Wordcloud

Berdasarkan hasil visualisasi kata dengan *wordcloud*, terlihat bahwa kata yang paling banyak dibicarakan dicetak lebih besar daripada kata yang lain. Dengan begitu kata *pantai* merupakan kata yang paling sering dibicarakan pada komentar yang diambil dari situs *Tripadvisor* dan aplikasi *Facebook*. Berdasarkan hasil visualisasi tersebut dapat disimpulkan

Pada proses ekstraksi fitur, tahap pertama yang dilakukan setelah tokenisasi yaitu mengubah dataset ke dalam sebuah representasi *vector* dengan menggunakan *library python* yang bernama *CountVectorizer*. Hasil dari representasi *vector* dapat dilihat pada gambar 4.4.

Gambar 4. 4 Data *Word Vector*

Seperti yang sudah dijelaskan pada bagian metodologi, setiap baris pada Gambar 4.19 mewakili baris setiap data komentar yang menjadi dataset. Kemudian setiap kolomnya merupakan seluruh kata. Data yang sudah menjadi *word vector* kemudian dihitung menggunakan rumus *N-Gram* dan *TF-IDF* sehingga menghasilkan *word vector* dengan nilai yang sudah terbobot. Hasil dari pembobotan word vector dapat dilihat pada gambar 4.5.

Gambar 4. 5 Data *Word Vector* yang sudah Terbobot

D. Implementasi Klasifikasi Naive Bayes

Proses ekstraksi fitur dan klasifikasi naive bayes digabung ke dalam satu class pipeline dengan urutan *vectorizer* => *transformer* => *classifier*. Proses tersebut dilakukan dengan menggunakan bantuan *library* pada bahasa pemrograman *Python3* yang bernama *scikit-learn* untuk proses klasifikasi, *numpy* dan *pandas* untuk pembacaan data. Adapun *library* *scikit-learn* yang dipakai diantaranya adalah *Pipeline*, *CountVectorizer*, *TfidfTransformer*, *Naive Bayes*, *MultinomialNB*, *confusion matrix*, dan *f1 score*.

Langkah pertama untuk melakukan ekstraksi fitur dan klasifikasi adalah menginstall *library* yang diperlukan. Selanjutnya setelah semua *library* yang diperlukan dipastikan sudah terinstall maka dilanjutkan dengan mendeklarasikan semua *library* yang dibutuhkan. Adapun kode program deklarasi tersebut dapat dilihat pada gambar 4.6.

Gambar 4. 6 Kode Program Deklarasi *Library* untuk Klasifikasi

Setelah semua *library* selesai dideklarasikan selanjutnya mengambil *dataset* yang akan dijadikan sebagai data *training* menggunakan *library pandas*. Adapun kode program deklarasi tersebut dapat dilihat pada potongan gambar 4.7.

Gambar 4. 7 Kode Program Pemanggilan *Dataset*

Proses selanjutnya membuat class pipeline yang di dalamnya terdapat 3 urutan proses yaitu mengubah *dataset* ke dalam sebuah *representasi vector* menggunakan library *CountVectorizer* dilanjut dengan pembobotan *word vector* dengan menggunakan library *TfidfTransformer* dan yang ketiga baru dilakukan klasifikasi dengan menggunakan library *MultinomialNaiveBayes*. Adapun implementasi dari ketiga proses tersebut dapat dilihat pada potongan kode program gambar 4.8.

Gambar 4. 8 Kode Program Implementasi Klasifikasi

Pada proses klasifikasi yang dilakukan, penulis menggunakan data tes yang diambil secara acak dari 33% data *training*. Setelah proses klasifikasi dilakukan dengan menggunakan perhitungan *probabilitas* antar kalimat terhadap setiap kelas, maka barulah bisa mendapatkan hasil jelas prediksi dari data yang dimasukkan. Setelah semua proses itu dilakukan, barulah bisa menghitung performa dari *algoritma* yang digunakan.

E. Uji Model

Untuk mengetahui performa dari *Algoritma Naive Bayes*, maka dilakukan pengujian terhadap model. Hasil klasifikasi akan ditampilkan dalam bentuk *confusion matrix*. Tabel *confusion matrix* terdiri dari *predicted class* dan *actual class*. Model *confusion matrix* 3x3 ditunjukkan pada tabel 4.1.

Tabel 4. 1 Model *Confusion Matrix*

		<i>Predict Class</i>		
		<i>Class A</i>	<i>Class B</i>	<i>Class C</i>
<i>Actual Class</i>	<i>Class A</i>	AA	AB	AC
	<i>Class B</i>	BA	BB	BC
	<i>Class C</i>	CA	CB	CC

Sedangkan untuk nilai akurasi model diperoleh dari jumlah data yang tepat hasil klasifikasi dibagi dengan total data, yang dapat dilihat pada gambar 4.9.

$$Akurasi = \frac{AA+BB+CC}{AA+AB+AC+BA+BB+BC+CA+CB+CC}$$

Gambar 4. 9 Perhitungan Akurasi Model

Pada proses uji model yang dilakukan menghasilkan nilai akurasi dan *confusion matrix* 3x3 yang dapat dilihat pada gambar 4.10.

```
Accuracy: 0.8275862068965517
Confusion matrix: [[232   5   0]
 [ 12 171   0]
 [ 67   1  5]]
```

Gambar 4. 10 Hasil Uji Model

Nilai akurasi yang didapatkan sebesar **0.8275** yang dihitung berdasarkan jumlah nilai dari diagonal *confusion matrix* dibagi dengan jumlah seluruh data. Karena jumlah data tiap class pada data training tidak seimbang, maka besarnya nilai akurasi bukanlah terpenting.

F. Evaluasi Model

Evaluasi model dilakukan setelah proses uji model selesai dilakukan. Evaluasi model dilakukan untuk menghitung

performa metode yang dipilih. Pada proses uji model yang dilakukan menghasilkan *confusion matrix* 3x3 yang dapat

		<i>Predict Class</i>		
		<i>Positive</i>	<i>Negative</i>	<i>Positive & Negative</i>
<i>Actual Class</i>	<i>Positive</i>	232	5	0
	<i>Negative</i>	12	171	0
	<i>Positive & Negative</i>	67	1	5

dilihat pada tabel 4.2.

Tabel 4. 2 Hasil *Confusion Matrix*

Seperti yang terlihat pada tabel 4.2, *confused matrix* berupa matriks dengan ukuran 3x3, dimana setiap kolomnya mewakili setiap kelas yaitu kelas *positif*, kelas *negatif*, dan kelas *positif negatif*. Apabila ingin mendapatkan nilai *true positif*, *true negatif*, *false positif* dan *false negatif* dalam *confused matrix* dengan ukuran matriks 2x2 di setiap kelas, maka akan menjadi seperti berikut:

seperti berikut:

- Kelas *Positif*

Tabel 4. 3 *Confusion Matrix* Kelas *Positif*

	<i>Positif</i>	<i>Bukan</i>
<i>Positif</i>	232	5
<i>Bukan</i>	79	176

- Kelas *Negatif*

Tabel 4. 4 *Confusion Matrix* Kelas *Negatif*

	<i>Negatif</i>	<i>Bukan</i>
<i>Negatif</i>	171	12
<i>Bukan</i>	6	237

- Kelas *Positif Negatif*

Tabel 4. 5 *Confusion Matrix* Kelas *Positif Negatif*

	<i>Positif Negatif</i>	Bukan
<i>Positif Negatif</i>	5	68
Bukan	0	403

Sehingga berdasarkan rumus pada bab sebelumnya nilai presisi pada keseluruhan sistem dapat dihitung dan sebesar **0.828** dan untuk nilai *recall* keseluruhan sistem sebesar **0.828**. Sedangkan untuk nilai *f-1 score* untuk evaluasi dalam informasi temu kembali yang mengkombinasikan nilai presisi dan *recall* sebesar **0.828**. Untuk menghitung nilai presisi, *recall* dan *f-1 score* pada sistem dapat menggunakan metode pada gambar 4.11

```
print("Precision score: {}".format(precision_score(y_test, predictions, average='micro')))
print("Recall score: {}".format(recall_score(y_test, predictions, average='micro')))
print("F1 Score: {}".format(f1_score(y_test, predictions, average='micro')))
```

Gambar 4. 11 Kode Program Perhitungan Presisi dan *Recall*

Dengan mengetahui besarnya nilai presisi, *recall* dan *f-1 score* pada kinerja keseluruhan sistem dapat dikatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem dan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebesar **82,8%**.

Selanjutnya untuk performa metode klasifikasi dari setiap kelasnya dapat dilihat melalui nilai presisi, *recall*, dan *f1-score* pada setiap kelasnya. Hasil nilai presisi, *recall*, dan *f1-score* memiliki nilai sebesar 0-1. Semakin tinggi nilainya maka semakin baik. Hasil dari keseluruhan proses evaluasi model dapat dilihat pada gambar 4.12.

```
Accuracy: 0.8275862068965517
Precision score: 0.8275862068965517
Recall score: 0.8275862068965517
F1 Score: 0.8275862068965517
Confusion matrix: [[232  5  0]
 [ 12 171  0]
 [ 67  1  5]]

              precision    recall  f1-score   support

0           0.75         0.98         0.85         237
1           0.97         0.93         0.95         183
2           1.00         0.07         0.13          73

micro avg       0.83         0.83         0.83         493
macro avg       0.90         0.66         0.64         493
weighted avg    0.87         0.83         0.78         493
```

Gambar 4. 12 Hasil Pengukuran Evaluasi Performa

Hasil presisi, *recall*, dan *f-1 score* di setiap kelasnya dapat dilihat pada tabel 4.6.

Tabel 4. 6 Nilai Presisi, *Recall*, dan *f1-Score*

Jenis Klasifikasi	presisi	<i>recall</i>	<i>f-1 score</i>
Positif	0.75	0,98	0.85
Negatif	0,97	0,93	0,95
Positif & Negatif	1.00	0.07	0.08

Hasil dari evaluasi model dapat dilihat nilai presisi dan *recall* di setiap kelasnya dapat dikatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna untuk kelas *positif* sebesar **75%**, untuk kelas *negatif* sebesar **97%**, kelas *positif negatif* sebesar **100%**. Sedangkan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi untuk kelas *positif* sebesar **98%**, untuk kelas *negatif* sebesar **93%**, kelas *positif negatif* sebesar **0.07%**. Artinya kinerja sistem keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai *positif negatif* dalam dokumen sangatlah rendah.

Untuk menentukan hasil uji dan evaluasi yang maksimal dilakukan pengujian kedua dengan menggunakan *k-fold cross validation*. Dimana jumlah nilai k yang digunakan pada penelitian ini yaitu 10. Dalam 10 *fold* CV, data dibagi menjadi 10 *fold* berukuran kira-kira sama, sehingga memiliki 10 *subset* data untuk mengevaluasi kinerja model atau algoritma. Hasil dari pengujian dengan menggunakan 10 *fold cross validation* dapat dilihat pada gambar 4.13.

```
Accuracy: 0.8580121703853956
Precision score: 0.8580121703853956
Recall score: 0.8580121703853956
F1 Score: 0.8580121703853956
Confusion matrix: [[199  5  33]
 [  5 178  0]
 [ 26  1  46]]

              precision    recall  f1-score   support

0           0.8652         0.8397         0.8522         237
1           0.9674         0.9727         0.9700         183
2           0.5823         0.6301         0.6053          73

micro avg       0.8580         0.8580         0.8580         493
macro avg       0.8050         0.8142         0.8092         493
weighted avg    0.8612         0.8580         0.8594         493
```

Gambar 4. 12 Evaluasi Model dengan *Cross Validation*

Dengan menggunakan *cross validation* dapat dilihat nilai akurasi meningkat menjadi **0.858** atau sebesar **85.8%**.

Untuk hasil presisi, *recall*, dan *f-1 score* di setiap kelasnya dapat dilihat pada Tabel 4.7.

Tabel 4. 6 Nilai Presisi, *Recall*, dan *f1-Score* Evaluasi Model

Jenis Klasifikasi	presisi	<i>recall</i>	<i>f-1 score</i>
Positif	0.87	0,84	0.85
Negatif	0,97	0,97	0,97
Positif & Negatif	0.58	0.63	0.61

Hasil dari evaluasi model dapat dilihat nilai presisi dan *recall* di setiap kelasnya dapat dikatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna untuk kelas *positif* sebesar **87%**, untuk kelas *negatif* sebesar **97%**, kelas *positif negatif* sebesar **58%**. Sedangkan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi untuk kelas *positif* sebesar **84%**, untuk kelas *negatif* sebesar **97%**, kelas *positif negatif* sebesar **0.63%**. Artinya kinerja sistem keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai *positif negatif* dalam dokumen sangatlah rendah.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian algoritman *Naive Bayes Classifier* yang telah dilakukan ada beberapa hal yang dihasilkan, antara lain :

1. Pada penelitian ini, algoritma *Naive Bayes Classifier* terbukti algoritma yang akurat karena menghasilkan nilai akurasi sebesar **0.828** atau **82.8%**.
2. Untuk memastikan hasil penelitian ini, dilakukan juga pengujian dengan *K-Fold Cross Validation* dengan k sebesar 10 yang hasil nilai akurasinya sebesar **0.858** atau **85.8%**.
3. Kelebihan pada penelitian ini adalah memiliki nilai akurasi, presisi dan *recall* yang baik, sehingga sudah cukup untuk bisa digunakan dalam sebuah sistem.
4. Kekurangan pada penelitian ini terdapat pada kinerja sistem dalam mencari keberhasilan sistem untuk menemukan kembali sebuah informasi pada kelas *positif negatif* , yaitu hanya sebesar **0.07%**. Hal itu disebabkan karena jumlah data training pada kelas *positif negatif* lebih sedikit dibandingkan dengan kelas *positif* dan kelas *negatif* atau bisa dikatakan data training yang digunakan pada penelitian ini tidak *balance*.

B. Saran

Dari hasil penelitian yang dilakukan masih banyak kekurangan, dengan demikian peneliti berharap penelitian ini untuk dikembangkan, beberapa saran dari penulis, diantaranya :

1. Menggunakan algoritma klasifikasi yang lain sehingga dapat membandingkan hasil uji model yang dilakukan untuk mencari algoritma klasifikasi terbaik.
2. Menambah data *training* dan mengupayakan untuk menyeimbangkan jumlah data di setiap kelasnya.
3. Membuat *interface* dari proses pengujian model dan *visualisasi* dari performa metode yang digunakan.

DAFTAR PUSTAKA

- Clayton, F., 2011. *Coarse-and Fine-Grained Sentiment Analysis of Social Media Text*. s.l.:s.n.
- Fannisa, S., Fauzi, M. A. & Adinugroho, S., 2018. Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Firmansyah, R. F. N., Fauzi, M. & Afirianto, T., 2016. Sentiment Analysis pada Review Aplikasi Mobile Menggunakan Metode Naive Bayes dan Query Expansion.
- G, V. & Chandrasekaran, D. R., 2015. *A comparative performance evaluation of neural network based approach for sentiment classification of online reviews*. s.l.:Journal of King Saud University.
- Haryanto, D. J., Muflikhah, L. & Fauzi, M. A., 2018. Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Hidayatullah, A. F. A. S. N., 2014. *ANALISIS SENTIMEN DAN KLASIFIKASI KATEGORI TERHADAP TOKOH PUBLIK PADA TWITTER*. Yogyakarta, s.n.
- Kumar, E., 2011. *NATURAL LANGUAGE PROCESSING*. New Delhi: LK. Intrnational Publishing House Pvt, Ltd.
- Muslehatin, W., Ibnu, M. & Mustakim, 2017. Penerapan Naïve Bayes Classification untuk Klasifikasi Tingkat Kemungkinan Obesitas Mahasiswa Sistem Informasi UIN Suska Riau. *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI)* 9, pp. 250-256.
- Prasetyo, E., 2012. *Data Mining Konsep dan Aplikasi menggunakan MATLAB*. Yogyakarta: Penerbit Andi.
- Susanti, N. D., 2016. Uji Perbandingan Akurasi Analisis Sentimen Pariwisata menggunakan Algoritma Support Vektor Machine dan Naive Bayes.
- Wilianto, L., Rakhmat Umbara, F. & Hendro Pudjiantoro, T., 2017. *ANALISIS SENTIMEN TERHADAP TEMPAT WISATA DARI KOMENTAR PENGUNJUNG DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER STUDI KASUS JAWA BARAT*. *Prosiding SNATIF Ke -4*.