

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



Disusun Oleh:

N a m a : Ali Imron

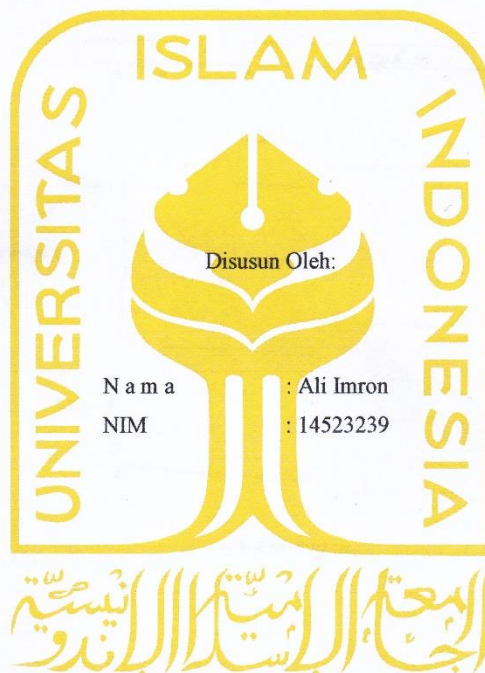
NIM : 14523239

**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2019

HALAMAN PENGESAHAN DOSEN PEMBIMBING

ii

HALAMAN PENGESAHAN DOSEN PEMBIMBING**ANALISIS SENTIMEN TERHADAP TEMPAT WISATA DI
KABUPATEN REMBANG MENGGUNAKAN METODE
*NAIVE BAYES CLASSIFIER*****TUGAS AKHIR**

Yogyakarta, 27 Februari 2019

Pembimbing,

(Ahmad Fathan Hidayatullah, S.T, M.Cs)

HALAMAN PENGESAHAN DOSEN PENGUJI

iii

HALAMAN PENGESAHAN DOSEN PENGUJI

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

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 27 Februari 2019

Tim Penguji

Ahmad Fathan Hidayatullah, S.T., M.Cs.

Anggota 1

Andhik Budi Cahyono, S.T., M.T.

Anggota 2

Septia Rani, S.T., M.Cs.

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

iv

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ali Imron

NIM : 14523239

Tugas akhir dengan judul:

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

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 27 Februari 2019


(Ali Imron)

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin atas segala nikmat yang telah diberikan kepada kita. Shalawat serta salam kita haturkan kepada junjungan kita Nabi Muhammad SAW yang kita nantikan safa'atnya di yaumul akhir nanti.

Terima kasih yang amat besar saya haturkan kepada kedua orang tua saya yang telah mengasuh dan mendidik saya sejak di dalam kandungan sampai pada saat ini juga. Semoga beliau senantiasa diberikan kesehatan, kebahagiaan dan panjang umur.

Terima kasih kepada Dosen Pembimbing saya, Pak Fathan yang selalu melatih, membimbing saya dengan sabar dan selalu memperhatikan saya di setiap pertemuan bimbingan maupun pada saat tidak dapat bertemu dengan langsung.

Terima kasih kepada semua pihak yang tidak bisa disebutkan satu persatu atas dukungan dan bantuannya baik secara langsung maupun tidak langsung.

HALAMAN MOTO

“Hidup ini adalah perlombaan. Kejarlah kesempurnaan maka sukses akan
mendatangimu.”

“Selalu berpikir positif terhadap segala hal akan membantumu dalam menyelesaikan
setiap permasalahan yang kamu alami.”

Sikap anda di masa lalu menjadikan anda hari ini, sikap anda hari ini akan menjadikan
anda di masa depan.”

“Rejeki tertarik kepada orang yang sikapnya baik dan hidup yang baik adalah orang yang
sikapnya baik.”

“Jangan pernah mengatakan bahwa kita mempunyai masalah besar, karena kita
mempunyai Allah Yang Maha Besar.”

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillah Robbil Alamin, adalah kalimat yang bisa terucap kepada Allah SWT, karena dengan segala rahmat dan karunia-Nya lah penulis bisa bertahan menahan segala gangguan dan godaan dalam menyelesaikan laporan penelitian dengan judul “Analisis Sentimen Terhadap Tempat Wisata di Kabupaten Rembang Menggunakan Metode Naive Bayes Classifier”. Shalawat serta salam juga tidak lupa selalu tucurahkan kepada Nabi Muhammad SAW, sebagai panutan seluruh umat di segala penjuru dunia.

Adapun dalam penyelesaian tugas akhir ini, banyak pihak yang terlibat dalam penyelesaiannya. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Dr. Raden Teduh Dirgahayu, S. T., M. Sc. selaku Kaprodi Teknik Informatika
2. Ahmad Fathan Hidayatullah, S. T., M. Sc. Selaku Dosen Pembimbing .
3. Seluruh jajaran staff dan dosen Teknik Informatika Universitas Islam Indonesia.
4. Kedua orang tua yang selalu sabar dan mendukung anaknya dalam menjalankan pendidikannya.
5. Kepada teman-teman Damen Group, yang selalu memberikan dukungan dan semangat positif.
6. Kepada semua teman-teman Jurusan Teknik Informatika angkatan 2014.
7. Serta semua orang yang selalu mendukung dan mendoakan yang tidak bisa disebutkan satu per satu.

Penulis sadar banyak terdapat kekurangan dalam pembuatan tugas akhir ini. Namun penulis selalu berharap tugas akhir ini dapat bermanfaat atau mungkin bisa dikembangkan menjadi hal yang lebih besar lagi, sehingga dapat memberikan dampak yang baik untuk dunia, terlebih khusus untuk negara tercinta Indonesia.

Yogyakarta, Desember 2018

(Ali Imron)

SARI

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 algoritma 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.

Uji model dilakukan dengan menggunakan library python yaitu MultinomialNaiveBayes. Dalam proses uji model, besarnya data tes diambil 33% dari data training yang dilakukan secara acak. Evaluasi model yang dilakukan pada penelitian ini menggunakan 10 fold cross validation dengan hasil akurasi 85,8%.

Kata kunci: Wisata, Feedback, Sentimen, Library, Naive Bayes Classifier, Cross Validation

GLOSARIUM

<i>Pre-processing</i>	Perlakuan awal terhadap data untuk dijadikan bahan <i>training</i> .
<i>Training</i>	Mengolah data untuk dijadikan model.
Model	Hasil dari <i>training</i> yang digunakan untuk mengklasifikasikan bahasa.
<i>Dataset</i>	Data yang digunakan dalam pembentukan model. <i>Word vector</i> Matriks kata yang menandakan kata tertentu terdapat dalam dokumen dengan membandingkan dokumen dengan seluruh kata dari seluruh dokumen.
Klasifikasi	Penentuan kelas secara otomatis menggunakan model.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO.....	vi
KATA PENGANTAR.....	vii
SARI	viii
GLOSARIUM	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Machine Learning	5
2.2 Pengolahan Bahasa Alami	5
2.3 <i>Text Mining</i>	6
2.4 <i>Sentiment Analysis</i>	7
2.5 <i>Classification</i>	8
2.6 <i>Naïve Bayes Classiffication</i>	9
2.7 <i>Cross Validation</i>	10
2.8 <i>Performance Evaluation Measure</i>	10
2.9 Penelitian Serupa.....	14
BAB III METODE PENELITIAN	16
3.1 Alur Pengerjaan Tugas Akhir	16
3.2 Uraian Metodologi	17
3.2.1 Pengambilan Data.....	17
3.2.2 Langkah-Langkah Preprocessing	19
3.2.3 Ekstraksi Fitur	23
3.2.4 Klasifikasi Naive Bayes	24
3.2.5 Uji Model	24
3.2.6 Evaluasi Model.....	24
BAB IV HASIL DAN PEMBAHASAN.....	25
4.1 Pengambilan Data	25
4.2 <i>Preprocessing</i>	30
4.3 Ekstraksi Fitur	33
4.4 Implementasi Klasifikasi Naive Bayes	38
4.5 Uji Model	39
4.6 Evaluasi Model	40
BAB V KESIMPULAN	45

5.1	Kesimpulan	45
5.2	Saran.....	45
DAFTAR PUSTAKA.....		xv
LAMPIRAN		xvi

DAFTAR TABEL

Tabel 2. 1 <i>Confusion Matrix</i>	11
Tabel 2. 2 Contoh Hasil <i>Confusion Matrix</i>	12
Tabel 3. 1 Contoh Data Hasil <i>Labelling</i>	18
Tabel 3. 2 Contoh Proses Penerapan <i>Cleaning</i>	19
Tabel 3. 3 Contoh Proses Penerapan <i>Remove Stopword</i>	20
Tabel 3. 4 Contoh Proses Penerapan <i>Tokenization</i>	21
Tabel 3. 5 Contoh Proses Penerapan <i>Stemming</i>	22
Tabel 4. 1 Pembuatan Word Vector	36
Tabel 4. 2 Tabel <i>TF (Term Frequency)</i>	36
Tabel 4. 3 Tabel <i>DF (Document Frequency)</i>	36
Tabel 4. 4 <i>IDF (Inverse Document Frequency)</i>	37
Tabel 4. 5 Contoh Hasil Perhitungan <i>TF-IDF</i>	37
Tabel 4. 6 Contoh dari <i>Word Vector</i> yang sudah terbobot	37
Tabel 4. 7 Model <i>Confusion Matrix</i>	40
Tabel 4. 8 Hasil <i>Confusion Matrix</i>	41
Tabel 4. 9 <i>Confusion Matrix</i> Kelas <i>Positif</i>	41
Tabel 4. 10 <i>Confusion Matrix</i> Kelas <i>Negatif</i>	41
Tabel 4. 11 <i>Confusion Matrix</i> Kelas <i>Positif Negatif</i>	42
Tabel 4. 12 Nilai Presisi, <i>Recall</i> , dan <i>f1-Score</i>	43
Tabel 4. 13 Nilai Presisi, <i>Recall</i> , dan <i>f1-Score</i> Evaluasi Model	44

DAFTAR GAMBAR

Gambar 2. 1 Proses Klasifikasi	9
Gambar 2. 2 Perbandingan Tingkat Akurasi dan Presisi	12
Gambar 2. 3 Perbandingan Hasil Nilai Presisi, <i>Recall</i> dan Akurasi	13
Gambar 3. 1 Alur Pengerjaan Tugas Akhir	16
Gambar 3. 2 Halaman Awal Situs <i>Tripadvisor</i>	17
Gambar 3. 3 Logo Aplikasi <i>Facebook</i>	18
Gambar 3. 4 Hasil <i>Crawling</i> Data.....	18
Gambar 4. 1 Ekstensi <i>Data Scraper</i>	25
Gambar 4. 2 Komentar Pengunjung Alun-Alun Kota Rembang	26
Gambar 4. 3 Pengambilan Data Menggunakan <i>Data Scraper</i>	27
Gambar 4. 4 Langkah Awal <i>Scraping Data</i>	27
Gambar 4. 5 Pengambilan Data Pada Halaman Selanjutnya	28
Gambar 4. 6 Menyimpan Hasil <i>Scraping Data</i>	28
Gambar 4. 7 Hasil <i>Scraping Data</i> dan <i>Labelling Data</i>	29
Gambar 4. 8 Persentase Data Hasil <i>Labelling</i>	30
Gambar 4. 9 Kode Program Tahap <i>Cleaning</i>	31
Gambar 4. 10 Perintah Instalasi <i>library nltk</i>	31
Gambar 4. 11 Deklarasi <i>library nltk</i>	31
Gambar 4. 12 Kode Program Tahap <i>Remove Stopword</i>	32
Gambar 4. 13 Kode Program Tahap <i>Tokenization</i>	32
Gambar 4. 14 Perintah <i>Instalasi Library Sastrawi</i>	32
Gambar 4. 15 Deklarasi <i>Library Sastrawi</i>	33
Gambar 4. 16 Kode Program Tahap <i>Stemming</i>	33
Gambar 4. 17 Hasil <i>Preprocessing</i>	33
Gambar 4. 18 Visualisasi Kata Terpopuler dengan <i>Wordcloud</i>	34
Gambar 4. 19 Data <i>Word Vector</i>	35
Gambar 4. 20 Data <i>Word Vector</i> yang sudah Terbobot.....	38
Gambar 4. 21 Kode Program Deklarasi <i>Library</i> untuk Klasifikasi	38
Gambar 4. 22 Kode Program Pemanggilan <i>Dataset</i>	39
Gambar 4. 23 Kode Program Implementasi Klasifikasi	39

Gambar 4. 24 Perhitungan Akurasi Model	40
Gambar 4. 25 Hasil Uji Model.....	40
Gambar 4. 26 Kode Program Perhitungan Presisi dan <i>Recall</i>	42
Gambar 4. 27 Hasil Pengukuran Evaluasi Performa	43
Gambar 4. 28 Evaluasi Model dengan <i>Cross Validation</i>	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

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.

Problematisa 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, di antaranya memilih *classifier* dan sumber data yang akan digunakan untuk analisis.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang, maka rumusan masalah yang menjadi fokus dari penelitian ini yaitu bagaimana melakukan analisis sentimen berdasarkan komentar pengunjung wisata untuk mengetahui performa algoritma *Naive Bayes Classifier*?

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah melakukan analisis sentimen terhadap komentar pengunjung wisata untuk mengetahui performa algoritma *Naive Bayes Classifier* dalam melakukan klasifikasi berdasarkan komentar pengunjung wisata di Kabupaten Rembang.

1.4 Batasan Masalah

Adapun batasan masalah dalam melakukan penelitian ini yaitu :

- a. Data yang digunakan adalah data komentar pengunjung wisata pada aplikasi *Tripadvisor* dan *Facebook*.
- b. Hasil akhir dalam penelitian ini adalah analisis sentimen dari metode *Naive Bayes Classifier*.

1.5 Manfaat

Manfaat dari penelitian ini yaitu mengetahui performa algoritma Naive Bayes Classifier dalam melakukan klasifikasi berdasarkan komentar pariwisata yang ada di Kabupaten Rembang.

1.6 Metodologi

Metodologi penelitian merupakan tahapan yang dilakukan pada penelitian agar dalam pelaksanaannya dapat terarah sesuai rencana dan mendapatkan hasil yang diharapkan. Metodologi yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Analisis Kebutuhan

Dalam melakukan analisis sentimen terhadap komentar pengunjung wisata dibutuhkan data-data komentar atau ulasan dari pengunjung wisata sebagai sumber data. Setelah data terkumpul kemudian mempersiapkan *tools* pendukung untuk melakukan analisis sentimen, yaitu seperti *PyCharm* sebagai IDE, *Python package library* sebagai bahasa pemrograman, dan *library* yang mendukung melakukan metode *Naive Bayes Classifier*.

2. Analisis Perancangan

Analisis perancangan ini dibutuhkan sebagai sebuah gambaran untuk memudahkan orang lain dalam memahami analisis sentimen ini. Cara yang dilakukan yaitu dengan membuat diagram alur analisis sentimen ini dilakukan.

3. Implementasi

Setelah melakukan analisis kebutuhan dan analisis perancangan maka selanjutnya yaitu mengimplementasikan kebutuhan analisis sentimen ini sesuai dengan alur yang dibuat dalam melakukan analisis perancangan.

4. Pengujian dan evaluasi sistem

Pengujian ini dilakukan apakah dalam melakukan analisis sentimen ini berjalan dengan baik atau tidak. Kemudian dilakukan pengujian untuk mengetahui apakah hasil dari analisis sentimen ini sesuai dengan tujuan atau tidak. Pengujian ini juga dilakukan untuk mengetahui kekurangan dan kelemahan analisis sentimen yang sudah berjalan. Jika dalam pengujian ini tidak berhasil dan tidak sesuai maka akan masuk lagi ke dalam proses implementasi dan dilakukan pengujian kembali.

1.7 Sistematika Penulisan

Sistematika penulisan ditujukan kepada pembaca agar lebih mudah dalam memahami isi laporan penelitian. Secara garis besar sistematika penulisan laporan ini terdiri dari :

BAB I PENDAHULUAN

Dalam bagian pendahuluan membahas latar belakang penelitian ini dimana untuk mengetahui sebab penelitian ini dilakukan dan selanjutnya membahas rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan langkah penyelesaian.

BAB II LANDASAN TEORI

Pada landasan teori membahas hal-hal dasar yang berisi teori-teori yang berkaitan dengan analisis sentimen dengan menggunakan metode Naive Bayes Classifier dan hal-hal yang mendukung dalam hal melakukan analisis sentimen ini.

BAB III METODOLOGI PENELITIAN

Bagian ini membahas tentang sumber data dan tahapan yang dilakukan dalam pembuatan penelitian serta gambaran umum sistem yang akan dikerjakan.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menjelaskan tentang hasil pengolahan data, pengujian, serta menjelaskan kelebihan dan kekurangan dari hasil olahan data.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan yang merupakan rangkuman dari hasil penelitian ini dan berisi saran-saran yang membangun untuk pengembangan yang lebih baik di masa depan.

BAB II

LANDASAN TEORI

2.1 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.

2.2 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.

PBA terdiri dari bagian utama yaitu :

a. *Parser*

Parser merupakan suatu sistem dimana dilakukan proses pengambilan kalimat input bahasa alami dan kemudian menguraikannya ke dalam beberapa bagian gramatikal (kata benda, kata kerja, kata sifat, dan lain-lain)

b. Sistem Representasi Pengetahuan

Sistem Representasi Pengetahuan merupakan sistem yang menganalisis *output parser* untuk menentukan maknanya yang nantinya akan menghasilkan *output* sesuai dengan *input* yang diberikan *parser*.

c. *Output Translator*

Output Translator merupakan terjemahan yang mempresentasikan sistem pengetahuan dan melakukan langkah-langkah yang bisa berupa jawaban atas bahasa alami atau output khusus yang sesuai dengan program komputer.

Pada zaman modern seperti saat ini, PBA sudah banyak diimplementasikan ke dalam berbagai bidang. Beberapa diantara penerapannya seperti penjawab pesan otomatis atau sering disebut dengan *chatbot*, *machine translation* (aplikasi translator), *spam filtering* (pemfilteran pesan sampah), dan *language identification* (identifikasi bahasa), dan lain-lain. Ada beberapa tingkat pengolahan kata dalam PBA antara lain : *fonetik*, *morfologi*, *sintaksis*, *semantik*, *discourse knowledge*, dan *pragmatik*.

2.3 *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.

Pada saat ini, text mining sudah diterapkan di berbagai bidang, di antaranya :

a. *Information Extraction* (Ekstraksi Informasi)

Identifikasi terhadap hubungan dan frase-frase kunci dalam teks dengan mencari urutan yang sudah ditetapkan dalam text menggunakan pencocokan pola.

b. *Topic Tracking* (Pelacakan Topik)

Berdasarkan pada profil user dan berbagai dokumen yang dilihat user, text mining bisa memprediksi dokumen-dokumen lain yang menjadi perhatian/minat user tersebut.

c. *Summarization* (Peringkasan)

Meringkas suatu dokumen untuk menghemat waktu dari sisi pembaca

d. *Clustering* (Penggugusan)

Mengelompokkan dokumen-dokumen yang mirip tanpa memiliki kategori yang sudah ditetapkan sebelumnya.

e. *Concept Linking* (Penautan Konsep)

Menghubungkan berbagai dokumen terkait dengan mengidentifikasi konsep yang digunakan berbsama dan dengan demikian membantu para user untuk menemukan informasi yang barangkali mereka tidak akan temukan dengan menggunakan metode-metode pencarian tradisional.

f. *Question Answering* (Penjawaban Otomatis)

Menemukan jawaban terbaik pada pertanyaan yang diberikan melalui pencocokan pola berbasis *knowledge*.

2.4 *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, 2015).

Sentiment Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian *Sentiment Analysis* adalah *Sentiment Analysis* pada level dokumen dan *Sentiment Analysis* pada level kalimat (Clayton, 2011). Berdasarkan level sumber datanya *Sentiment Analysis* terbagi menjadi 2 kelompok besar yaitu :

1. *Coarse-grained Sentiment Analysis*

Pada *Sentiment Analysis* jenis ini, *Sentiment Analysis* yang dilakukan adalah pada level dokumen. Secara garis besar fokus utama dari *Sentiment Analysis* jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah *sentiment positif* atau *sentiment negatif* (Clayton, 2011).

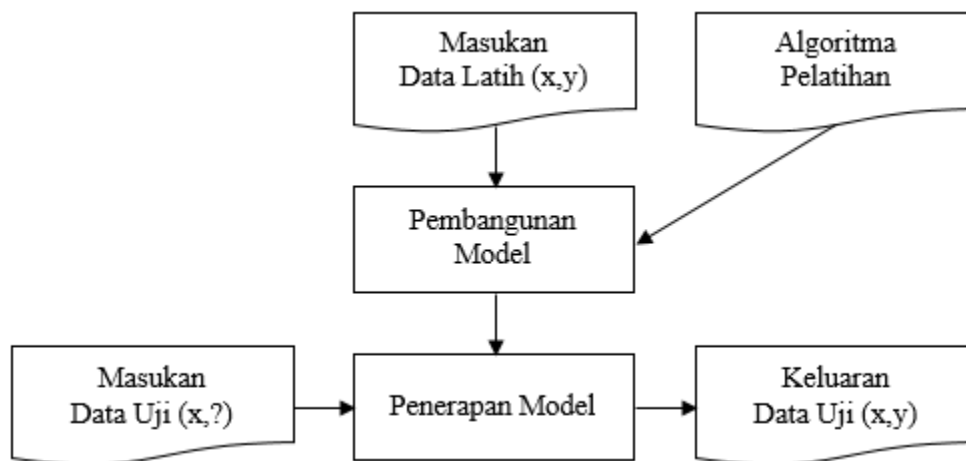
2. *Fined-grained Sentiment Analysis*

Fined-grained Sentiment Analysis adalah *Sentiment Analysis* pada level kalimat. Fokus utama *fined-greined Sentiment Analysis* adalah menentukan sentiment pada setiap kalimat pada suatu dokumen, dimana kemungkinan yang terjadi adalah terdapat sentiment pada level kalimat yang berbeda pada suatu dokumen (Clayton, 2011).

2.5 *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.

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, di mana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja (*framework*) klasifikasi ditunjukkan pada Gambar 2.2. Pada gambar tersebut disediakan sejumlah data latih (x,y) untuk digunakan sebagai data pembangun model. Model tersebut kemudian dipakai untuk memprediksi kelas dari data uji (x,?) sehingga diketahui kelas y yang sesungguhnya.



Gambar 2. 1 Proses Klasifikasi

Kerangka kerja seperti yang ditunjukkan pada gambar 2.1 meliputi dua langkah proses, yaitu induksi dan deduksi. Induksi merupakan langkah untuk membangun model klasifikasi dari data latih yang diberikan, disebut juga proses pelatihan, sedangkan deduksi merupakan langkah untuk menerapkan model tersebut pada data uji sehingga kelas yang sesungguhnya dari data uji dapat diketahui, disebut juga proses prediksi.

2.6 Naïve 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.

Rumus Bayes dalam (Muslehatin dkk, 2017) secara umum dapat diberikan sebagai berikut:

$$P(H | X) = \frac{P(H | X) P(H)}{P(X)} \quad (2.1)$$

Keterangan :

- X = Data dengan class yang belum diketahui
- H = Hipotesis data X merupakan suatu class spesifik
- $P(H|X)$ = Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)
- $P(H)$ = Probabilitas hipotesis H (prior prob.)
- $P(X|H)$ = Probabilitas X berdasarkan kondisi tersebut
- $P(X)$ = Probabilitas dari X

Aturan Bayes adalah sebagai berikut :

Jika $P(h1|x) < P(h2|x)$, maka x diklasifikasikan sebagai h2. Pernyataan $P(h1|x)$ mengindikasikan probabilitas hipotesis h1 berdasarkan kondisi x terjadi, begitu pula dengan h2. Sehingga dapat klasifikasi dari x sesuai dengan probabilitas terbesar diantara probabilitas x terhadap semua kelas.

2.7 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.

2.8 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) :

$$pre = \frac{TP}{FP + TP} \quad (2.2)$$

Rumus *accuration* (acc) :

$$acc = \frac{TN + TP}{FN + FP + TN + TP} \quad (2.3)$$

Rumus *recall* (rec) :

$$rec = \frac{TP}{FN + TP} \quad (2.4)$$

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

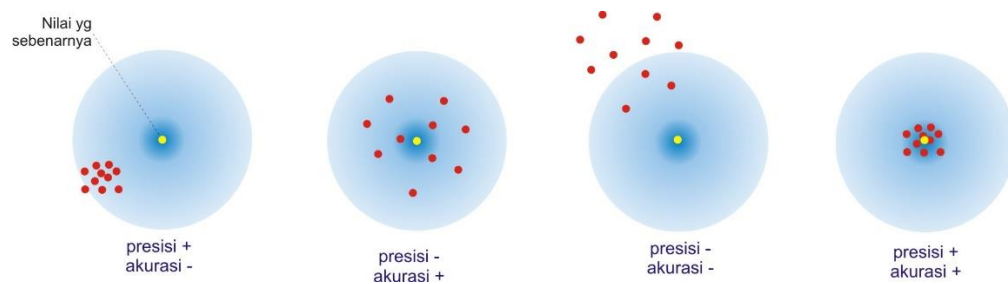
Keterangan :

TP (*true positive*) : contoh data bernilai positif yang diprediksi benar sebagai positif

TN (*true negative*) : contoh data bernilai negatif yang diprediksi benar sebagai negatif

FP (*false positive*) : contoh data bernilai negatif yang diprediksi salah sebagai positif

FN (*false negative*) : contoh data bernilai positif yang diprediksi salah sebagai negatif



Gambar 2. 2 Perbandingan Tingkat Akurasi dan Presisi

(Sumber: <https://dataq.wordpress.com/perbedaan-precision-recall-accuracy/>)

Dari gambar di atas bisa dilihat gambaran persebaran data dengan *accuracy* dan *precision*. Atau dapat juga dijelaskan dengan permisalan sebagai berikut:

Misalkan kita ingin mengukur kinerja dari sebuah mesin pemisah buah yang bertugas memisahkan buah-buah apel dari semua buah yang telah dikumpulkan. Untuk mengujinya akan dimasukkan 100 buah apel dan 900 buah lainnya (bukan buah apel). Hasilnya mesin tersebut memisahkan 110 yang dideteksi sebagai buah apel. Ke 110 buah tersebut kemudian dicek kembali oleh manusia, ternyata dari 110 buah tersebut hanya 90 buah yang merupakan buah apel, sedangkan 20 lainnya merupakan buah lainnya.

Dari kasus tersebut maka dapat disimpulkan bahwa mesin tersebut memiliki *precision* sebesar 82%, *recall* sebesar 90% dan *accuracy* sebesar 97% yang didapatkan dari perhitungan berikut:

Tabel 2. 2 Contoh Hasil *Confusion Matrix*

		Nilai Sebenarnya	
		True	False
Nilai Prediksi	True	90	20
	False	10	880

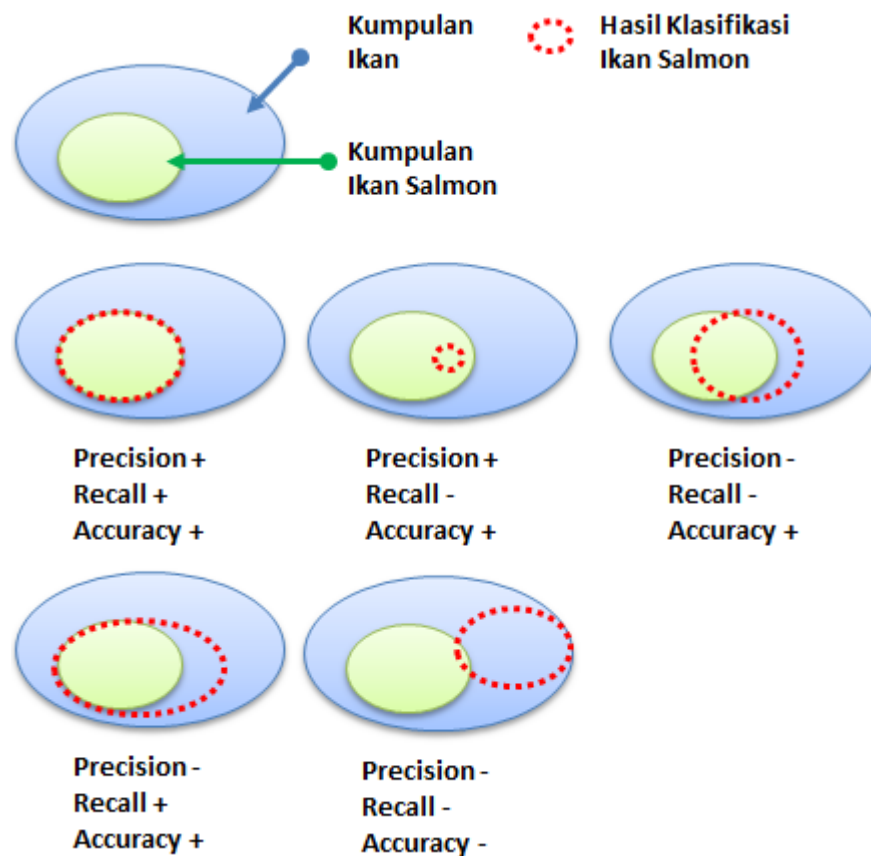
Dari tabel di atas bisa dihitung :

$$precision = \frac{90}{90 + 20} = 0.82 = 82\%$$

$$recall = \frac{90}{90 + 10} = 0.9 = 90\%$$

$$accuracy = \frac{90 + 880}{90 + 880 + 20 + 10} = 0.97 = 97\%$$

Sehingga bisa mendapat kesimpulan gambaran seperti pada gambar 2.3 apabila membandingkan nilai dari ketiga hal tadi:



Gambar 2. 3 Perbandingan Hasil Nilai Presisi, *Recall* dan Akurasi

(Sumber: <https://dataq.wordpress.com/perbedaan-precision-recall-accuracy/>)

2.9 Penelitian Serupa

Dalam pembuatan penelitian ini ada penelitian-penelitian sebelumnya yang sudah pernah dilakukan oleh orang lain yang mirip dan bahkan menjadi acuan daripada penelitian ini. Beberapa penelitian yang serupa bisa dilihat sebagai berikut:

1. Wilianto, dkk (2017) dalam penelitiannya menggunakan metode *Naive Bayes Classifier* untuk mendapatkan informasi tentang topik-topik yang sering dibahas mengenai wisata di Jawa Barat dan memantau perkembangan di tiap tempat wisata yang ada, dengan adanya sistem ini diharapkan mempermudah Dinas Pariwisata dan Kebudayaan dalam mendapatkan informasi. Dan dapat cepat menentukan langkah apa yang harus diambil kedepannya.
2. Fanissa, dkk (2018) melakukan suatu penelitian menggunakan metode *Naive Bayes Classifier* yang dapat diterapkan pada proses analisis sentimen pariwisata Malang menggunakan data latih dari ulasan-ulasan pada website *TripAdvisor*. Data latih juga harus dilabeli kelas sentimennya oleh pakar untuk proses klasifikasi. Dalam penyempurnaan penelitian ini maka penelitian selanjutnya disarankan untuk memperhatikan singkatan, gabungan dua kata atau lebih (*bigram*, *trigram*, *n-gram*), kata-kata ambigu, dan kalimat sarkastik supaya hasil klasifikasinya lebih sempurna.
3. Haryanto, dkk (2018) melakukan penelitian menggunakan metode *Support Vector Machine* yang diterapkan yaitu *Sequential Training SVM* dengan tambahan metode *Query Expansion* dapat diterapkan pada analisis sentimen review barang berbahasa Indonesia. *Query Expansion* digunakan untuk mendapatkan hasil yang lebih bagus dalam klasifikasi review barang berbahasa Indonesia dengan menggunakan daftar sinonim kata. Akurasi yang diperoleh dengan menggunakan metode *Support Vector Machine* dan *Query Expansion* sebesar 96.25%, sedangkan akurasi yang diperoleh dengan menggunakan metode *Support Vector Machine* tanpa *Query Expansion* sebesar 94.75%.
4. Susanti (2016) melakukan analisis sentimen review wisata dengan membandingkan metode *Algoritma Support Vektor Machine* dan *Naive Bayes*. Hasil dari paper ini menunjukkan bahwa nilai akurasi dengan menggunakan metode SVM lebih baik dari pada menggunakan NB dengan selisih 10,96%, ini dikarenakan pada NB, TP berbanding terbalik dengan TN yaitu nilai TP besar dan TN kecil. Sedangkan nilai TP dan TN Svm tidak terlalu besar selisihnya. Nilai Precision pada SVM baik yang *positif* ataupun *negatif*, bernilai lebih baik dari pada

NB. Dikarenakan nilai *Precision* atau tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem tidak banyak perubahan seperti yang ada pada NB. Yaitu *true positif* bernilai tinggi dan *true negatif* pada NB bernilai kecil. Nilai *Recall* pada NB *positif* lebih tinggi daripada SVM dikarenakan nilai *True positif* nilainya juga sangat tinggi. Karena nilai TP sangat tinggi maka mengakibatkan *recall*nya juga semakin tinggi dan nilai FN yang dihasilkan kecil.

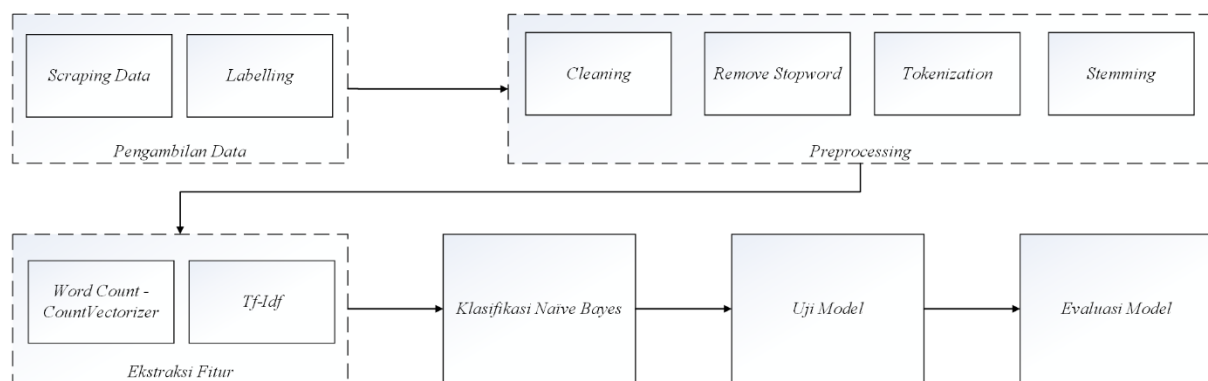
5. Firmansyah (2016) melakukan penelitian tentang analisis sentimen *review* aplikasi *mobile* dari pengguna aplikasi dengan membandingkan metode *naive bayes* dan *query expansion*. Analisis sentimen digunakan untuk menganalisis emosi dari semua komentar dan mengklasifikasikan ke dalam semua sentimen. Metode *naive bayes* digunakan untuk membagi atau memisahkan teks yang mengandung emosi *positif* atau *negatif*. Dalam metode *naive bayes* hasil klasifikasi akan tergantung pada probabilitas nilai *frekuensi* kemunculan semua dokumen karena komentar yang diberikan tergolong singkat maka perlu dilakukan optimasi dengan menggunakan *query expansion recognition of synonym*. Dengan model tersebut *short text* yang mengandung kata emosi akan dicari sinonim pada dokumen yang diujikan. Pengujian akurasi yang dihasilkan oleh sistem mendapatkan nilai terbaik sebesar 95,49% untuk pengujian akurasi dengan metode *naive bayes*. Sementara untuk *query expansion* memiliki tingkat akurasi 75,88%.

BAB III

METODE PENELITIAN

3.1 Alur Pengerjaan Tugas Akhir

Perancangan alur pengerjaan tugas akhir merupakan gambaran umum terkait alur penelitian yang akan dilakukan dalam pengerjaan tugas akhir ini dari awal hingga akhir. Alur kerja dari pengerjaan tugas akhir dapat dilihat pada gambar 3.1 berikut :



Gambar 3. 1 Alur Pengerjaan Tugas Akhir

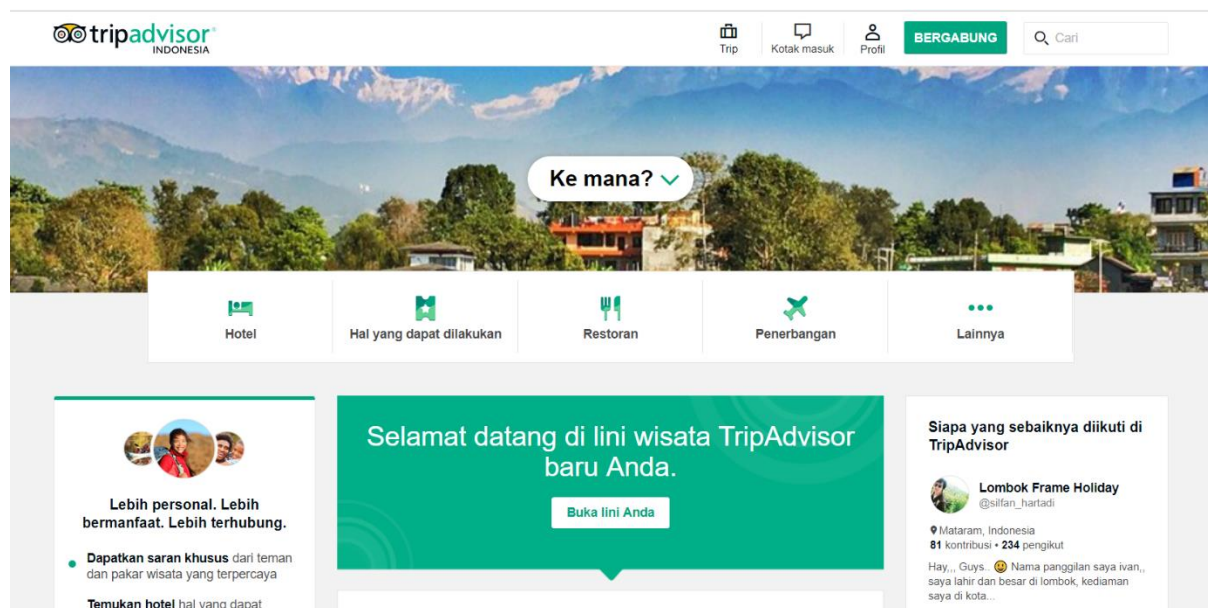
Alur atau langkah pertama dari pengerjaan tugas akhir ialah mendapatkan data komentar dari situs *Tripadvisor* dan aplikasi *Facebook* dengan menggunakan teknik *crawling* data kemudian setelah semua data terkumpul dilakukan *labelling* untuk menentukan sentimen komentar yang didapatkan. Langkah kedua, dilakukan *preprocessing* yang berguna untuk menyeleksi data dan mengubahnya menjadi data yang terstruktur. Pada tahap *preprocessing* terdiri dari 4 tahapan, yaitu *Cleaning*, *Remove Stopword*, *Tokenization* dan *Stemming*. Proses *Cleaning* digunakan untuk membersihkan kata-kata yang tidak diperlukan guna mengurangi *noise*. Selain kata-kata juga tanda baca seperti titik(.), koma(,) dan tanda baca yang lainnya akan dihilangkan. Selain itu pada proses *Cleaning* juga dilakukan perubahan bentuk kata menjadi lower-case semua. Proses *Remove Stopword* ialah tahap kata-kata yang kurang bermakna atau tidak memiliki arti dilakukan penghapusan seperti kata: saya, dan, atau. Proses *Tokenization* digunakan untuk mengidentifikasi kata-kata dalam teks menjadi beberapa urutan yang terpotong oleh spasi atau karakter spesial. Proses *Stemming* ialah tahap mengubah kata yang berimbuhan kembali ke bentuk aslinya. Pada langkah ketiga ialah ekstraksi fitur dimana dilakukan pembuatan fitur untuk mempermudah jalannya proses *learning Naive Bayes Classifier*. Langkah keempat ialah langkah yang paling penting yaitu *Naive Bayes Classifier*

dimana pada tahap ini dimulainya proses pengklasifikasian berdasarkan sentimen yang ada di dalam dokumen. Setelah langkah satu sampai dengan langkah keempat dilakukan akan menghasilkan model yang nantinya akan digunakan untuk menunjukkan ketepatan hasil klasifikasi. Pada tahap selanjutnya ialah uji model yaitu tahap dimana untuk mengukur nilai performa klasifikasi yang telah dilakukan. Setelah uji model selesai, selanjutnya dilakukan evaluasi model dengan cara melihat tingkat akurasi metode melalui *confusion matrix* dan tabel akurasi serta presisi untuk tiap model.

3.2 Uraian Metodologi

3.2.1 Pengambilan Data

Data yang digunakan dalam penelitian ini adalah data komentar yang didapatkan dari situs *Tripadvisor*. Data yang dikumpulkan berupa data teks yang diambil dengan teknik *crawling* menggunakan ekstensi *Google Chrome*, yaitu *Data Scraper* versi 3.299.84.



Gambar 3. 2 Halaman Awal Situs *Tripadvisor*

Selain dari situs *Tripadvisor*, data yang dikumpulkan juga berasal dari komentar pengunjung wisata pada aplikasi *Facebook*. Data yang berasal dari aplikasi *Facebook* cenderung lebih banyak dikarenakan masih kurangnya pengunjung wisata yang belum *familiar* terhadap situs *Tripadvisor* beda lain dengan aplikasi *Facebook* yang dimana banyak dari semua pengunjung sudah mengenal aplikasi ini dari semua kalangan.



Gambar 3. 3 Logo Aplikasi Facebook

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Komentar															
2	mampir di alun2 krn lapar dan nyobain bakso yg jual di lingkungan alun2, sebrang favehotel rembang. surprise bakso nya enaaakk..															
3	Tempatnya luas dan bersih.Tersedia penjual makanan di sekitar alun-alun dan tempat parkir yang cukup untuk mobil maupun motor.Waktu itu kami mampir agak siang di alun															
4	Bersih dan terawat begitu memasuki area alun-alun kota Rembang.. aneka kuliner di malam hari banyak di jajakan .. tapi begitu lagi langsung bersih...															
5	Banyak penjual makanan, setiap malam akan ramai dengan penaja makanan, sehingga mudah untuk menemukan makanan ketika tidak ingin makanan hotel.															
6	Berjalan-jalan malam hari di alun-alun ini karena letak nya hanya diseberang hotel tempat saya bermalam. Tamannya sekilas bersih tertata dan banyak pedagang makanan. Ke															
7	Jangan lewatkan makanan khas daerah ketika anda berkunjung ke rembang. Terutama lontong Tuyuhan. Masakan lontong dengan kuah rasa seperti kare dan lauk ayam kampi															
8	Saat siang hari alun-alun tidak banyak yang bisa kita jumpai. Saat malam, kita bisa kuliner disana, dari nasi goreng, ayam goreng, nasi gandum (kesukaan) bisa kita coba disana															
9	Alun2 ini Ada didepan hotel tempat Saya menginap. Sekeliling pinggir lapangan dipenuhi berbagai aneka Warung makanan, mayoritas nasi goreng. Dengan harga murah, rasa r															
10	alun alun rembang adalah pusat kegiatan berkumpul masyarakat. Lokasi ini dirawat dengan baik oleh pemda setempat dan suasana cukup rindang.															
11	menikmati kuliner malam hari dikota rembang, lontong tuyuhan, bubur sumsum, nasi gandum hemm yummy deh hanya di alun2 saja gak jauh-jauh															
12	Wisata kuliner saat ini menjadi salah satu kegiatan wisata. Wisata kuliner di Rembang dapat dinikmati di Alun-alun Rembang yang menjadi salah satu tempat tujuan kuliner. M															
13	Pertama kemari dibula november 2016 dan pada bulan dec 2016 saya menginap di fave hotel depan persis alun alun rembang. Alun alun ini aktive dikala matahari telah tergel															
14	Letak Alun-alun strategis di tepi jalan raya lintas pantura & di lalui berbagai jalan kota Rembang,ada bangunan atap & tiang pilar beton ala puing 'Colosseum di Roma' di gerb															
15	Tempat ini lumayan enak untuk kongkow santai dengan teman, lokasinya mudah dicapai dan di tempuh, strategis sekali, sayang fasilitasnya belum ada wifi															
16	alun alun rembang terletak di pinggir jalan pantura.banyak yang jual makanan di area alun alun.terdapat masjid besar agung rembang disebalah barat, favehotel dan terminal															
17	dikenal sebagai alun-alun Rembang, sayang masih sedikit kulinernya. Yang unik adalah kopi leletnya. ampas kopi dipakai melukis rokok.															
18	Alun - alun kota rembang bisa menjadi alternatif tempat untuk kongkow, tempatnya bersih dan tertata. di sekitarnya juga banyak terdapat pedagang makanan dan minuman.															
19	Tempat ini hanya di depan hotel favorit saya.itu bagus untuk pergi di malam hari sampai nite, banyak makanan lokal kopi di sekitar sana, hanya untuk bersantai.ketika siang ha															

Gambar 3. 4 Hasil Crawling Data

Hasil dari *crawling* data disimpan dalam *file* bertipe *xlsx* dan kemudian dilakukan *labelling* untuk menentukan pendapat atau pandangan dari komentar yang diambil. Pada proses *labelling* dibedakan menjadi 3 *class*, yaitu *class positif*, *class negatif*, dan *class positif negatif*. Contoh dari data yang didapat dan hasil dari *labelling* seperti terlihat pada tabel 3.1.

Tabel 3. 1 Contoh Data Hasil Labelling

komentar	label
Bagus dan indah sekali.kami bersama teman2 sangat puas	0
Murah meriah bersih dan nyaman tiket masuk Cuma 5000 rupiah saja	0
Tempatnya enak nyaman dan aman, angin semilir cocok banget, suasana indah banget :)	0
Kurang perawatan lokasi wisatanya :(1

Tempat bilas tidak memadai dan toiletnya sangat kotor	1
Tempat parkir terlalu jauh, harga parkirnya juga mahal, wahananya perlu ditambah lagi	1
Tempatnya bagus sayang kurang dikembangkan	2
Tempat parkirnya luas banget, tapi jangan harap dapat parkir saat liburan tiba	2
Pemandangannya sangat indah cuman banyak sampah dari pengunjung yang kurang sadar kebersihan	2

Pada proses pelabelan untuk mempermudah klasifikasi, kelas sentimen *positif* diganti dengan label 0, kelas sentimen *negatif* diganti dengan label 1, sedangkan untuk kelas sentimen *positif negatif* diganti dengan label 2.

3.2.2 Langkah-Langkah Preprocessing

Tujuan dari tahap ini adalah untuk membersihkan kata-kata yang tidak perlu atau kata-kata yang tidak memiliki makna. Langkah-langkah *preprocessing* yang dilakukan disesuaikan berdasarkan kondisi dari data komentar pengunjung. Adapun urutan dari tahapan *preprocessing* yang dilakukan ialah:

1. *Cleaning*

Seringnya muncul simbol, tanda baca dan bilangan angka pada komentar pengunjung situs Tripadvisor membuat data menjadi tidak efektif dan tidak memiliki arti. Pada tahap ini akan menghapus simbol, bilangan angka, dan tanda baca tersebut. Dan proses ini dilakukan menggunakan program, sehingga dilakukan secara otomatis sebelum menyimpan hasil *decode* ke dalam bentuk *xlsx*. Contoh penerapan pada tahap *cleaning* dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Contoh Proses Penerapan *Cleaning*

Sebelum	Sesudah
Bagus dan indah sekali.kami bersama teman2 sangat puas	bagus dan indah sekali kami bersama sangat puas
Murah meriah bersih dan nyaman tiket masuk Cuma 5000 rupiah saja	murah meriah bersih dan nyaman tiket masuk cuma rupiah saja

Tempatnya enak nyaman dan aman, angin semilir cocok banget, suasana indah banget :)	tempatnnya enak nyaman dan aman angin semilir cocok banget suasana indah banget
Kurang perawatan lokasi wisatanya ;(kurang perawatan lo kasi wisatanya
Tempat bilas tidak memadai dan toiletnya sangat kotor	tempat bilas tidak memadai dan toiletnya sangat kotor
Tempat parkir terlalu jauh, harga parkirnya juga mahal, wahananya perlu ditambah lagi	tempat parkir terlalu jauh harga parkirnya juga mahal wahananya perlu ditambah lagi
Tempatnya bagus sayang kurang dikembangkan	tempatnnya bagus sayang kurang dikembangkan
Tempat parkirnya luas banget, tapi jangan harap dapat parkir saat liburan tiba	tempat par kirnya luas banget tapi jangan harap dapat parkir saat liburan tiba
Pemandangannya sangat indah cuman banyak sampah dari pengunjung yang kurang sadar kebersihan	pemandangannya sangat indah cuman banyak sampah dari pengunjung yang kurang sadar kebersihan

2. *Remove Stopword*

Pada tahap ini kata-kata yang kurang bermakna atau tidak memiliki arti dilakukan penghapusan seperti kata: saya, dan, atau. Contoh penerapan pada tahap *Remove Stopword* dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Contoh Proses Penerapan *Remove Stopword*

Sebelum	Sesudah
bagus dan indah sekali kami bersama sangat puas	bagus indah puas
murah meriah bersih dan nyaman tiket masuk cuma rupiah saja	murah meriah bersih nyaman tiket masuk cuma rupiah
tempatnnya enak nyaman dan aman angin semilir cocok banget suasana indah banget	tempatnnya enak nyaman aman angin semilir cocok banget suasana indah banget

kurang perawatan lokasi wisatanya	kurang perawatan lokasi wisatanya
tempat bilas tidak memadai dan toiletnya sangat kotor	tempat bilas memadai toiletnya kotor
tempat parkir terlalu jauh harga parkirnya juga mahal wahananya perlu ditambah lagi	tempat parkir jauh harga parkirnya mahal wahananya ditambah
tempatnyanya bagus sayang kurang dikembangkan	tempatnyanya bagus sayang kurang dikembangkan
tempat parkirnya luas banget tapi jangan harap dapat parkir saat liburan tiba	tempat parkirnya luas banget harap dapat parkir saat liburan tiba
pemandangannya sangat indah cuman banyak sampah dari pengunjung yang kurang sadar kebersihan	pemandangannya indah cuman sampah pengunjung kurang sadar kebersihan

3. Tokenization

Tokenization adalah tahap memecah kalimat menjadi bagian-bagian dinamakan token. Sebuah token dianggap sebagai satu bentuk sebuah kata, frasa, atau suatu elemen yang berarti. Contoh penerapan pada tahap *tokenization* dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Contoh Proses Penerapan *Tokenization*

Sebelum	Sesudah
bagus indah puas	['bagus', 'indah', 'puas']
murah meriah bersih nyaman tiket masuk cuma rupiah	['murah', 'meriah', 'bersih', 'nyaman', 'tiket', 'masuk', 'cuma', 'rupiah']
tempatnyanya enak nyaman aman angin semilir cocok banget suasana indah banget	['tempatnyanya', 'enak', 'nyaman', 'aman', 'angin', 'semilir', 'cocok', 'banget', 'suasana', 'indah', 'banget']
kurang perawatan lokasi wisatanya	['kurang', 'perawatan', 'lokasi', 'wisatanya']

tempat bilas memadai toiletnya kotor	['tempat', 'bilas', 'memadai', 'toiletnya', 'kotor']
tempat parkir jauh harga parkirnya mahal wahananya ditambah	['tempat', 'parkir', 'jauh', 'harga', 'parkirnya', 'mahal', 'wahananya', 'ditambah']
tempatnyanya bagus sayang kurang dikembangkan	['tempatnyanya', 'bagus', 'sayang', 'kurang', 'dikembangkan']
tempat parkirnya luas banget harap dapat parkir liburan tiba	['tempat', 'parkirnya', 'luas', 'banget', 'harap', 'dapat', 'parkiran', 'liburan', 'tiba']
pemandangannya indah cuman sampah pengunjung kurang sadar kebersihan	['pemandangannya', 'indah', 'cuman', 'sampah', 'pengunjung', 'kurang', 'sadar', 'kebersihan']

4. Stemming

Setelah *tokenization* dilakukan, kemudian peneliti melakukan *stemming*. *Stemming* adalah tahap mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata. Contoh penerapan pada tahap *stemming* dapat dilihat pada Tabel 3.5.

Tabel 3. 5 Contoh Proses Penerapan *Stemming*

Sebelum	Sesudah
['bagus', 'indah', 'puas']	bagus indah puas
['murah', 'meriah', 'bersih', 'nyaman', 'tiket', 'masuk', 'cuma', 'rupiah']	murah riah bersih nyaman tiket masuk cuma rupiah

['tempatnya', 'enak', 'nyaman', 'aman', 'angin', 'semilir', 'cocok', 'banget', 'suasana', 'indah', 'banget']	tempat enak nyaman aman angin milir cocok banget suasana indah banget
['kurang', 'perawatan', 'lokasi', 'wisatanya']	kurang awat lokasi wisata
['tempat', 'bilas', 'memadai', 'toiletnya', 'kotor']	tempat bilas pada toilet kotor
['tempat', 'parkir', 'jauh', 'harga', 'parkirnya', 'mahal', 'wahananya', 'ditambah']	tempat parkir jauh harga parkir mahal wahana tambah
['tempatnya', 'bagus', 'sayang', 'kurang', 'dikembangkan']	tempat bagus sayang kurang kembang
['tempat', 'parkirnya', 'luas', 'banget', 'harap', 'dapat', 'parkiran', 'liburan', 'tiba']	tempat parkir luas banget harap dapat parkir libur tiba
['pemandangannya', 'indah', 'cuman', 'sampah', 'pengunjung', 'kurang', 'sadar', 'kebersihan']	pandang indah cuman sampah ujung kurang sadar bersih

3.2.3 Ekstraksi Fitur

Setelah semua data sudah melewati tahap *preprocessing*, maka langkah selanjutnya adalah pembuatan fitur untuk mempermudah proses klasifikasi. Pada tahap pembuatan fitur dilakukan dua proses yaitu pembuatan *word vector* dimana dilakukan pengubahan fitur teks menjadi sebuah representasi *vector* dari dan pembobotan kata dengan *Tf-Idf*. *Word vector* atau dalam Bahasa Indonesia bisa disebut vektor kata, yaitu membuat kalimat yang sudah menjadi kumpulan *array* menjadi suatu matriks, yang mana setiap baris matriks tersebut mewakili baris dokumen, sedangkan kolom pada matriks mewakili seluruh kata yang ada di seluruh teks yang ada. Setelah sudah berubah menjadi vektor kata, maka selanjutnya adalah memberikan pembobotan terhadap setiap kata pada setiap kalimat atau dokumen menggunakan *Unigram*

dan *tf-idf* menggunakan rumus yang bisa dilihat pada bab sebelumnya, maka *dataset* sudah siap untuk digunakan dalam *training* menggunakan perhitungan pada *naive bayes*.

3.2.4 Klasifikasi Naive Bayes

Metode *naive bayes* merupakan metode yang digunakan untuk mengkasifikasikan data komentar untuk memperoleh sentimen analisis. Untuk melakukan klasifikasi sentimen akan menggunakan data hasil proses *preproceesing* sampai dengan pembobotan kata dengan *tf-idf*. Setelah data berhasil di-*training* kemudian akan dilakukan pengujian menggunakan data *test* untuk menguji hasil ketepatan klasifikasi yang dilakukan.

3.2.5 Uji Model

Proses uji model dilakukan setelah proses training data dilakukan. Uji model sendiri dilakukan untuk mengetahui kinerja model. Jumlah data yang dijadikan untuk uji model diambil dari data *training* sebesar 0,33%. Pengambilan data dilakukan secara acak dengan bantuan *library* dari Python. Setelah uji model dilakukan maka akan tampil seberapa besar akurasi metode yang dilakukan.

3.2.6 Evaluasi Model

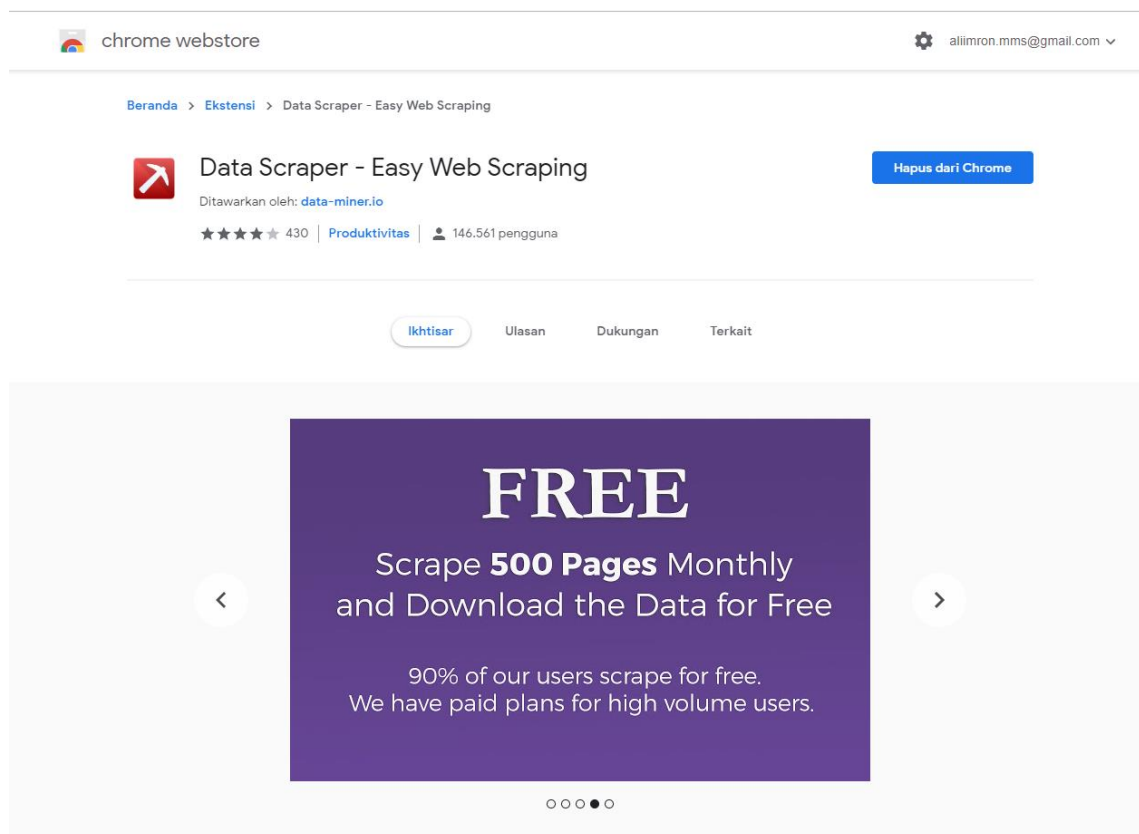
Evaluasi model dilakukan untuk mengetahui kinerja model. Evaluasi model dilakukan dengan cara melihat tingkat akurasi metode melalui *confusion matrix* dan tabel akurasi serta presisi untuk tiap model. Setelah *data test* diujikan terhadap *data training*, maka akan menghasilkan daftar kelas-kelas dari *data test*, sebut sata prediksi kelas. Kemudian prediksi kelas dibandingkan dengan kelas yang sebenarnya dari *data test* yang disembunyikan sebelumnya. Sehingga dapat dilihat dan dihitung nilai *accuration*, *precision*, *recall*, dan *f1-score*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 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. Sebagai contoh pada gambar 4.2 komentar atau ulasan wisata Alun-Alun Kota Rembang.

Ikhtisar Ulasan Terdekat T&J

1-10 dari 18 ulasan




anggunirene
Jakarta, Indonesia

39 4

Diulas pada 10 April 2018 via perangkat selular


Alun Alun Kota Rembang

mampir di alun2 km lapar dan nyobain bakso yg jual di lingkungan alun2, sebrang favehotel rembang. surprise bakso nya enaaakk..



Tanggal pengalaman: Desember 2017

Terima kasih anggunirene




Aufarelqofadi
Surabaya, Indonesia

90 24

Diulas pada 17 Maret 2018


Tempat yang cocok untuk olahraga pagi atau sekedar berkumpul

Tempatnya luas dan bersih. Tersedia penjual makanan di sekitar alun-alun dan tempat parkir yang cukup untuk mobil maupun motor. Waktu itu kami mampir agak siang di alun-alun dan sempat makan bakso serta soto ayam. Rasa makanannya mungkin khas kota rembang, untuk baksonya enak dan ada tetelannya sedangkan... [Selengkapnya](#)



Tanggal pengalaman: Desember 2017

Terima kasih Aufarelqofadi



SUWARDONO
Solo, Indonesia

254 42


Diulas pada 28 November 2017 via perangkat selular

BERSIH...

Bersih dan terawat begitu memasuki area alun-alun kota Rembang.. aneka kuliner di malam hari banyak di jajakan .. tapi begitu lagi langsung bersih... 🍕

Tanggal pengalaman: November 2017

Terima kasih SUWARDONO



Benbernavita

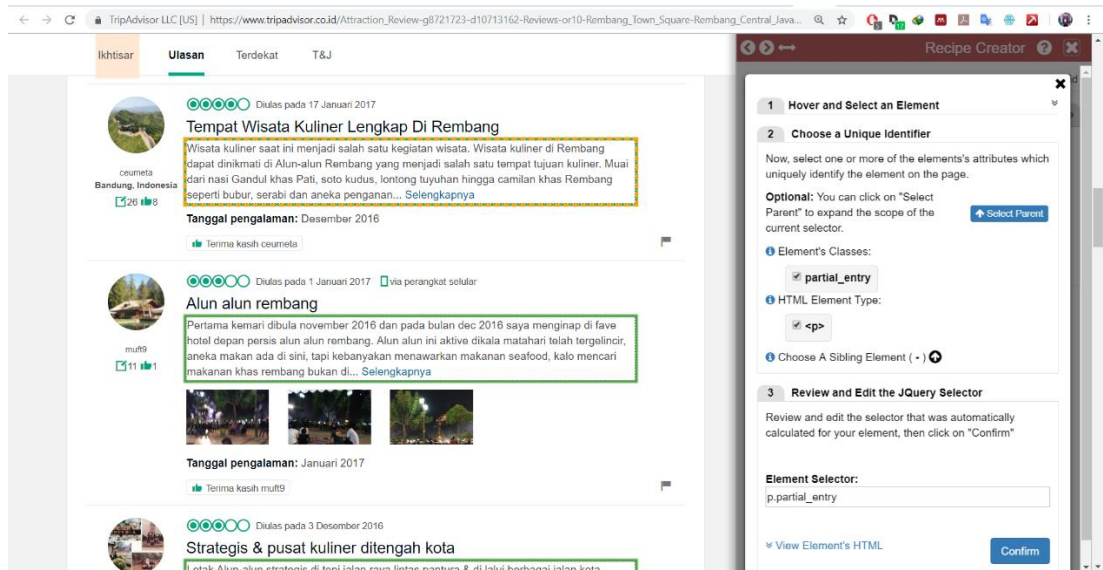
Diulas pada 23 Oktober 2017

Tidak berjudul

Banyak penjual makanan, setiap malam akan ramai dengan penaja makanan, sehingga mudah untuk menemukan makanan ketika tidak ingin makanan hotel.

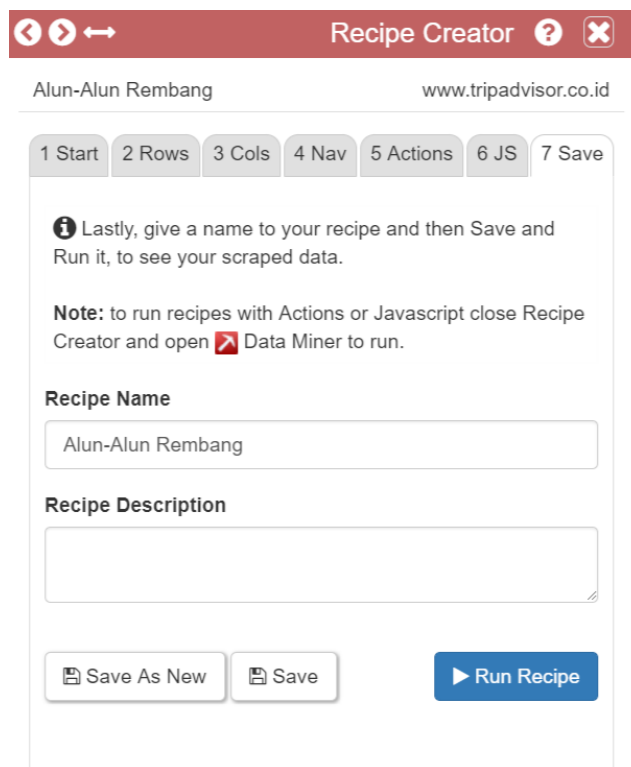
Gambar 4. 2 Komentar Pengunjung Alun-Alun Kota Rembang

Selanjutnya, pengambilan data dimulai dari mengaktifkan *Data Scraper* dan membuat sebuah *query* untuk menentukan bagian mana saja yang akan diambil datanya.

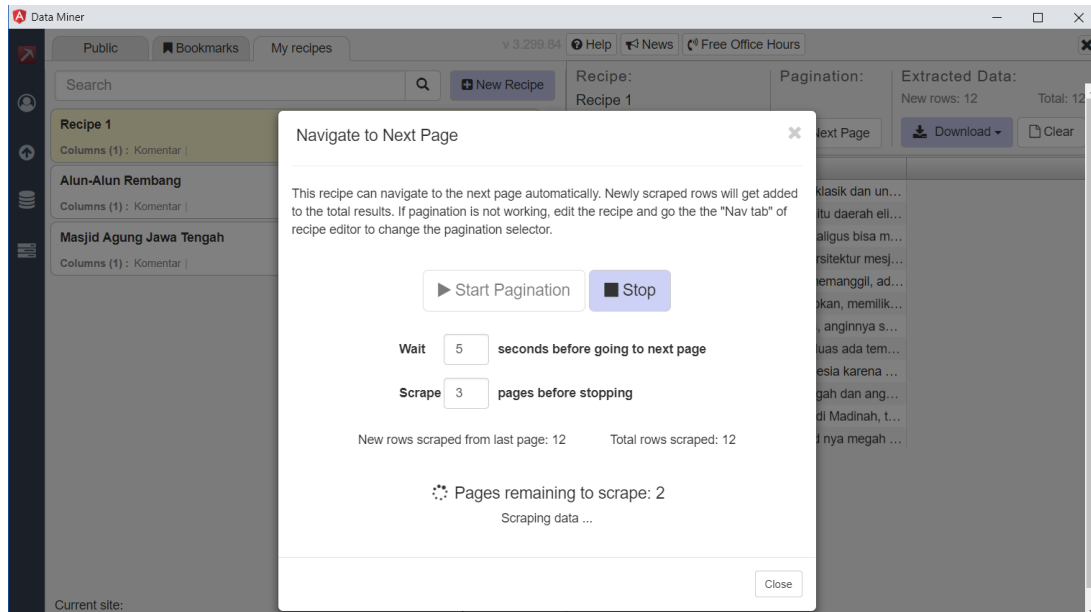


Gambar 4. 3 Pengambilan Data Menggunakan *Data Scraper*

Kemudian untuk melakukan *scraping data* dapat dimulai dengan mengklik tombol *run recipe* yang dapat dilihat pada gambar 4.4.

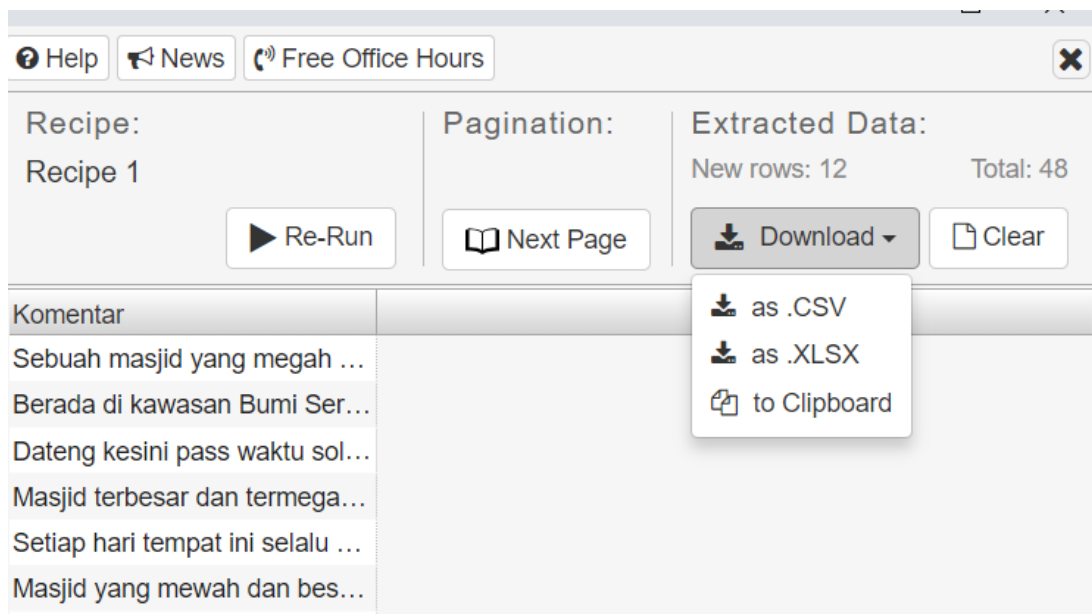


Gambar 4. 4 Langkah Awal *Scraping Data*



Gambar 4. 5 Pengambilan Data Pada Halaman Selanjutnya

Setelah semua komentar terkumpul, selanjutnya menyimpan hasil *scraping data* ke dalam *file excel* dengan *format xlsx*.



Gambar 4. 6 Menyimpan Hasil *Scraping Data*

Data yang tersimpan hasil dari *scraping data* selanjutnya dilakukan labelling untuk membedakan *sentimen positif*, *sentimen negatif* dan *sentimen positif negatif*. Untuk mempermudah proses klasifikasi pada sistem maka *sentimen positif* ditandai dengan label 0, *sentimen negatif* ditandai dengan label 1, sedangkan *sentimen positif negatif*

ditandai dengan label 2. Hasil *scraping data* dan *labelling data* dapat dilihat pada gambar 4.7

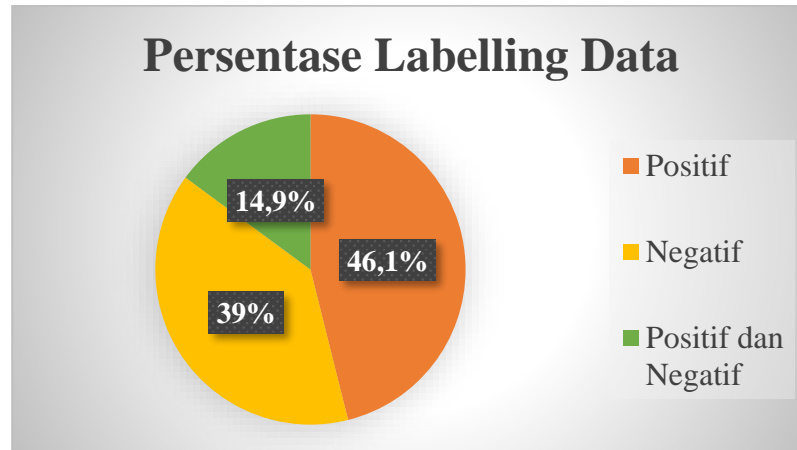
1	text	label
2	bagus dan indah sekali.kami bersama teman2 sangat puas	0
3	Sangat bagus tempat dan pantainya bersih.....klo bisa kedepannya d tmbah lagi spot2 buat foto2ny.....	0
4	cukup menyenangkan sekali banyak pohon2nya di pinggir pantai bisa istirahat....	0
5	Murah meriah bersih dan nyaman.saran spy kedepannya bisa bekerjasama dg perbankan shg bisa dilengkapi dengan atm atau agen brilink.makasih.	0
6	tempatny adem.kulinernya terjangkau.buat liburan asiklah👍	0
7	Pokoknya kjb sangt bgus skali pantainya besrsih indh nyman siiip lah pokoknya	0
8	tempatny nyaman,parkir luas.	0
9	saya suka dengan pantai nya, tpi tolong kamar mandi bilas harus di perbanyak biar ndak ngantri kayak ngantri beras...Air ny juga harus dikondisikan jangan sampai susah.. permainan nya di tambah lagi dan jalan menuju pantai harus di kondisikan supaya lebih baik... yang paling saya suka banyak pohon nya yang begitu bagus.....pasir putih ny yang lembut... yang paling saya gk suka knapa mobil parkir di pinggir pantai kok di biarin aja (ngrusak pemandangan) trus juga pembuangan sampah yang kocar kacir sana sini kurang tertib..buat selfi2	1
10	pohonya sangat banyak Dan pantainya indah	0
11	muantab..... the best..... soal menu makanan wajar tidak dijual lebih mahal alias kata orang jawa "Gak Ditengkek" harga wajar umum. layanan warung makan ramah. dan apabila pinjam tiker kita makan dibelakange warung geruaaaaaaatis tis..... dah ak buktikan. tempate klo g salah arah foto2 yg love love.	0
12	so..... masui lokasi g terlalu nguras kantong T O B. top abis..... moga slamanya gini tros g mehong. ... tyus.....	0
13	Tempat yg nyaman dan tenang,cocok untuk tempat santai berwisata dgn keluarga atau outbond training	0
14	Mantaap..... the best pokonya???lautnya bersih saya suka...saya suka	0
15	Cukup menyenangkan	0
16	Jos tenang	0
17	Yang pasti murah meriah.. Tempatnya bagus.. Top lahh	0
18	Saya senang berkunjung disitu kalau lgie liburan bersama keluarga,, enak tempatnya mg makin maju kedepanya dan bisa lebih baik lgie	0
19	Bagus bnget pantainya sy suka dn puas jajannya murah	0

Gambar 4. 7 Hasil *Scraping Data* dan *Labelling Data*

Pada proses pengambilan data yang dilakukan berdasarkan 2 sumber yaitu situs *Tripadvisor* dan aplikasi *Facebook* untuk wisata Rembang dihasilkan data dengan jumlah 1491 komentar.

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 4.8



Gambar 4. 8 Persentase Data Hasil *Labelling*

4.2 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, di antaranya:

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, di antaranya ialah:

- a. Menghapus tanda baca
- b. Menghapus angka
- c. Menyeragamkan huruf menjadi huruf kecil semua
- d. Menghapus kelebihan spasi

Adapun kode program yang memperlihatkan implementasi dari 4 tahap *cleaning* data ditunjukkan oleh gambar 4.9.

```

10 def cleaning(str):
11
12     # remove punctuations
13     str = re.sub(r'[^\\w]|_', ' ', str)
14     # remove digit from string
15     str = re.sub("\\S*d\\S*", "", str).strip()
16     # remove digit or numbers
17     str = re.sub(r"\\b\\d+\\b", " ", str)
18     # to lowercase
19     str = str.lower()
20     # Remove additional white spaces
21     str = re.sub('[\\s]+', ' ', str)
22
23     return str

```

Gambar 4. 9 Kode Program Tahap *Cleaning*

Pada potongan kode tersebut, keseluruhan proses tahap *cleaning* dilakukan dengan memanfaatkan *regex* atau *regular expression*.

2. Remove Stopword

Tahap *remove stopwords* merupakan tahapan yang dilakukan untuk menghapus stopwords. Proses penghapusan stopwords dari data dilakukan dengan mendefinisikan kata-kata yang didaftar sebagai stopwords terlebih dahulu. Dalam hal ini, seluruh daftar stopwords disimpan ke dalam sebuah *file* yang diberi nama *stopword_id*. *File* tersebut disimpan pada sebuah folder di dalam *corpora* yang terdapat pada *nlk_data*. Tahap *remove stopwords* dilakukan dengan menggunakan bantuan *library* pada bahasa pemrograman *Python3* yang bernama *nlk*. Dalam hal ini, peneliti melakukan instalasi *library* menggunakan perintah *pip* sebagaimana ditunjukkan oleh gambar 4.10.

```

\\PycharmProjects\\sentimenanalisiswisata>pip install nltk

```

Gambar 4. 10 Perintah Instalasi *library nltk*

Sebelum proses *remove stopwords* dilakukan, *library nltk* dideklarasikan terlebih dahulu sebagaimana potongan *source code* pada gambar 4.11.

```

2 import nltk
3 from nltk import word_tokenize
4 from nltk.corpus import stopwords

```

Gambar 4. 11 Deklarasi *library nltk*

Adapun kode program yang memperlihatkan implementasi dari tahap *remove stopwords* ditunjukkan oleh gambar 4.12.

```

24
25 def removeStopword(str):
26     stop_words = set(stopwords.words('stopwords_id'))
27     word_tokens = word_tokenize(str)
28     filtered_sentence = [w for w in word_tokens if not w in stop_words]
29     return ' '.join(filtered_sentence)
30

```

Gambar 4. 12 Kode Program Tahap *Remove Stopword*

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 *nlk*. Adapun kode program yang memperlihatkan implementasi dari tahap *tokenization* ditunjukkan oleh gambar 4.13.

```

36
37 def word_tokenization(str):
38     str = word_tokenize(str)
39
40     return str
41

```

Gambar 4. 13 Kode Program Tahap *Tokenization*

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*. Dalam hal ini, peneliti melakukan instalasi *library* menggunakan perintah *pip* sebagaimana ditunjukkan oleh gambar 4.14.

```

\\PycharmProjects\\sentimenanalisiswisata>pip install sastrawi

```

Gambar 4. 14 Perintah *Instalasi Library Sastrawi*

Sebelum proses *stemming* dilakukan, *library sastrawi* dideklarasikan terlebih dahulu sebagaimana potongan *source code* pada gambar 4.15.

```

2   import nltk
3   from nltk import word_tokenize
4   from nltk.corpus import stopwords

```

Gambar 4. 15 Deklarasi *Library Sastrawi*

Adapun kode program yang memperlihatkan implementasi dari tahap *stemming* ditunjukkan oleh gambar 4.16.

```

24
25 def removeStopword(str):
26     stop_words = set(stopwords.words('stopwords_id'))
27     word_tokens = word_tokenize(str)
28     filtered_sentence = [w for w in word_tokens if not w in stop_words]
29     return ' '.join(filtered_sentence)
30

```

Gambar 4. 16 Kode Program Tahap *Stemming*

Setelah data dilakukan *preprocessing*, selanjutnya hasil *preprocessing* disimpan menjadi *file* baru yang nantinya akan digunakan sebagai *dataset* dalam proses klasifikasi. Adapun hasil *preprocessing* dapat dilihat pada gambar 4.17.

		cleanText	label
1			
2	0	bagus indah puas	0
3	1	bagus tempat pantainya bersih depan tmah	0
4	2	cukup senang pinggir pantai istirahat	0
5	3	murah riah bersih nyaman saran depan bekerjasama perban shg lengkap atm agen brilink	0
6	4	tempat adem kuliner jangkau libur asiklah	0
7	5	pokok kjb sangt skali pantai besrsih indh nyaman siiip pokok	0
8	6	tempat nyaman parkir luas	0
9	7	suka pantai tpi tolong kamar mandi bilas banyak biar ngantri kayak ngantri beras air kondisi su	1
10	8	pohonya pantai indah	0
11	9	muantab best soal menu makan wajar jual mahal alias kata orang jawa tengkek harga wajar u	0
12	10	tempat nyaman tenang cocok tempat santai wisata keluarga outbond training	0
13	11	mantaap best poko laut bersih suka suka	0
14	12	cukup menyenangkan	0
15	13	jos tenan	0
16	14	murah riah tempat bagus top lahh	0
17	15	senang kunjung situ lgie libur keluarga enak tempat maju depa baik lgie	0
18	16	bagus bnget pantai suka puas jajannya murah	0
19	17	maen karang jahe beach tempat bersih harga makan main jangkau tempat wisata cocok bawa	0
20	18	pantai bersih pandang lumayan asri	0

Gambar 4. 17 Hasil *Preprocessing*

4.3 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

Tabel 4. 1 Pembuatan *Word Vector*

	Pantai	Bagus	Taman	Indah
(D1)	1	1	0	0
(D2)	0	0	2	1
(D3)	1	0	0	1

Data yang sudah menjadi *word vector* kemudian dihitung menggunakan rumus *TF-IDF* sehingga menghasilkan *word vector* dengan nilai yang sudah terbobot. Adapun *TF* (*Term Frequency*) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan, sedangkan *IDF* (*Inverse Document Frequency*) merupakan sebuah perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Proses pembobotan kata dilakukan dengan menghitung *TF* (*Term Frequency*) terlebih dahulu. Adapaun contohnya dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Tabel *TF* (*Term Frequency*)

	(D1)	(D2)	(D3)
Pantai	1	0	1
Bagus	1	0	0
Taman	0	2	0
Indah	0	1	1

Setelah terbentuk *TF* (*Term Frequency*) selanjutnya menentukan *DF* (*Document Frequency*) yaitu banyaknya *term* (*t*) muncul dalam semua dokumen. Adapun hasilnya dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Tabel *DF* (*Document Frequency*)

<i>T</i> (<i>Term</i>)	<i>DF</i> (<i>Document Frequency</i>)
Pantai	2
Bagus	1
Taman	2
Indah	2

Kemudian menghitung nilai *IDF* (*Inverse Document Frequency*) dengan menghitung nilai log dari hasil D (jumlah dokumen) dibagi dengan nilai *DF* (*Document Frequency*). Adapun hasilnya dapat dilihat pada Tabel 4.4.

Tabel 4. 4 *IDF (Inverse Document Frequency)*

<i>T (Term)</i>	<i>DF (Document Frequency)</i>	<i>D/DF</i>	<i>IDF (Inverse Document Frequency)</i>
Pantai	2	1.5	$\log 1.5 = 0.176$
Bagus	1	3	$\log 3 = 0.477$
Taman	2	1.5	$\log 1.5 = 0.176$
Indah	2	1.5	$\log 1.5 = 0.176$

Setelah mengetahui nilai *IDF (Inverse Document Frequency)* langkah selanjutnya langsung dapat menghitung *TF-IDF*, adapun hasil dari contoh di atas dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Contoh Hasil Perhitungan *TF-IDF*

<i>Q</i>	<i>TF</i>			<i>DF</i>	<i>D/DF</i>	<i>IDF</i>	<i>IDF+1</i>	<i>W = TF*(IDF+1)</i>		
	<i>D1</i>	<i>D2</i>	<i>D3</i>					<i>D1</i>	<i>D2</i>	<i>D3</i>
Pantai	1	0	1	2	1.5	0.176	1.176	1.176	0	1.176
Bagus	1	0	0	1	3	0.477	1.477	1.477	0	0
Taman	0	2	0	2	1.5	0.176	1.176	0	2.352	0
Indah	0	1	1	2	1.5	0.176	1.176	0	1.176	1.176
Nilai Bobot Setiap Dokumen								2.653	3.528	2.352

Adapun hasil *word vector* yang sudah terbobot dapat dilihat pada Tabel 4.6

Tabel 4. 6 Contoh dari *Word Vector* yang sudah terbobot

	Pantai	Bagus	Taman	Indah
(D1)	1.176	1.477	0	0
(D2)	0	0	2.352	1.176
(D3)	1.176	0	0	1.176

Pada penelitian ini proses pembuatan *word vector* dan pembobotan kata menggunakan bantuan *library Python 3* yaitu *TfidfVectorizer*, adapun hasil dari pembobotan *word vector* dapat dilihat pada gambar 4.20.

0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.20565485	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.3070992	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.

Gambar 4. 20 Data *Word Vector* yang sudah Terbobot

4.4 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 di antaranya 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 4.21.

```

13 import pandas as pd
14 import numpy as np
15 from sklearn.naive_bayes import MultinomialNB
16 from sklearn.feature_extraction.text import CountVectorizer
17 from sklearn.feature_extraction.text import TfidfTransformer
18 from sklearn.pipeline import Pipeline
19 from sklearn.model_selection import train_test_split

```

Gambar 4. 21 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 4.22.

```
26 data = pd.read_excel('datasetreimbang.xlsx')
27 len(data)
28 .
```

Gambar 4. 22 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 4.23.

```
#Multinomial Naive Bayes
pipeline_mnb = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer(use_idf=True, smooth_idf=True)),
    ('clf', MultinomialNB(alpha=1))
])

X_train, X_test, y_train, y_test = train_test_split(data['cleanText'], data['label'], test_size=0.33, random_state=0)
pipeline_mnb.fit(X_train, y_train)
predictions = pipeline_mnb.predict(X_test)
```

Gambar 4. 23 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.

4.5 Uji Model

Untuk mengetahui performa dari *Algoritma Naive Bayes*, maka dilakukan pengujian terhadap model. Hasil klasifikasi akan ditampilkan dalam bentuk *confusion matix*. Tabel *confusion matrix* terdiri dari kelas *predicted* dan kelas *actual*. Model *confusion matrix* 3x3 ditunjukkan pada Tabel 4.7.

Tabel 4. 7 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 4.24.

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

Gambar 4. 24 Perhitungan Akurasi Model

Pada proses uji model yang dilakukan menghasilkan nilai akurasi dan *confusion matrix* 3x3 yang dapat dilihat pada gambar 4.25.

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

Gambar 4. 25 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 kelas pada data *training* tidak seimbang, maka besarnya nilai akurasi bukanlah terpenting.

4.6 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* dengan ukuran 3x3 yang dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Hasil *Confusion Matrix*

		<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

Seperti yang terlihat pada tabel 4.8, *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:

- Kelas *Positif*

Tabel 4. 9 *Confusion Matrix* Kelas *Positif*

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

- Kelas *Negatif*

Tabel 4. 10 *Confusion Matrix* Kelas *Negatif*

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

- Kelas *Positif Negatif*

Tabel 4. 11 *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 4.26

```
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. 26 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 4.27.


```

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. 27 Hasil Pengukuran Evaluasi Performa

Hasil presisi, *recall*, dan *f-1 score* di setiap kelasnya dapat dilihat pada Tabel 4.11.

Tabel 4. 12 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 keduadengan 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 4.28.

```

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. 28 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 4.13.

Tabel 4. 13 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.

BAB V

KESIMPULAN

5.1 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*.

5.2 Saran

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

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*.
- Fannisa, S., Fauzi, M., & 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., Fauzi, M., & Afirianto, T. (2016). Sentiment Analysis pada Review Aplikasi Mobile Menggunakan Metode Naive Bayes dan Query Expansion.
- G, V., & Chandrasekaran, D. (2015). *A comparative performance evaluation of neural network based approach for sentiment classification of online reviews*. Journal of King Saud University.
- Haryanto, D., Muflikhah, L., & Fauzi, M. (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. (2014). ANALISIS SENTIMEN DAN KLASIFIKASI KATEGORI TERHADAP TOKOH PUBLIK PADA TWITTER. *Seminar Nasional Informatika 2014 (semnasIF 2014)*. Yogyakarta.
- 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*, 250-256.
- Prasetyo, E. (2012). *Data Mining Konsep dan Aplikasi menggunakan MATLAB*. Yogyakarta: Penerbit Andi.
- Susanti, N. (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*.

LAMPIRAN