TUGAS AKHIR – KI141502

DETEKSI EMOSI MANUSIA PADA TWEET BAHASA INDONESIA DENGAN KLASIFIKASI NAIVE BAYES

MAHARDHIKA MAULANA
NRP 5111 100 052

Dosen Pembimbing
Diana Purwitasari, S.Kom., M.Sc.
Dr. Eng.Chastine Fatichah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

FINAL PROJECT – KI141502

EMOTION DETECTION OF INDONESIAN TWEETS USING NAIVE BAYES CLASSIFICATION

MAHARDHIKA MAULANA
NRP 5111 100 052

Advisor
Diana Purwitasari, S.Kom., M.Sc.
Dr. Eng.Chastine Fatichah, S.Kom., M.Kom.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Segala puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **"*Deteksi Emosi Manusia pada Tweet Bahasa Indonesia dengan Klasifikasi Naive Bayes*"**.

Dalam pelaksanaan tugas akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Ibu dan ayah, serta dua kakak penulis yang senantiasa memberikan semangat, dukungan dan doa agar penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Ibu Diana Purwitasari, S.Kom, M.Sc selaku dosen pembimbing tugas akhir pertama yang telah membimbing dengan penuh kesabaran dan dukungan, memotivasi dan memberikan banyak masukan dalam pengerjaan tugas akhir ini.
3. Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. selaku dosen pembimbing tugas akhir kedua yang telah membimbing, memotivasi dengan penuh senyuman dan memberikan banyak masukan dalam pengerjaan tugas akhir ini.
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah mengajarkan banyak ilmu berharga kepada penulis.
5. Bapak dan Ibu karyawan Jurusan Teknik Informatika ITS atas berbagai bantuan yang telah diberikankepada penulis selama masa perkuliahan.

6. Teman-teman mahasiswa bidang minat Komputasi Cerdas dan Visi yang telah menemani perjuangan mencari ilmu selama mengambil mata kuliah RMK KCV.
7. Teman-teman administrator dan sahabat Laboratorium Komputasi Cerdas dan Visi yang selalu menemani hari-hari saya, Hayam, Farhan, Askary, Andre, Yudha, Ghozi, Addien, Rizok, Haqiqi, Reza, Ihsan, Nida, Nela, Mustofa, Ano dan Jono.
8. Teman-teman Teknik Informatika ITS angkatan 2011, yang telah memberikan warna-warni kehidupan mahasiswa mulai sejak mahasiswa baru hingga lulus.
9. Pihak-pihak lain yang tidak sempat penulis sebutkan, yang telah membantu kelancaran pengerjaan tugas akhir ini.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak, khususnya bagi diri penulis sendiri dan seluruh *civitas academica* Teknik Informatika ITS, serta bagi agama, bangsa, dan negara. Tak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan tugas akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Januari 2016

Mahardhika Maulana

DETEKSI EMOSI MANUSIA PADA TWEET BAHASA INDONESIA DENGAN KLASIFIKASI NAIVE BAYES

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
MAHARDHIKA MAULANA
NRP : 5111 100 052

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Diana Purwitasari, S.Kom., M.Sc. .....
NIP: 197804102003122001 (pembimbing 1)

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. .....
NIP: 197512202001122002 (pembimbing 2)

SURABAYA
JANUARI 2016

DETEKSI EMOSI MANUSIA PADA TWEET BAHASA INDONESIA DENGAN KLASIFIKASI NAIVE BAYES

Nama Mahasiswa : MAHARDHIKA MAULANA
NRP : 5111 100 052
Jurusan : Teknik Informatika ITS
Dosen Pembimbing I : Diana Purwitasari, S.Kom., M.Sc.
Dosen Pembimbing II : Dr. Eng. Chastine Fatichah, S.Kom., M.Kom

Abstrak

Emosi manusia memegang peranan penting dalam kehidupan sehari-hari. Mengenali emosi manusia mampu membantu manusia dalam proses pengenalan kepribadian, rekomendasi produk, dan deteksi tingkat kriminalitas pada suatu tempat. Twitter sebagai salah satu sosial media terbesar memberikan wadah dimana manusia dapat berinteraksi dan menyampaikan opini kepada manusia lain dengan cepat. Oleh karena itu, diperlukan deteksi emosi manusia dari tweet untuk memahami bagaimana emosi manusia dalam berinteraksi di sosial media.

Pada tugas akhir ini sistem yang diimplementasikan berupa sistem yang mampu mendeteksi emosi dari pengguna dengan klasifikasi naive bayes. Data yang diambil dari tweet bahasa Indonesia dengan tenggang waktu tertentu. Emosi yang digunakan adalah emosi yang didefinisikan oleh Paul Ekman yaitu emosi senang, sedih, marah, terkejut, takut dan jijik. Tahap pertama adalah pemberian label kelas dilakukan berdasarkan penanda emoticon dan hashtag yang berada di dalam tweet untuk menghindari pemberian label secara manual pada data yang sangat besar. Tahap kedua adalah preprocessing untuk menghapus tweet yang tidak diperlukan seperti tweet duplikat dan retweet lalu dilakukan stemming untuk mencari akar kata. Tahap ketiga klasifikasi naive bayes untuk menciptakan model klasifikasi yang dapat melakukan deteksi emosi pada tweet.

Uji coba pada tugas akhir ini menggunakan data tweet yang dibagi 80% untuk training dan 20% untuk testing. Uji coba dilakukan dengan jumlah data yang berbeda dan penanda yang

berbeda. Hasil uji coba sistem bahwa sistem dapat melakukan deteksi emosi cukup baik pada kelas emosi netral, senang dan sedih dengan fscore masing-masing 77%, 75% dan 65%. Sedangkan performa pada kelas marah hanya mencapai 37%. Pada kelas terkejut dan takut sebesar 27% dan 23% dan pada kelas jijik, model klasifikasi tidak dapat melakukan deteksi sama sekali.

Kata kunci: deteksi emosi, klasifikasi teks., naive bayes, Twitter

EMOTION DETECTION OF INDONESIAN TWEET USING NAIVE BAYES CLASSIFICATION

Name : MAHARDHIKA MAULANA
NRP : 5111 100 052
Major : Informatics Department – ITS
Supervisor I : Diana Purwitasari, S.Kom., M.Sc.
Supervisor II : Dr. Eng. Chastine Fatichah, S.Kom., M.Kom

Abstract

Human emotion plays an important role in daily life. Human emotions recognition could help people in personality assesment, product recommendations, and detection of the crime rate. Twitter as one of the biggest social media provide a place where people can interact and express opinions to other humans quickly. Therefore, it is possible to detect emotion from Twitter to understand how human emotions role interacting in social media.

In this final project, a system that is able to detect the emotions of users with Naive Bayes classification is created. Data taken from Indonesian tweet with a certain grace period. Emotion class defined by Paul Ekman is happy, sad, angry, surprised, scared and disgusted. First class labelling are conducted based on emoticons and hashtags markers inside tweet to avoid manual annotation on very large data. The second stage is preprocessing to remove unneeded tweet and word, The third stage is using Naive Bayes classification to create a classification model that can detect emotions in a tweet.

The evaluation in this final project uses data that is with division of 80% for training and 20% for testing. The test is done with a number of different data and different markers to label the emotion. The results is the system can detect emotions well enough in class of neutral emotion, happy and sad with fscore respectively 77%, 75% and 65%. While the performance of the angry class only reached 37%. On the class surprised and scared by 27% and 23% and in disgust class can not detect at all.

Keywords: emotion detection, Naive Bayes, text classification, Twitter

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Berkembangnya teknologi informasi dan jejaring sosial menciptakan tren baru dalam berinteraksi dengan media online. Sosial media seperti *Facebook* dan *Twitter* digunakan dalam basis harian oleh pengguna dari seluruh dunia untuk mengungkapkan opini dan berkomunikasi. Hal ini memberikan sebuah tantangan baru pada bidang klasfikasi sentimen dan opini. Tantangan ini berupa bagaimana cara mendapatkan sentimen dan opini dari teks yang pendek dan tidak terstruktur. Twitter adalah salah satu situs media sosial terkenal dimana pengguna membagikan pesan terbaru sepanjang 140 karakter(*tweet*) tiap pesan. Tugas akhir ini bertujuan untuk mendapatkan emosi dari pengguna sosial media *twitter* berdasarkan *tweet* sepanjang 140 karakter yang dibagikan secara online. Manfaat dari tugas akhir ini adalah untuk memberi Twitter data untuk memberi rekomendasi terkait dengan kondisi emosi pengguna. Rekomendasi ini dapat berupa rekomendasi produk, rekomendasi orang yang dapat diikuti.

Terdapat beberapa pendekatan dalam klasifikasi emosi menggunakan data teks, pendekatan tersebut seperti pendekatan *keyword-based*, *learning-based* dan gabungan dari dua metode tersebut [1]. Pendekatan *keyword-based* dapat menggunakan fitur OMCS (*Open Mind Common Sense Knowledge*) dan WordNet-Affect DB. Pendekatan OMCS menggunakan database yang berisi kalimat-kalimat sederhana yang memiliki *knowledge* tertentu. *Knowledge* kemudian digunakan secara komputasional sehingga dapat direpresentasikan secara terstruktur. WordNet-Affect DB

menggunakan database dimana terdapat makna kata, contoh penggunaan kata dan sinonim kata yang terkait, dalam kata lain WordNet-Affect DB merupakan gabungan dari kamus dan *thesaurus* [2]. Untuk klasifikasi sentimen dari *Twitter* terdapat beberapa penelitian, penelitian yang dilakukan oleh Alec Go memetakan tweet menjadi tiga kelas yaitu kelas sentimen positif, sentimen negatif dan sentimen netral, Go menggunakan *emoticon* untuk mendapatkan label kelas pada data reduksi fitur, dan metode klasifikasi *naive bayes,* metode *maximum entropy* dan metode *Support Vector Machines* [3].

Emosi yang digunakan dalam tugas akhir ini sebanyak enam jenis emosi yaitu senang, sedih, marah, takut, terkejut, jijik/muak dan ditambah dengan satu emosi netral untuk data yang tidak memiliki kecenderungan kepada emosi manapun. Emosi ini didefinisikan oleh psikologis Paul Ekman. Paul Ekman telah mempelajari emosi manusia dengan mengunjungi seluruh dunia dari perkotaan hingga ke daearah terpencil untuk menemukan kesamaan emosi apa saja yang terdapat pada seluruh manusia terlepas dari lokasi dan budayanya [4]. Emosi dan hal-hal yang digunakan untuk menilai emosi yang didefinisikan oleh Paul Ekman telah banyak digunakan dalam bidang klasifikasi emosi, salah satunya adalah pengenalan emosi dari ekspresi wajah manusia [5]

Untuk pengambilan data twitter digunakan API (*Application Programming Interface*) dari twitter yang memberikan data dengan format JSON [6]. JSON diolah oleh library *Twitter4j* dan digunakan. Pemberian label kelas merupakan masalah yang cukup sulit pada data tanpa label dengan jumlah yang sangat besar, oleh karena itu digunakan metode pelabelan otomatis yaitu dengan memberikan label secara otomatis berdasarkan penanda tertentu [7]. Metode klasifikasi yang digunakan adalah metode klasifikasi Naive Bayes, metode Naive Bayes merupakan salah satu metode klasifikasi yang memiliki performa baik untuk klasifikasi data teks [8].

Hasil yang diharapkan dari tugas akhir ini adalah klasifikasi emosi user berdasarkan *tweet* atau pesan terakhir yang dikirimkan di twitter. Data emosi pengguna ini dapat berguna rekomendasi terkait *mood*(suasana hati) pengguna tersebut. Rekomendasi terkait mood dapat dapat berupa rekomendasi musik [9] atau rekomendasi film yang dapat ditonton [10].

1.2. Rumusan Masalah

Rumusan masalah yang terdapat pada tugas akhir ini adalah sebagai berikut

1. Bagaimana mendapatkan, mengolah dan memberi label secara otomatis pada data Twitter dalam skala besar?
2. Bagaimana mengekstraksi fitur penting dari teks pendek sepanjang 140 karakter?
3. Bagaimana sistem dapat memberikan prediksi emosi seseorang berdasarkan *tweet* terakhirnya?

1.3. Batasan Masalah

1. Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:
2. *Tweet* yang diambil untuk dataset adalah *tweet* yang berbahasa Indonesia.
3. Pemberian label menggunakan kata kunciyang sudah ditentukan sebelumnya [11].
4. Kelas emosi yang digunakan adalah enam kelas yang didefinisikan oleh Ekman, yaitu senang, sedih, marah, takut, terkejut dan jijik/muak [4].
5. Emosi yang dapat diklasifikasikan adalah emosi yang bersifat eksplisit.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir iniadalah untuk menciptakan sebuah aplikasi yang dapat memberikan kondisi

emosi pengguna twitter berdasarkan tweet menggunakan metode klasifikasi Naive Bayes.

1.5. Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. **Penyusunan proposal tugas akhir**
   Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang d iangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat darihasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. **Studi literatur**
   Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi ini yaitu Twitter API,*preprocessing data*, metode klasifikasi Naive Bayes, ekstraksi fitur bag of wordsdari teks dan distant supervision untuk pemberian label data yang belum memiliki label.

3. **Implementasi**
   Aplikasi ini akan dibangun dengan bahasa pemrograman java dengan bantuan library *twitter4j* untuk mendapatkan data *tweet* dalam bentuk objek. Aplikasi ini akan dibangun dengan menggunakan *Integrated Development Environment*

(IDE) Netbeans IDE 7.3.0 untuk melakukan pengambilan data, *preprocessing* dan Microsoft SQL Server 2008 R2 untuk menyimpan data *tweet,* menyimpan data kata-kata dari *tweet,* melakukan klasifikasi*,* menyimpan peluang kemunculan kata-kata dari *tweet*, menghapus *tweet* yang tidak penting.

4. **Uji Coba dan Evaluasi**

   Pada tahap ini dilakukan uji coba aplikasi dan evaluasi terhadap implementasi metode pada aplikasi. Pengujian ini mengukur kemampuan aplikasi dalam melakukan klasifikasi emosi dari tweet. Pengujian ini meliputi pengujian akurasi secara keseluruhan, pengujian *precision* dan *recall* tiap kelas dan pengujian akurasi klasifikasi berdasarkan daerah tertentu.

5. **Penyusunan Buku Tugas Akhir**

   Tahap ini merupakan tahap dokumentasi dari tugas akhir. Buku tugas akhir berisi dasar teori, perancangan, implementasi dan hasil uji coba dan evaluasi dari aplikasi yang dibangun.

1.6. Sistematika Penulisan

Buku tugas akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan

   Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir. Tinjauan pustaka pada tugas akhir ini meliputi pembahasan mengenai emosi manusia, pembahasan mengenai *preprocessing* yang meliputi pembersihan data,*distant supervision*, metode klasifikasi naive bayes dan penerapannya dalam klasifikasi teks, dan metode evaluasi dari klasifikasi.

3. Bab III. Analisis dan Perancangan

   Bab analisis dan perancangan berisi penjelasan mengenai pengambilan data *tweet*, perancangan data *tweet*, perancangan *preprocessing tweet* yang meliputi *stemming,* URL *Removal,* penghapusan *tweet* duplikat, penghapusan *retweet*, perancangan sistem klasifikasi naive bayes yang meliputi *training* dan *testing*, perancangan halaman antarmuka pengguna dan perangkat yang digunakan dalam pengerjaan tugas akhir

4. Bab IV. Implementasi

   Bab implementasi berisi pembangunan implementasi deteksi emosi manusia menggunakan klasifikasi Naive Bayes sesuai dengan rumusan dan batasan yang sudah dijelaskan pada bagian pendahuluan. Implementasi berisi lingkungan perangkat implementasi, *pseudocode* dari algoritma-algoritma yang digunakan, proses apa saja yang terlibat, keluaran dari masing-masing proses, diagram alir untuk menampilkan urutan proses, query-query database yang terkait dengan proses klasifikasi dan implementasi antarmuka pengguna.

5. Bab V. Pengujian dan Evaluasi

   Bab uji coba dan evaluasi berisi pembahasan mengenai hasil dari uji coba yang dilakukan terhadap aplikasi deteksi emosi manusia menggunakan metode klasifikasi Naive Bayes. Selain pengujian akurasi, dilakukan pengujian *recall,*

precision dan *fscore* untuk mengetahui performa dari setiap kelas dikarenakan jumlah data pada masing-masing kelas tidak seimbang. Pada tahap pengujian digunakan *Confusion Matrix* digunakan untuk mempermudah perhitungan *recall, precision* dan *fscore* dan untuk mempermudah analisis distribusi data.

6. Bab VI. Kesimpulan dan Saran

   Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan tugas akhir.

[Halaman ini sengaja dikosongkan]

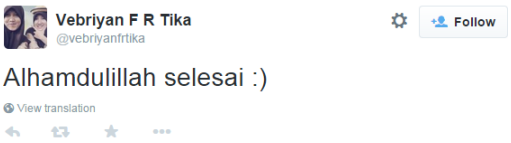
BAB II
TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

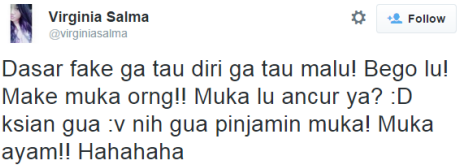
2.1. Emosi Manusia

Emosi manusia adalah suatu keadaan jiwa kompleks yang terdiri dari tiga komponen berbeda yaitu pengalaman subjektif, respon psikologis dan respon ekspresif [12]. Untuk lebih memahami emosi, peneliti melakukan identifikasi dan klasifikasi pada beberapa jenis emosi. Pada 1972, psikologis Paul Ekman menyatakan bahwa terdapat enam emosi dasar yang terdapat pada budaya manusia secara universal. Ekman melakukan penilitian di seluruh pelosok dunia dan menemukan enam emosi yang sama melalui ekpresi wajah.

Enam emosi dasar tersebut antara lain bahagia, sedih, terkejut, marah, jijik dan takut. Ekman menemukan bahwa enam ekspresi ini bersifat universal kepada seluruh manusia di dunia [4]. Selain menggunakan ekspresi wajah, terdapat juga studi semantik terhadap emosi yang telah didefinisikan oleh Ekman [13]. Studi ini meneliti kata apa saja yang berhubungan dengan emosi yang terkait. Kata-kata yang ada dalam emosi ter.Pada tugas akhir ini *tweet* akan dilakukan klasifikasi berdasarkan enam kelas tersebut.



Gambar 2.1 Contoh *Tweet* yang Tergolong dalam Emosi Senang oleh akun @vebriyanfrtika



Gambar 2.2 Contoh *Tweet* yang Tergolong dalam Emosi Marah oleh akun @virginiasalma

2.2. Preprocessing

Preprocessing adalah proses yang penting dalam proses data mining. Proses ini dilakukan dengan menghilangkan data yang tidak diperlukan dalam komputasi [14]. Secara umum hal yang termasuk dalam *preprocessing* adalah *cleaning,* normalisasi, transformasi, pemilihan fitur dan ekstraksi fitur [15]. *Preprocessing* pada tugas akhir ini dibagi menjadi tiga langkah yaitu *URL removal, stopwords removal* dan *stemming.URL Removal* adalah proses menghilangkan URL yang ada pada *tweet* karena tidak memberikan informasi yang terkait dengan deteksi emosi. *Stop Words Removal* adalah penghapusan kata-kata yang sering muncul sehingga dianggap tidak penting untuk mempercepat komputasi pada pemrosesan teks. Kosa kata yang termasuk dalam *stop words* bahasa indonesia yang digunakan dalam tugas akhir ini adalah *stop words* yang didefinisikan oleh Fadhillah Z Tala [16]. *Stemming* adalah proses pengambilan kata dasar dari suatu kata yang sudah diberi imbuhan. Algoritma terkenal dalam proses *stemming* adalah Porter-Stemming [17].

Porter stemmer menghilangkan imbuhan-imbuhan yang dianggap tidak penting. Porter stemmer adalah algoritma yang terkenal dalam *stemming* menggunakan bahasa Inggris. Algoritma stemming khusus untuk bahasa Indonesia yang akan digunakan untuk tugas akhir ini adalah algoritma stemming oleh Nazief-Adriani [18]. Nazief-Adriani mengemukakan bahwa pada umumnya kata dalam bahasa Indonesia terdiri dari kombinasi Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks

1. Dari kombinasi tersebut dibentuk sebuah aturan-aturan untuk membuang imbuhan kata dan mendapatkan kata dasar.

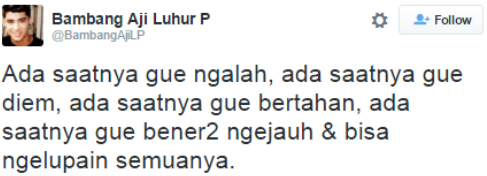
Secara garis besar algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka algoritma berhenti.
2. Hapus akhiran *Inflection Suffixes*(“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) dan *Possesive Pronouns*(“-ku”, “-mu”, atau “-nya”) jika ada.
3. Hapus akhiran *Derivation Suffixes*(“-i”, “-an” atau “-kan”).
4. Hapus awalan *Derivation Prefix*(“be-”, “pe-”, “di-”,”ke-”,”me-”,”ter-’).
5. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word.

Pada setiap tahap, kata akan dicocokan dengan kamus kata dasar, apabila ditemukan maka proses berhenti.

2.3. N-Gram Feature

Fitur N-gram adalah fitur yang dapat berupa huruf, kata atau kalimat yang berjumlah sebanyak *N*. Fitur dengan *N* berjumlah satu disebut unigram. Fitur dengan *N* berjumlah dua disebut bigram **[19]**. Fitur ini dapat digunakan untuk dalam tahap klasifikasi. Pada tugas akhir ini satuan N-Gram yang digunakan adalah satuan kata dan *N* yang digunakan sebanyak satu yaitu unigram. Penggunaan fitur unigram tidak memperhatikan urutan kemunculan kata namum memperhatikan jumlah kemunculan suatu kata. Fitur unigram juga memastikan bahwa setiap fitur tidak memiliki kebergantungan dengan fitur lainnya. Contoh *tweet* dan fitur unigramnya dapat dilihat pada Gambar 2.7 dan Tabel 2.1



Gambar 2.3 Contoh *Tweet* yang Akan Dilakukan Ekstraksi Fitur

Tabel 2.1 Contoh Fitur Unigram dari *Tweet*

Kata	Jumlah
Ada	4
Saatnya	4
Gue	4
Ngalah	1
Diem	1
Ngejauh	1
Bener2	1
Ngelupain	1
Semuanya	1

2.4. Pelabelan Otomatis

Pelabelan Otomatis (*distant supervision*) adalah proses pemberian label kelas yang bersifat otomatis dan *noisy* pada dataset yang belum memiliki label kelas agar data dapat digunakan untuk membangun model klasifikasi. Pemberian label ini dapat berdasarkan kata kunci, relasi, *emoticon* atau hal yang bersifat *knowledge-based* [7]. Metode ini digunakan untuk menghindari pemberian label kelas untuk dataset yang sangat besar. Kata kunci yang digunakan sebagai penanda untuk tiap kelas telah didefinisikan sebelumnya, apabila ditemukan penanda pada satu *tweet*, maka *tweet* tersebut akan diberi label kelas berdasarkan penanda. Penggunaan *automatic labelling* pada analisis sentimen yang dilakukan oleh Alec Go menggunakan emoticon, apabila

terdapat emoticon :) maka tweet mengandung sentimen positif dan sebaliknya apabila terdapat emoticon :(maka tweet mengandung sentimen negatif [3]. Hal ini memberikan kata-kata yang berada pada sekitar penanda memiliki bobot fitur lebih pada kelas yang terkait. Pada tugas akhir ini akan digunakan *hashtag* dan *emoticon* untuk memberikan label kelas secara otomatis kepada data yang berjumlah besar. Evaluasi dari pemberian label secara otomatis akan dilakukan pada tahap pengujian akurasi klasifikasi.

Contoh pemetaan *emoticon* yang dilakukan oleh Alec Go untuk memberikan label kelas dapat dilihat pada Tabel 2.2

Tabel 2.2 Pemetaan Dua Kelas dengan *Emoticon* oleh Go

Kelas Senang	Kelas Sedih
:) :-) :) :D =)	:(:-(: (

Tabel 2.3 Contoh *Tweet* dengan Pemetaan Dua Kelas Menggunakan *Emoticon*

Tweet	Kelas
2 Minggu lagi Ulang tahun saya :)	Senang
Capee :(hari kerjaan banyak beud :(Pulang kehujanan :(	Sedih

Pada kasus diatas, apabila klasifikasi yang digunakan adalah klasifikasi naive bayes dan pencarian akar kata sudah dilakukan maka kata yang berada pada satu *tweet* dengan penanda ":(" seperti kata "pulang", "hujan", "capee" dan "kerja" akan memiliki peluang kemunculan pada kelas sedih yang ditambah sebesar $\Pr(w_n|sedih) = \frac{1}{N+\sum_{x=1}^{N} F_{xsedih}}$, yaitu sebesar kemunculan kata

pada tweet tersebut yaitu masing-masing satu dibagi dengan jumlah kata unik dan jumlah kata pada kelas sedih. Apabila tidak terdapat penanda *emoticon* ":(" maka *tweet* tersebut akan tergolong pada kelas netral dan peluang kemunculan kata tersebut diketahui kelas sedih tidak akan bertambah, namun akan bertambah pada kelas netral sebesar $\Pr(w_n|netral) = \frac{1}{N+\sum_{x=1}^{N} F_{xnetral}}$

2.5. Naive Bayes

Naive Bayes adalah metode yang mampu melakukan klasifikasi data text dengan baik [8]. Tugas akhir ini menggunakan *multinomial Naive Bayes*.

$$\Pr(c|t_i) = \frac{\Pr(c)\,\Pr(t_i|c)}{\Pr(t_i)}, c \in C \qquad (2.1)$$

Dengan asumsi

$\Pr(c|t_i)$: probabilitas kelas c diketahui *tweet* t_i
C : himpunan seluruh kelas emosi.
c : kelas emosi
$\Pr(c)$: probabilitas kemunculan data dengan kelas c
$\Pr(t_i|c)$: probabilitas kemunculan *tweet* t_i diketahui kelas c
$\Pr(t_i)$: probabilitas *tweet* t_i

Metode Multinomial Naive Bayes akan memberi *tweet* yang diuji t_i kelas yang memiliki probabilitas kelas tertinggi $\Pr(c|t_i)$. $\Pr(c)$ didapatkan dengan membagi jumlah *tweet* yang termasuk dalam kelas *c* dengan jumlah *tweet* keseluruhan. $\Pr(t_i|c)$ adalah kemungkinan *tweet* t_i jika diketahui kelas *c*. $\Pr(t_i|c)$ didapatkan dengan menghitung perkalian antar probabilitas kata pada tweet t_i. [20].

$$\Pr(t_i|c) = \alpha \prod_n \Pr(w_n|c)^{\mathrm{fni}}, \qquad (2.2)$$

Dengan asumsi
$\Pr(t_i|c)$: probabilitas kemunculan *tweet* t_i diketahui kelas c
n : jumlah kata dalam satu tweet
$\Pr(w_n|c)$: probabilitas kata w_n diketahui kelas c
f_{ni} : jumlah kata w_n pada *tweet* t_i

Dimana f_{ni} adalah jumlah kata n pada *tweet* yang diuji t_i dan $\Pr(w_n|c)$ adalah probabilitas kata n apabila diketahui kelas *c*. $\Pr(w_n|c)$ didapatkan dengan menghitung jumlah kata tersebut pada kelas yang bersangkutan dibagi dengan jumlah kata unik pada seluruh *tweet* ditambah dengan jumlah kata pada kelas tersebut.

$$\Pr(w_n|c) = \frac{1 + F_{nc}}{N + \sum_{x=1}^{N} F_{xc}}, \quad (2.3)$$

Dengan asumsi
$\Pr(w_n|c)$: probabilitas kata w_n diketahui kelas c
F_{nc} : jumlah kata w_n pada kelas c
N : jumlah kata unik pada data training (*vocabulary*)
$\sum_{x=1}^{N} F_{xc}$: jumlah seluruh kata pada kelas c.

Untuk proses *smoothing* agar tidak terdapat perkalian dengan angka nol pada persamaan 2.2, ditambahkan angka satu pada jumlah kata w_n pada kelas c. Hal ini dapat disebabkan suatu kata terdapat pada satu kelas namun tidak terdapat pada kelas lain.

$$c_{ti} = argmax \ \Pr(c|t_i) \quad (2.4)$$

Kelas dari *tweet* t_i (c_{ti}) didapatkan dengan membandingkan probabilitas dari masing-masing lalu cari kelas dengan probabilitas tertinggi.

Apabila terdapat suatu dataset yang terdiri dari empat dokumen untuk membangun model klasifikasi, satu dokumen untuk pengujian seperti ditunjukan pada Tabel 2.4 dan pada dataset

tersebut memiliki dua kelas yaitu kelas a dan b, maka $\Pr(a) = {}^{3}/_{4}$ dan $\Pr(b) = {}^{1}/_{4}$.

Tabel 2.4 Contoh Kumpulan Dokumen untuk Klasifikasi

	No	Isi Dokumen	Kelas
Data Training	1	Chinese Beijing Chinese	a
	2	Chinese Chinese Shanghai	a
	3	Chinese Macao	a
	4	Tokyo Japan Chinese	b
Data Uji	5	Chinese Chinese Chinese Tokyo Japan	?

$$\Pr(w_n|c) = \frac{1 + F_{nc}}{N + \sum_{x=1}^{N} F_{xc}}, \qquad (2.3)$$

Untuk probabilitas kata *chinese* di kelas *a*:
Jumlah kata chinese di kelas a $F_{chinese\ a}$ = 5
Jumlah kata unik N = 6
Jumlah kata chinese di kelas a $\sum_{x=1}^{N} F_{xc}$ = 8

$$\Pr(Chinese|a) = \frac{1 + 5}{6 + 8} = \frac{3}{7}$$

Dengan cara yang sama, probabilitas kata di kelas a:
$\Pr(Tokyo|a) = 1/14$
$\Pr(Japan|a) = 1/14$

Probabilitas kata di kelas b
$\Pr(Japan|b) = 2/9$
$\Pr(Chinese|b) = 2/9$
$\Pr(Tokyo|b) = 2/9$

Sehingga kemungkinan kelas tweet nomor 5 $\Pr(c|t_5)$ adalah

$$\Pr(a|t_5) = \Pr(\text{a}) * \Pr(Chinese|a)^3 * \Pr(Tokyo|a) * \Pr(Japan|a)$$

$$\Pr(a|t_5) = 3/4 * 3/7^3 * 1/14 * 1/14 \approx 0.0003$$

$$\Pr(b|t_5) = \Pr(b) * \Pr(Chinese|b)^3 * \Pr(Tokyo|b) * \Pr(Japan|b)$$

$$\Pr(b|t_5) = 1/4 * 2/9^3 * 2/9 * 2/9 \approx 0.0001$$

Karena $\Pr(a|t_5)$ memiliki jumlah lebih banyak dari $\Pr(b|t_5)$ maka model klasifikasi akan memberikan dokumen lima kelas a.

2.6. Perhitungan Kinerja Aplikasi

Perhitungan kinerja aplikasi pada sistem ini adalah dengan menggunakan akurasi, *recall, precision* dan *f-score*. Akurasi digunakan untuk mengukur performa keseluruhan sistem. Akurasi didapatkan dengan membagi jumlah data yang dapat diklasifikasikan dengan benar pada seluruh kelas dengan total data. Persamaan untuk pengukuran akurasi dapat dilihat pada persamaan 2.5

$$Akurasi = \frac{Data\ yang\ diprediksi\ dengan\ benar}{Total\ seluruh\ data} \qquad (2.5)$$

Untuk mengukur performa klasifikasi pada masing-masing kelas digunakan metode perhitungan *recall, precision* dan f-score. Ketiga metode tersebut digunakan karena jumlah data pada tiap kelas yang tidak seimbang. *Recall* didapatkan dengan menghitung jumlah data yang dapat diklasifisikan dengan benar pada suatu kelas dibagi dengan jumlah data yang tergolong pada kelas tersebut. *Precision* didapatkan dengan menghitung jumlah data yang dapat diklasifikan dengan benar pada suatu kelas dibagi dengan jumlah data yang diklasifikasikan sebagai kelas tersebut. *Precision* dan *recall* terkadang memiliki nilai yang bertolak belakang. *Fscore* digunakan untuk menghitung performa gabungan antara *precision* dan *recall*. [21]

$$Precision_c = \frac{Jumlah\ prediksi\ data\ kelas\ c\ yang\ benar}{Jumlah\ data\ yang\ diprediksi\ sebagai\ c} \quad (2.\ 6)$$

$$Recall_c = \frac{Jumlah\ prediksi\ data\ kelas\ c\ yang\ benar}{Jumlah\ seluruh\ data\ pada\ kelas\ c} \quad (2.\ 5)$$

Pada persamaan 2.5, $Recall_c$ merupakan *recall* dari kelas *c,* pada persamaan 2.6 $Precision_c$ merupakan *precision* dari kelas *c* dan persamaan 2.7 merupakan *FScore* dari kelas *c.*

$$FScore = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.\ 7)$$

Untuk membantu perhitungan *precision, recall* dan *Fscore,* digunakan *confusion matrix*. *Confusion matrix* adalah sebuah tabel yang sering digunakan untuk menghitung performa model klasifikasi pada data uji. *Confusion Matrix* dapat membantu analisis data dan membantu memetakan kelas mana saja yang memiliki *precision* atau *recall* yang rendah. Contoh *confusion matrix* dapat dilihat pada Tabel 2.5

Tabel 2.5 Contoh *Confusion Matrix* dengan Tiga Kelas

Kelas Sebenarnya			Prediksi Kelas
Netral	Senang	Sedih	
2312	1452	200	Netral
225	**1400**	39	Senang
5835	596	**1342**	Sedih

Pada Tabel 2.5, angka yang dicetak tebal merupakan jumlah data yang mampu diklasifikasikan dengan benar, pada kolom kedua baris pertama terdapat angka 1452, angka tersebut merupakan data yang kelas sebenearnya adalah senang, namun

diprediksi sebagai kelas netral. Data pada kolom pertama pada baris kedua yang berjumlah 225 adalah data yang memiliki kelas sebenarnya netral namun oleh model klasifikasi diklasifikasikan sebagai kelas senang. Data pada kolom pertama baris ke tiga yang berjumlah 5835 adalah jumlah data dengan kelas netral namun oleh model klasifikasi diklasifikasikan sebagai kelas sedih. Pada contoh ini dapat disimpulkan bahwa kelas netral memiliki *recall* yang rendah sebesar 27%. Data pada kelas senang memiliki *recall* rendah sebesar 40% namun *precision* yang tinggi sebesar 84%. Data pada kelas sedih memiliki *recall* yang tinggi sebesar 85% karena hanya 239 data yang gagal diklasifikasikan sebagai sedih namun *precision* yang rendah karena terdapat 5835 data yang diklasifikan sebagai sedih. Jumlah *precision* dan *recall* yang memiliki perbedaan jauh ini lah yang mendorong penggunaan *fscore.* Pada tiga kelas tersebut, kelas senang memiliki *fscore* paling tinggi sebesar 54,7% karena *recall* dan *precisionnya* paling seimbang yaitu masing-masing sebesar 40,6% dan 84,1%. Kelas sedih walaupun memiliki recall yang paling tinggi sebesar 85% namun memiliki *fscore* paling rendah sebesar 28,69% karena selisih antara *recall* dan *precision* yang cukup jauh. Kelas netral walaupun dengan angka *precision* dan *recall* yang rendah dapat memiliki *fscore* yang lebih tinggi dari kelas sedih yaitu sebesar 37,4% Untuk tabel *precision, recall, dan fscore* dari contoh *confusion matrix* diatas dapat dilihat di Tabel 2.6

Tabel 2.6 Penilaian *Recall, Precision* dan *Fscore*

Kelas	Recall	Precision	Fscore
Netral	27,62%	**58,32%**	37,48%
Senang	40,60%	**84,13%**	54,77%
Sedih	**84,88%**	17,26%	28,69%

[Halaman ini sengaja dikosongkan]

BAB III
ANALISIS DAN PERANCANGAN

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari tugas akhir. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

3.1. Analisis Implementasi Metode Secara Umum

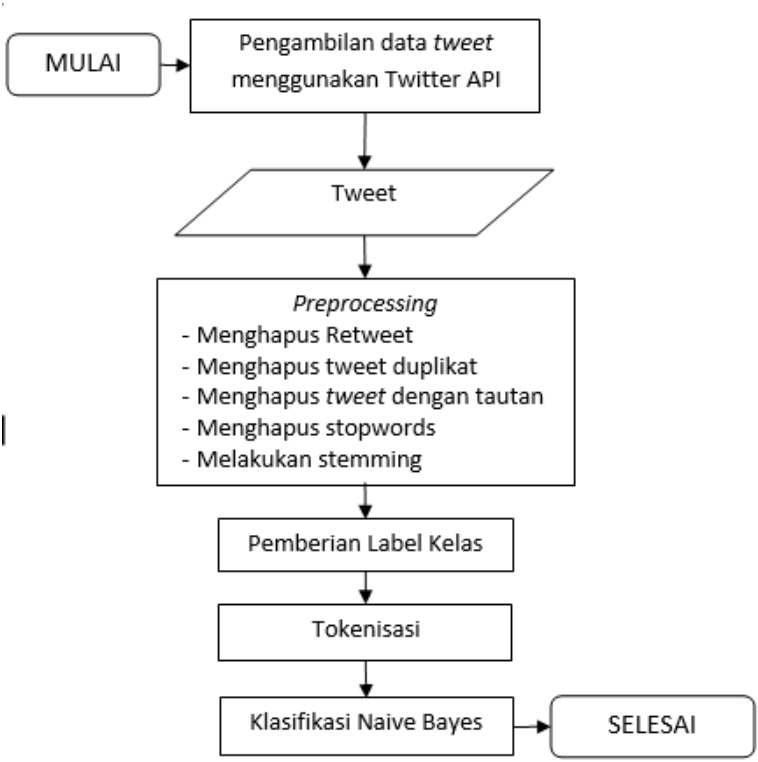
Pada tugas akhir ini akan dibangun sebuah sistem untuk melakukan deteksi emosi manusia pada *tweet* Bahasa Indonesia dengan metode klasifikasi *Naive Bayes*. Proses-proses yang terlibat di dalam implementasi sistem ini meliputi tahap pengambilan data *tweet* dari Twitter, tahap *preprocessing*, tahap pelabelan otomatis, ekstraksi fitur, tahap *training,testing* dan yang terakhir tahap evaluasi. Tahap pertama adalah tahap pengambilan data *tweet* dari Twitter, pada tahap ini digunakan *library* Java *Twitter4j* untuk mempermudah pengambilan data. Setelah *tweet* dikumpulkan, dilakukan *preprocessing*.

Tahap preprocessing dilakukan dengan membuang tweet yang tergolong sebagai spam, tweet yang mengandung tautan ke halaman lain, tweet yang merupakan *retweet* dari *tweet* lain. *Retweet* adalah *tweet* pengguna lain yang dibagikan oleh pengguna lain dalam bentuk *tweet.* Setelah *tweet* yang tidak perlu dibuang, dilakukan penghapusan kata yang dianggap tidak penting (*stopwords)* dan proses pencarian akar kata (*stemming*) menggunakan algoritma *stemming* oleh Nazief-Adriani.

Setelah tahap *preprocessing,* dilakukan pemberian label kelas otomatis berdasarkan kemunculan *hashtag* kata dan *emoticon.* Pemberian label kelas ini dilakukan secara otomatis karena banyaknya jumlah data. Setelah data diberi label, dilakukan ektraksi fitur. Fitur yang digunakan adalah unigram dengan satuan

gram adalah kata. Setiap kata dihitung probabilitas kemunculannya di setiap kelas.

Untuk proses *testing,* dihitung probabilitas setiap kelas dan setiap kata dalam tweet lalu dikalikan, kelas dengan jumlah perkalian tertinggi akan digunakan sebagai hasil akhir klasifikasi. Proses evaluasi dilakukan dengan melakukan perhitungan akurasi keseluruhan, akurasi setiap kelas, *precision, recall* dan *f-score* setiap kelas.



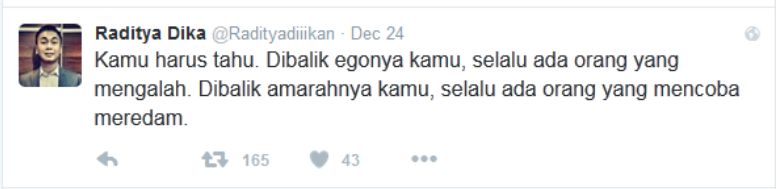
Gambar 3.1 Implementasi Proses Sistem Secara Umum

3.2. Perancangan Data

Pada subbab ini akan dibahas mengenai perancangan data yang merupakan bagian penting karena data sebagai objek yang akan diolah oleh perangkat lunak dalam tugas akhir ini dan menghasilkan sebuah informasi. Data yang akan digunakan pada sistem ini adalah data masukan yang berupa *tweet* yang diambil dari Twitter.

3.2.1. Data Masukan

Data masukan merupakan data awal yang akan diproses oleh sistem untuk melakukan deteksi emosi. Data masukan tersebut berupa *tweet* berbahasa Indonesia yang diambil dari Twitter menggunakan *Search API*. *Tweet* merupakan pesan berbasis teks yang terdiri dari maksimal 140 huruf. Contoh *tweet* yang digunakan sebagai data masukan ditunjukkan oleh Gambar 3.1.



Gambar 3.2 *Tweet* yang dibagikan oleh akun Twitter @Radityadiiikan

3.2.2. Proses Pengambilan Data

Proses pengambilan data menggunakan *Search API* dari *Twitter*. *Search API Twitter* memiliki batasan 180 *request* setiap 15 menit, satu *request* dapat mengambil maksimal 100 *tweet*.

3.2.3. Penyimpanan Data

Data *tweet* yang telah diambil disimpan dalam tabel database SQL Server 2008. Tujuan dari penyimpanan dalam

database adalah untuk mempermudah mengakses data, mempercepat proses perhitungan dan menyimpan data lebih terstruktur. *Tweet* beserta informasi yang terkait akan disimpan pada tabel *Twitter.* Struktur tabel Twitter dapat dilihat dapat dilihat pada Tabel 3.1

Tabel 3.1 Struktur Tabel Database Penyimpanan *Tweet*

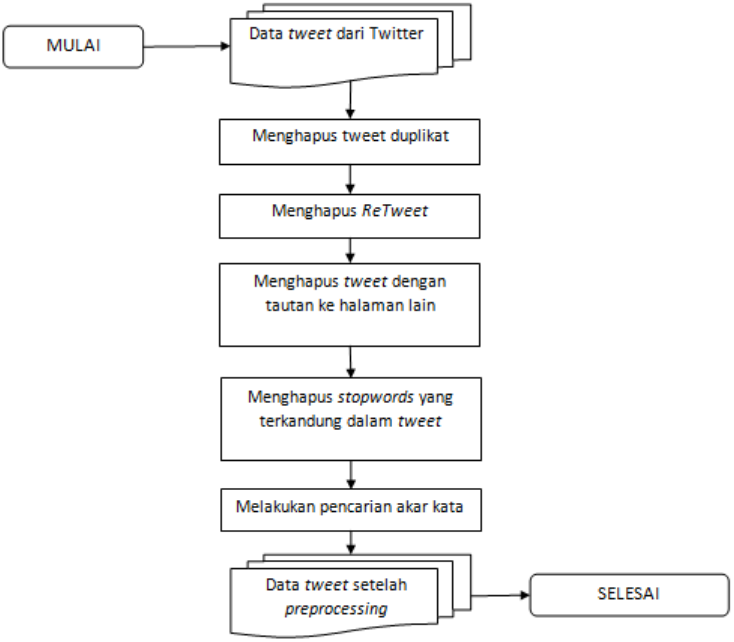
Atribut	Tipe Data	Keterangan
DateCreated	Datetime	Menunjukkan kapan *tweet* diciptakan
IdUser	Varchar(25)	Menunjukkan id pengguna *Twitter*
ScreenName	Varchar(25)	Menunjukkan nama @ pengguna, bersifat unik
Name	Varchar(30)	Menunjukkan nama pengguna, tidak bersifat unik
Text	Varchar(300)	Menunjukkan *Tweet* yang dikirim oleh pengguna
GeoLocationX	Float	Menunjukkan koordinat lintang pengguna apabila mengaktifkan lokasi
GeoLocationY	Float	Menunjukkan koordinat bujur pengguna apabila mengaktifkan lokasi
Class	Varchar(15)	Menunjukkan kelas dari tweet,
IdTweet	Int	Menunjukkan id dari *tweet*
Classification	Varchar(15)	Menunjukkan apakah data merupakan data *training* atau *testing*
CleanedText	Varchar(300)	Menunjukkan data *tweet* setelah *preprocessing*

3.3. Perancangan Proses

Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan untuk memberikan gambaran secara rinci pada setiap alur implementasi metode pada aplikasi deteksi emosi manusi. Alur tersebut nantinya akan digunakan dalam tahap implementasi.

3.3.1. Tahap Preprocessing

Di dalam tahap *preprocessing* terdapat beberapa proses antara lain menghapus *tweet* yang mengandung tautan ke halaman lain, menghapus *tweet* duplikat, menghapus *tweet* yang merupakan *ReTweet,* menghapus *stopwords* yang terdapat dalam *tweet* dan mencari akar kata (*stemming*) dari kata yang tersisa. Diagram alir mengenai tahap ini dapat dilihat pada Gambar 3.2.



Gambar 3.3 Diagram alir tahap *preprocessing*

3.3.1.1. Proses Penghapusan *Tweet* Duplikat

Penghapusan *tweet* duplikat adalah proses pertama dalam tahap *preprocessing*. Data *tweet* yang didapat dari Twitter dikelompokkan berdasarkan citra teks *tweet* masing-masing. *Tweet* pertama dari masing-masing kelompok akan diambil sementara sisanya akan dihapus karena dianggap sebagai duplikat. Contoh tweet yang memiliki duplikat beserta jumlahnya dapat dilihat pada Gambar 3.4.

	Tweet	Jumlah Duplikat
1	RT @GMEAgency: Sebastian Mikael - As Low As Me https://t.co...	1005
2	RT @gmailoilo: Kanta pa more @aldenrichards02 #KapusoFansD...	351
3	RT @epuleusoff: Fikiran wanita vs lelaki.. https://t.co/lneYALRagO	308
4	RT @SMACKHighTX: Siri a freak ???? https://t.co/UUhDhLZeq3	246
5	Orang yang tak bersyukur adalah orang lemah dan belum mampu ...	223
6	Janganlah berkubang di dalam penyesalanmu, tapi gunakanlah ia ...	222
7	Suka? pujilah Dia. Sulit? carilah Dia. Senang? sembahlah Dia. Se...	222
8	Jangan berharap segalanya menjadi lebih mudah, tetapi berharapl...	222
9	RT @IzzatiTaufek: Bagi yang selalu/suka ambik gambar airline bo...	189
10	RT @HatiAkuEgo: My mom kata, "Vape kat China dah kene reje...	179

Gambar 3.4 Contoh Tweet yang Memiliki Duplikat Beserta Jumlah Duplikatnya.

3.3.1.2. Proses Penghapusan *Retweet*

Retweet merupakan *tweet* suatu pengguna yang dibagikan oleh pengguna lain, karena isi *tweet* yang sama maka *Retweet* akan dihapus. *Retweet* dibagi menjadi dua yaitu *Native Retweet* dan manual *Retweet*. *Native Retweet* akan membagikan *tweet* dalam bentuk tautan ke *tweet* asli, *Native Retweet* tidak dihitung sebagai *tweet* baru sehingga tidak akan muncul sebagai *tweet* duplikat. Manual *Retweet* membagikan tweet pengguna lain dengan menyalin *tweet* dan menambahkan kata 'RT' didepan tweet salinan tersebut. Manual *Retweet* akan dihapus karena dianggap sebagai duplikat dari tweet lain. Manual *retweet* dapat ditemukan dengan cara mencari *tweet* dengan awalan RT pada data *tweet*. Contoh tweet yang merupakan manual retweet dapat dilihat pada Gambar 3.5.

ScreenName	text
Luqmannazri97	RT @amar_aizat: Jangan hujan malam ni udah lahh
stksmwhyni	RT @IcanCahyani: Andai aku bisa
Aisyilaist	RT @Bayangno: Fix kowe parasit ?
yeojas	RT @Sir_KTH: @_yeojas_ /bunuh pake cinta/????
Maytica_	RT @Zenghelis: PP - PSOE - C's : CIS tema.
makyein	RT @makjuhyun: aku gpp nyai, seul, yein
ttzuyupreme	RT @krystpreme: banyak yang unver aku sedih
cahyoxe	RT @kjstal_: Yang on retweet.pen dipolow??
dimaasan	RT @rezaarchm: @dimaasan klpnto

Gambar 3.5 Contoh Tweet yang Tergolong Manual Retweet

Gambar 3.6 Contoh Native Retweet oleh @RiendaFauriza

Gambar 3.7 Contoh Manual Retweet oleh @IndahJanuarti25

3.3.1.3. Proses Penghapusan *Tweet* dengan Tautan

Tweet yang mengandung tautan dianggap sebagai *tweet* yang tidak mengandung emosi. Hal in dikarenakan *tweet* dikirimkan dengan tujuan agar pengguna lain membuka tautan yang terdapat pada *tweet* tersebut, bukan untuk menyampaikan pendapat yang terkait dengan emosi. Tautan pada tweet dapat ditemukan dengan format "http://" atau https://. Apabila ditemukan kata dengan format seperti diatas maka *tweet* akan dihapus dari

proses klasifikasi. Contoh *tweet* yang mengandung tautan dapat dilihat di Gambar 3.8

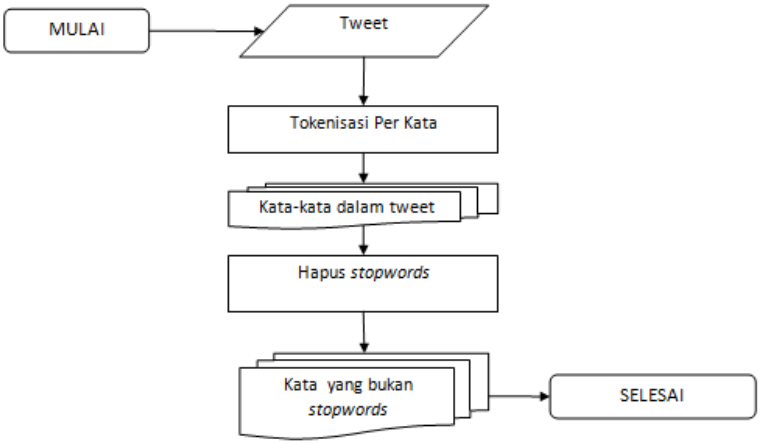
ScreenName	text
frady08	Perayaan Ultah Istri Mantan Menteri Era SBY Ini Dibiayai Negara https://t.co/2jluxiflg9
GustavoSaprut	#ESUNAFANO HTTPS://T.CO/3IKRAV9CQU
ifdvr	#dlvrit [Art] Yuka Ichijo https://t.co/kRmoy3Y3SD #dlvrit
farapiqa	Jom isyak https://t.co/vclkIMckPA
madiunpos	KABAR DUKA : Misye Arsita Dimakamkan Malam Hari https://t.co/qHI3u3pzGQ
iAmsterdamNews	Ajax Amsterdam vs Fenerbahce Preview https://t.co/KQdWkwNXe2 #Amsterdam #News
buekyeon	Siapa tuhh https://t.co/xmmMkFqX6S
frady08	Malaysia dan Jepang Masuki Pasar Terigu Indonesia https://t.co/l4QqorhCXn
cik_eyyza	boleh.. blh dlm mimpi ???? https://t.co/GenIJj1BJS
RingkasBerita	Perayaan Ultah Istri Mantan Menteri Era SBY Ini Dibiayai Negara https://t.co/TbHRmHJTSM [Rep...
Travelonesia	Perkara pencemaran Fadli Zon dilimpahkan ke pengadilan https://t.co/hdmRFmVTZt

Gambar 3.8 Contoh Teks *Tweet* yang Mengandung Tautan

3.3.1.4. Proses Penghapusan *Stopwords*

Stopwords merupakan kata-kata yang sering muncul sehingga dianggap tidak penting untuk mempercepat komputasi pada pemrosesan teks. Kosa kata yang termasuk dalam *stopwords* bahasa indonesia yang digunakan dalam tugas akhir ini adalah *stopwords* yang didefinisikan oleh Fadhillah Z Tala [16]. *Stopwords* tambahan ditambahkan karena terdapat kata tidak baku yang memiliki frekuensi tinggi tetapi tidak memiliki tingkat kepentingan yang tinggi pada suatu *tweet.* Kata-kata ini bisa merupakan kata disingkat, kata yang sering digunakan sehari-hari namun tidak sesuai dengan ejaan yang disempurnakan. *Stopwords* bahasa Inggris juga ditambahkan dikarenakan penggunaan beberapa bahasa dalam satu *tweet* walaupun *tweet* sudah dideteksi sebagai *tweet* dengan bahasa Indonesia. Daftar lengkap kata yang termasuk dalam *stopwords* akan terlampir pada lampiran A.1. Proses penghapusan *stopwords* dilakukan dengan melakukan tokenisasi per kata, apabila kata merupakan *stopwords* maka kata akan dihapus. Diagram alir proses penghapusan stopwords dapat dilihat pada Gambar 3.6. Sedangkan contoh *tweet* setelah dan sebelum dilakukan penghapusan stopwords ada pada Tabel 3.2. Beberapa kata yang termasuk *stopwords* bahasa Indonesia yang

didefinisikan oleh Fadhilah Z Tala dapat dilihat pada [16], untuk stopwords lengkap dapat dilihat pada lampiran.



Gambar 3.9 Diagram Alir Proses Penghapusan Stopwords

Tabel 3.2 Contoh *Tweet* Setelah Stopwords Dihapus

Tweet	Setelah Penghapusan Stopword
Pertahankan orang yg mempertahankanmu. Banggakan lah dia yg membanggakanmu. Dan lepaskan lah dia yg tak pernah menghargaimu. :)	pertahankan mempertahankanmu banggakan membanggakanmu lepaskan menghargaimu :)
Kalau kau merasakan cinta kasih, perasaan itu sudah tentu diberkati Gusti Allah. Bersukacitalah karena hatimu masih mampu merasa :)	merasakan cinta kasih perasaan diberkati gusti allah bersukacitalah hatimu merasa :)
Jangan membalas mereka yg membencimu. Tersenyum dan berbahagialah di depan mereka, tak ada yg lebih menyakiti mereka daripada itu :)	membalas membencimu tersenyum berbahagialah menyakiti :)
Orang yg kamu sayang, belum tentu selalu ada untuk kamu. Orang yang sayang sama kamu, udah pasti dia selalu ada untuk kamu! :)	sayang sayang :)

Tabel 3.3 Contoh Stopwords yang Digunakan

apa	siapa	ada
karena	anda	dia
oleh	untuk	tiap
selain	seperti	se
selalu	agar	harus

3.3.1.5. Proses Stemming

Stemming merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (root word) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan distem ke root wordnya yaitu "sama". Proses *stemming* pada teks berBahasa Indonesia berbeda dengan stemming pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks dan prefiks dihilangkan.

Algortima *stemming* yang digunakan dalam tugas akhir ini adalah algoritma *stemming* teks bahasa Indonesia yang diusulkan oleh Nazief & Adriani *[18]*. Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah root word. Maka algoritma berhenti.
2. *Inflection Suffixes*("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa *particles* ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns*("-ku", "-mu", atau "-nya"), jika ada.
3. Hapus *Derivation Suffixes*("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a

   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
   b. Tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word.
6. Selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
3. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.

Contoh *Tweet* setelah dilakukan stemming dapat dilihat pada Tabel 3.4

Tabel 3.4 Contoh *Tweet* Setelah Dilakukan Stemming

Tweet	Setelah *Stemming*
pertahankan mempertahankanmu banggakan membanggakanmu lepaskan menghargaimu :)	tahan tahan bangga bangga lepas harga :)
rasanya memperjuangkan rasanya mempertahankan mudah melepaskan :)	rasa juang rasa tahan mudah lepas :)

3.3.2. Tahap Pelabelan Otomatis

Di dalam tahap ini dilakukan proses pemberian label secara otomatis berdasarkan kemunculan penanda *hashtag* dan *emoticons*. Metode ini digunakan untuk menghindari pemberian label kelas untuk dataset yang sangat besar. Kata kunci yang digunakan sebagai penanda untuk tiap kelas telah didefinisikan sebelumnya, apabila ditemukan penanda pada satu *tweet*, maka *tweet* tersebut akan diberi label kelas berdasarkan penanda. Penggunaan pelabelan otomatis (*distant supervision*) pada analisis sentimen yang dilakukan oleh Alec Go menggunakan emoticon, apabila terdapat emoticon :) maka tweet mengandung sentimen positif dan sebaliknya apabila terdapat emoticon :(maka tweet mengandung sentimen negatif. Hal ini dilakukan pada training set, pada testing set *emoticon* tidak lagi diperlukan [3]. Pada tugas akhir ini akan digunakan kata kunci untuk memberikan label kelas secara otomatis kepada data yang berjumlah besar. Evaluasi dari pemberian label secara otomatis akan dilakukan pada tahap pengujian akurasi klasifikasi.

Kamus kata kunci *hashtag* pemetaan emosi yang digunakan merujuk kepada penelitian yang dilakukan oleh Matthew Purver [7] dan penelitian semantik kepada kamus emosi kata bahasa Indonesia dengan kelas emosi yang didefinisikan oleh Ekman [13]. Apabila penanda tidak ditemukan maka data akan dilabeli sebagai kelas netral. Apabila terdapat beberapa penanda dari kelas yang berbeda maka data akan dilabeli sesuai dengan kelas dengan anggota paling sedikit.

Tabel pemetaan emosi dengan keyword yang menggunakan *hashtag* dapat dilihat di Tabel 3.5

Tabel 3.5 Pemetaan Emosi dengan *Hashtag*

Senang	#senang# Girang #Gembira #Bahagia #Riang #Puas #Sayang #Geli #Cinta
Sedih	#sedih #Pilu #Sesal #Putus asa #Sedih #Murung #Haru #Duka #Rindu
Marah	#marah#Kesal #Murka #Dongkol #Gemas #Dengki #Sebal #Benci #Curiga #Suntuk #Bosan #Marah #Cemburu #Jengkel #Kecewa
Takut	#takut #was-was #ngeri #gugup #ragu #takut #gentar #khawatir #ciut
Terkejut	#terkejut #henyak #heran
Jijik	#jijik #najis #risih #muak #antipati

Tabel 3.6 Pemetaan Emosi dengan *Emoticon*

Senang	:-) :) ;) ;-) :D :p 8) 8-| <@o
Sedih	:-(:(;-(;(:-< :< :'(
Marah	:-@ :@
Takut	:|
Terkejut	:s :S
Jijik	:$ +0(

3.3.3. Tahap Klasifikasi Naive Bayes

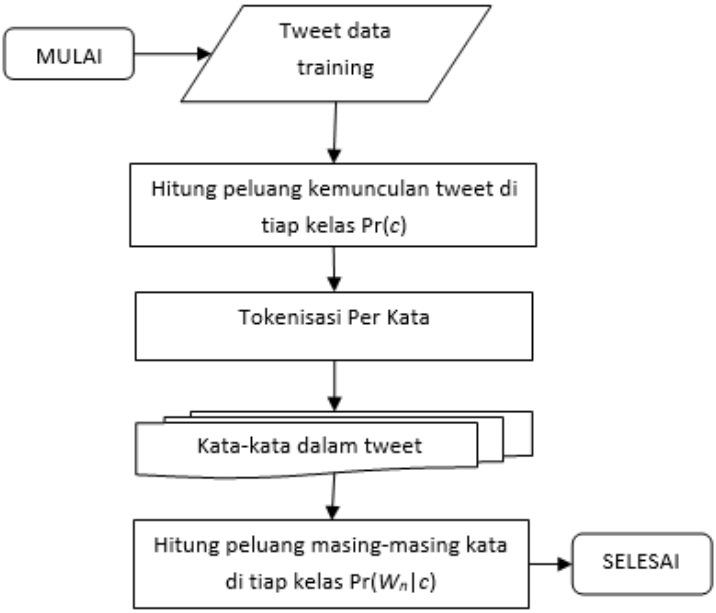
Setelah tahap pelabelan, selanjutnya dilakukan klasifikasi Naive Bayes untuk mendapatkan model yang dapat melakukan prediksi kelas terhadap *tweet* baru. Tahap klasifikasi ini dibagi menjadi dua yaitu *training* dan *testing*. Tahap *training* menggunakan data *training* untuk membangun model klasifikasi.

Tahap *testing* menggunakan data *testing* untuk menguji model klasifikasi yang sudah dibangun pada tahap *training*.

3.3.4. Proses *Training* Klasifikasi Naive Bayes

Proses *training* digunakan untuk membangun model klasifikasi. Proses *training* pada klasifikasi Naive Bayes dimulai dengan menghitung peluang kemunculan *tweet* berdasarkan kelasnya. Peluang ini dihitung dengan membagi jumlah data yang tergolong pada suatu kelas dibagi dengan total data Pr(c). Kemudian hitung peluang masing-masing kata terhadap suatu kelas. Kedua peluang ini akan digunakan pada tahap *testing* untuk memberikan label pada *tweet* yang belum diketahui kelasnya. Apabila terdapat suatu kata yang ada di satu ada namun ada di kelas lain dilakukan *smoothing* dengan menambahkan jumlah kata menjadi satu untuk menghindari probabilitas suatu kata kosong di satu kelas namun ada di kelas lain.

Data probabilitas kata pada masing-masing kelas akan disimpan pada database untuk mengklasifikasikan *tweet* dengan memberi label *tweet* proses testing. Semakin sering suatu kata terdapat pada suatu kelas maka probabilitas dari kata tersebut pada kelas yang bersangkutan akan semakin tinggi sehingga meningkatkan peluang suatu *tweet* yang meningkatkan peluang suatu tweet tergolong pada kelas tertentu pada proses klasifikasi. Diagram alir proses ini dapat dilihat pada Gambar 3.10

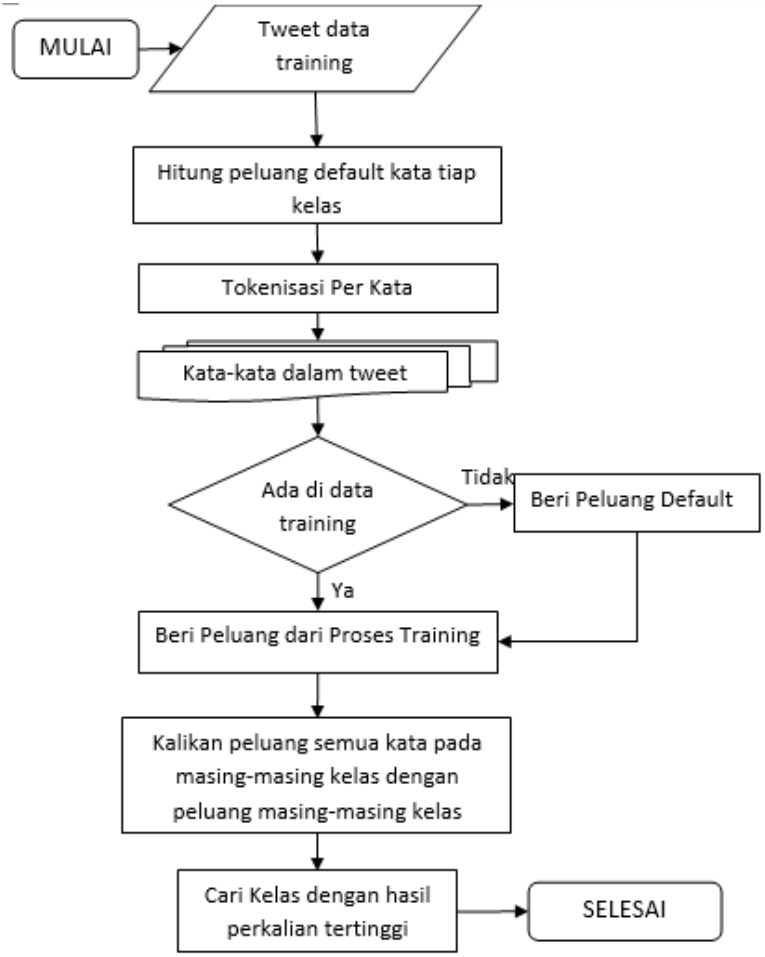


Gambar 3.10 Diagram Alir Proses *Training* Klasifikasi Naive Bayes

3.3.5. Proses *Testing* Klasifikasi Naive Bayes

Proses *testing* memberikan label kelas pada *tweet* yang belum memiliki label kelas dengan tujuan untuk menghitung performa model klasifikasi yang sudah dibangun pada proses *training.* Proses menggunakan perhitungan peluang yang sudah didapatkan dari proses *training.* Pertama *tweet* yang belum diketahui label kelasnya dipecah per kata, lalu masing-masing kata tersebut dicari keberadaannya di data *training*, apabila kata terdapat pada data *training* maka ambil peluang kata tersebut dari data *training,* apabila kata tidak ditemukan maka gunakan peluang *default* masing-masing kelas. Semua peluang kata dalam satu *tweet* dikalikan berdasarkan kelasnya, lalu hasil perkaliannya dikalikan peluang masing-masing kelas. Kelas dengan hasil perkalian

tertinggi akan digunakan sebagai label kelas untuk *tweet* tersebut. Diagram alir proses *testing* klasifikasi Naive Bayes dapat dilihat pada Gambar 3.8



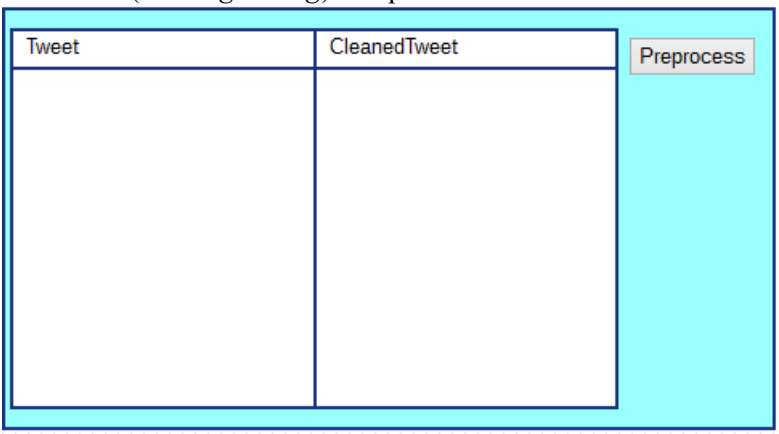
Gambar 3.11 Diagram Alir Proses *Training* Klasifikasi Naive Bayes

3.4. Perancangan Antarmuka Perangkat Lunak

Pada subbab ini akan dibahas mengenai perancangan antarmuka perangkat lunak yang bertujuan untuk dapat mempermudah interaksi antara perangkat lunak dengan pengguna. Sistem ini memiliki beberapa halaman preprocessing, hitung peluang kata (*training*), prediksi tweet yang pada tahap pengujian *(testing)*

3.4.1. Halaman Preprocessing

Halaman ini merupakan halaman yang pertama kali muncul pada sistem karena digunakan untuk mendapatkan *template* mata yang nantinya akan diproses di halaman-halaman berikutnya. Pada halaman ini, pengguna melihat data *tweet,* data *tweet* yang sudah dilakukan *preprocessing,* kelas, pembagian klasifikasi (*training/testing*) dan prediksi kelas

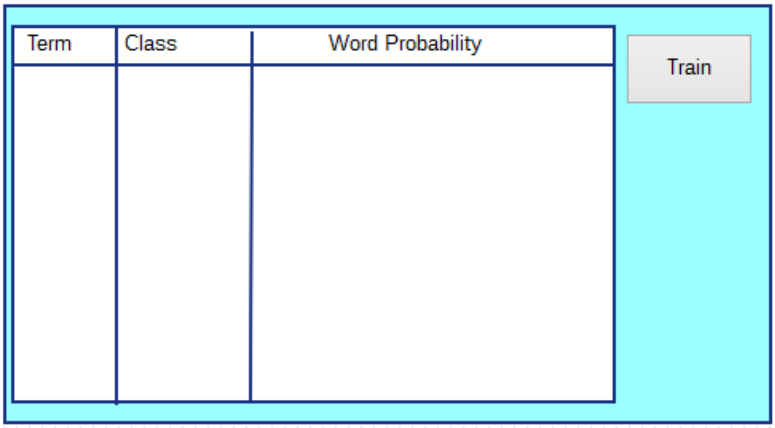


Gambar 3.12 Rancangan Halaman preprocessing

3.4.2. Halaman Training

Halaman ini adalah halaman yang digunakan untuk melakukan perhitungan peluang. Pada halaman ini, data peluang akan ditampilkan dan terdapat tombol untuk melakukan perhitungan peluang. Data yang ditampilkan adalah kata yang ada

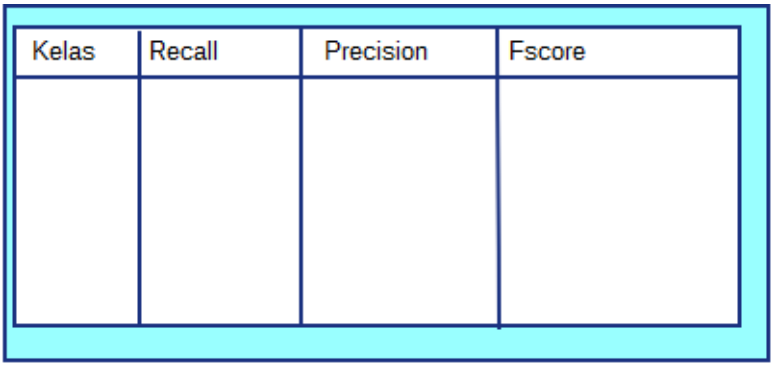
dalam *tweet*, kelas dari kata tersebut dan peluang kemunculan kata tersebut dalam suatu kelas.



Gambar 3.13 Rancangan Halaman Training Data

3.4.3. Halaman Evaluasi

Halaman ini adalah halaman yang digunakan untuk melihat hasil dari klasifikasi. Pada halaman ini, data *recall, precision,* akan ditampilkan dan terdapat tombol untuk melakukan perhitungan peluang.



Gambar 3.14 Rancangan Halaman Evaluasi

BAB IV
IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman Java dengan database Microsoft SQL Server 2008 R2 untuk menyimpan data *tweet*.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada **Error! eference source not found.**.

Tabel 4.1 Lingkungan Implementasi Sistem

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i3-4150 CPU @ 3.50GHz (4 CPUs) , ~3.5GHz Memori: 4096 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows Embedded 8.1 Industry Pro 64-bit Perangkat Pengembang: IDE NetBeans7.3 Microsoft SQL Server 2008 R2 Perangkat Pembantu: Notepad++, Microsoft Excel 2013, Microsoft Word 2013
Library	*Twitter4j*

4.2. Implementasi Pengambilan Data

Pengambilan data dilakukan dengan menggunakan bahasa pemrograman java dan library *Twitter4j*. Pengambilan data dilaksanakan selama dua minggu dan menghasilkan sekitar 800.000 *tweet.* Berikut adalah *pseduocode* program pengambilan *tweet*. Untuk kode dalam bahasa Java dapat dilihat pada lampiran.

```
FUNCTION tweetCrawler()
    INITIALIZE twitterConnection as tweetConnection
    GET requestLimit from tweetConnection
    SET query and filter TO query
    WHILE true
       SET request = 0
       WHILE request < requestLimit-1
          SET tweet data TO Database
          SET request+=1
        END WHILE
       DO Wait 15 minutes
    END WHILE
END FUNCTION
```

Gambar 4.1 Pseudocode Pengambilan Tweet ke Database

4.2.1. Analisis Pengambilan Data

Proses pengambilan data menggunakan *Search API* dari *Twitter*. *Search API Twitter* memiliki batasan 180 *request* setiap 15 menit, satu *request* dapat mengambil maksimal 100 *tweet*.

API Twitter memungkinkan pengguna untuk melakukan mendapatkan *tweet* dengan filter bahasa. Dalam tugas akhir ini digunakan filter bahasa Indonesia, walaupun telah menggunakan filter Bahasa Indonesia terdapat beberapa tweet dengan bahasa selain Bahasa Indonesia. Bahasa tersebut antara lain *tweet* dengan bahasa asing seperti Bahasa Melayu, Bahasa India dan bahasa daerah seperti Bahasa Jawa dan Bahasa Sunda. *Tweet* dengan bahasa asing ini dapat mengurangi performa klasifikasi karena kosa kata yang digunakan dengan Bahasa Indonesia berbeda.

Contoh *tweet* dengan bahasa asing dapat dilihat pada Gambar 4.2, Gambar 4.3

	TEXT
1	@melodyJKT48 yahh tehh .. Abdi teu tiasa ka banjarmasin na .. Keur di gawe.. Mun arek mh ka kab. Tabalong na.. Tat
2	Napoli 5-0 Dortmund 4-0 Pool 1-0 aja udh seneng abdi mah~ da abdi mah apa atuh
3	RT @GalauKeunnn: Muhun da abdi mah sakieu ayana , bade ingkah ? Mangga moal di ulah ulah :) #GLWKN
4	Mustaf Abdi Noor, Aka Shafana, #Somalia https://t.co/47tqfbl05l
5	MAJALIS KHAMSA-E-AZA 21-TILL> 25-MUHARRAM 7:30PM DAILY-NIGHT SPK> ALLAMA ZUHAIR-ABBAS ABDI A
6	Pitnah etamah mang, abdi mah teu ngiringan @olegunnnn https://t.co/q1QT9uC57m
7	Salah deui ?? kudu kumaha atuh abdi teh gusti.
8	ya abdi mah naon atuh :')
9	RT @banyolanhidup: nepang keun atuh, abdi yayat. WAAAAAAAWWWWWWW, seluruh warga kaget bukan main.
10	RT @GalauKeunnn: Muhun da abdi mah sakieu ayana , bade ingkah ? Mangga moal di ulah ulah :) #GLWKN

Gambar 4.2 Contoh *Tweet* yang Menggunakan Bahasa Sunda

TEXT
RT @urassa23: Wenye CCM wamekula menu hii Leo...We unakaa kutusumbua hapa kazi tu huku unashindia Utumbo na Vichw
maze RT @AmMstephano: Manze imeamua kunonesha mhindi "Erimniqque_: Europa league si mchezo"
@sardanarohit ka twitter account block KR diya gya...ye baat kuchh hazam nhi ho rahi hai @ZeeNewsHindi @sudhirchaudhary
RT @urassa23: Wenye CCM wamekula menu hii Leo...We unakaa kutusumbua hapa kazi tu huku unashindia Utumbo na Vichw
Wenye CCM wamekula menu hii Leo...We unakaa kutusumbua hapa kazi tu huku unashindia Utumbo na Vichwa vya kuku. http
Manje ngoba ngilala nge skhindi nje https://t.co/yBgQoy6EQp
Our comedians uenda USA kuchoma mahindi ama, Sasa utaenda kuambia wazungu jokes za tribes waelewe nini
@Slash, Meegan, @franksidoris @brentfitz in India for @MtviXtreme #RNFNR #worldonfire #SlashIndia #slash #smkc https://t.c
JADINE YAN UY JADINE YAAAAAN!!! hindi na yan Cleah!!!! :)))))) #OTWOLLoveShots
@ilahyapcastro luhbaliwhindizone

Gambar 4.3 Contoh *Tweet* yang Menggunakan Bahasa India

	TEXT
1	RT @_RasahCangkeman: Nek misale mboh sesok kapan koe ora bahagia mbek pilihanmu, noleho mburi,
2	Ash mboh karep mu wae ?? ncen nyesel ki teko ne keri
3	Hahaha...kudune lek ngerti aturan yo gk sampai drama adu pinalti sam. Mboh lek ngeyel njaluk menang :) h
4	@aqylah___ hahah ye qylah. Dio tamboh tolok jah cumo keno pehe ayat dio hk belit supo ular sawo tu
5	"Akan ada bangkit setelah jatuh" - emboh iku mari dijungkrakno koncomu, opo njlungup dewe perkoro... htt
6	@ShauqiAlief omboh haha bukan selalu kan hahaha
7	@M_DesyJKT48 mboh ndes
8	@rasyakura mboh yo
9	Berada di tahap emboh hahaha
10	Emboh mumet tok!!

Gambar 4.4 Contoh Tweet yang Menggunakan Bahasa Jawa

TEXT
@eykadyy hahaha.ada haaa kj jom ar pd.aku balik kj 10 hb nanti
RT @FandaZulkamain: "Hai, jom couple?" "Pahala aku tak cukup lagi nak masuk syurga. Kau
So, tunggu apa lagi? Jom lah jadi pengguna @myaltel! #IniBaruReal https://t.co/zwW0Q8EZrY
Jom isyak https://t.co/vclkIMckPA
RT @MukminOmar: @OhMyTranung jom admin https://t.co/Dwn74nUUmw
Ang pakai minyak wangi apa ni, busuk ngat aih.. smpai pening kepala mak.. Sat lgi jom bli yg lain.
RT @FandaZulkamain: "Hai, jom couple?" "Pahala aku tak cukup lagi nak masuk syurga. Kau
RT @SerojaUITM: Sapa free mlm ni angkat tangan ?? Jom join?? FORUM PERDANA EHWAL I
Salam abg @faizdickie jom kawin ? Mesti hari hari hidup saya funny tengok abg buat lawak.
RT @ameyyken: First semester "Weh jom holiday ramai2 satu kelas" "Korang okay aku okay je"

Gambar 4.5 Contoh Tweet yang Menggunakan Bahasa Melayu

4.3. Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.

4.3.1. Implementasi Tahap Preprocessing

Subbab ini membahas implementasi tahap preprocessing. Implementasi tahap ini menggunakan *Structured Query Language* (SQL) dan bahasa pemorgraman Java. Proses awal dari tahap ini adalah dengan menghapus tweet yang memiliki duplikat, menghapus tweet yang mengandung link di dalamnya, menghapus tweet yang merupakan sebuah retweet, menghilangkan stopwords yang terkandung dalam tweet dan mencari akar kata di setiap tweet dengan melakukan *stemming*. Query untuk melakukan penghapusan data duplikat dapat dilihat pada Gambar 4.6

```
1  update dbo.Twitter
2  set isNoise = 1
3  from dbo.Twitter
4  LEFTOUTERJOIN(
5     SELECT MIN(idTweet)as RowId,Text
6     FROM dbo.Twitter
7     GROUP BY Text
```

```
)as KeepRows ON
   dbo.Twitter.IdTweet = KeepRows.RowId
where isNoise=0 and KeepRows.RowId ISNULL
```

Gambar 4.6 Query untuk Menghapus Tweet Duplikat

Setelah *tweet* duplikat dihapus, dilakukan proses penghapusan *Retweet*. *Retweet* memiliki ciri-ciri frase "RT" diawal kalimat. Query untuk menghapus *retweet* dapat dilihat pada Gambar 4.7

```
update dbo.Twitter set isNoise=1 where TEXT
Like 'RT%' and isNoise = 0
```

Gambar 4.7 Query Proses Penghapusan *Retweet*

Setelah *Retweet* dihapus, dilakuan proses penghapusan *tweet* yang mengandung tautan ke halaman lain. *Tweet* yang mengandung tautan memiliki kata "http://" atau "https://". Query untuk menghapus *tweet* yang mengandung tautan dapat dilihat Gambar 4.8

```
update dbo.Twitter set isNoise=1 whereText
like'%http%'
```

Gambar 4.8 *Query* proses penghapusan tweet yang mengandung link

Proses selanjutnya adalah melakukan penghapusan *stopwords* pada *tweet* yang tersisa. *Tweet* dibagi menjadi kata-kata penyusunnya kemudian diperiksa apakah kata tersebut termasuk dalam *stopwords*. Apabila kata termasuk dalam *stopwords* maka kata tersebut akan dihapus. Kata yang tidak termasuk ddalam *stopwords* akan dilakukan *stemming* lalu kata akan dimasukkan ke dalam database tabel Term. Pseudocode untuk menghapus *stopwords* dapat dilihat pada Gambar 4.9.

```
FUNCTION removeStopwords()
    GET stopwords list THEN SET TO wordsList
    SET list of tweet TO tweetList
    FOR tweet in tweetList
```

```
5          FOR words IN tweet
6             FOR stopwords IN wordsList
7               IF stopwords==words THEN
8                 DELETE words
9                 CONTINUE
10              END IF
11            END FOR
12            DO Stemming(words)
13         END FOR
14      END FOR
15  END FUNCTION
```

Gambar 4.9 *Pseudocode* penghapusan stopwords dan stemming

Dari proses penghapusan *stopwords*, ditemukan bahwa terdapat banyak kata yang tergolong dalam *stopwords* namun karena bentuk dan pengejaannya tidak baku. *Stopwords* memiliki karakteristik memiliki jumlah kemunculan pada dokumen yang tinggi. Setelah melakukan analisis dibentuklah *stopwords* untuk kata tidak baku. Contoh stopwords yang merupakan singkatan dapat dilihat di Tabel 4.2

Tabel 4.2 Contoh *Stopwords* tidak Baku

yg	nya	haha	gak
ya	la	gk	gue
nak	tau	gak	eh
kau	iya	jd	jgn
aja	ku	deh	udh
dah	dgn	udah	kal
jadi	banget	kalo	gitu
ga	lg	gua	ha
orang	ko	lu	

Proses stemming dilakukan dengan tiga fungsi, fungsi *deleteDerivationPrefixes* untuk menghapus awalan, fungsi *deleteDerivationSuffixes* dan fungsi *deleteInflictionSuffixes*

Fungsi-fungsi tersebut dikombinasikan untuk mencari kata dasar dari suatu kata.

```
1   FUNCTION naziefAdrianiStemmer(String word)
2       GET words list THEN SET TO dictionary
3       IF word IN dictionary
4           RETURN word
5       word = deleteDerivationPrefixes(word)
6       word = deleteInflictionSuffixes(word)
7       word = deleteDerivationSuffixes(word)
8       IF word IN dictionary
9          RETURN word
10      ELSE
11         RETURN originalWord
12  END FUNCTION
```

Gambar 4.10 *Code* untuk Melakukan Stemming

Fungsi *deleteDerivationPrefixes* digunakan untuk menghapus awalan kata, fungsi *deleteDerivationSuffixes* digunakan untuk menghapus akhiran kata "-i","-an" dan "-kan". Fungsi *deleteInflectionSuffixes* digunakan untuk menghapus akhiran yang berupa kepemilikan atau akhiran yang berupa "-lah","-kah","-tah". Untuk kode lengkap dari proses ini dapat dilihat pada lampiran.

4.3.2. Implementasi Tahap Pelabelan Otomatis

Subbab in membahas implementasi tahap automatic labelling. Dalam tahap ini dari suatu tweet dicari peluang apakah terdapat penanda yang telah didefinisikan di Tabel 3.5 dan Tabel 3.6. Contoh query untuk proses ini ditunjukkan pada Gambar 4.11, Gambar 4.12, Gambar 4.13 dan Gambar 4.14

```
update twitter
set Class='Senang'
where isNoise =0 and (text like
'%#senang%'
or text like '%#girang%'
or text like '%#gembira%'
or text like '%#bahagia%'
```

```
or TEXT like '%#riang%'
or text like '%#puas%'
or text like '%#sayang%'
or text like '%#alhamdulillah%'
or text like '%#geli%'
or text like '%#cinta%'
or text like '%:-)%'
or text like '%:)%'
or text like '%:-D%'
or text like '%:D %'
or text like '%;)%'
or text like '%;)%'
or text like '%:p %'
or text like '%8)%'
or text like '%8-|%')
```

Gambar 4.11 *Query* proses Pemberian Label pada Kelas Senang

```
update dbo.Twitter
set Class='Takut'
where isNoise=0 and (TEXT like
'%#takut%'
or Text like '%#menakutkan%'
or text like '%#cemas%'
or text like '%#gugup%'
or text like '%#waswas%'
or text like '%#ngeri%'
or text like '%#seram%'
or text like '%#ragu%'
or TEXT like '%#takut%'
or TEXT like '%#gentar%'
or TEXT like '%#khawatir%'
or TEXT like '%#ciut%'
or TEXT like '%#malu%'
or TEXT like '%#segan%'
or text like '%:|%')or text like '%;(%')
```

Gambar 4.12 *Query* proses Pemberian Label pada Kelas Takut

```
update Twitter
set Class='Marah'
where isNoise=0 and (TEXT like '%#marah%'
or text like '%#kesal%'
or text like '%#murka%'
or text like '%#dongkol%'
or text like '%#gemas%'
or text like '%#dengki%'
or text like '%#sebal%'
or text like '%#benci%'
or text like '%#curiga%'
or text like '%#suntuk%'
or text like '%#bosan%'
or text like '%#cemburu%'
or text like '%#jengkel%'
or text like '%#kecewa%'
or text like '%:@ %'
or text like '%:-@%'
or text like '%x(%')
```

Gambar 4.13 *Query* proses Pemberian Label pada Kelas Marah

```
update dbo.Twitter
set Class='Terkejut'
where isNoise=0 and (TEXT like '%#takjub%'
or text like '%#cengang%'
or text like '%#tercengang%'
or text like '%#kejut%'
or text like '%#terkejut%'
or text like '%#tegun%'
or text like '%#tertegun%'
or text like '%#henyak%'
or text like '%#heran%'
or text like '%:-o%'
or text like '%:o %'
or TEXT like '%xO %'
or TEXT like '%x-o%')
```

Gambar 4.14 *Query* Proses Pemberian Label pada Kelas Terkejut

4.3.3. Implementasi Tahap Klasifikasi Naive Bayes

Subbab ini membahas implementasi klasifikasi *naive bayes,* tahap ini menggunakan database SQL Server 2005 untuk menyimpan data peluang, data daftar kata di tiap kelas dan data prediksi kelas. Query untuk menghitung peluang tiap kata dan memasukkanya ke tabel wordProbability dapat dilihat pada gambar 4.3

```
select g.term as term, g.class as class,
g.wordProbability as wordProbability into
WordProbability from(
      select distinct a.Term,b.Class,

case when b.class='Netral' then
(isnull(j,0)+1) /
(@netralWordCount+@vocabularySize)

when b.class='Senang' then (isnull(j,0)+1) /
(@senangWordCount + @vocabularySize)
when b.class='Sedih' then
(isnull(j,0)+1)/(@sedihWordCount+@vocabularySi
ze)
when b.class='Marah' then
(isnull(j,0)+1)/(@marahWordCount+@vocabularySi
ze)
when b.class='Terkejut' then
(isnull(j,0)+1)/(@terkejutWordCount+@vocabular
ySize)
when b.class='Jijik' then
(isnull(j,0)+1)/(@jijikWordCount+@vocabularySi
ze)
when b.class='Takut' then
(isnull(j,0)+1)/(@takutWordCount+@vocabularySi
ze) end as wordProbability
      FROM(
      SELECT DISTINCT (Term) from term,
dbo.Twitter where
```

```
term.IdTweet=dbo.Twitter.IdTweet and
Classification='Training') a
      cross join (
      SELECT DISTINCT class from class) b
      left join
      (select count(*) as 'j',Term, class from
dbo.Twitter left join Term on
term.IdTweet=Twitter.IdTweet where
Classification='Training' group by Term,class)
c on a.Term=c.Term and b.Class=c.class
   Group by a.Term,b.Class,j) g
```

Gambar 4.15 Query untuk *Training* Klasifikasi *Naive Bayes*

Setelah proses *training*, proses selanjutnya adalah proses *testing*. Testing melakukan perkalian dari peluang yang sudah dihitung pada tahap klasifikasi. Apabila data tidak ada di tahap klasifikasi maka kata tersebut tetap memiliki peluang tergantung pada kelasnya.

```
insert into #tempTable2
select
t1.idTweet,t1.class,exp(sum(log(ISNULL(w.wordP
robability,
case when t1.class='Netral' then
@netralDefaultWeight
when t1.class='Senang' then
@senangDefaultWeight
when t1.class='Sedih' then @sedihDefaultWeight
when t1.class='Marah' then @marahDefaultWeight
when t1.class='Jijik' then @jijikDefaultWeight
when t1.class='Terkejut' then
@terkejutDefaultWeight
when t1.class='Takut' then @takutDefaultWeight
end)))) *
case when t1.class='Netral' then
@netralProbability
when t1.class='Senang' then @senangProbability
when t1.class='Sedih' then @sedihProbability
```

```
when t1.class='Marah' then @marahProbability
when t1.class='Jijik' then @jijikProbability
when
t1.class='Terkejut'then@terkejutProbability
when t1.class='Takut' then @takutProbability
end as 'Hasil'
from (select a.idTweet,b.Term, c.Class
from dbo.Twitter a,dbo.Term b, dbo.Class c
where a.IdTweet=b.IdTweet and
a.Classification='Testing') t1
left join dbo.WordProbability w on
t1.Class=w.class and t1.Term=w.term
group by IdTweet,t1.class
```

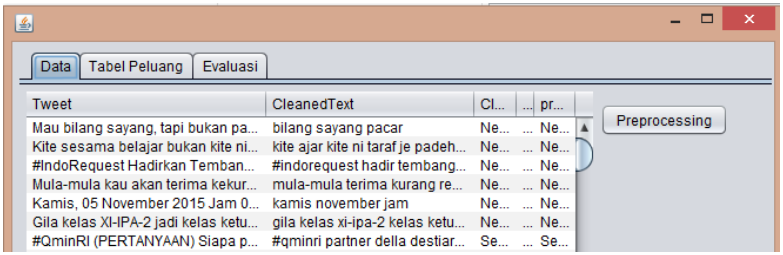
Gambar 4.16 Query untuk *Testing* Klasifikasi *Naive Bayes*

4.4. Implementasi Antar Muka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang sudah dijelaskan pada bab analisis dan perancangan.

4.4.1. Implementasi Halaman Preprocessing

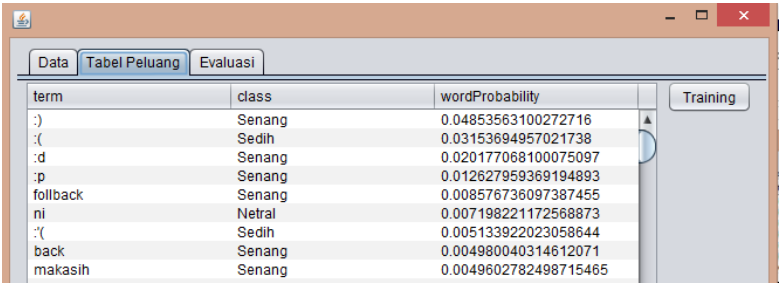
Halaman ini digunakan untuk melihat data setelah dan sebelum dilakukan preprocessing dengan tombol untuk melakukan preprocessing. Gambar untuk halaman ini dapat dilihat di Gambar 4.11



Gambar 4.17 Halaman Preprocessing

4.4.2. Impelentasi Halaman Training

Halaman ini digunakan untuk melihat data setelah dan sebelum dilakukan preprocessing dengan tombol untuk melakukan preprocessing. Gambar untuk halaman ini dapat dilihat di gambar 4.12



Gambar 4.18 Halaman Training Data

BAB V
PENGUJIAN DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari deteksi emosi manusia pada *tweet* bahasa Indonesia menggunakan klasifikasi Naive Bayes.

5.1. Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan tugas akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan untuk melakukan uji coba deteksi emosi manusia pada *tweet* berbahasa Indonesia. Lingkungan uji coba merupakan komputer dengan spesifikasi sebagai berikut:

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i3-4150 CPU @ 3.50GHz (4 CPUs) , ~3.5GHz Memori: 4096 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows Embedded 8.1 Industry Pro 64-bit Perangkat Pengembang: IDE NetBeans7.3 Microsoft SQL Server 2008 R2 Perangkat Pembantu: Notepad++, Microsoft Excel 2013, Microsoft Word 2013

5.2. Data Uji Coba

Data yang digunakan untuk uji coba implementasi sistem ini adalah *tweet* yang diambil menggunakan *search* API *Twitter* dengan batasan bahwa *tweet* berbahasa Indonesia. *Tweet* memiliki panjang maksimal 140 karakter. Jumlah data yang digunakan dalam uji coba bergantung pada jumlah data keseluruhan yang

digunakan dalam model dan skenario uji coba. Data memiliki satu label kelas.

Tabel 5.1 Contoh Data Masukan Uji Coba

No	Citra Masukan	Kelas
1	Aku memilih kamu bkn karna apa yg kamu miliki, bkn juga ketertarikan fisik. Aku memilih kamu karna aku nyaman dan aku bahagia :)<3	Senang
2	baru juga sejam udah kebangun dan gbsa tidur lagi dong :(	Sedih
3	Pencemaran lingkungan karena limbah industri gaduh, kenapa aturan bagi pelanggar pencemaran lingkungan tdk gaduh? #Heran..	Terkejut
4	Uda jijik ngelihat muka hina nya.... #Muak	Jijik
5	Pengen kuliah.. Pengen risen ... Pen pergi dari rumah #muak	Jijik
6	polisi tangerang pada sadis #ngeri	Takut
7	Lindungi dia dimana pun dia berada pada saat ini, tunjukkan yg salah itu salah yg benar itu benar #khawatir	Takut
8	Jika kau tidak mengenali, kenapa perlu menilai? Jika kau tidak mengenali, kenapa perlu mengandai?	Netral
9	jerawat dimanamana hissss :@	Marah
10	@baerhbie joyieee ndaa jahatt:((	Sedih

5.3. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Perhitungan hasil akurasi, *recall, precision* dan *fscore* pada data dengan distribusi tiap kelas yang belum diubah.
2. Perhitungan hasil akurasi, *recall, precision* dan *fscore* pada data dengan distribusi jumlah data tiap kelas yang lebih merata

5.4. Skenario Pengujian 1: Perhitungan Hasil Akurasi, Recall, Precision dan Fscore pada Data dengan Distribusi tiap Kelas yang Belum Diubah.

Pada skenario pengujian ini dilakukan pengambilan data selama tujuh hari sejak tanggal 28 Mei 2015 hingga 4 Juni 2015. Data yang didapatkan sebanyak 600.000 *tweet*. Setelah dilakukan proses penghapusan *tweet*, tersisa 299.458. *Tweet* dilakukan tokenisasi, dan stemming yang menghabiskan waktu hingga 12 jam. Data kemudian diberi label kelas dengan *automatic labelling.* Pembagian data *training* dengan *testing* adalah 80% data *training* dan 20% data *testing.* Jumlah masing-masing data pada tiap kelas dapat dilihat di Tabel 5.2.

Tabel 5.2 Pembagian Data Klasifikasi Skenario Pengujian 1

Kelas	Training	Testing	Jumlah
Netral	209.963	5.2491	262.454
Senang	16.568	4.142	20.710
Sedih	5.967	1.492	7.459
Terkejut	2.897	724	3.621
Takut	2.000	500	2.500
Marah	1.917	480	2.396
Jijik	254	64	318

Akurasi dihitung dengan jumlah data yang mampu diklasifikasikan dengan benar. Sedangkan *recall, precision* dan *fscore* dihitung berdasarkan confusion matrix. *Confusion matrix* untuk hasil skenario uji coba ini dabat dilihat pada Tabel 5.3. Tabel *recall, precision* dan *fscore* dapat dilihat pada Tabel 5.4.

Tabel 5.3 *Confusion Matrix* Skenario Pengujian 1.

Kelas Sebenarnya							Prediksi Kelas
Netral	Senang	Sedih	Marah	Jijik	Terkejut	Takut	
45.024	1.452	732	315	46	560	307	Netral
1.225	**2.093**	39	10	1	26	13	Senang
5.835	596	**719**	77	16	91	104	Sedih
6	0	0	**78**	0	0	0	Marah
197	0	0	0	**1**	3	0	Jijik
67	0	2	0	0	**42**	0	Terkejut
5	1	0	0	0	0	**76**	Takut

Tabel 5.4 Tabel *Precision,Recall* dan *Fscore* Skenario Pengujian 1

Kelas	**Recall**	**Precision**	**Fscore**
Netral	85,9%	92,9%	89,3%
Senang	50,5%	61,4%	55,4%
Sedih	48,1%	9,6%	16,1%
Marah	16,2%	92,8%	27,6%
Jijik	1,5%	0,5%	0,7%
Terkejut	5,8%	37,8%	10,0%
Takut	15,2%	92,6%	26,1%

Akurasi dari skenario pengujian ini mencapai 80% namun performa klasifikasi pada masing-masing kelas menunjukkan angka yang cukup rendah. Hal ini dikarenakan akurasi yang tinggi pada kelas netral yang memiliki porsi data yang jauh lebih tinggi dibandingkan kelas-kelas lainnya.

5.5. Skenario Pengujian 2: Perhitungan Hasil Akurasi, Recall, Precision dan Fscore pada Data dengan Komposisi Data yang Berbeda

Pada skenario pengujian ini dilakukan pengambilan data selama 3 hari sejak pada November 2015. Data yang didapatkan sebanyak 300,000 *tweet*. Setelah dilakukan proses penghapusan, data kemudian diberi label kelas dengan *automatic labelling*, dan dianalisa. Setelah proses analisis emosi pada *tweet*, diambil 20.810 *tweet* dengan pemberian label kelas yang diperketat dan jumlah data yang lebih berimbang. Komposi kelas dibuat lebih berimbang dengan tetap memperhatikan bahwa kelas netral memiliki jumlah paling banyak, kelas senang memiliki kelas lebih sedikit, sedih memiliki jumlah data yang lebih sedikit dari senang, marah terkejut dan takut memiliki jumlah data yang relatif dekat dan jijik memiliki jumlah data paling sedikit. *Tweet* dilakukan tokenisasi, dan stemming yang menghabiskan waktu 20 menit Pembagian data *training* dengan *testing* adalah 80% data *training* dan 20% data *testing*. Jumlah masing-masing data pada tiap kelas dapat dilihat di Tabel 5.5.

Tabel 5.5 Pembagian Data Klasifikasi Skenario Pengujian 2

Kelas	Training	Testing	Jumlah
Netral	8.000	2.000	10.000
Senang	5201	1300	6501
Sedih	2397	599	2996
Marah	392	98	490
Terkejut	327	82	409
Takut	279	70	349

Kelas	Training	Testing	Jumlah
Jijik	52	13	65

Akurasi dihitung dengan jumlah data yang mampu diklasifikasikan dengan benar. Akurasi dari skenario pengujian ini mencapai 72 persen. Sedangkan *recall, precision* dan *fscore* dihitung berdasarkan confusion matrix. *Confusion matrix* untuk hasil skenario uji coba ini dabat dilihat pada Tabel 5.6. Tabel *recall, precision* dan *fscore* dapat dilihat pada Tabel 5.7.

Tabel 5.6 *Confusion Matrix* Skenario Pengujian 2.

Kelas Sebenarnya							Kelas Prediksi
Netral	Senang	Sedih	Marah	Jijik	Terkejut	Takut	
1643	309	166	58	2	58	37	Netral
200	**962**	57	8	3	3	26	Senang
141	14	**369**	3	0	2	2	Sedih
5	3	0	**26**	6	1	0	Marah
0	0	0	0	**0**	0	0	Jijik
7	11	2	1	0	**16**	0	Terkejut
4	2	1	2	2	1	**12**	Takut

Tabel 5.7 Tabel *Precision, Recall* dan *Fscore* Skenario Pengujian 2

Kelas	Recall	Precision	Fscore
Netral	82,1%	72,3%	76,9%
Senang	73,9%	76,4%	75,1%
Sedih	62,0%	69,4%	65,5%
Marah	26,5%	63,4%	37,4%
Jijik	0,0%	0,0%	0,0%
Terkejut	19,7%	43,2%	27,1%

Kelas	Recall	Precision	Fscore
Takut	15,5%	50,0%	23,7%

Pada skenario pengujian 2, akurasi keseluruhan berkurang hingga 8 persen tetapi performa klasifikasi pada tiap kelas meningkat. Hal ini dapat dilihat dari *fscore* kelas senang, sedih, marah dan terkejut yang meningkat secara signifikan. Hal ini dapat dikarenakan komposisi data yang lebih berimbang dibandingkan skenario pengujian 1. Sampel kinerja model klasifikasi untuk melakukan prediksi kelas senang dapat dilihat pada Tabel 5.8 dan untuk prediksi kelas sedih pada tabel 5.9. Sampel menunjukkan kinerja yang baik pada *tweet* dengan ukuran teks yang panjang. Tabel 5.10 menunjukkan *tweet* yang panjang teksnya relatif pendek. *Tweet* yang panjang teksnya relatif pendek setelah dilakukan *preprocessing* menghasilkan kinerja yang buruk pada proses klasifikasi.

Tabel 5.8 Sampel Kinerja Prediksi pada Kelas Senang

No	Tweet	Kelas	Prediksi
1	Wes adus wes dandan wes siap2 lungo, bajigur bulno yo gur ning omh, ngarep bnget ono sing metok dolan ,jilaak :(	Sedih	Senang
2	Selamat Malam Generasi Muda Tulang Punggung Keropos Bangsa :) Bagaimana kabar V. Rossi, Marquez, Game COC & Hal2 gak penting Lainnya? :v	Senang	Senang
3	@jmmagsi48 @shahzadahmeds13 bardi akalmandi ki bat ki he apne::) dunno bar amal krke dekhaya lekin apke Imran Khan ne dunno bar dhoka dya	Senang	Senang
4	ngga sengaja liat foto kamu di salah satu account sosmed kamu, ituuuuu	Senang	Senang

No	Tweet	Kelas	Prediksi
	bikin baper lagii hihiii :D skrg tambah gendut yaa kamu?? :')		
5	Di Handphone banyakan foto PRILLY LATUCONSINA dari pada yg punya.ahaha. Jika memori full yaa foto gw yng gw hapus! gw mah gitu orangnya:-D	Senang	Senang
6	Mimin baik ko bktinya ngsh info trskan tntang ka ricky:) RT @KIRUNERSS: Mau bales2in mention ahh biar gk dibilang .... Aahhsyudahlahhhhhh ???	Senang	Senang
7	Assalamualaikum wr. wb. campus brainers yang budiman. #KHASANAHPETANG edisi #CEMAS kali ini ditemani Safia hingga nanti pukul 7 malam ya :)	Takut	Senang

Tabel 5.9 Sampel Kinerja Prediksi pada Sedih

No	Tweet	Kelas	Prediksi
1	adek kangen abang adek kangen abang adek kangen abang adek kangen abang :(adek kangen abang :(adek kangen abang :(adek kangen abang :(	Sedih	Sedih
2	arisia meni lila muka panto teh wan tanya dayat sambil ngelapan tembok :((, eh hampura anjis kieu da imah aing mah pinuh tantangan yat :(	Sedih	Sedih
3	wan wan jawab ath arisia wan ieu aing dayat buru buka panto imah, iwan pun menjawab "keula yat dagoan aing mersihan korong heula" :(	Sedih	Sedih
4	Dari zaman bahtera nabi nuh nepika ayeuna zaman cabe-cabean U19 merajalela, buuk si kak seto asa teu robah-robah anjis. Kayak cinta aku. :(	Sedih	Sedih

No	Tweet	Kelas	Prediksi
5	Keur bobogohan mh jjs jng jjm th kanu ninja euy,naha ai gs kawin jjs atawa jjm th kana delman? Hemm agul ku payung batur tea :((	Sedih	Sedih
6	Ternyata benar apa yg orang-orang katakan. Lieur kucinta tuh leuwih2 ti mabok arak anying :(. Sakitnya tuuhh.... Kela poho deui euy :((	Sedih	Sedih
7	#YEUHDANGUKEUNRP cik atuh ayaangg...aa teh kangen km nu dulu:(nisedih rek didangukeun pek, teu ge kajeun lah:(NUHUN	Sedih	Sedih
8	Geusss ayeuna mah engeeesss Kumaha bentuk bareungeut maraneh we rek robert, rek opick terseraahh :(#eh bhaaakkkss	Sedih	Sedih

Tabel 5.10 Sampel Kinerja Pada *Tweet* dengan Panjang Relatif Pendek

No	Tweet	Kelas	Prediksi
1	@naeyond @Tailurswift @svtjunh @I_Taehyung95 @rvltirene @I7JWWU @bearmillerr kamu juga :(	Sedih	Sedih
2	@Tailurswift @fkaccpard @svtjunh @I_Taehyung95 @rvltirene @I7JWWU @bearmillerr kesian :(	Sedih	Sedih
3	@soojvnx elap ingusnya sana nong wkwk	Netral	Sedih
4	@S_SiscaJKT48 cup cup sini aku temenin wkwk	Netral	Sedih
5	@anissazwaa seronoknyaa ishh	Netral	Sedih
6	#ON ygstan / yg rpers?	Netral	Sedih
7	Kgn u hm peka tlng	Netral	Sedih

5.6. Analisis Skenario Pengujian 1

Dari skenario uji coba, jumlah data mempengaruhi performa klasifikasi *naive bayes*, pada skenario pengujian satu didapatkan akurasi yang cukup tinggi sejumlah 80% namun *fscore* pada kelas selain kelas netral memiliki angka yang cukup rendah. Hal ini disebabkan jauhnya jumlah data yang dimiliki oleh kelas netral dibandingkan dengan kelas-kelas lainnya.

Dapat dilihat pada Tabel 5.3, terdapat 1.452 data kelas senang yang diprediksi sebagai kelas netral, jumlah tersebut sebesar 35% dari total data senang, hal ini dikarenakan jumlah data pada kelas netral yang tinggi sehingga apabila terdapat *tweet* yang tidak terdapat pada kosa kata data *training* maka kelas akan dimasukkan sebagai kelas netral. Pada Tabel 5.3 dapat dilihat bahwa terdapat 5.835 tweet kelas netral yang masuk kedalam kelas sedih, hal ini dikarenakan kosa kata yang termasuk dalam kelas sedih juga terdapat di dalam kosa kata yang termasuk dalam kelas netral namun dalam kata-kata tersebut tidak termasuk dalam penanda label kelas. Hal ini membuat *precision* dari kelas sedih terendah kedua setelah jijik. Contoh data kelas netral yang diklasifikasikan sebagai kelas sedih dapat dilihat pada Tabel 5.11. Dapat dilihat pada *tweet* nomor 3 terdapat *tweet* bahasa asing yang salah diklasifikasikan. Pada *tweet* nomor 1 terdapat kata "kenangan" yang mungkin pada kelas sedih memiliki bobot tinggi sehingga *tweet* diklasifikasikan sebagai kelas sedih. Pada *tweet* nomor 6, dan 8 juga terdapat kata-kata yang mungkin memiliki peluang besar pada kelas sedih.

Tabel 5.11 Data Kelas Netral yang Diklasifikasikan Sebagai Sedih

No	Tweet
1	Nyapu kenangan itu kayak nyapu sterofoam pasti ADALAGI.. ADALAGI! Yuk mending Ngalay bareng biar hidup lebih berwarna! :v cc: @indosatmania
2	Stss.. RT @darilandrean: Permenkecil: Atashi no Taiyou Jangan lupa sholjum." No mensyen bgt neh.."

No	Tweet
3	*Shab e Baraat*Mulla Ali Qari R.H Ny Farmaya;Mujhy Un logon Pe Taajjub Hota Hy Jo Hadees Ka Thora Boht Elm Rakhty Hen Mgr Phir ...>next
4	Tengah tido tetiba budak kecik ni masuk "akakkkk jomlah pergi buai laju laju lajuuu" ????????
5	@natashaasmam @twt_kecantikan pakai shampoo plus conditioner avon yg olive oil. Jgn lap rmbut pkai tuala just biar dia kering sndri.
6	Dulu sekolah rendah kawan kelas aku ada buka spa kat belakang kelas. Dorang ambil pensel colour dorang warnakan kening aku.

Kelas marah memiliki *recall* yang rendah yaitu 16,2% namun memiliki *precision* yang tinggi yaitu 92,8%. Dari 480 data *tweet* marah hanya 78 yang dapat diklasifikan dengan benar, namun model klasifikasi hanya mengklasifikan 6 data yang salah sebagai kelas marah. Hal ini dapat menandakan bahwa kosa kata yang berada pada data *training* kelas marah cukup unik dan tidak ada pada kelas lain sehingga model klasifikasi hanya mengklasifikasikan 84 data sebagai kelas marah. Tingkat *precision* pada kelas marah menjadi cukup tinggi, mencapai hingga 92,8%. Namun angka *recall* pada kelas marah sangat rendah karena terdapat 315 *tweet* kelas marah yang diklasifikasikan sebagai kelas netral. Hal ini dapat dikarenakan kata yang digunakan ada pada kosa kata kelas netral dan kosa kata kelas netral cukup besar dikarenakan jumlah data yang jauh lebih tinggi. Contoh data yang diklasifikasikan dengan benar pada kelas marah dapat dilihat pada Tabel 5.12

Tabel 5.12 Data Kelas Marah yang Diklasifikan dengan Benar

No	Tweet
1	Bawaanya emosi
2	@EX0HOON jgn marah huhu

3	Camiiii jgn lah marah aku juz pos apa yg org bagi jaaa ??
4	bodo benci
5	@Deviped hahhahhahha. Marah ap cin???
6	@parkjunghwaa marah aja yaa
7	Pembenci yang sukses adalahpembenci yang berhasilmembuatmu membencinya.
8	Pembenci akan membenci. Itu memang peran
9	Emosi tidak akan membimbingmu pada suatu pemikiran atau tindakan positif
10	@allkjm91: Cho ngambek ama gua? Etdah-_-aku juga !
11	Kecewa itu pasti mau marah tapi percuma...

Pada Tabel 5.13, *tweet* nomor 1 dan 7 terjadi kesalahan klasifikasi dikarenakan terdapat kata "cinta" yang mungkin memiliki peluang pada kelas senang lebih tinggi, *tweet* nomor 2 tidak terdapat spasi yang baik antar kata sehingga beberapa kata dihitung menjadi satu, tidak terdapat pada kosa kata dan akhirnya masuk sebagai kelas netral, tweet nomor 4 merupakan bahasa asing dan kata "bencinyaa" memiliki akhiran tidak baku sehingga gagal dilakukan *stemming*.

Tabel 5.13 Data Kelas Marah yang Dilasifikasikan dengan Salah

No	Tweet
1	Kesalahanku karena tlah mencintaimu
2	Bisagaksih gakusahbuatorangkesal??
3	dari semua hal aku paling benci yang namanya nunggu!
4	comel ah chen ish bencinyaa
5	Lawan Madrid Rakitic Sempat Dibuat Kesal
6	Tak salah pun aku bawak motor.takkan disebabkan motor kau benci.????

No	Tweet
7	Jangan terlalu benci sama orang nanti kalo jadi cinta malu sendiri lo.
8	Yg dewasa yg ngalah. Yg dewasa yg harus kasih solusi. Yg dewasa yg harus bisa ngendaliin keadaan. Pakai otak bukan emosi atau otot.
9	Hidup bukan tentang seberapa besar kesalahanmu di masa lalu tapi tentang bagaimana kamu memperbaiki diri dan kuat menjalani hari.
10	@h0ngbin_ biar penasaran /? Kalo situasi digodain terus2an ngambek gak?
11	#AutoFreeFollowers Inisial seseorang yang kamu benci ?

Kelas takut memiliki *recall* yang rendah yaitu hanya 15,2% namun memiliki *precision* yang tinggi. Dari 500 data *tweet* takut hanya 76 yang dapat diklasifikan dengan benar, namun model klasifikasi hanya mengklasifikan 6 data yang salah sebagai kelas takut. Hal ini dapat menandakan bahwa kosa kata yang berada pada data *training* kelas takut cukup unik dan tidak ada pada kelas lain sehingga model klasifikasi hanya mengklasifikasikan 82 data sebagai kelas takut, kelas ini memiliki pola yang sama seperti kelas marah. Tingkat *precision* pada kelas takut menjadi cukup tinggi, mencapai hingga 92%. Namun angka *recall* pada kelas takut sangat rendah karena terdapat 307 *tweet* kelas takut yang diklasifikasikan sebagai kelas netral dan 104 *tweet* sebagai kelas sedih. Hal ini dapat dikarenakan kata yang digunakan ada pada kosa kata kelas netral dan kosa kata kelas netral cukup besar dikarenakan jumlah data yang jauh lebih tinggi. Contoh data yang diklasifikasikan dengan salah pada kelas takut dapat dilihat pada Tabel 5.14. Pada data pada nomor 1 terdapat kata “bahagia” yang memiliki bobot tinggi pada kelas senang. Data nomor 2 terdapat kata “cinta” sehingga mungkin terjadi kesalahan ke kelas senang dan pada data nomor 10 terdapat kata “tenang” dan “positif” yang mungkin memiliki bobot besar pada kelas netral dan senang.

Tabel 5.14 Tabel Kelas Takut yang Diklasifikasikan dengan Salah

No	Tweet
1	Yaiyalah gue cemas lo sama dia dia lebih cantik & bisa bikin lo bahagia daripada gue #KamusCewek
2	Hati melompat-lompat kemana kamu akan terpikat? Jangan terus meragu cinta tak bisa menunggu.
3	@KUNINGSSI bkn gitu takutnya nnt abang rugi sih
4	ki amat mah ga galak :v gausah tkt @miaaw_22 @yulian_mutiaraa aku tadi liat yutup ti : sampe nangis. Takut kiamat:(aneh kan?:3
5	Cewek itu kadang suka ngelarang cowoknya deket sama cewek lain bukan karna apa2...Dia cuma takut cowoknya selingkuh sm yg lain. :)
6	Pengennya sii bareng² sama lo. Tapi takut ngerasain kedulu.haha konyol.:|
7	Walaupun dia bukan lah pencinta kucing and sometime penakut jugak dkt kucing tapi tiap kali aku terdengar bunyi anak kucing so apa lagi bang
8	Ada benda sebenarnya mmg dah nak kena berubah.. tapi diri sendiri tak berani lakukan.. senang je.. kau Penakut!!
9	Sabar itu capek sabar itu emosi sabar itu kesel sabar itu susah tapi sabar itu indah. | loh indahnya kapan? | ya sabar aja :|
10	Berhentilah mencemasan sesuatu secara berlebihan. Cemas adalah buah dari pemikiran. Berpikir positif akan membuat hidup jadi lebih tenang.
11	Zionis pu negara namanya israel katolik punya negara namanya vatikan komunis juga punya negara hindu juga punya negara knp takut islam?
12	Jangan ragu pakai product fashion terkeren karya anak bangsa.Semua bisa kamu dapatin di @WaydeeStore Follow ya

Kelas terkejut memiliki *recall* dan *precision* yang rendah. Hal ini dapat dikarenakan sedikitnya jumlah data training yaitu 2.897 dan kosa kata dari kelas memiliki kesamaan yang dengan kelas netral sehingga dari 724 data *testing,* sebanyak 560 masuk ke kelas

netral dan hanya 42 yang masuk dalam kelas terkejut. Contoh data kelas terkejut yang didefinisikan sebagai salah dapat dilihat pada

Tabel 5.15 Contoh Data Kelas Terkejut yang Diklasifikasikan dengan Salah

No	Tweet
1	Iya:v "@PShinHye90: Aduhh jdi itu kamu >< ocidaakk RT @GaexmGyu: Semalem.kmu ga inget/?"PShinHye90: Omo kita kapan bkina? :o
2	@adilla096 .HAHH:o siapaa yang jadi putri pangeran wkwk *muka polos
3	@folksyoon: folksooj: "folksyoon: eeeebrarti kalo ngomongin ailee di acc atunya aja .g"acc mana lagi ogt y"ljhwvixx/?"oohh :o
4	@Sevan_Dev gimana kabarnya hari ini :o
5	@insom_cy omegodeeeeh:o diaa tak mengakuiku -__- aku udh biasa diginiin loh min aku kuat ko:")"D
6	Heran... kari akeh masalah dino iki?? Maegot :o
7	@skukzkky gimana rasanya ps?:o
8	@realsanamjung ab kia hoga Haya ka :O ab kia krygi wo :(rameez ni kro humri Haya k sath ese .. Hadi aa jaen apni Haya ko bachany .. #Alvida
9	@btsbwiv @bsuga_twt wah kalian suka mangkal :oOoOo
10	Ngantuk :O
11	. Segmen 6 ya..? :O. Jam brpa tuh pass Segmen 6..? Ini ajja udh jam 23.14 Wib :(
12	@icescoklat wah kenapah? :o masalah tugas akhir yah?
13	@Liestaaa loh aku doramania lo :o
14	Emg syp yg ngegangguin situ : iyuhh Geer bgt ANDA :o
15	Sakit kakinya ragara naik ke atas beko :o turunnya aduh kefleset :v

Kelas jijik gagal diklasifikasikan karena jumlah data yang jauh lebih sedikit dan jumlah penanda kelas yang jauh lebih sedikit dibandingkan dengan kelas lain.

5.7. Analisis Skenario Pengujian 2

Pada skenario pengujian dua, jumlah data pada kelas netral ditekan dari 262.454 data menjadi 10.000 data. Pemberian label kelas pun dijadikan lebih selektif dengan hanya menggunakan *hashtag* dan *emoticon* tanpa menggunakan kata kunci. Pada skenario dua akurasi menurun menjadi 72% tetapi nilai *fscore* pada masing-masing kelas lebih merata dan tidak terdapat satu kelas yang performanya menonjol. Pada kedua kasus pengujian ditemukan bahwa kelas jijik memiliki performa yang sangat rendah bahkan pada kasus kedua tidak mampu mengklasifikan sama sekali. Hal ini dikarenakan jumlah data yang sedikit dan penanda pemberi kelas yang belum cukup baik untuk kelas jijik. Pada pengujian dua, performa kelas senang dan sedih cukup baik apabila panjang dari *tweet* setelah *preprocessing* mendekati batas maksimal karakter dalam *tweet* yaitu 140. Namun pada sampel *tweet* yang relatif pendek didapatkan akurasi kelas yang rendah.

Pada kelas netral, *fscore* menurun dari 89% menjadi 76,9% namun pada kelas senang, *fscore* meningkat menjadi 75% dari 55% dan pada kelas sedih *fscore* meningkat menjadi 65% dari 16%. Kelas marah memiliki fscore yang meningkat menjadi 37,4% dari 27,6% namun *precision* menurun dari 92% menjadi 63%. Hal ini berarti model klasifikasi lebih sensitif untuk mengklasifikasikan kelas marah dan oleh karenanya *recall* kelas marah meningkat. Kelas jijik tetap rendah sama seperti skenario pengujian pertama. Sedangkan pada kelas terkejut *fscore* meningkat menjadi 27% dari 10% tapi pada kelas takut *fscore* menurun tipis dari 26% menjadi 23,7%.

Pada kelas senang, terjadi kenaikan yang signifikan, setelah analisis data dilakukan, ditemukan bahwa terdapat 962 data *tweet*

yang mampu diklasifikasikan dengan benar dari 1301 *tweet* data testing dengan kelas yang benar. Dari 962 data *tweet* yang benar ditemukan bahwa 772 *tweet* memiliki panjang teks lebih dari 50 huruf setelah teks dilakukan preprocessing. Jadi terdapat 110 *tweet* kelas senang yang diklasifikasikan dengan benar dan memilki panjang teks kurang dari 51 huruf. Pola yang sama juga ditemukan di kelas sedih. Kelas sedih mengalami peningkatan yang cukup besar dari skenario pengujian pertama. Dari 369 *tweet* kelas sedih yang diklasifikasikan dengan benar ditemukan 226 *tweet* dengan panjang teks setelah *preprocessing* lebih dari 50 huruf. Hanya 143 tweet yang memiliki panjang teks setelah *preprocessing* kurang dari 50 huruf.

[Halaman ini sengaja dikosongkan]

BAB VI
KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan tugas akhir ini. Selain itu, juga terdapat beberapa saran terhadap tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik lagi.

6.1. Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan pada tugas akhirantara lain:

1. Dengan dataset yang seimbang, deteksi emosi menggunakan naive bayes memiliki performa yang baik pada kelas netral, senang dan sedih. Hal ini ditunjukkan oleh fscore pada kelas netral sebesar 76,9%, kelas senang 75,1% dan kelas sedih 65,5%.
2. Pada data set yang tidak seimbang, metode naive bayes akan cenderung memberikan prediksi kelas untuk data *testing* sesuai dengan kelas yang memiliki jumlah data lebih banyak, pada kasus ini adalah kelas netral.
3. Pada kelas senang dan kelas sedih, *tweet* dengan panjang teks setelah *preprocessing* yang lebih tinggi cenderung memiliki *fscore* lebih tinggi. Hal ini dapat diterapkan untuk membangun model klasifikasi baru dengan panjang *tweet* sebagai salah satu faktor yang diperhitungkan.
4. Evaluasi menggunakan akurasi tidak dapat digunakan pada kasus klasifikasi ini dikarenakan tidak imbangnya jumlah data pada setiap kelas. Skenario 1 menunjukkan perhitungan akurasi yang baik sebesar 80% namun pada pada enam kelas selain kelas netral menunjukkan performa *fscore* yang buruk. Pada kelas senang fscore bernilai sebesar 55%, kelas sedih 16%, kelas marah 27,6%, kelas jijik 0,7%, kelas terkejut 10%, dan kelas takut 26,1%

5. Kelas jijik tidak dapat diklasifikasikan sama sekali dengan fscore dibawah satu persen. Hal ini dikarenakan sedikitnya jumlah yang terkait dengan kelas tersebut.

6.2. Saran

Terdapat beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Perlu dilakukan *preprocessing* lebih lanjut untuk mendapatkan data yang lebih baik dan lebih toleran kepada bahasa tidak baku.
2. Penanda pemberi label kelas dapat ditambah dengan menggunakan penanda lain seperti kata kunci dan *emoji* .
3. Perlu dilakukan penelitian lebih lanjut untuk membedakan kelas jijik, terkejut dan takut dalam teks.
4. Perlu dilakukan pencarian data *tweet* yang lebih dalam untuk kelas jijik, terkejut dan takut dalam *tweet.*
5. Perlu dilakukan pembersihan bahasa selain Bahasa Indonesia seperti Bahasa Malaysia, Bahasa Sunda, Bahasa Jawa dan Bahasa India yang terambil dalam proses pengambilan data.

LAMPIRAN

Lampiran A. 1 Daftar Kata yang Termasuk dalam Stopwords oleh Fadhilah Z Tala

ada	ataupun	sebelum
adanya	bagai	sebelumnya
adalah	bagaikan	sebenarnya
adapun	sebagai	berapa
agak	sebagainya	berapakah
agaknya	bagaimana	berapalah
agar	bagaimanapun	berapapun
akan	sebagaimana	betulkah
akankah	bagaimanakah	sebetulnya
akhirnya	bagi	biasa
aku	bahkan	biasanya
akulah	bahwa	bila
amat	bahwasanya	bilakah
amatlah	sebaliknya	bisa
anda	banyak	bisakah
andalah	sebanyak	sebisanya
antar	beberapa	boleh
diantaranya	seberapa	bolehkah
antara	begini	bolehlah
antaranya	beginian	buat
diantara	beginikah	bukan
apa	beginilah	bukankah
apaan	sebegini	bukanlah
mengapa	begitu	bukannya
apabila	begitukah	cuma
apakah	begitulah	percuma
apalagi	begitupun	dahulu
apatah	sebegitu	dalam
atau	belum	dan
ataukah	belumlah	dapat

dari
daripada
dekat
demi
demikian
demikianlah
sedemikian
dengan
depan
di
dia
dialah
dini
diri
dirinya
terdiri
dong
dulu
enggak
enggaknya
entah
entahlah
terhadap
terhadapnya
hal
hampir
hanya
hanyalah
harus
haruslah
harusnya
seharusnya
hendak
hendaklah
hendaknya
hingga
sehingga
ia
ialah
ibarat
ingin
inginkah
inginkan
ini
inikah
inilah
itu
itukah
itulah
jangan
jangankan
janganlah
jika
jikalau
juga
justru
kala
kalau
kalaulah
kalaupun
kalian
kami
kamilah
kamu
kamulah
kan
kapan
kapankah
kapanpun
dikarenakan
karena
karenanya
ke
kecil
kemudian
kenapa
kepada
kepadanya
ketika
seketika
khususnya
kini
kinilah
kiranya
sekiranya
kita
kitalah
kok
lagi
lagian
selagi
lah
lain
lainnya
melainkan
selaku
lalu
melalui
terlalu
lama
lamanya
selama
selama
selamanya
lebih

terlebih
bermacam
macam
semacam
maka
makanya
makin
malah
malahan
mampu
mampukah
mana
manakala
manalagi
masih
masihkah
semasih
masing
mau
maupun
semaunya
memang
mereka
merekalah
meski
meskipun
semula
mungkin
mungkinkah
nah
namun
nanti
nantinya
nyaris
oleh
olehnya
seorang
seseorang
pada
padanya
padahal
paling
sepanjang
pantas
sepantasnya
sepantasnyalah
para
pasti
pastilah
per
pernah
pula
pun
merupakan
rupanya
serupa
saat
saatnya
sesaat
saja
sajalah
saling
bersama
sama
sesama
sambil
sampai
sana
sangat
sangatlah
saya
sayalah
se
sebab
sebabnya
sebuah
tersebut
tersebutlah
sedang
sedangkan
sedikit
sedikitnya
segala
segalanya
segera
sesegera
sejak
sejenak
sekali
sekalian
sekalipun
sesekali
sekaligus
sekarang
sekarang
sekitar
sekitarnya
sela
selain
selalu
seluruh
seluruhnya
semakin
sementara
sempat

semua
semuanya
sendiri
sendirinya
seolah
seperti
sepertinya
sering
seringnya
serta
siapa
siapakah
siapapun
disini
disinilah
sini
sinilah
sesuatu
sesuatunya
suatu
sesudah
sesudahnya
sudah
sudahkah
sudahlah
supaya
tadi
tadinya
tak
tanpa
setelah
telah
tentang
tentu
tentulah
tentunya
tertentu
untuk
seterusnya
tapi
tetapi
setiap
tiap
setidaknya
tidak
tidakkah
tidaklah
toh
waduh
wah
wahai
sewaktu
walau
walaupun
wong
yaitu
yakni
yang

Lampiran A. 2 Daftar Kata yang Termasuk dalam Stopwords Singkatan dan Kata tidak Baku

yg	iya	kak
ya	ku	follow
nak	dgn	hai
kau	banget	salam
aja	lg	kenal
dah	ko	terimakasih
jadi	eh	yaa
ga	jgn	unfoll
orang	udh	hahaha
gak	kal	ah
gue	gitu	wkwk
udah	ha	si
kalo	haha	viceennt22
gua	gk	lo
lu	gak	
nya	jd	
la	deh	
tau	rt	

Lampiran A. 3 Daftar Kata yang Termasuk dalam Stopwords Bahasa Inggris

a	anyhow	behind
a's	anyone	being
able	anything	believe
about	anyway	below
above	anyways	beside
according	anywhere	besides
accordingly	apart	best
across	appear	better
actually	appreciate	between
after	appropriate	beyond
afterwards	are	both
again	aren't	brief
against	around	but
ain't	as	by
all	aside	c
allow	ask	c'mon
allows	asking	c's
almost	associated	came
alone	at	can
along	available	can't
already	away	cannot
also	awfully	cant
although	b	cause
always	be	causes
am	became	certain
among	because	certainly
amongst	become	changes
an	becomes	clearly
and	becoming	co
another	been	com
any	before	come
anybody	beforehand	comes

concerning
consequently
consider
considering
contain
containing
contains
corresponding
could
couldn't
course
currently
d
definitely
described
despite
did
didn't
different
do
does
doesn't
doing
don't
done
down
downwards
during
e
each
edu
eg
eight
either
else
elsewhere
enough
entirely
especially
et
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
f
far
few
fifth
first
five
followed
following
follows
for
former
formerly
forth
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had
hadn't
happens
hardly
has
hasn't
have
haven't
having
he
he's
hello
help
hence
her
here
here's
hereafter
hereby
herein
hereupon
hers

herself
hi
him
himself
his
hither
hopefully
how
howbeit
however
i
i'd
i'll
i'm
i've
ie
if
ignored
immediate
in
inasmuch
inc
indeed
indicate
indicated
indicates
inner
insofar
instead
into
inward
is
isn't
it
it'd
it'll
it's
its
itself
j
just
k
keep
keeps
kept
know
knows
known
l
last
lately
later
latter
latterly
least
less
lest
let
let's
like
liked
likely
little
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
n
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone

nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn't
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry

specified
specify
specifying
still
sub
such
sup
sure
t
t's
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that's
thats
the
their
theirs
them
themselves
then
thence
there
there's
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they'd
they'll
they're
they've
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
wasn't
way
we
we'd
we'll
we're
we've
welcome

well
went
were
weren't
what
what's
whatever
when
whence
whenever
where
where's
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who's
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won't
wonder
would
would
wouldn't
x
y
yes
yet
you
you'd
you'll
you're
you've
your
yours
yourself
yourselves
z
zero

Lampiran B.1 Kode Pengambilan Data Twitter

```
import data.database.DBConnection;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Map;
import twitter4j.GeoLocation;
import twitter4j.Query;
import twitter4j.QueryResult;
import twitter4j.RateLimitStatus;
import twitter4j.Status;
import twitter4j.TwitterException;

/**
 *
 * @author Mahardhika
 * crawling tweet menggunakan search API
 */
public class TweetCrawler {

//    private static Connection connection;

   public static void main(String[] args)
throws TwitterException,
ClassNotFoundException, SQLException,
InterruptedException {
        boolean nextData=true;
        QueryResult result = null;
        boolean useProxy=true;
        Map<String,RateLimitStatus> limit =
TwitterHelper.getInstanceTwitter(useProxy).get
RateLimitStatus();
        Query query = new Query();

        query.setQuery("lang:id");
        query.setCount(100);
        DBConnection connection;
        connection = new DBConnection();
        connection.connect();
```

```
        RateLimitStatus r =
limit.get("/search/tweets");
        int i = r.getRemaining()-2;
        int c = 0;
     while (nextData) {
                if(i!=0){
                 result =
TwitterHelper.getInstanceTwitter(useProxy).sea
rch(query);
                 i--;}

                 if(i==0){

System.out.println("Waiting for twitter
limitation; " + i);
                     Thread.sleep(900000);
                     i=180;}
                 nextData=result.hasNext();
                 query=result.nextQuery();

                 for (Status status :
result.getTweets()) {
                     System.out.println(" => @"
+ status.getUser().getScreenName() + " : " +
status.getText());
                     c++;
                     java.sql.Date date;
                     date = new
java.sql.Date(status.getCreatedAt().getTime())
;
                     if
(status.getGeoLocation()!=null)
                     {
                           GeoLocation geo =
status.getGeoLocation();

connection.InsertTweet(date,
String.valueOf(status.getUser().getId()),
status.getUser().getScreenName(),
```

```
status.getUser().getName(), status.getText(),
geo.getLongitude(), geo.getLatitude(), null);
                        }
                         else

connection.InsertTweet(date,
String.valueOf(status.getUser().getId()),
status.getUser().getScreenName(),
status.getUser().getName(), status.getText(),
-9999, -9999, null);
                    }
                 System.out.println("Tweet
fetched :"+c);
               }
        }
                           }
```

Lampiran B.2 Kode Stemming Nazief Adriani

```
package data.preprocessing;

import data.database.DBConnection;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.StringTokenizer;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author Mahardika Maulana
 */
public class Stemming {

    DBConnection connection = new
DBConnection();
    ArrayList<String> wordsList = new
ArrayList<String>();

    public Stemming(){
        try {
            this.loadDictionary();
        } catch (ClassNotFoundException ex) {

Logger.getLogger(Stemming.class.getName()).log
(Level.SEVERE, null, ex);
        } catch (IOException ex) {

Logger.getLogger(Stemming.class.getName()).log
(Level.SEVERE, null, ex);
```

```
        }
    }

    public boolean checkDictionary(String
word) throws FileNotFoundException,
ClassNotFoundException, IOException
    {

        for(String words: wordsList)
        {
            if(words.equalsIgnoreCase(word))
                return true;
        }
        return false;
    }

    public ArrayList<String> loadDictionary()
throws FileNotFoundException,
ClassNotFoundException, IOException
    {
        File file = new File("kata-
dasar.txt");
        BufferedReader buffer;
        buffer = new BufferedReader(new
FileReader(file));

        String CurrentLine;


        while ((CurrentLine =
buffer.readLine()) != null) {


            CurrentLine= CurrentLine.split("
",2)[0];
            wordsList.add(CurrentLine);

            }
        return wordsList;
```

```
            }

    public String
deleteInflictionSuffixes(String word)
    {
        //regex pertama merupakan akhiran
particle
        String regex="([klt]ah|pun)$";
        Pattern r =
Pattern.compile(regex,Pattern.CASE_INSENSITIVE
);
        Matcher m = r.matcher(word);
        if(m.find())
            word = word.replaceAll(regex, "");

        // regex kedua merupakan akhiran
kepemilikan
        String regex2="([km]u|nya)$";
        Pattern r2=
Pattern.compile(regex2,Pattern.CASE_INSENSITIV
E);
        Matcher m2 = r2.matcher(word);
        if(m2.find())
            return word.replaceAll(regex2,
"");
        else return word;
    }


    public String
deleteDerivationSuffixes(String word) throws
ClassNotFoundException, IOException
    {

        word = deleteInflictionSuffixes(word);

        String regex="((i|an)$)";
```

```
        Pattern r =
Pattern.compile(regex,Pattern.CASE_INSENSITIVE
);
        Matcher m = r.matcher(word);
        if(m.find())
        {
             word = word.replaceAll(regex, "");
             if(checkDictionary(word))
                 return word;
             else if (word.length()>1)
                 if(word.charAt(word.length()-
1)=='k' )
                     return
word.substring(0,word.length()-1);

        }
        return word;
    }

    public String
deleteDerivationPrefixes(String word) throws
ClassNotFoundException, IOException
    {
        String originalWord = word;
        String word1="";
        String regex="^(me|pe|per)";
        Pattern r0 =
Pattern.compile(regex,Pattern.CASE_INSENSITIVE
);
        Matcher m0 = r0.matcher(word);
        if(m0.find())
        {
        word = word.replaceAll(regex,"");

        }
        if(checkDictionary(word))
           return word;
        else word = originalWord;
```

```
        String
regex1="^([mp](eng|em|eny|en)|di)";
        Pattern r =
Pattern.compile(regex1,Pattern.CASE_INSENSITIV
E);
        Matcher m = r.matcher(word);
        if(m.find())
        {

        word = word.replaceAll(regex1,"");

        if(word.matches("[aiueo](.*)"))
        {

if(originalWord.matches("[mp]eng(.*)"))
            {word="k"+word;
                    }
            else
if(originalWord.matches("[mp]eny(.*)"))
            {word="s"+word;
                    }
            else
if(originalWord.matches("[mp]en(.*)"))
            {word="t"+word;

            }
            else
if(originalWord.matches("[mp]em(.*)"))
            {word="p"+word;
                    }
            if(checkDictionary(word))
                return word;
            else
                word = word.substring(1);
            }
        else if(checkDictionary(word))
               return word;}

       if(checkDictionary(word))
```

```
            return word;

        originalWord=word;
        String regex2="^([pkst]e)";
        Pattern r2 =
Pattern.compile(regex2,Pattern.CASE_INSENSITIV
E);
        Matcher m2 = r2.matcher(word);
        if(m2.find())
        {word = word.replaceAll(regex2,"");
        if(checkDictionary(word))
           return word;
        else word=originalWord;

        }

        String regex3="^([tbps]e[rl]{0,1})";
        Pattern r3 =
Pattern.compile(regex3,Pattern.CASE_INSENSITIV
E);
        Matcher m3 = r3.matcher(word);
        if(m3.find())
        {word = word.replaceAll(regex3,"");
        if(checkDictionary(word))
           return word;
                else word=originalWord;

        }

        return word;
    }

    public String naziefAdrianiStemmer(String
word) throws ClassNotFoundException,
IOException
    {
       word = word.toLowerCase();
       String originalWord = word;
```

```
        String word1 =
deleteDerivationPrefixes(word);
        String word2 =
deleteInflictionSuffixes(word);
        String word3 =
deleteDerivationSuffixes(word);

        if(checkDictionary(word))
            return word;

        if(checkDictionary(word1))
             return word1;
        else if(checkDictionary(word2))
             return word2;
        else if(checkDictionary(word3))
             return word3;

        word1 =
deleteInflictionSuffixes(word1);
        word2 =
deleteDerivationSuffixes(word2);
        word3 =
deleteDerivationPrefixes(word3);
        if(checkDictionary(word1))
             return word1;
        else if(checkDictionary(word2))
             return word2;
        else if(checkDictionary(word3))
             return word3;

        word1 =
deleteDerivationSuffixes(word1);
        word2 =
deleteDerivationPrefixes(word2);
        word3 =
deleteInflictionSuffixes(word3);
        if(checkDictionary(word1))
             return word1;
        else if(checkDictionary(word2))
```

```
            return word2;
        else if(checkDictionary(word3))
            return word3;

         return originalWord;
    }
}
```

Lampiran B.3 Kode Preprocessing dan Penghapusan Stopwords

```
/*
 * To change this license header, choose
License Headers in Project Properties.
 * To change this template file, choose Tools
| Templates
 * and open the template in the editor.
 */

package data.preprocessing;

import data.database.DBConnection;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.StringTokenizer;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Mahardika Maulana
 * menghilangkan stop words di database
 * input: Tweet di database, daftar stopwords
 * output:
 *
 */

/**
 *
 * @author Mahardika Maulana
```

```
 menghilangkan stop words di database
 input: Tweet di database, daftar stopwords
 output:
 */
public class StopwordsRemoval {
    BufferedReader buffer;
    ArrayList<String> stopWordsList = new
ArrayList<String>();
    String query2;
    File file = new File("stopwords_id.txt");
    private DBConnection connection = new
DBConnection();
    Stemming stemmer = new Stemming();

    public void removeStopwords() throws
FileNotFoundException, IOException,
ClassNotFoundException, SQLException{
        buffer = new BufferedReader(new
FileReader(file));

        connection.connect();
        String CurrentLine;
        while ((CurrentLine =
buffer.readLine()) != null) {
            if(!CurrentLine.contains("#"))
           stopWordsList.add(CurrentLine);
            }
        boolean nextData = true;
        int startIndex = 1;
        String text2=""; //string yang telah
dibersihkan
        String text3=""; // string yang
dibersihkan tanpa di stemming
        String nonStemmedWord;
        String temp;
        int counter=0;
        int resultCount = 10000;
        ResultSet rs;
        String text;
```

```
        String classification;
        int idTweet;
        boolean found=false;
        while(nextData)
        {
            rs = selectTweet(startIndex,
resultCount);
            //hentikan loop apabila rs tidak
mengembalikan hasil
            if(!rs.isBeforeFirst())
            {
                break;
            }

            System.out.println(startIndex + "
" + resultCount);
            if(rs.first()){
                while(!rs.isAfterLast())
                {
                    // remove stopwords here
                    text =
rs.getString("Text");
                    idTweet =
rs.getInt("IdTweet");
                    classification =
rs.getString("Classification");

//                    text =
text.replaceAll("\\?", " ");
//                    text =
text.replaceAll("", " ");
//                    text =
text.replaceAll("", " ");
//                    text =
text.replaceAll("", " ");
                    text =
text.replaceAll("[/,!.?]", " ");
```

```
                    StringTokenizer st = new
StringTokenizer(text);
                    while
(st.hasMoreElements()) {
                        found=false;
            = st.nextToken().toLowerCase();
                        if(temp.contains("@")
|| !temp.matches("^[A-Za-z#:;].*$"))
                        {
//
System.out.println(temp);
                            continue;
                        }

                      for(String words:
stopWordsList)
                        {

if(words.equalsIgnoreCase(temp)){
                            found = true;
                            break;
                        }
                        }
                        if(!found)

{//System.out.print(temp+" ");
                            //text 2 adalah
text yang sudah dibersihkan

                        if(temp.length() > 25
|| temp.length() <2)
                            continue;
                        nonStemmedWord=temp;
                        text3=text3+temp+" ";
//text yang tidak dilakukan stemming

                        temp =
stemmer.naziefAdrianiStemmer(temp); //
stemming text
```

```
//
System.out.print(classification);

//
if(classification.equals("Training")){

connection.InsertTerm(idTweet, temp);

connection.InsertNonStemmedTerm(idTweet,
nonStemmedWord);
//                          }

                        if(temp.length() < 20)
                        text2=text2+temp+" ";
// text yang di stemming
                    }
                }
                   counter++;

System.out.println(counter);
                    String query = "update
dbo.twitter set CleanedText=?,nonStemmedText=?
where IdTweet=?";

connection.executeUpdate(query,text2,text3,idT
weet);
                    text2="";
                    text3="";
                    rs.next();
            }
        }
            startIndex = startIndex +
resultCount;
    }
}

    public ResultSet selectTweet(int
startIndex,int numberOfRow) throws
SQLException
```

```
    {
        String query = String.format("select
text,idTweet,Classification from dbo.Twitter
where IdTweet>=%d and IdTweet<=%d and
classification is not
null",startIndex,startIndex+numberOfRow);
//        String query = String.format("select
* from dbo.Twitter where IdTweet>=%d and
IdTweet<=%d and
classification='Training'",startIndex,startInd
ex+numberOfRow);

        return
connection.executeSelect(query);
    }


    public static void main(String[] args)
throws IOException
    {
        StopwordsRemoval a= new
StopwordsRemoval();
        try {
            a.removeStopwords();
//
System.out.println(System.getProperty("user.di
r"));
        } catch (ClassNotFoundException |
SQLException ex) {

Logger.getLogger(StopwordsRemoval.class.getNam
e()).log(Level.SEVERE, null, ex);
        }

    }

}
```

DAFTAR PUSTAKA

[1] H. Binali, W. Chen and V. Potdar, "Computational Approaches for Emotion Detection in Text," in *4th IEEE International Conference on Digital Ecosystems and Technologies*, 2010.

[2] M. Chunling, H. Prendinger and M. Ishizuka, "Emotion Estimation and Reasoning on Affective Textual Interaction," *Affective Computing and Intelligent Interaction,* vol. 3784, pp. 622-628, 2005.

[3] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," 2009.

[4] P. Ekman, Universals and cultural differences in facial expressions of emotion, vol. 19, Nebraska: University of Nebraska Press, 1972.

[5] I. Cohan, A. Garg and T. S. Huang, "Emotion Recognition from Facial Expressions using Multilevel HMM," in *in Neural Information Processing Systems*, 2000.

[6] "REST APIs | Twitter Developers," Twitter, 22 Maret 2015. [Online]. Available: https://dev.twitter.com/rest/public. [Diakses 22 Maret 2015].

[7] M. Purver and B. Stuart, Experimenting with Distant Supervision for Emotion Classification, London: Association for Computational Linguistics, 2012.

[8] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing., 1999: MIT Press.

[9] M.-K. Shan, K. Fang-Fei, M.-F. Chiang and L. Suh-Yin, "Emotion-based Music Recommendation by Affinity Discovery from Film Music," *Expert Systems with Applications,* no. 36, pp. 7666-7674, 2009.

[10] A. T. Ho, Menezes, I. L.L. and Y. Tagmouti, E-MRS: Emotion-based Movie Recommender System, Department of Informatics and Operations Research.

[11] P. R. Shaver, U. Murdaya and R. C. Fraley, "Structure of the Indonesian emotion lexicon," *Asian Journal of Social Psychology,* vol. 4, pp. 201-224, 2001.

[12] D. Hockenbury and S. Hockenbury, Discovering Psychology, New York: Worth Publishers.

[13] W. Wahyu and H. J. Prawitasari, "Struktur Semantik Kata Emosi dalam Bahasa Indonesia," *Jurnal Psikologi,* vol. 37, no. 9012, pp. 153-164, Desember 2010.

[14] D. Pyle, Data Preparation for Data Mining, Los Altos, California: Morgan Kaufmann Publishers, 1999.

[15] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, "Data Preprocessing for Supervised Learning," *International Journal of Computer Science,* vol. 1 N. 2, pp. 111-117, 2006.

[16] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," 2003.

[17] M. Porter, An Algorithm for Suffix Stripping, Cambridge, 1980.

[18] A. M, A. J and B. Nazief, Stemming Indonesian: A confix-stripping approach, Jakarta, 2007.

[19] A. Z. Broder, S. C. Glassman, M. S. Manasse and G. Zweig, Syntactic clustering of the web, 1997.

[20] A. M. Kibriya, E. Frank, B. Pfahringer and G. Holmes, "Multinomial Naive Bayes for Text Categorization Revisited," *Lecture Notes in Computer Science,* vol. 3339, pp. 488-499, 2005.

[21] D. M. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Corelation," Adelaide, 2007.

BIODATA PENULIS

Penulis lahir di Banyuwangi, 6 Juni 1994. Penulis telah menempuh pendidikan dasar di SDN 22 Palu dan SDN Jember Lor 1, kemudian untuk pendidikan menengah pertama di SMPN 2 Jember dan di jenjang menengah atas di SMAN 1 Jember. Sejak kecil, penulis memiliki ketertarikan yang besar pada bidang komputer sehingga penulis memutuskan untuk mengambil pendidikan sarjana S1 di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Komputasi Cerdas dan Visi serta memiliki ketertarikan di bidang Analisis Media Sosial, Data Mining, Pengolahan Citra Digital, Visi Komputer dan Sistem Temu Kembali Informasi. Untuk komunikasi, penulis dapat dihubungi melalui surel: mahardhikamaulana12@gmail.com