

TUGAS AKHIR - KI141502

Pemodelan Multilabel Tweet Media Sosial Mahasiswa untuk Klasifikasi Keluhan

Muhammad Faris Musthafa
NRP 5113 100 131

Dosen Pembimbing I
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D

Dosen Pembimbing II
Victor Hariadi, S.Si., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

Pemodelan Multilabel Tweet Media Sosial Mahasiswa untuk Klasifikasi Keluhan

Muhammad Faris Musthafa
NRP 5113 100 131

Dosen Pembimbing I
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D

Dosen Pembimbing II
Victor Hariadi, S.Si., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**Multilabel Modeling of Students' Social Media
Tweets for the Classification of Complaints**

**Muhammad Faris Musthafa
NRP 5113 100 131**

**Supervisor I
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D**

**Supervisor II
Victor Hariadi, S.Si., M.Kom.**

**INFORMATICS DEPARTMENT
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PEMODELAN MULTILABEL TWEET MEDIA SOSIAL MAHASISWA UNTUK KLASIFIKASI KELUHAN

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
Muhammad Faris Musthafa
NRP. 5113 100 186

Disetujui oleh Dosen Pembimbing

Prof. Ir. Joko Lianto Buliali, M.Eng., PhD
NIP: 19670727 199203 1 002

Victor Hariadi, S.Si., M.Kom.
NIP: 19691228 199412 1 001



SURABAYA
JANUARI, 2018

[Halaman ini sengaja dikosongkan]

PEMODELAN MULTILABEL TWEET MEDIA SOSIAL MAHASISWA UNTUK KLASIFIKASI KELUHAN

Nama : Muhammad Faris Musthafa
NRP : 5113100131
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.
Dosen Pembimbing II : Victor Hariadi, S.Si., M.Kom

Abstrak

Pada umumnya sistem informasi akademik di sebuah perguruan tinggi memiliki fitur umum bagi dosen untuk memantau proses perkembangan akademik anak walinya secara aktif. Namun jika dosen wali ataupun orang tua tidak melakukan pantauan secara aktif maka mahasiswa wali yang memiliki permasalahan akademik berisiko drop out dalam proses evaluasi tingkat 1 universitas karena rendahnya pemahaman dosen terhadap mahasiswa walinya. Tujuan dari Tugas Akhir ini adalah membuat rancangan model prediksi keluhan dalam data tweet mahasiswa. Aspek keluhan bisa dibagi menjadi empat kategori: keluhan personal, keluhan subjek, keluhan relasi, dan keluhan institusi.

Metode multilabel yang digunakan adalah Binary Relevance dengan dengan pilihan *classifier* Naïve Bayes, Simple Logistic, KStar Decision Table, dan Naïve Bayes.

Berdasarkan hasil pengujian pada berbagai *classifier* Naïve Bayes memiliki performa tertinggi baik dalam aspek akurasi maupun waktu eksekusi. Hasil implementasi sistem Deteksi Multilabel Keluhan menggunakan *classifier* Naïve Bayes pada delapan puluh data uji terhadap label keluhan personal, subjek, relasi, dan institusi memiliki akurasi masing-masing bernilai 76.47%, 75%, 80%, dan 80%. Hasil deteksi multilabel keluhan yang ditemukan berpotensi digunakan lebih lanjut pada konteks yang lebih luas.

Kata kunci: Deteksi keluhan, kegagalan akademik, permodelan prediksi, multilabel, media sosial

[Halaman ini sengaja dikosongkan]

MULTILABEL MODELING OF STUDENTS' SOCIAL MEDIA TWEETS FOR THE CLASSIFICATION OF COMPLAINTS

Student Name : Muhammad Faris Musthafa
NRP : 5113100131
Major : Informatics Department – FTIf ITS
Supervisor I : Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.
Supervisor II : Victor Hariadi, S.Si., M.Kom

Abstract

In general, academic information systems in a university have a common feature for lecturers to monitor the process of academic development of their students actively. However, if the lecturers or parent do not actively monitor, the students who have academic problems are at risk of dropping out in the university in level one evaluation process because of the low understanding of the lecturers towards the students. The purpose of this Final Project is to construct detection model of complaint in student tweet data. The complaint aspects can be divided into four categories: personal complaints, subject complaints, relationship complaints, and institutional complaints.

The multilabel method used is Binary Relevance Transformation with classifier choice of Naïve Bayes classifier, Simple Logistic, KStar Decision Table, and Naïve Bayes.

Based on test results on various classifier, Naïve Bayes has the highest performance both in the aspect of accuracy and execution time. The implementation results of the Multilabel Detection of Complaints using Naïve Bayes classifier on eighty test data on personal complaint label, subject, relation, and institution have accuracy of 76.47%, 75%, 80%, and 80% respectively. The results of multilabel detection of found complaints are potentially used further in the wider context.

Keywords: Early Detection, academic failure, complaint detection modelling, multilabel, social media

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Multilabelisasi Permasalahan Mahasiswa dari Data Media Sosial untuk Prediksi Keluhan dalam Tweet Mahasiswa”

Pengerjaan Tugas Akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku Tugas Akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu, kakak, dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Joko Lianto Buliali dan bapak Victor Hariadi selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Segenap dosen rumpun mata kuliah Manajemen Informasi.
6. Rekan-rekan mahasiswa S2 dan S3 Teknik Informatika yang telah banyak meluangkan waktu dan bersedia membantu penulis dalam memahami pengerjaan Tugas Akhir ini.
7. Teman-teman seperjuangan anak didik Tugas Akhir Prof. Joko Lianto Buliali

8. Teman-teman Teknik Informatika Angkatan 2013 yang selalu mendukung, menyemangati, membantu dan mendengarkan suka duka selama proses pengerjaan Tugas Akhir.
9. Serta semua pihak yang turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Januari 2018

Muhammad Faris Musthafa

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak	i
<i>Abstract</i>	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Permasalahan	4
1.4 Tujuan	4
1.5 Manfaat.....	5
1.6 Metodologi	5
1.7 Sistematika Penulisan.....	6
2 BAB II DASAR TEORI.....	9
2.1 Teks Mining	9
2.2 NLP (Natural Language Processing).....	9
2.2.1 Information Retrieval	10
2.2.2 Morphological Analysis.....	11
2.2.3 Stemming & Lemmatization	11
2.2.4 InaNLP	12
2.2.5 <i>Part-of-Speech-Tagger</i> (POS Tagger).....	12
2.3 MEKA.....	14
2.4 Binary Relevance For Multilabel Classification	15

2.5 Naïve Bayes Classifier	15
2.6 <i>Java</i>	16
2.6.1 Java Swing	17
2.7 MySQL.....	17
3 BAB III ANALISIS DAN PERANCANGAN SISTEM	19
3.1. Tahap Analisis.....	19
3.1.1 Deskripsi Umum	19
3.1.2 Arsitektur Perangkat Lunak.....	19
3.1.3 Analisis Aktor	20
3.1.4 Analisis Permasalahan.....	20
3.1.5 Spesifikasi Kebutuhan Perangkat Lunak.....	20
3.1.6 Kasus Penggunaan	21
3.1.7 Data Uji Pengujian Sistem	28
3.1.8 Penentuan Classifier Latih	28
3.2. Tahap Perancangan Sistem.....	35
3.1.1 Perancangan Basis Data	35
3.1.2 Perancangan Antarmuka.....	37
3.1.3 Pengolahan Data Latih	39
3.1.4 Pelatihan Data Latih.....	46
3.1.5 Prediksi dan Hasil	54
4 BAB IV IMPLEMENTASI SISTEM.....	57
4.1 Lingkungan Implementasi.....	57
4.2 Tahap Implementasi	57
4.2.1 Tahap Pelatihan Data Latih	57
4.2.2 Tahap Prapemrosesan Data latih.....	58
4.2.3 Tahap Prediksi	60

4.3 Implementasi Antarmuka	60
4.3.1 Antarmuka Kolom <i>Main Menu</i>	61
5 BAB V PENGUJIAN DAN EVALUASI	63
5.1 Lingkungan Uji Coba	63
5.2 Data Uji Coba.....	63
5.3 Skenario Pengujian.....	68
5.3.1 Membuka aplikasi	68
5.3.2 Memilih file data latih	69
5.3.3 Melakukan prediksi pada data latih.....	69
5.3.4 Membuka file hasil prediksi.....	70
5.4 Hasil uji coba.....	70
5.5 Evaluasi Pengujian	79
6 BAB VI KESIMPULAN DAN SARAN.....	81
6.1 Kesimpulan.....	81
6.2 Saran.....	82
DAFTAR PUSTAKA.....	83
BIODATA PENULIS.....	85

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Proses Utama Teks Mining.....	9
Gambar 3.1 Diagram Kasus Penggunaan.....	21
Gambar 3.2 Diagram Aktivitas UC-001.....	23
Gambar 3.3 Diagram Aktivitas UC-002.....	25
Gambar 3.4 Diagram Aktivitas UC-003.....	27
Gambar 3.5 Perbandingan performa akurasi classifier.....	34
Gambar 3.6 Perbandingan build time classifier.....	35
Gambar 3.9 Rancangan antarmuka sistem.....	38
Gambar 3.10 Data Flow Diagram Prapemrosesan.....	39
Gambar 3.11 Diagram inaNLP.....	43
Gambar 3.12 Representasi ARFF.....	46
Gambar 3.13 Penerapan Binary Classifier.....	47
Gambar 3.14 Flowchart Pelatihan Naïve Bayes.....	49
Gambar 3.15 Flowchart Pengujian Naïve Bayes.....	54
Gambar 4.1 Antarmuka perangkat lunak.....	70
Gambar 4.1 Kode Pelatihan Data Latih.....	58
Gambar 4.2 Kode Pembersihan Simbol.....	59
Gambar 4.3 Kode Penggunaan NLP Formalization.....	59
Gambar 4.4 Kode Penggunaan Stemmer.....	60
Gambar 4.5 Kode Prediksi.....	60
Gambar 4.6 Antarmuka perangkat lunak.....	61
Gambar 5.1 Antarmuka halaman utama.....	68
Gambar 5.2 Antarmuka filechooser.....	69
Gambar 5.3 Antarmuka notifikasi prediksi sukses.....	69
Gambar 5.4 Isi file hasil prediksi.....	70

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Struktur Kelas kata POS-Tagging	13
Tabel 3.1 Daftar Kebutuhan Fungsional Sistem.....	200
Tabel 3.2 Keterangan Kode Kasus Penggunaan.....	222
Tabel 3.3 Spesifikasi Kasus Memilih file data uji.....	222
Tabel 3.4 Spesifikasi Kasus melakukan prediksi	244
Tabel 3.5 Spesifikasi Kasus melihat prediksi.....	266
Tabel 3.6 Daftar Respon Survei	288
Tabel 3.7 Daftar hashtag.....	40
Tabel 3.8 Daftar aspek seleksi.....	41
Tabel 3.9 Deskripsi proses pengolahan	43
Tabel 3.10 Daftar imbuhan relevan	44
Tabel 3.11 Daftar POS Tagger	44
Tabel 3.12 Contoh Data Latih	45
Tabel 3.13 Independensi perkata	51
Tabel 3.14 Pembobotan Positive Outcome	52
Tabel 3.15 Daftar Negative Outcome.....	53
Tabel 5.1 Tabel data uji coba	63
Tabel 5.2 Tabel hasil prediksi	70
Tabel 5.3 Tabel legenda hasil deteksi.....	78
Tabel 5.4 Tabel statistik hasil uji coba	78

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pengerjaan Tugas Akhir, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Perguruan Tinggi merupakan sebuah institusi pendidikan penyelenggara pendidikan tinggi. Pendidikan tinggi menjadi hal yang sangat penting dikarenakan mencerminkan kondisi intelektual masyarakat dari sebuah negara. Membutuhkan perhatian dan evaluasi lebih dari pemerintah untuk dapat menghasilkan sumberdaya manusia yang baik dari Perguruan Tinggi. Hal ini tentunya untuk menghasilkan generasi yang mempunyai intelektual dan keterampilan tinggi ketika bersaing di dunia kerja nantinya. Kualitas pendidikan tinggi yang baik menjadi prioritas dan pusat perhatian pendidikan internasional.

Pencapaian tujuan strategis Perguruan Tinggi bergantung pada faktor konstektual, yang meliputi kepemimpinan suatu negara, ideologi negara, finansial, sosial budaya, komunikasi dan bahasa, serta proses pengambilan keputusan. Selain itu Perguruan Tinggi juga didorong oleh norma yang terkait dengan lingkungan dan nilai pendidikan itu sendiri.

Salah satu upaya untuk meningkatkan kualitas Perguruan Tinggi adalah dengan melakukan evaluasi pada mahasiswa yang terdaftar. Evaluasi bertujuan untuk menganalisis performa akademik mahasiswa. Salah satu tujuan dari analisis tersebut adalah untuk melakukan deteksi dini terhadap adanya potensi kegagalan akademik mahasiswa yang terkait.

Deteksi dini kegagalan akademik merupakan hal yang penting didalam operasional sebuah perguruan tinggi. Deteksi tersebut bertujuan untuk mengetahui mahasiswa yang berpotensi mengalami permasalahan akademik secara dini. Semakin awal proses deteksi mahasiswa bermasalah dilakukan, maka semakin kecil resiko mahasiswa bermasalah tersebut mengalami kegagalan

akademik / *drop-out* (tentunya jika didukung dengan adanya penanganan yang cepat dan tepat oleh pihak perguruan tinggi). Dari literatur-literatur terkait dengan kegagalan akademik, pemodelan deteksi dini yang terbaik adalah pemodelan deteksi kegagalan akademik ditahun pertama. Data di lingkungan Universitas Telkom menunjukkan hal tersebut, yaitu rata-rata prosentase *drop out* terbesar semua angkatan terjadi di tahun pertama sebesar 43.32 persen. Oleh karena itu penanganan terbaik untuk mencegah kegagalan tersebut adalah memaksimalkan penanganan di tahun pertama.

Berdasarkan A. Zhang and C. L. Aasheim, dari tiga puluh delapan literatur di penelitian yang telah disurvei, Zhang mengusulkan pengelompokan faktor-faktor penentu kesuksesan akademik menjadi 7 kelompok:

1. Faktor Personal (motivasi, komitmen, dsb)
2. Faktor Keluarga (level pendapatan, status pernikahan, jml saudara, dsb)
3. Faktor Relasi / Pertemanan (teman, teman sekelas, anggota studi, dsb)
4. Subjek / konten (struktur matakuliah, kebijakan, buku teks, dsb)
5. Agen Institusi (dosen, dosen wali, staff, dsb)
6. Reputasi Insitusi (lokasi, beasiswa, dsb)
7. Sosial (ekonomi, tingkat kriminal, dsb)

Seluruh faktor tersebut tidak bisa digali hanya dari data akademik saja, akan tetapi melibatkan data lain sebagai pelengkap data akademik. Faktor partisipasi dan proses belajar dapat dilihat dari data akademik mahasiswa yang bersangkutan, baik dalam hal presensi kehadiran, nilai tugas, nilai UTS, nilai UAS dan nilai akhir. Sedangkan faktor yang lain, yaitu motivasi, komitmen, serta dukungan keluarga dan rekan tidak dapat dilihat didalam data akademik. Oleh karena itu dibutuhkan adanya potensi sumber data lain yang dapat membantu memberikan gambaran motivasi, komitmen, dukungan keluarga dan rekan. Media sosial diharapkan mampu melengkapi kebutuhan data tersebut.

Salah satu potensi pemanfaatan media sosial adalah ekstraksi permasalahan yang dihadapi mahasiswa. Pada saat ini,

penelitian yang ada melakukan penggalian data posting media sosial mahasiswa berbahasa inggris dengan metoda klasifikasi multilabel untuk membantu mengenali beberapa kelompok keluhan permasalahan yang dihadapi mahasiswa secara umum di media sosial terkait perkuliahan seperti beban studi yang berat, permasalahan gangguan tidur, kurang mampu bersosialisasi, watak temperamen, dsb. Pengembangan metode dengan domain bahasa Indonesia belum pernah dilakukan. Domain teks mining permasalahan mahasiswa dari data media sosial berbahasa Indonesia mempunyai tantangan tersendiri, yaitu banyaknya bahasa alay, singkatan, tidak baku, dsb... Beberapa penelitian terkait preproses bahasa tidak baku dari media sosial adalah penelitian yang menggunakan metoda “*clean number, convert number, remove repeat, dan translate*”. Metoda formalisasi bahasa baku tersebut dapat digunakan untuk mendukung ekstraksi multilabel permasalahan mahasiswa dari data media sosial yang akan dibuat.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana merancang dan memodelkan klasifikasi multilabel untuk deteksi keluhan dengan memanfaatkan data media sosial *Twitter* sebagai model input dengan menggunakan metode transformasi multilabelisasi *Binary Relevance* dan *Naïve Bayes* sebagai *classifier*-nya.
2. Bagaimana melakukan seleksi, penyaringan, dan pra-pemrosesan data *Tweet* sehingga siap digunakan sebagai data latih.
3. Bagaimana melakukan deteksi multilabel terhadap kumpulan data uji dengan menggunakan model poin (1).
4. Bagaimana membuat perangkat lunak antarmuka deteksi multilabel pada model poin (1).

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Sistem dibangun dengan basis aplikasi desktop. Bahasa pemrograman yang digunakan adalah *Java* dengan menggunakan *library Swing* untuk perancangan desain antarmuka, *inaNLP* sebagai *Natural Language Processor* dengan bahasa Indonesia untuk melakukan pembakuan ekstraksi kata dalam prapemrosesan data, baik data latih maupun data uji, dan *weka meka* yang merupakan *application* dan *java library* untuk *machine learning* yang digunakan baik pada pengujian metode *classifier* maupun pada implementasi pemrograman.
2. Database relasional yang digunakan adalah MySQL.
3. Data latih yang digunakan merupakan postingan (disebut *tweet*) data media sosial *Twitter*. Data latih didapat dari data posting *Twitter* mahasiswa Universitas Telkom angkatan 2014 dan 2015 yang telah memberikan ijin.
4. Perangkat lunak dapat melakukan deteksi multilabel pada data uji *Tweet* yang sudah disiapkan dalam file (.csv). Proses pengambilan (*crawling*) data uji pada *Twitter* diasumsikan dilakukan pada sistem yang berbeda.
5. Dari tujuh faktor penentu kesuksesan akademik yang diuraikan yang dipaparkan pada Latar Belakang, kategori kelas label untuk multilabel dibatasi menjadi empat faktor yang berkonsentrasi pada empat keluhan, yaitu:
 - 5.1 faktor personal
 - 5.2 faktor subjek
 - 5.3 faktor relasi
 - 5.4 faktor agen institusi

Pemilihan empat faktor tersebut didasarkan pada label data yang ditemukan di dalam data latih media sosial yang dijadikan studi kasus.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang dan memodelkan klasifikasi multilabel dengan empat label dengan memanfaatkan data media sosial *Twitter* sebagai model input.
2. Melakukan deteksi multilabel terhadap data uji dengan menggunakan model klasifikasi multilabelisasi yang telah dibuat.
3. Membuat perangkat lunak antarmuka yang jelas dan mudah dimengerti oleh pengguna perangkat lunak.

1.5 Manfaat

Tugas Akhir ini diharapkan mampu memfasilitasi para dosen wali ataupun staff pengevaluasi universitas sebagai pengguna untuk melakukan deteksi dini risiko kegagalan mahasiswa pada fase evaluasi universitas berdasarkan keluhan-keluhan yang disampaikan mahasiswa dalam media sosial.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Pada tahap ini, akan dicari studi literatur yang relevan untuk dijadikan referensi dalam pengerjaan tugas akhir. Studi literatur dapat diambil dari buku, internet, ataupun materi dalam suatu mata kuliah yang berhubungan dengan metode yang akan digunakan.

2. Analisis dan desain perangkat lunak

Aktor yang menjadi pelaku adalah pengguna perangkat lunak yang dibangun oleh penulis. Kemudian beberapa kebutuhan fungsional dari sistem ini adalah sebagai berikut:

- a. Analisis aktor yang terlibat di dalam sistem.
- b. Perancangan diagram *use case* yang merupakan analisis kebutuhan pada aplikasi yang akan dibangun.
- c. Analisis kebutuhan non-fungsional.

- d. Prapemrosesan data uji
 - e. Uji performa berbagai *classifier*
3. Implementasi perangkat lunak

Pada tahap ini dilakukan pembuatan perangkat lunak berupa model klasifikasi multilabel dengan data latih yang telah digunakan menggunakan *classifier* terbaik berdasarkan uji performa yang sudah dilakukan. Aplikasi juga memiliki sistem prapemrosesan data uji dan memiliki tampilan antarmuka.
 4. Pengujian dan evaluasi

Di tahap ini dilakukan pengujian terhadap perangkat lunak menggunakan data uji final berupa delapan puluh baris *tweet* yang diambil secara acak melalui *Twitter*. Sistem melakukan prapemrosesan pada data uji untuk mengekstrak kata-kata yang relevan untuk dideteksi. Lalu sistem melakukan deteksi multilabel pada data uji dan menyimpan hasilnya ke dalam file (.csv).
 5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

- Bab II Dasar Teori
Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan yang menjadi dasar dari pembuatan Tugas Akhir ini.
- Bab III Analisis dan Perancangan Sistem
Bab ini membahas mengenai perancangan perangkat lunak. Perancandaftagan perangkat lunak meliputi perancangan alur, proses dan perancangan antarmuka pada perangkat lunak.
- Bab IV Implementasi
Bab ini berisi implementasi dari perancangan perangkat lunak perangkat lunak dan implementasi fitur-fitur penunjang perangkat lunak.
- Bab V Pengujian dan Evaluasi
Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-fitur perangkat lunak.
- Bab VI Kesimpulan
Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.
- Bab VII Daftar Pustaka
Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

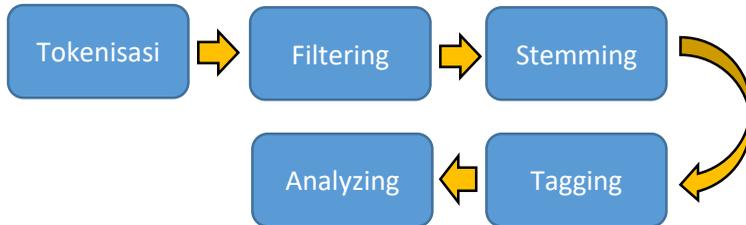
[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 Teks Mining

Teks mining merupakan proses menggali data teks yang didapatkan dari sumber data berupa dokumen (word, pdf, kutipan, dan sejenisnya). Teks mining disebut juga dengan teknik mengekstraksi pola dari sebuah dokumen. Tujuan utama dari teks mining adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen dengan melakukan pencarian kata-kata yang dapat mewakili isi dari sebuah dokumen yang kemudian dapat dianalisa. Kegunaan dari penggunaan teks mining adalah sebagai pengkategorian teks (*text categorization*) dan pengelompokan teks (*text clustering*) [12]. Proses umum dalam teks mining seperti pada Gambar 2.1.



Gambar 2.1 Proses Utama Teks Mining

2.2 NLP (Natural Language Processing)

Natural Language Processing (NLP) adalah cara untuk mengekstrak arti yang lebih lengkap dari representasi teks bebas. NLP umumnya menggunakan konsep linguistik seperti (kata benda, kata kerja, kata sifat, dll) dan struktur gramatikal seperti (Kata ganti preposisi, kata ganti benda dan kata ganti objek).

Perkembangan NLP menghasilkan kemungkinan dari interface bahasa natural menjadi knowledge base dan penterjemahan bahasa natural. Poole dan Mackworth (2010) menjelaskan bahwa ada 3 (tiga) aspek utama pada teori pemahaman mengenai natural language yang pertama adalah syntax untuk menjelaskan bentuk dari bahasa. Syntax biasa dispesifikasikan oleh sebuah grammar. Natural language jauh lebih daripada formal language yang digunakan untuk logika kecerdasan buatan dan program computer. Yang kedua adalah Semantics untuk menjelaskan arti dari kalimat dalam satu bahasa. Meskipun teori semantics secara umum sudah ada, ketika membangun sistem natural language understanding untuk aplikasi tertentu, akan digunakan representasi yang paling sederhana. Yang ketiga adalah pragmatics menjelaskan bagaimana pernyataan yang ada berhubungan dengan dunia. Untuk memahami bahasa, agen harus mempertimbangan lebih dari hanya sekedar kalimat. Agen harus melihat lebih ke dalam konteks kalimat, keadaan dunia, tujuan dari speaker dan listener, konvensi khusus, dan sejenisnya.

2.2.1 Information Retrieval

Information Retrieval (IR) adalah pekerjaan untuk menemukan dokumen yang relevan dengan kebutuhan informasi yang dibutuhkan oleh user. Contoh sistem IR yang paling populer adalah search engine pada World Wide Web. Seorang pengguna Web bisa menginputkan query berupa kata apapun ke dalam sebuah search engine dan melihat hasil dari pencarian yang relevan. Karakteristik dari sebuah sistem IR (Russel & Norvig, 2010) diantaranya adalah:

- A corpus of documents. Setiap sistem harus memutuskan dokumen yang ada akan diperlakukan sebagai apa. Bisa sebagai sebuah paragraf, halaman, atau teks multipage.
- Queries posed in a query language. Sebuah query menjelaskan tentang apa yang user ingin peroleh. Query language dapat berupa list dari kata-kata, atau bisa juga

menspesifikasikan sebuah frase dari kata-kata yang harus berdekatan

- A result set. Ini adalah bagian dari dokumen yang dinilai oleh sistem IR sebagai yang relevan dengan query.
- A presentation of the result set. Maksud dari bagian ini adalah tampilan list judul dokumen yang sudah di ranking.

2.2.2 Morphological Analysis

Proses dimana setiap kata yang berdiri sendiri (individual words) dianalisis kembali ke komponen pembentuk mereka dan token nonword seperti tanda baca dsb dipisahkan dari kata tersebut.

Syntactic analysis harus menggunakan hasil dari morphological analysis untuk membangun sebuah deskripsi yang terstruktur dari kalimat. Hasil akhir dari proses ini adalah yang sering disebut sebagai parsing. Parsing adalah mengkonversikan daftar kata yang berbentuk kalimat ke dalam bentuk struktur yang mendefinisikan unit yang diwakili oleh daftar tadi. Representasi yang deklaratif, yang disebut juga sebagai Grammar, dari fakta sintaktis mengenai bahasa yang digunakan. Procedure, yang disebut juga sebagai Parser, yang membandingkan grammar dengan kalimat yang diinputkan untuk menghasilkan struktur kalimat yang telah di parsing. Cara yang paling umum digunakan untuk merepresentasikan grammar adalah dengan sekumpulan production rule. Rule yang paling pertama bisa diterjemahkan sebagai “Sebuah Sentence terdiri dari sebuah Noun Phrase, diikuti oleh Verb Phrase”, garis vertical adalah OR, sedangkan ϵ mewakili string kosong.

2.2.3 Stemming & Lemmatization

Stemming merupakan sebuah proses yang bertujuan untuk mereduksi jumlah variasi dalam representasi dari sebuah kata (Kowalski, 2011). Resiko dari proses stemming adalah hilangnya informasi dari kata yang di-stem. Hal ini menghasilkan menurun-

nya akurasi atau presisi. Sedangkan untuk keuntungannya adalah, proses stemming bisa meningkatkan kemampuan untuk melakukan recall. Tujuan dari stemming sebenarnya adalah untuk meningkatkan performace dan mengurangi penggunaan resource dari sistem dengan mengurangi jumlah unique word yang harus diakomodasikan oleh sistem. Jadi, secara umum, algoritma stemming mengerjakan transformasi dari sebuah kata menjadi sebuah standar representasi morfologi (yang dikenal sebagai stem).

Proses parsing menggunakan aturan-aturan yang ada pada Grammar, kemudian membandingkannya dengan kalimat yang diinputkan. Struktur paling sederhana dalam melakukan parsing adalah Parse Tree, yang secara sederhana menyimpan rule dan bagaimana mereka dicocokkan satu sama lain. Setiap node pada Parse Tree berhubungan dengan kata yang dimasukkan atau pada nonterminal pada Grammar yang ada. Setiap level pada Parse Tree berkorespondensi dengan penerapan dari satu rule pada Grammar.

2.2.4 InaNLP

InaNLP adalah Natural Language Processing toolkit untuk Bahasa Indonesia yang dapat digunakan untuk memproses formal text atau social text yang berbasis Bahasa Indonesia. Beberapa NLP module sudah terintegrasi kedalam system InaNLP untuk memudahkan orang dalam membangun system NLP berbahasa Indonesia. Toolkit ini berisikan pemisah kata, tokenization, POS (Part of Speech) Tagger, pemotong kata, named entity (NE) tagger, syntactic parser, semantic analyzer dan penormalan kata.

2.2.5 *Part-of-Speech-Tagger* (POS Tagger)

Part-of-Speech-Tagger (POS Tagger) merupakan teknik yang dapat menerapkan penanda tag dari dokumen teks (corpus) dalam beberapa bahasa untuk menjadi kelas kata berdasarkan tipe struktur kalimat dalam dokumen. Struktur kata ini seperti kata benda (*noun*), kata kerja (*verb*), kata sifat (*adjective*) dan

sebagainya. POS Tagger disebut juga *grammatical tagging*. Proses menandai sebuah kata dalam sebuah teks (*corpus*) yang sesuai dengan bagian pembicaraan tertentu. Berdasarkan definisi konteks, yaitu hubungan dengan kata-kata yang berdekatan yang terkait dalam frase, kalimat atau paragraf [28].

POS Tagging pada zaman dahulu dilakukan menggunakan cara manual yaitu dengan menggunakan jasa ahli bahasa untuk memetakan kelas dari kata dalam *corpus*. Untuk saat ini POS Tagging dilakukan dalam konteks komputasi linguistik, menggunakan algoritma yang mengasosiasikan istilah diskrit, serta bagian pembicaraan yang tersembunyi (*latent*). Salah satu pengembangan POS Tagger yang populer saat ini adalah Stanford POS Tagger yang dibangun dalam bahasa Java. Struktur kelas kata POS Tagging digambarkan pada Tabel 2.1 [28].

Tabel 2.1 Struktur Kelas kata POS-Tagging

Tag	Deskripsi	Contoh Penerapan
CC	Coordin, conjunction	And, but, or
CD	Cardinal number	One, two
DT	Determiner	A, the
EX	Existential 'there'	There
FW	Foreign word	Mea culpa
IN	Preposition/sub-conjunction	Of, in, by
JJ	Adjective	Yellow
JJR	Adj., comparative	Bigger
JJS	Adj., superlative	Wildest
LS	List item marker	1, 2, one
MD	Modal	Can, should
NN	Noun, sing. Or mass	Llama
NNS	Noun, plural	Llamas
NNP	Proper noun, sing.	IBM
NNPS	Proper noun, plural	Carolinas
PDT	Predeterminer	All, both
POS	Possessive ending	's
PRP	Personal pronoun	I, you, he

PRP\$	Possessive pronoun	Your, one's
RB	Adverb	Quickly, never
RBR	Adverb, comparative	Faster
RBS	Adverb, superlative	Fastest
RP	Participle	Up, off
SYM	Symbol	, %, &
TO	“to”	to
UH	Interjection	Ah, oops
VB	Verb base form	Eat
VBD	Verb past tense	Ate
VBG	Verb gerund	Eating
VBN	Verb past participle	Eaten
VBP	Verb on-3sg pres	Eat
VBZ	Verb 3-sg pres	Eats
WDT	Wh-determiner	Which, that
WP	Wh-pronoun	What, who
WP\$	Possessive wh-	Whose
WRB	Wh-adverd	How, where
\$	Dollar sign	\$
#	Pound sign	#
“	Left quote	‘ or “
”	Right quote	‘ or “
(Left parenthesis	[, (, {, <
)	Right parenthesis],), }, >
,	Comma	,
.	Sentence-final punc	. ! ?
:	Mid-sentence punc	: ; ... -- -

2.3 MEKA

MEKA project merupakan implementasi open-source untuk metode multi-label learning and evaluation. Dalam melakukan pengklasifikasian multi-label harus dapat mendeteksi multilabel beberapa input dari variable untuk tiap input yang berkaitan.

Perbedaan dari standard binary atau multi-class classification adalah hanya melibatkan satu target variable. MEKA dibuat berdasarkan WEKA machine learning Toolkit yang termasuk beberapa metode multi-label dari literature scientific dan sebagai penghubung terhadap MULAN framework.

Scopus meng-index artikel jurnal akademik dari berbagai bidang ilmu. Maka dari itu banyak publikasi pemeringkatan perguruan tinggi lain seperti Scimago menggunakan jasa dari Scopus sebagai pusat database jurnal akademik. Scopus sendiri sangat berhati-hati dan selektif untuk meng-index artikel jurnal akademik. Tidak semua artikel jurnal di database lain seperti Google Scholar, ataupun Web of Science dapat ter-index oleh Scopus. Scopus hanya meng-index berdasarkan kualitas artikel jurnal tersebut.

2.4 Binary Relevance For Multilabel Classification

Klasifikasi Multilabel (ML) bertujuan untuk mendapatkan model yang dapat menghasilkan label untuk tiap objek, tidak seperti klasifikasi multiclass yang melibatkan prediksi dari hanya sebuah class. Metode pembelajaran ini tumbuh dalam persoalan seperti media dokumen (tulisan, music dan video) yang biasanya terdapat *tag* dengan beberapa label to memberitahu user mengenai isi konten tersebut. Binary Relevane Efficacy adalah metode dekomposisi berdasarakan mempelajari asumsi yang di label secara independen. Selanjutnya, tiap label tersebut di klasifikasikan menjadi relevan dan irelevan oleh binary classifier yang dipelajari secara individu dari label yang lain.

2.5 Naïve Bayes Classifier

NaïveBayes Classifier merupakan sebuah metode klasifikasi yang berakar pada teorema *Bayes*. Metode ini menggunakan metode probabilitas dan statistik yg dikembangkan oleh ilmuwan Inggris Thomas Bayes. Metode ini memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr *Naïve Bayes*

Classifier adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi/kejadian.

Menurut Olson Delen (2008), *Naïve Bayes* menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar mengingat vektor informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek bersifat independen. Probabilitas yang terlibat dalam produksi perkiraan akhir dihitung sebagai jumlah frekuensi master tabel keputusan.

Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan oleh Xhemali, Hinde Stone dalam jurnalnya "*Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages*" yang memaparkan, *Naïve Bayes Classifier* memiliki tingkat akurasi yg lebih baik dibanding model classifier lainnya.

Keuntungan penggunaan *Metode Naïve Bayes Classifier* adalah metoda ini hanya membutuhkan jumlah data latih yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian.

2.6 Java

Java adalah Bahasa pemrograman yang dapat dijalankan pada berbagai perangkat mulai dari computer ataupun telepon genggam. Bahasa ini awalmula dibuat oleh James Gosling saat masih dalam Sun Microsystem yang saat ini merupakan bagian dari Oracle dan dirilis pada tahun 1995. Aplikasi-aplikasi berbasis java umumnya dikompilasi kedalam p-code(bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java adalah Bahasa pemrograman yang bersifat umum/tidak spesifik, dan secara khusus didesain untuk dapat di jalankan di beberapa platform sistem operasi yang berbeda. Ada beberapa kelebihan lain dari java yaitu OOP (Object Oriented Programming – Pemrogram Berorientasi Objek), Library atau Class yang lengkap, lebih familiar karena Bahasa yang digunakan menyerupai C++.

2.6.1 Java Swing

Java swing adalah library java yang digunakan untuk membuat Graphic User Interface (GUI). Dengan menggunakan java swing, user interface yang telah dibuat dapat digunakan pada system cross platform. Swing memberikan tampilan dan nuansa asli yang tidak menampakan platform yang mendasarinya. Swing dapat menampilkan beberapa komponen yang lebih *advance* seperti panel tab, scroll panes, trees, tables dan list.

2.7 MySQL

MySQL adalah Relational Database Management System (RDBMS) dapat digunakan secara gratis dibawah lisensi GPL (General Public License. Dimana semua orang dapat meggunakan MySQL namun tidak dapat dijadikan produk turunan yang bersifat komerisal. SQL (Structured Query Language) adalah sebuah cara untuk mengoperasikan database terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dapat dilakukan dengan mudah dan otomatis.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan Tugas Akhir. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir. Solusi yang ditawarkan oleh penulis juga dicantumkan pada tahap permasalahan analisis ini. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan perangkat lunak klasifikasi multilabel.

3.1.1 Deskripsi Umum

Tugas akhir ini adalah pembangunan perangkat lunak untuk melakukan deteksi multilabel pernyataan keluhan ke dalam empat label keluhan yaitu: keluhan personal, keluhan subjek, keluhan relasi, dan keluhan institusi dari posting media sosial *Twitter* atau biasa disebut *tweet* milik mahasiswa. Aplikasi ini dapat digunakan bagi evaluator atau dosen wali untuk melakukan deteksi dini bagi mahasiswa yang memiliki masalah dan berisiko drop out pada evaluasi tingkat 1 perguruan tinggi. Data keluaran aplikasi ini berupa deteksi multilabel keluhan berdasarkan tweet yang dimasukkan oleh evaluator ataupun dosen wali sebagai pengguna. Data keluaran tersebut digunakan sebagai salah satu gambaran dari kondisi mahasiswa yang memiliki masalah.

3.1.2 Arsitektur Perangkat Lunak

Seluruh proses yang terdapat pada aplikasi klasifikasi multilabel dijalankan pada komputer. Aplikasi menerima data uji

masukkan berupa post media sosial *Twitter* atau disebut *tweet* milik mahasiswa yang telah dipilih dalam format (.csv). Aplikasi deteksi multilabel dibangun dengan menggunakan bahasa pemrograman *Java*. Aplikasi ini akan berjalan pada platform desktop dengan sistem operasi *Microsoft Windows*. Setelah menerima data uji, aplikasi melakukan deteksi klasifikasi multilabel pada data uji. Hasil prediksi i pada *file (.csv)*.

3.1.3 Analisis Aktor

Aktor yang terdapat pada aplikasi deteksi multilabel ini adalah dosen wali atau evaluator akademik mahasiswa di perguruan tinggi.

3.1.4 Analisis Permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana mengimplementasikan suatu aplikasi berbasis desktop untuk memberikan deteksi keberadaan dan jenis keluhan yang terdapat pada data posting *tweet* mahasiswa. Pada perancangan sistem akan dilakukan penentuan jenis *classifier* dengan membandingkan performa beberapa macam kategori *classifier*. Dengan penggunaan *classifier* yang memiliki performa terbaik, sistem diharapkan dapat memberikan deteksi keberadaan dan jenis keluhan yang terdapat pada *tweet* mahasiswa dengan akurat.

3.1.5 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan dalam sistem ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.1

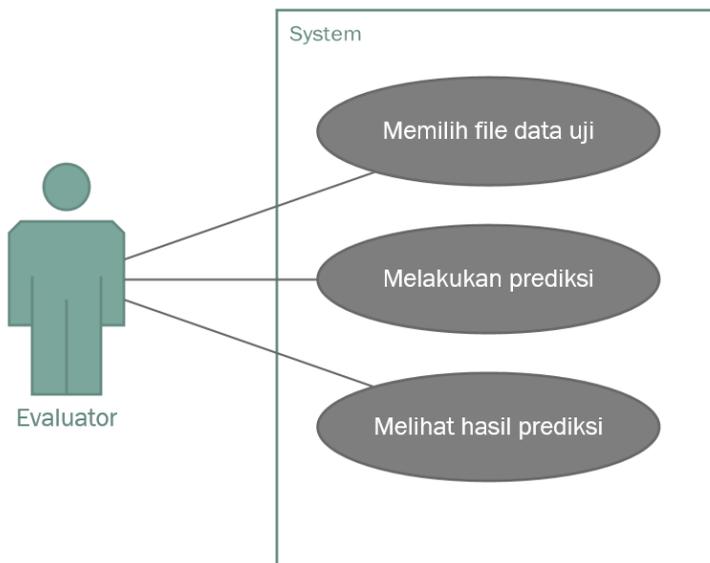
Tabel 3.1 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Memilih file data uji	Aktor mengunggah file yang akan dideteksi oleh sistem dengan cara memilih lokasi file (.csv) yang berisi satu atau lebih baris <i>tweet</i> pada sistem

F-002	Melakukan prediksi	Aktor menginstruksikan sistem untuk melakukan prediksi terhadap data uji yang telah diunggah sebelumnya
F-003	Melihat hasil prediksi	Aktor melihat hasil prediksi multilabel dalam bentuk file (.csv) yang telah dilakukan oleh sistem

3.1.6 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.1 dan kode kasus penggunaan ada pada tabel 3.2.



Gambar 3.1 Diagram Kasus Penggunaan

Tabel 3.2 Keterangan Kode Kasus Penggunaan

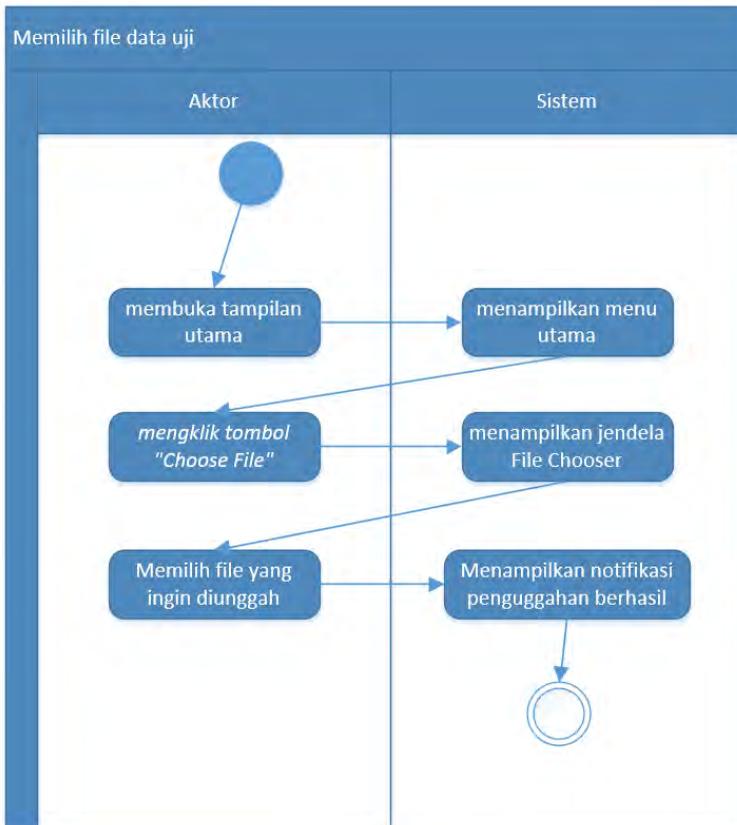
Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Memilih file data uji
UC-002	Melakukan prediksi
UC-003	Melihat hasil prediksi

3.1.6.1 UC-001 Memilih file data test

Tabel 3.3 Spesifikasi Kasus Memilih file data uji

Nama	Memilih file data uji
Nomor	UC-001
Deskripsi	Kasus penggunaan ini digunakan untuk mengunggah data uji dalam bentuk <i>file (.csv)</i>
Tipe	Fungsional
Aktor	Evaluator atau dosen wali
Kondisi Awal	Aktor sudah memasuki tampilan awal aplikasi
Kondisi Akhir	Sistem sukses menerima file unggahan
Alur Normal	<ol style="list-style-type: none"> 1. Aktor membuka tampilan utama aplikasi prediksi multilabel 2. Sistem menampilkan tampilan utama aplikasi 3. Aktor mengklik tombol "<i>Choose file</i>" 4. Sistem menampilkan jendela <i>File Chooser</i> dan memberikan instruksi untuk memilih file yang ingin diunggah 5. Aktor memilih file yang ingin diunggah 6. Sistem menampilkan notifikasi bahwa pengunggahan telah berhasil 7. Kasus penggunaan berakhir
Alur Alternatif	-

Tabel 3.3 menguraikan proses kasus memilih file data uji. Kasus penggunaan ini digunakan untuk mengunggah data uji dalam bentuk file csv. Alur berisi tujuh tahap interaksi antar aktor dan sistem. Tidak ada alur alternatif dari kasus memilih file data uji.



Gambar 3.2 Diagram Aktivitas UC-001

Gambar 3.2 merupakan diagram aktivitas UC-001. Diagram ini menjelaskan alur dari kasus penggunaan memilih file data uji. Tahapan yang dijalankan adalah pertama-tama aktor membuka tampilan utama. Lalu sistem akan memberi tanggapan dengan menampilkan tampilan utama. Kemudian aktor akan meng-klik tombol *Choose File* yang ada pada tampilan utama. Sistem akan memberi respons dengan membuka jendela *filechooser*. Berikutnya aktor memilih *file* yang ingin diunggah melalui jendela *filechooser*.

Sistem memberi respons dengan menampilkan notifikasi *file* berhasil diunggah.

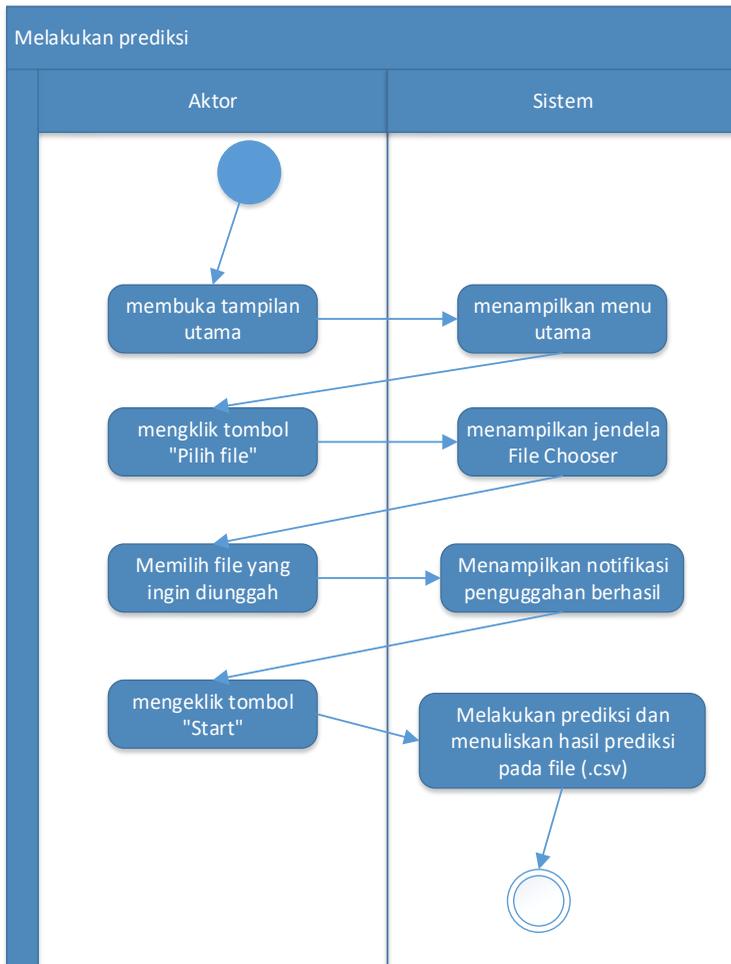
3.1.6.2 UC-002 Melakukan prediksi

Tabel 3.4 Spesifikasi Kasus melakukan prediksi

Nama	Melakukan prediksi
Nomor	UC-002
Deskripsi	Kasus penggunaan ini digunakan untuk memerintahkan sistem melakukan prediksi pada data uji yang sebelumnya telah diunggah.
Tipe	Fungsional
Aktor	Evaluator atau dosen wali
Kondisi Awal	Aktor sudah memasuki tampilan awal aplikasi dan sudah berhasil melakukan kasus penggunaan UC-001
Kondisi Akhir	Sistem menampilkan notifikasi bahwa prediksi berhasil dilakukan dan hasil prediksi telah disimpan dalam <i>file (.csv)</i>
Alur Normal	<ol style="list-style-type: none"> 1. Aktor melakukan kasus penggunaan UC-001 2. Aktor mengklik tombol "<i>Start</i>" 3. Sistem melakukan prediksi dan menuliskan hasil prediksi ke <i>file (.csv)</i> 4. Kasus penggunaan berhasil
Alur Alternatif	-

Tabel 3.4 menguraikan proses kasus melakukan prediksi. Kasus penggunaan ini digunakan untuk memerintahkan sistem untuk melakukan prediksi pada data uji yang sebelumnya telah berhasil diunggah. Alur berisi empat tahap interaksi antar aktor dan sistem. Tidak ada alur alternatif dari kasus memilih file data uji. Kondisi awal dari spesifikasi ini adalah aktor telah membuka tampilan utama perangkat lunak dan telah berhasil melakukan

kasus penggunaan UC-001, yaitu kasus penggunaan untuk melakukan punggahan data latih.



Gambar 3.3 Diagram Aktivitas UC-002

Gambar 3.3 merupakan diagram aktivitas UC-002 yang menjelaskan alur dari kasus penggunaan melakukan prediksi. Aktor harus terlebih dahulu melakukan aktivitas UC-001. Proses

selanjutnya adalah aktor meng-klik tombol *Start* yang ada pada tampilan utama. Sistem akan merespon dengan menampilkan notifikasi berhasil melakukan prediksi.

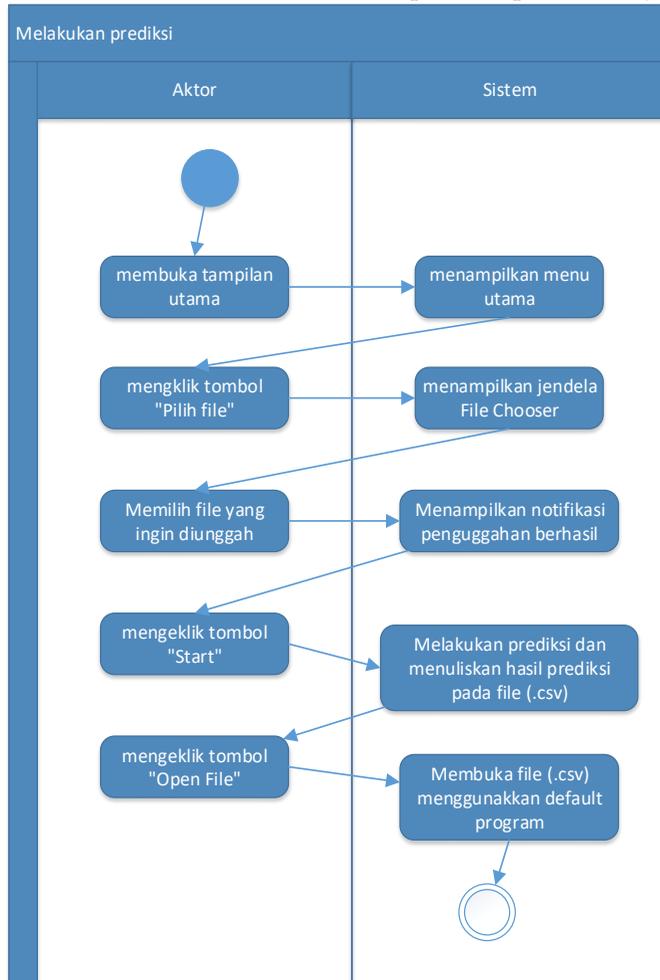
3.1.6.3 UC-003 Melihat hasil prediksi

Tabel 3.5 Spesifikasi Kasus melihat prediksi

Nama	Melihat hasil prediksi
Nomor	UC-003
Deskripsi	Kasus penggunaan ini digunakan untuk membuka hasil prediksi dalam bentuk <i>file (.csv)</i>
Tipe	Fungsional
Aktor	Evaluator atau dosen wali
Kondisi Awal	Aktor sudah memasuki tampilan awal aplikasi dan sudah berhasil melakukan kasus penggunaan UC-001 dan UC-002
Kondisi Akhir	Sistem membuka <i>file</i> hasil prediksi
Alur Normal	<ol style="list-style-type: none"> 1. Aktor melakukan kasus penggunaan UC-001 2. Aktor melakukan kasus penggunaan UC-002 3. Aktor mengeklik tombol "<i>Open file</i>" 4. Sistem membuka <i>file (.csv)</i> hasil prediksi menggunakan <i>default program</i> dari sistem 5. Kasus penggunaan berhasil
Alur Alternatif	-

Tabel 3.5 menguraikan proses kasus melihat hasil prediksi. Kasus penggunaan ini digunakan untuk membuka hasil prediksi yang sudah tersimpan dalam bentuk *file csv*. Alur terdiri dari lima tahap interaksi antar aktor dan sistem. Tidak ada alur alternatif dari kasus memilih file data uji. Kondisi awal dari spesifikasi ini adalah aktor telah membuka tampilan utama perangkat lunak dan telah

berhasil melakukan kasus penggunaan UC-001, yaitu kasus penggunaan untuk melakukan pengunggahan data latih dan kasus penggunaan UC-002, yaitu kasus penggunaan untuk memerintahkan sistem untuk melakukan prediksi pada data uji.



Gambar 3.4 Diagram Aktivitas UC-003

Gambar 3.4 merupakan diagram aktivitas UC-003 yang menjelaskan alur dari kasus penggunaan melihat hasil prediksi. Aktor harus terlebih dahulu melakukan aktivitas UC-001 dan UC-002. Proses selanjutnya adalah aktor meng-klik tombol *Open File* yang ada pada tampilan utama. Sistem akan merespon dengan membukan *file csv* dengan menggunakan *default program* yang ada pada sistem.

3.1.7 Data Uji Pengujian Sistem

Data uji yang digunakan untuk *training* pembuatan model prediksi adalah data posting *Twitter* mahasiswa Universitas Telkom angkatan 2014 dan 2015. Survei perizinan diadakan untuk mendapatkan perizinan penggunaan data *tweet* milik mahasiswa. Data yang diambil adalah *tweet* dari mahasiswa yang memberikan respon positif terhadap survei perizinan. Tabel 3.5 menjabarkan hasil dari survei yang telah dilakukan (pengambilan data dilakukan hingga tanggal 13 Desember 2016 pukul 13.50).

Data diambil menggunakan API Twitter (<https://dev.twitter.com/rest/reference>).

Tabel 3.6 Daftar Respon Survei

Angkatan	Status Jawab
2014	Menjawab : 59
	Mengizinkan : 49
	Tidak mengizinkan : 10
	Tidak menjawab : 5663
2015	Menjawab : 107
	Mengizinkan : 87
	Tidak mengizinkan : 20
	Tidak menjawab : 7025

3.1.8 Penentuan Classifier Latih

Dalam pertimbangan penentuan jenis *classifier* terbaik dilakukan pengujian *Train/test split*, yaitu pengujian performa dari beberapa jenis *classifier* yang disediakan oleh aplikasi *machine*

learning meka. Pengujian dilakukan dengan menggunakan data latih yang telah di praproses. Metode pengujian adalah dengan cara membagi data latih dan data uji dengan rasio 80% (367 data): 20% (92 data) dari keseluruhan 459 data latih. Pengujian dilakukan pada beberapa kategori *classifier* : kategori berbasis Bayes yaitu *classifier Naive Bayes*, kategori berbasis fungsi yaitu *classifier Simple Logistic*, kategori berbasis *Lazy* yaitu *classifier KStar*, kategori berbasis *Rule* yaitu *classifier Desicion Table*, dan kategori berbasis *tree* yaitu *classifier j48*.

3.1.8.1.1 Classifier Basis Bayes – NaiveBayes

== Predictive Performance	
Number of test instances (N)	
Accuracy	0.663
Jaccard index	0.663
Hamming score	0.861
Exact match	0.587
Jaccard distance	0.337
Hamming loss	0.139
ZeroOne loss	0.413
Harmonic score	0.709
One error	0.261
Rank loss	0.128
Avg precision	0.853
Log Loss (lim. L)	0.283
Log Loss (lim. D)	0.364
F1 (micro averaged)	0.727
F1 (macro averaged by example)	0.688
F1 (macro averaged by label)	0.746
AUPRC (macro averaged)	0.79
AUROC (macro averaged)	0.892
Curve Data	
Macro Curve Data	
Micro Curve Data	
Label indices	[0 1 2 3]
Accuracy (per label)	[0.826 0.815 0.826 0.978]
Empty labelvectors (predicted)	0.141
Label cardinality (predicted)	1.022
Levenshtein distance	0.136
== Additional Measurements	
Number of training instances	367
Number of test instances	92
Label cardinality (train set)	1.035
Label cardinality (test set)	1.011
Build Time	0.212
Test Time	0.22
Total Time	0.432

3.1.8.1.2 Classifier Basis Fungsi – SimpleLogistic

```

== Evaluation Info

Classifier                meka.classifiers.multilabel.BR
Options                  [-W,
weka.classifiers.functions.SimpleLogistic, --, -I, 0, -M, 500, -H, 50, -W,
0.0]
Additional Info
Dataset                  filteringResult
Number of labels (L)    4
Type                     ML
Threshold                0.31672460184730994
Verbosity                3

== Predictive Performance

Number of test instances (N)
Accuracy                 0.636
Jaccard index            0.636
Hamming score            0.821
Exact match              0.511
Jaccard distance         0.364
Hamming loss             0.179
ZeroOne loss             0.489
Harmonic score           0.711
One error                 0.261
Rank loss                0.117
Avg precision             0.858
Log Loss (lim. L)        0.297
Log Loss (lim. D)        0.339
F1 (micro averaged)      0.68
F1 (macro averaged by example) 0.678
F1 (macro averaged by label) 0.726
AUPRC (macro averaged)  0.758
AUROC (macro averaged)  0.852
Curve Data
Macro Curve Data
Micro Curve Data
Label indices            [ 0 1 2 3 ]
Accuracy (per label)     [ 0.587 0.837 0.880 0.978 ]
Empty labelvectors (predicted) 0.022
Label cardinality (predicted) 1.228
Levenshtein distance     0.179

== Additional Measurements

Number of training instances 367
Number of test instances     92
Label cardinality (train set) 1.035
Label cardinality (test set) 1.011
Build Time                   2.683
Test Time                    0.007
Total Time                   2.69

```

3.1.8.1.3 Classifier Basis Lazy – KStar

```

== Evaluation Info

Classifier                meka.classifiers.multilabel.BR
Options                  [-W, weka.classifiers.lazy.KStar, --, -B, 20,
-M, a]
Additional Info
Dataset                  filteringResult
Number of labels (L)    4
Type                     ML
Threshold                0.4051813846372896
Verbosity                3

== Predictive Performance

Number of test instances (N)
Accuracy                 0.69
Jaccard index            0.69
Hamming score            0.845
Exact match              0.674
Jaccard distance         0.31
Hamming loss             0.155
ZeroOne loss             0.326
Harmonic score           0.699
One error                 0.293
Rank loss                 0.151
Avg precision             0.829
Log Loss (lim. L)        0.277
Log Loss (lim. D)        0.365
F1 (micro averaged)      0.695
F1 (macro averaged by example) 0.696
F1 (macro averaged by label) 0.715
AUPRC (macro averaged)  0.803
AUROC (macro averaged)  0.878
Curve Data
Macro Curve Data
Micro Curve Data
Label indices            [ 0 1 2 3 ]
Accuracy (per label)     [ 0.750 0.804 0.848 0.978 ]
Empty labelvectors (predicted) 0.022
Label cardinality (predicted) 1.022
Levenshtein distance     0.155

== Additional Measurements

Number of training instances 367
Number of test instances    92
Label cardinality (train set) 1.035
Label cardinality (test set) 1.011
Build Time                  0.014
Test Time                   23.093
Total Time                  23.107

```

3.1.8.1.4 Classifier Basis Rule – DecisionTable

```

== Evaluation Info

Classifier                meka.classifiers.multilabel.BR
Options                  [-W, weka.classifiers.rules.DecisionTable, -
-, -X, 1, -S, weka.attributeSelection.BestFirst -D 1 -N 5]
Additional Info
Dataset                  filteringResult
Number of labels (L)     4
Type                     ML
Threshold                 0.33613445378151263
Verbosity                 3

== Predictive Performance

Number of test instances (N)
Accuracy                  0.56
Jaccard index             0.56
Hamming score             0.78
Exact match               0.435
Jaccard distance          0.44
Hamming loss              0.22
ZeroOne loss              0.565
Harmonic score            0.635
One error                 0.348
Rank loss                 0.175
Avg precision              0.8
Log Loss (lim. L)         0.349
Log Loss (lim. D)         0.407
F1 (micro averaged)       0.609
F1 (macro averaged by example) 0.601
F1 (macro averaged by label) 0.661
AUPRC (macro averaged)   0.621
AUROC (macro averaged)   0.755
Curve Data
Macro Curve Data
Micro Curve Data
Label indices             [ 0 1 2 3 ]
Accuracy (per label)      [ 0.467 0.826 0.848 0.978 ]
Empty labelvectors (predicted) 0.011
Label cardinality (predicted) 1.239
Levenshtein distance      0.22

== Additional Measurements

Number of training instances 367
Number of test instances    92
Label cardinality (train set) 1.035
Label cardinality (test set) 1.011
Build Time                  11.358
Test Time                    0.006
Total Time                   11.364

```

3.1.8.1.5 Classifier Basis Tree – j48

```

j48 trees

== Evaluation Info

Classifier                meka.classifiers.multilabel.BR
Options                  [-W, weka.classifiers.trees.J48, --, -C, 0.25,
-M, 2]
Additional Info
Dataset                  filteringResult
Number of labels (L)    4
Type                     ML
Threshold                0.2875
Verbosity                3

== Predictive Performance

Number of test instances (N)
Accuracy                 0.513
Jaccard index            0.513
Hamming score            0.75
Exact match              0.435
Jaccard distance         0.487
Hamming loss             0.25
ZeroOne loss             0.565
Harmonic score           0.559
One error                 0.446
Rank loss                0.236
Avg precision             0.738
Log Loss (lim. L)        0.429
Log Loss (lim. D)        0.476
F1 (micro averaged)     0.545
F1 (macro averaged by example) 0.54
F1 (macro averaged by label) 0.559
AUPRC (macro averaged)  0.495
AUROC (macro averaged)  0.683
Curve Data
Macro Curve Data
Micro Curve Data
Label indices            [ 0 1 2 3 ]
Accuracy (per label)     [ 0.467 0.783 0.783 0.967 ]
Empty labelvectors (predicted) 0.011
Label cardinality (predicted) 1.185
Levenshtein distance     0.247

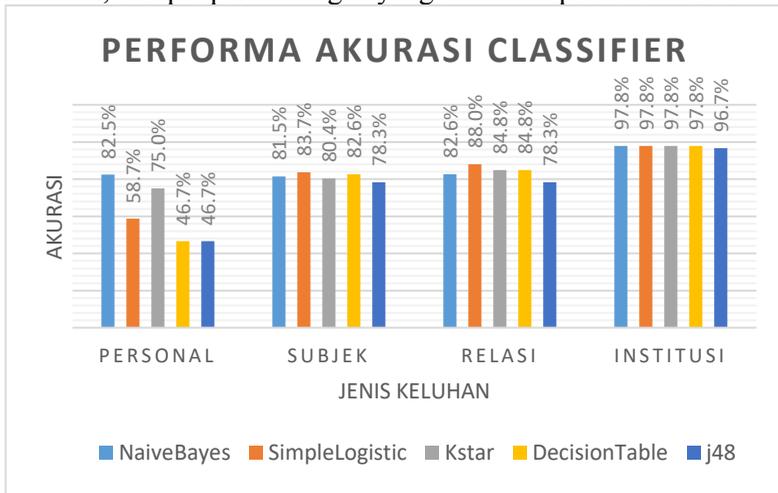
== Additional Measurements

Number of training instances 367
Number of test instances    92
Label cardinality (train set) 1.035
Label cardinality (test set) 1.011
Build Time                  1.279
Test Time                   0.002
Total Time                  1.281

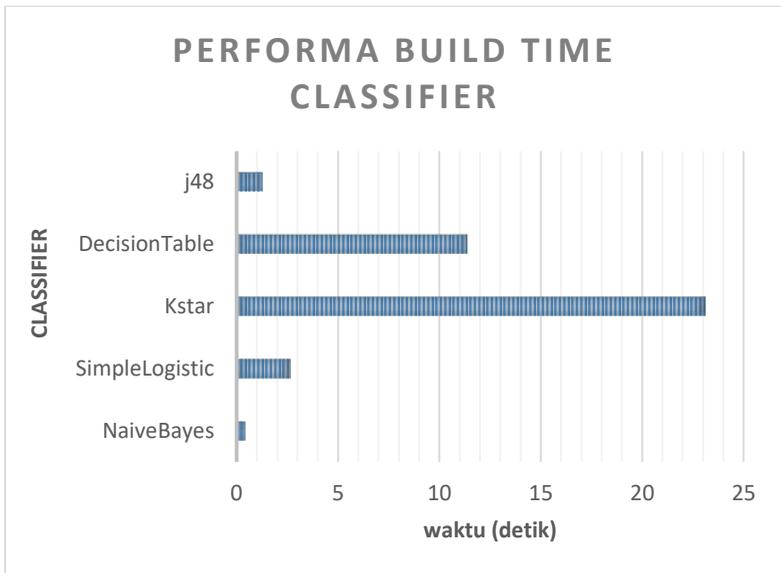
```

Berdasarkan lima pengujian diatas, didapat performa akurasi *classifier NaiveBayes* untuk masing-masing label personal, subjek, relasi, dan institusi bernilai 82.5%, 81,5%, 82,6%, dan 97.8%. Sedangkan akurasi untuk *classifier SimpleLogistic* untuk masing-masing label personal, subjek, relasi, dan institusi bernilai 58.7%, 83.7%, 88%, dan 97.8%. Akurasi untuk *classifier KStar* untuk masing-masing label personal, subjek, relasi, dan institusi bernilai 75%, 80.4%, 84.8%, dan 97.8%. Akurasi untuk *classifier DesicionTable* untuk masing-masing label personal, subjek, relasi, dan institusi bernilai 46.7%, 82.6%, 84.8%, dan 97.8%. Yang terakhir, akurasi untuk *classifier j48* untuk masing-masing label personal, subjek, relasi, dan institusi bernilai 46.7%, 78.3%, 78.3%, 78.3%, dan 96.7%.

Jika dilihat dari aspek penggunaan *resource* dari lamanya waktu yang dibutuhkan dalam proses Train/Test split yang dilakukan, didapat perbandingan yang diuraikan pada Gambar 3.7.



Gambar 3.5 Perbandingan performa akurasi *classifier*



Gambar 3.6 Perbandingan performa *build time classifier*

Dari analisis yang telah dilakukan, didasari dengan faktor akurasi dan efisiensi penggunaan *resource* dapat disimpulkan bahwa classifier dengan performa paling handal adalah *classifier NaiveBayes*.

3.2. Tahap Perancangan Sistem

Tahap ini meliputi perancangan basis data, perancangan tampilan antarmuka (*Interfaces*), prapemrosesan data latih, *training* data latih menggunakan metode *Binary Relevance* dengan *classifier NaiveBayes*, dan prapemrosesan data uji. Diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini.

3.1.1 Perancangan Basis Data

Metode penyimpanan dalam pra-pemrosesan data menggunakan campuran dari basis data relasional MySQL dan file (.arff). Basis data relasional digunakan untuk menampung crawling

raw data dari twitter menggunakan API Twitter, data kriteria hashtag, dan data tweet berdasarkan hashtag. Format file (.arff) digunakan untuk menyimpan data tweet setelah fase pra-pemrosesan bahasa dengan menggunakan inaNLP dan word to word matching dengan Kamus Bahasa Indonesia.

3.1.1.1 Tabel Data Mentah

Identifikasi : raw_data

Deskripsi isi : tabel untuk menyimpan seluruh data yang diambil langsung dari Twitter dengan menggunakan API Twitter

ID Field	Deskripsi	Type	Null	Auto-increment
ID	Variabel primary key	Int	No	Yes
Tweet	Variabel penampung isi tweet	Text	No	No

3.1.1.2 Tabel Hashtag

Identifikasi : hashtag

Deskripsi isi : tabel untuk menyimpan kata hashtag yang digunakan sebagai dasar filtering raw_data

ID Field	Deskripsi	Type	Null	Auto-increment
ID	Variabel primary key	Int	No	Yes
Hashtag	Variabel penampung kandidat hashtag	Text	No	No

3.1.1.3 Tabel Tweet berdasarkan hashtag

Identifikasi : tweet_hashtag

Deskripsi isi : tabel tweet yang sudah disaring berdasarkan hashtag

ID_Field	Deskripsi	Type	Null	Auto-increment
ID	Variabel primary key	Int	No	Yes
Hashtag	Variabel penampung isi hashtag	Text	No	No
Tweet	Variabel penampung isi tweet	Text	No	No

3.1.1.4 Tabel Tweet Berlabel

Identifikasi : tweet_label

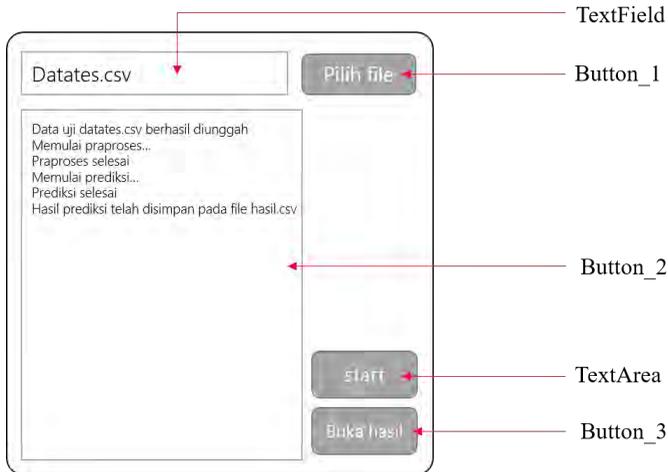
Deskripsi isi : tabel untuk menyimpan tweet yang sudah diseleksi dan diberi label keluhan

ID_Field	Deskripsi	Type	Null	Auto-increment
ID	Variabel primary key	Int	No	Yes
Tweet	Variabel penampung isi tweet	Text	No	No
Keluhan_personal	Variabel indikator keluhan personal	Boolean	No	No
Keluhan_subjek	Variabel indikator keluhan subjek	Boolean	No	No
Keluhan_relasi	Variabel indikator keluhan relasi	Boolean	No	No
Keluhan_institusi	Variabel indikator keluhan institusi	Boolean	No	No

3.1.2 Perancangan Antarmuka

Perangkat lunak dibangun dalam basis aplikasi desktop. Antarmuka perangkat lunak dibangun dengan menggunakan bahasa pemrograman Java menggunakan library Java Swing. Berikut rincian antarmuka yang dibangun.

3.1.2.1 Antarmuka Home



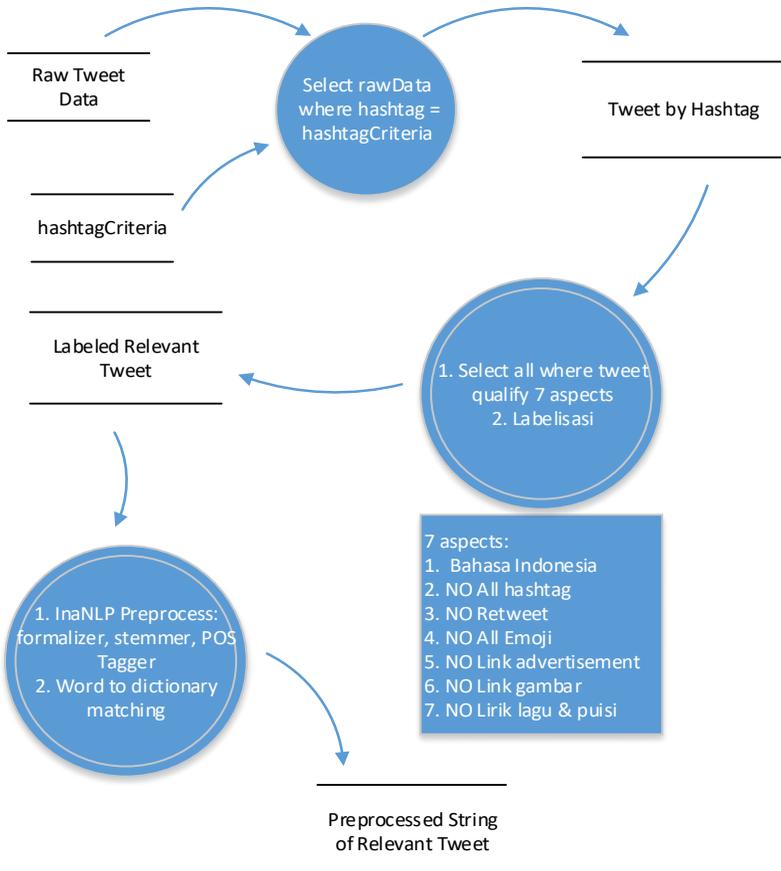
Gambar 3.7 Rancangan antarmuka sistem

Identifikasi: antarmuka home

Deskripsi isi: antarmuka ini berfungsi sebagai halaman utama perangkat lunak. Semua kebutuhan fungsional di sistem dioperasikan pada halaman utama ini. Komponen yang digunakan dalam antarmuka home adalah:

- TextField berguna untuk menampilkan nama data uji yang akan dideteksi,
- Button_1 adalah tombol yang digunakan untuk memilih dan mengunggah *file* (.csv) data uji,
- Button_2 adalah tombol yang digunakan untuk memerintah sistem untuk melakukan prediksi pada data uji yang telah diunggah
- Button_3 adalah tombol yang digunakan untuk membuka *file* hasil dari prediksi yang telah dilakukan oleh sistem, dan
- TextArea adalah area teks yang berguna untuk menampilkan status dari proses yang dilakukan sistem.

3.1.3 Pengolahan Data Latih



Gambar 3.8 Data Flow Diagram prapemrosesan Prapemrosesan data latih diuraikan pada *Data Flow Diagram* pada Gambar 3.8 yang terdiri dari: pengelompokan berdasarkan *hashtag*, eliminasi *tweet* non relevan, labelisasi data latih, *Formalizer*, *Stemmer*, dan *POS Tagger*, dan penyimpanan format ARFF.

Poin-poin berikut ini menjelaskan masing-masing prapemrosesan:

3.1.3.1 Pengelompokan Berdasarkan Hashtag

Seluruh data latih *tweet* yang didapat menggunakan API Twitter disaring berdasarkan hashtag yang relevan. Hal ini dilakukan untuk mempersempit topik dari jenis *tweet* yang berkategori keluhan yang dialami mahasiswa. Jenis *hashtag* diseleksi berdasarkan kata atau frasa yang memiliki hubungan erat dengan bentuk keluhan yang biasa disampaikan oleh mahasiswa. *Hashtag* yang digunakan diuraikan pada Tabel 3.7.

Tabel 3.7 Daftar hashtag

No	Hashtag	28	#dead
1	#penat	29	#deadlinetugas
2	#algorithm	30	#demam
3	#algoritma	31	#deritamahasiswa
4	#amarah	32	#dikejardeadline
5	#anakteknik	33	#dosa
6	#apes	34	#dosenselalubenar
7	#badmood	35	#ego
8	#banyakpikiran	36	#emosi
9	#banyaktugas	37	#fluberat
10	#baper	38	#fuck
11	#belajar	39	#gabisatidur
12	#besokujian	40	#gagalmoveon
13	#bete	41	#gakadakerjaan
14	#bimbang	42	#jenuh
15	#bingung	43	#koding
16	#bored	44	#lelah
17	#boring	45	#mumet

18	#bosan	46	#nasibmahasiswa
19	#brokenheart	47	#nguli
20	#butuhhiburan	48	#pusing
21	#butuhpiknik	49	#resikokuliah
22	#capek	50	#sakit
23	#cemburu	51	#salahjurusan
24	#coding	52	#suntuk
25	#cumaiseng	53	#tugas
26	#curhat	54	#tugasgila
27	#dampaktugas	55	#tugasmenumpuk

3.1.3.2 Eliminasi tweet non relevan

Pada tahap ini data uji *tweet* yang tidak relevan dieliminasi. Kualifikasi data didasari pada aspek-aspek yang dipaparkan Tabel 3.8. Seleksi berdasarkan tujuh aspek ini berfungsi untuk menyaring dan mengeliminasi data *tweet* berisi *spam* yang tidak relevan dengan pesan keluhan yang disampaikan oleh mahasiswa.

Tabel 3.8 Daftar aspek seleksi

Nomor	Aspek kualifikasi
1	Berbahasa asing
2	Hanya berisi hashtag
3	Berisi retweet
4	Berisi emotikon ataupun emoji
5	Berisi iklan dan promosi produk
6	Berisi gambar
7	Berisi kutipan lirik lagu ataupun puisi

3.1.3.3 Labelisasi data latih

Data yang berhasil lolos seleksi akan diklasifikasikan dengan berdasarkan empat label keluhan:

- a. keluhan personal
- b. keluhan subjek
- c. keluhan relasi
- d. keluhan institusi

Klasifikasi yang dilakukan adalah klasifikasi multilabel, yaitu klasifikasi dengan cara menentukan masing-masing label dengan nilai biner terhadap setiap *data instance*. Setiap *data instance* bisa memiliki satu atau beberapa label keluhan sekaligus.

3.1.3.4 *inaNLP Formalizer, Stemmer, dan POS Tagger dan word to word dictionary matching*

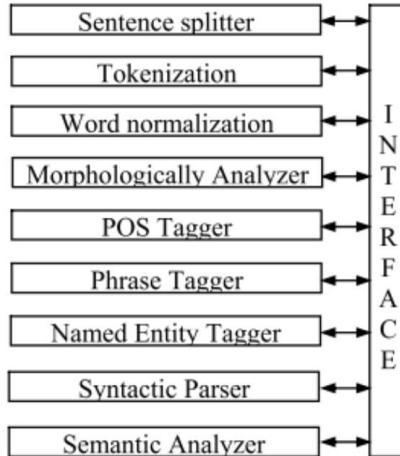
Di dalam domain bahasa Indonesia, penelitian mengenai ekstraksi permasalahan mahasiswa dengan data Twitter belum ada. Terdapat beberapa permasalahan yang ada didalam data Twitter yaitu tidak bakunya bahasa yang digunakan di dalam Twitter, penggunaan emoticon dan smmiley, penggunaan singkatan kata, dsb. Penelitian “*Sentiment classification for Indonesian message in social media*” oleh A. R. Naradhipa and A. Purwarianti mengawali preproses pada data media sosial dengan domain bahasa Indonesia dengan memberikan beberapa tahap preproses sebagai berikut:

1. Penggunaan tanda baca yang tidak standar
2. Adanya konversi angka ke huruf (bahasa alay)
3. Penyalahgunaan kata (perulangan kata untuk penekanan)
4. Penyingkatan kata yang tidak standar

Beberapa teknik preprosesing yang diusulkan untuk mengatasi variasi ketidakbakuan umum dalam Twitter berbahasa Indonesia adalah pembersihan angka, konversi angka ke huruf, menghilangkan perulangan, dan penerjemahan bahasa *alay* menjadi bahasa baku.

Salah satu output dari penelitian terkait *Natural Language Processing* dengan domain bahasa Indonesia adalah tools *inaNLP*. *inaNLP* menyediakan berbagai *library* untuk pemrosesan NLP di dalam domain bahasa Indonesia. Gambar 3.9 menunjukkan arsitektur dari *inaNLP* yang terdiri dari *sentence splitter*, *tokenization*, *word normalization*, *morphologically analyzer*, *pos*

tagger, phrase tagger, named entity tagger, syntactic parser, serta semantic analyzer.



Gambar 3.9 Diagram inaNLP

Tabel 3.9 Deskripsi dari masing-masing proses pengolahan

Nama Proses	Penjelasan
inaNLP Formalizer	Formalisasi atau perbaikan dari kata maupun kalimat yang memiliki typo untuk diganti ke kata dengan kemiripan tertinggi. Jika tidak terdapat kata dengan kemiripan yang cukup maka kata dianggap kata asing dan tidak diolah.
inaNLP Stemmer	Ekstraksi kata dasar dari kata berimbuhan.
inaNLP POS Tagger	Klasifikasi kata ke berbagai kategori jenis kata.
Dictionary matching	Pencocokan masing-masing kata ke daftar kata dalam Kamus Besar Bahasa Indonesia.

Data Tweet diproses dengan inaNLP Formalizer sebagai perbaikan kata-kata *typo* dan penggunaan singkatan pada kalimat. inaNLP Formalizer juga berfungsi untuk melakukan perbaikan terhadap bahasa alay dan menerjemahkannya ke kata baku.

Tabel 3.10 Daftar imbuhan yang berelevansi tinggi

Imbuhan	Kata ganti
-nya	dia
-ku	aku
-mu	kamu

Tahap berikutnya adalah ekstraksi kata dasar dengan inaNLP Stemmer yang berfungsi untuk mengambil kata dasar dari kata berimbuhan. Dilakukannya ekstraksi kata dasar akan berakibat pada tidak terikutsertakannya kata-kata imbuhan, padahal kata imbuhan kepemilikan berdasarkan tabel 3.10 memiliki bobot besar pada pengaruh keberadaan keluhan pada kalimat, khususnya keluhan relasi. Oleh karena itu, inaNLP possessivepronounsuffix digunakan untuk mendeteksi adanya imbuhan-imbuhan tersebut.

Tahap selanjutnya adalah ekstraksi kata-kata yang memiliki bobot peranan dan relevansi yang tinggi menggunakan inaNLP POS Tagger berdasarkan tabel 3.11.

Tabel 3.11 Daftar POS Tagger yang digunakan

POS Tagger	Arti	Contoh
VBI	Kata kerja intransitif	Duduk, Tidur
VBT	Kata kerja transitif	Membeli
JJ	Kata Sifat	Baik, Bagus
NN	Kata Benda	Meja, Kulkas

Pada fase ini ditemui masalah akurasi dari POS Tagger inaNLP. Penyaringan kata benda (NN) dari inaNLP seringkali meloloskan kata-kata yang tidak baku dan tidak ada pada Kamus Besar Bahasa Indonesia. Oleh karena itu diperlukan adanya pencocokan kata per kata dengan daftar kata dalam Kamus Besar

Bahasa Indonesia (<http://www.indodic.com/IndoWordList.zip>) dan melakukan eliminasi kata-kata yang tidak ada dalam Kamus Besar Bahasa Indonesia.

```

Num i = 0
WHILE dictionaryWord hasNextLine
    IF inWord EQUALS dictionaryWord[i]
        SAVE inWord INTO dataTrain.arff
    ENDIF
    i++
ENDWHILE

```

3.1.3.5 Penyimpanan ke format (.arff)

Kumpulan kata-kata yang sudah melewati tahap pre-pemrosesan disimpan ke bentuk file (.arff). ARFF (*Attribute-Relation File Format*) merupakan file teks ASCII yang menggambarkan kumpulan dari *data instance* yang berbagi kumpulan atribut. ARFF dikembangkan oleh *Machine Learning Project* dari *Department of Computer Science of The University of Waikato* untuk digunakan pada *Weka machine learning software*. ARFF memiliki dua bagian yaitu *header* dan *data*. *Header* dari ARFF berisi nama dari *relation* dan kumpulan *attribute* beserta tipe datanya. Sedangkan *data* berisi nilai dari setiap atribut pada masing-masing *data instance*.

Tabel 3.12 Contoh data latih multilabel

Message	Personal	Subjek	Relasi	Institusi
pusing pikir ujian matematika	1	1	0	0
bosan seharian hujan penat banget	1	0	0	0
depresi kamu tinggal aku	0	0	1	0
dosen terlambat	0	0	0	1

```

%
%Example of ARFF file
%
@RELATION dataLatih
@ATTRIBUTE Personal      NUMERIC
@ATTRIBUTE Subjek       NUMERIC
@ATTRIBUTE Relasi       NUMERIC
@ATTRIBUTE Institusi    NUMERIC
@ATTRIBUTE Message      STRING
@DATA
1, 1, 0, 0, 'pusing pikir ujian matematika'
1, 0, 0, 0, 'bosan seharian hujan penat banget'
0, 0, 1, 0, 'depresi kamu tinggal aku'
0, 0, 0, 1, 'dosen terlambat'

```

Gambar 3.10 Representasi data latihan pada format *arff*.

3.1.3.6 Konversi String to Vector

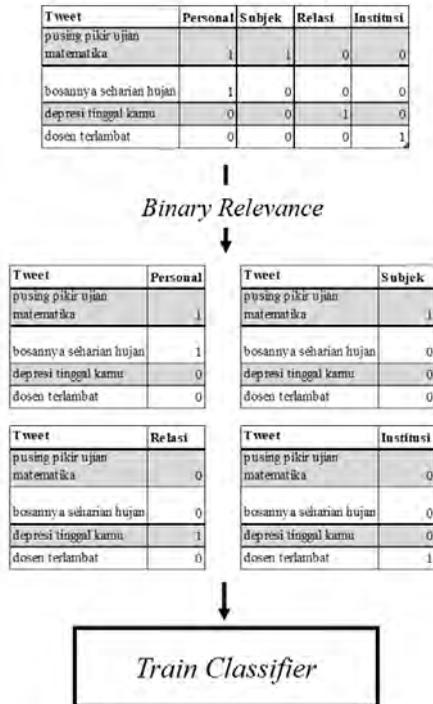
String to Vector merupakan metode konversi atribut yang berbentuk kalimat menjadi atribut kata. Konversi *string to vector* digunakan sebagai *input* dari *classifier NaïveBayes* yang menggunakan atribut yang independen. Konversi dilakukan pada data latihan yang telah di preproses dan disimpan dalam format *arff*. Konversi dilakukan dengan cara mengumpulkan setiap kata dari dalam data dan mengasumsikan masing-masing kata merupakan atribut yang terpisah. Atribut-atribut tersebut kemudian diurutkan berdasarkan abjad dan dituliskan pada *header file arff*. Pada kolom data, susunan masing-masing kata pada suatu baris diurutkan berdasarkan abjad dan direpresentasikan dengan menggunakan indeks urutan dari atribut yang terdapat pada *header file*. Lalu dalam satu baris data yang sama tidak diperbolehkan terdapat pengulangan kata. Jika terdapat pengulangan kata maka hanya satu kata yang digunakan.

3.1.4 Pelatihan data latihan menggunakan *Naïve Bayes classifier*

Data *tweet* yang dilakukan praproses akan dilatih dengan menggunakan metode transformasi multilabelisasi *Binary Relevance* dengan *train classifier NaiveBayes* sebagai untuk membangun model multilabel keluhan permasalahan mahasiswa.

3.1.4.1 *Tranformasi Binary Relevance*

Binary Relevance (BR) melakukan transformasi data permasalahan multilabel menjadi permasalahan satu label. Data hasil transformasi yang sudah berbentuk satu label baru akan dilatih dengan menggunakan classifier yang ditentukan. Proses BR berfungsi untuk mengadaptasi data latih pada algoritma klasifikasi NaiveBayes.



Gambar 3.11 Penerapan *Binary Classifier*

Gambar 3.13 menggambarkan penerapan *binary classifier* yang dilakukan pada contoh data latih yang terdapat pada Tabel 3.12.

3.1.4.2 *Pelatihan menggunakan classifier NaïveBayes*

Tahapan dari algoritma *NaïveBayes* adalah:

- a. Menentukan jumlah label
- b. Menghitung jumlah kasus per label
- c. Mengalikan semua variabel label
- d. Membandingkan hasil per kelas

$$P(c|x) = \frac{P(x|c).P(c)}{P(x)} \dots\dots\dots(1)$$

Teorema *Bayes* diuraikan oleh rumus 1 yaitu peluang masuknya karakteristik tertentu dalam kelas *c* (*posterior* pada rumus 2) adalah peluang munculnya kelas *c* (sebelum masuknya sampel tersebut disebut *prior* pada rumus 2) dikalikan dengan peluang kemunculan karakteristik sampel pada kelas *c* (*likelihood* pada rumus 2) dibagi dengan peluang kemunculan karakteristik sampel secara global (*evidence* pada rumus 2).

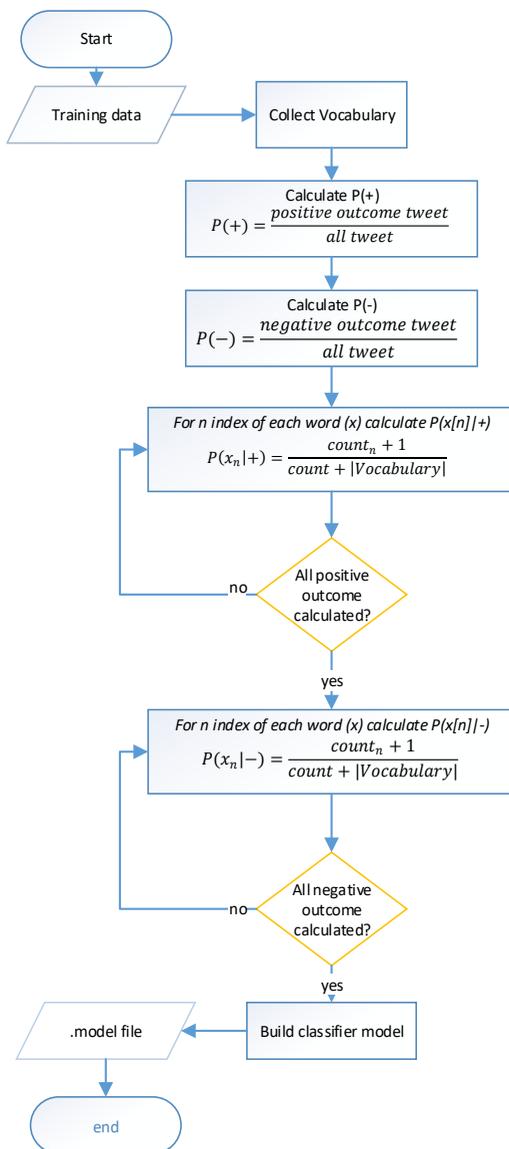
$$posterior = \frac{prior \times likelihood}{evidence} \dots\dots\dots(2)$$

Nilai *evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai *posterior* kelas lainnya untuk menentukan kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut dari rumus (1) digambarkan pada bentuk (3).

$$p(c, x_1, \dots, x_n) \dots\dots\dots(3)$$

Hasil penjabaran bentuk (3) bersifat kompleks dan menyulitkan proses analisis secara satu persatu yang berakibat pada sulitnya perhitungan yang dilakukan. Hal itu diatasi dengan mengasumsikan independensi (*naïve*) dari masing-masing atribut yang terlibat antara satu sama lain. Asumsi tersebut menghasilkan persamaan yang digambarkan pada rumus (4).

$$p(c, x_1, \dots, x_n) = P(c) \prod_{i=1}^n P(x_i|c) \dots\dots\dots(4)$$



Gambar 3.12 Flowchart Pelatihan menggunakan *Naïve Bayes*

Gambar 3.11 merupakan flowchart proses pembangunan model *classifier Naïve Bayes*. Pertama-tama dilakukan proses *collect vocabulary* pada data latih contoh yang terdapat pada Tabel 3.12. Dari proses tersebut diperoleh lima belas jumlah kata. Kata-kata terdiri dari:

{“aku”, “bosan”, “banget”, “depresi”, “dosen”, “hujan”, “kamu”, “matematika”, “penat”, “pikir”, “pusing”, “sehariannya”, “terlambat”, “tinggal”, “ujian”}

Tahap berikutnya adalah menghitung nilai $P(+)$ dan $P(-)$. Nilai-nilai tersebut dijabarkan dalam rumus berikut:

$$P(+)=\frac{\text{positive outcome tweet}}{\text{all tweet}}\dots\dots\dots(5)$$

$$P(-)=\frac{\text{negative outcome tweet}}{\text{all tweet}}\dots\dots\dots(6)$$

Dengan menggunakan transformasi *binary classifier*, proses pelatihan multilabel dilakukan secara independen pada masing-masing labelnya. Pada perhitungan ini dilakukan pelatihan pada label keluhan personal sebagai contoh.

$P(+)$ merupakan nilai probabilitas dari keluhan personal yang bernilai positif. Berdasarkan tabel 3.12 *tweet* yang positif memiliki keluhan personal berjumlah dua *tweet* dari total data latih berjumlah empat *tweet*. Maka berdasarkan rumus (5) nilai dari $P(+)$ adalah 2 (dua) *tweet* positif keluhan dibagi dengan 4 (empat) jumlah keseluruhan data latih. Maka $P(+)$ bernilai 0,5.

Sedangkan $P(-)$ adalah nilai probabilitas dari keluhan personal yang bernilai negatif. Berdasarkan tabel 3.12 dengan menggunakan rumus (6) $P(-) = 2 \div 4$ yang menghasilkan nilai 0,5.

Tahap berikutnya adalah melakukan pembobotan ke masing-masing kata. Pembobotan dilakukan menjadi fase *positive outcome* dan pembobotan fase *negative outcome*.

Tabel 3.13 Contoh data latih multilabel

<i>X</i>	Tweet_1	Tweet_2	Tweet_3	Tweet_4
aku				
bosan		+		
banget				
depresi			-	
dosen				-
hujan		+		
kamu			-	
matematika	+			
penat	+			
pikir	+			
pusing	+			
seharian		+		
terlambat				-
tinggal			-	
ujian	+			
Keluhan personal	1	1	0	0

Tabel 3.13 merupakan bentuk dari independensi kata per kata dari data *tweet*. Tanda (+) merepresentasikan kata yang merupakan atribut dari *tweet* yang positif memiliki keluhan personal. Sedangkan tanda (-) merepresentasikan kata yang merupakan atribut dari *tweet* yang negatif memiliki keluhan personal.

Tahap selanjutnya adalah perhitungan pembobotan ke masing-masing *vocabulary* kata yang berjumlah lima belas dengan

menggunakan rumus (7) untuk positif keluhan (*positive outcome*) dan rumus (8) untuk negatif keluhan (*negative outcome*). Rumus tersebut merupakan pengembangan *Naïve Bayes* yang dijuluki *Multinomial Naïve Bayes*.

$$P(x_n|+) = \frac{\text{count}_{n+1}}{\text{count}(+)+|\text{Vocabulary}|} \dots \dots \dots (7)$$

Tabel 3.14 Pembobotan *positive outcome*

P(word +)	$P(x_n +)$
P(aku +)	0.043
P(bosan +)	0.087
P(banget +)	0.043
P(depresi +)	0.043
P(dosen +)	0.043
P(hujan +)	0.087
P(kamu +)	0.043
P(matematika +)	0.087
P(penat +)	0.087
P(pikir +)	0.087
P(pusing +)	0.087
P(seharian +)	0.087
P(terlambat +)	0.043
P(tinggal +)	0.043
P(ujian +)	0.087

Tabel 3.14 merupakan hasil perhitungan pembobotan dari masing-masing kata terhadap *positive outcome*. count_n adalah jumlah dari kata indeks ke n yang bernilai positif (+). Sedangkan $\text{count}(+)$ merupakan jumlah kata yang terdapat pada label positif (+) yakni berjumlah delapan. *Vocabulary* adalah jumlah seluruh kata yang terdapat dalam data latih yakni berjumlah lima belas. Masing-masing kata dilakukan pembobotan satu persatu secara

iteratif. Jika seluruh kata telah diberi bobot maka dilanjutkan ke proses berikutnya yaitu pembobotan terhadap *negative outcome*.

$$P(x_n|-) = \frac{count_n+1}{count(-)+|Vocabulary|} \dots \dots \dots (8)$$

Tabel 3.15 Pembobotan *negative outcome*

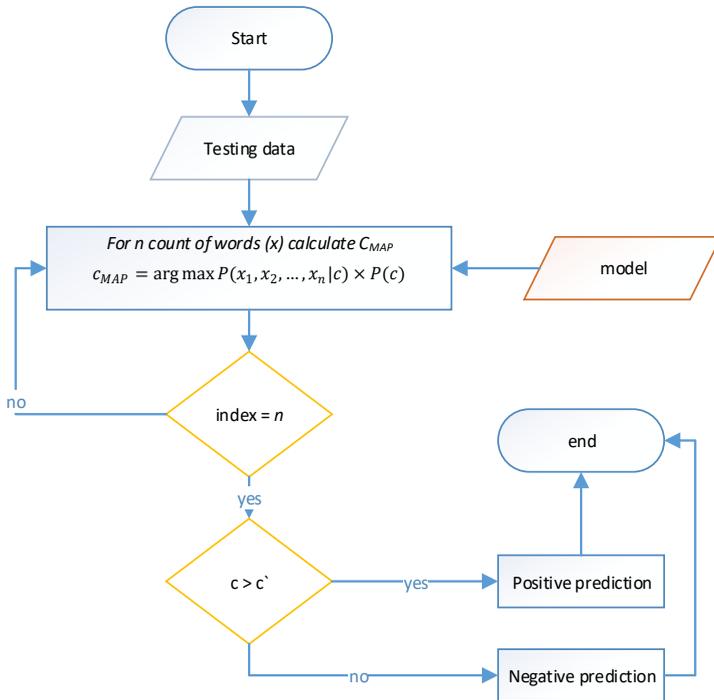
P(word -)	P(x_n -)
P(aku -)	0.1
P(bosan -)	0.05
P(banget -)	0.1
P(depresi -)	0.1
P(dosen -)	0.1
P(hujan -)	0.05
P(kamu -)	0.1
P(matematika -)	0.05
P(penat -)	0.05
P(pikir -)	0.05
P(pusing -)	0.05
P(seharian -)	0.05
P(terlambat -)	0.1
P(tinggal -)	0.1
P(ujian -)	0.05

Tabel 3.15 merupakan hasil perhitungan pembobotan dari masing-masing kata terhadap *negative outcome*. $count(-)$ memiliki nilai lima dan *Vocabulary* berjumlah lima belas.

Proses berikutnya adalah melakukan pembobotan terhadap masing-masing kategori label subjek, relasi, dan insititusi dengan menggunakan metode yang sama. Kemudian nilai-nilai $P(+)$, $P(-)$, $P(x_n|+)$, dan $P(x_n|-)$ yang telah didapat disimpan dalam bentuk file .model.

3.1.5 Prediksi dan hasil

Prediksi *classifier Naïve Bayes* menggunakan rumus (9) yaitu titik dimana fungsi memiliki nilai tertinggi (*arg max*) dari fungsi probabilitas *outcome* dari masing-masing atribut ($P(x_1, x_2, \dots, x_n|c)$) dikalikan dengan probabilitas *outcome* ($P(c)$)
 $c_{MAP} = \text{arg max } P(x_1, x_2, \dots, x_n|c) \times P(c) \dots \dots \dots (9)$



Gambar 3.13 Flowchart pengujian *Naïve Bayes*

Sebagai contoh berikut merupakan kalimat yang akan dideteksi label keluhan personal:

“Aku pusing dan depresi pikir kamu”

Tahap pertama adalah mencari nilai untuk *outcome* positif.

$$c = P(x_1, x_2, \dots, x_n|c) \times P(c)$$

$$c = P(\text{aku}|c) \times P(\text{pusing}|c) \times P(\text{depresi}|c) \times P(\text{pikir}|c) \times P(\text{kamu}|c) \times P(c)$$

$$c = 0.043 \times 0.087 \times 0.043 \times 0.087 \times 0.043 \times 0.5$$

$$c = 3.008 \times 10^{-7}$$

Tahap berikutnya adalah mencari nilai untuk *outcome* negatif.

$$c' = P(x_1, x_2, \dots, x_n|c') \times P(c')$$

$$c' = P(\text{aku}|c') \times P(\text{pusing}|c') \times P(\text{depresi}|c') \times P(\text{pikir}|c') \times P(\text{kamu}|c') \times P(c')$$

$$c' = 0.1 \times 0.05 \times 0.1 \times 0.05 \times 0.1 \times 0.5$$

$$c' = 1.25 \times 10^{-6}$$

Nilai *outcome* positif kurang dari nilai *outcome* negative sehingga maka hasil deteksi multilabel keluhan personal adalah tidak terdeteksi keluhan.

Tahapan berikutnya adalah melakukan prediksi terhadap masing-masing label keluhan subjek, relasi, dan institusi dengan metode yang sama. Setelah prediksi berhasil dilakukan, prediksi akan dituliskan sistem pada *file csv*. Perangkat lunak menyiapkan fitur untuk langsung membuka file hasil prediksi dengan meng-klik tombol *Open File* pada tampilan utama

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem deteksi multilabel. Berikut uraian dari kode sumber sistem. Implementasi perangkat lunak dibedakan menjadi beberapa subbab, yaitu lingkungan implementasi, tahap prapemrosesan dan labelisasi, tahap data training, dan implementasi antarmuka.

4.1 Lingkungan Implementasi

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. *Database* yang digunakan pada server adalah MySQL.
2. 1 PC untuk pengembangan *User* menggunakan Intel i5-4460 4 Cores x 3.2 GHZ
3. 1 PC untuk uji coba menggunakan Intel i5-4460 4 Cores x 3.2 GHZ
4. Microsoft Visio untuk pembuatan diagram, Sublime sebagai teks editor.
5. Eclipse Oxygen sebagai compiler pemrograman Java
6. Microsoft Excel untuk proses labelisasi.

4.2 Tahap Implementasi

Pada bagian ini terdapat beberapa subproses di dalamnya yaitu tahap pelatihan data latih, tahap prapemrosesan data uji, dan tahap deteksi multilabel data latih.

4.2.1 Tahap Pelatihan Data Latih

Implementasi pelatihan data latih menggunakan metode *Binary Relevance* yang merupakan metode transformasi multilabelisasi. Meka telah mengembangkan library untuk melakukan transformasi *Binary Relevance* dalam bahasa pemrograman Java. Implementasi kode program dituliskan pada gambar 4.1.

Pembangunan model *Naive Bayes* dan transformasi *Binary Relevance* dibuat menggunakan bantuan tools *meke*.

```

public class TrainTestSet {

    public static void main(String[] args) throws Exception {
        if (args.length != 2)
            throw new IllegalArgumentException("Required arguments: <train> <test>");

        System.out.println("Loading train: " + args[0]);
        Instances train = DataSource.read(args[0]);
        MLUtils.prepareData(train);

        System.out.println("Loading test: " + args[1]);
        Instances test = DataSource.read(args[1]);
        MLUtils.prepareData(test);

        // compatible?
        String msg = train.equalHeadersMsg(test);
        if (msg != null)
            throw new IllegalStateException(msg);

        System.out.println("Build BR classifier on " + args[0]);
        BR classifier = new BR();
        // further configuration of classifier
        NaiveBayes myModel = new NaiveBayes();
        classifier.setClassifier(myModel);
        classifier.buildClassifier(train);

        System.out.println("Evaluate BR classifier on " + args[1]);
        String top = "PCut1";
        String vop = "3";
        Result result = Evaluation.evaluateModel(classifier, train, test, top, vop);
    }
}

```

Gambar 4.1 Kode pelatihan data latih

4.2.2 Tahap Prapemrosesan Data Uji

Aplikasi akan membaca data uji yang berbentuk file .txt yang sebelumnya sudah dipilih oleh aktor. Setelah file data uji sukses dipilih, aktor akan menekan tombol start dan program akan melakukan proses preprosesing kepada file data.

4.2.2.1 *Membersihkan symbol dari teks*

Tahapan preprosesing pertama adalah membersihkan karakter-karakter simbol dan tanda baca untuk memudahkan proses klasifikasi. Implementasi kode diuraikan pada Gambar 4.2.

```

private String cleanSymbol(String text) {
    String escaped = text.replaceAll("[\\<\\(\\[\\{\\|\\^\\-\\
\\=\\$\\!\\|\\/\\]\\\\\\\\)\\\\?\\\\*\\\\+\\\\.\\\\>0123456789,?@#%&]", " ");
    escaped = escaped.replace(" ", " ");
    escaped = escaped.trim();
    escaped = escaped.toLowerCase();
    return escaped;
}

```

Gambar 4.2 Kode pembersihan simbol

4.2.2.2 Formalisasi per kata

Tahapan preprosesing selanjutnya adalah menggunakan inaNLP Formalization untuk memperbaiki kata-kata typo ataupun tidak baku. Implementasi kode diuraikan pada Gambar 4.3.

```

private String Formalization(String text) {
    String result = "";
    IndonesianSentenceFormalization formalizer = new
    IndonesianSentenceFormalization();
    for (int i = 0, j = 0; i < text.length(); i = j + 1) {
        j = text.indexOf(" ", i);
        if (j == -1) {
            String word = text.substring(i, text.length());
            result += word;
            break;
        }
        String word = text.substring(i, j);
        word = formalizer.formalizeWord(word);
        result += word;
        result += " ";
    }
    return result;
}

```

Gambar 4.3 Kode penggunaan inaNLP Formalization

4.2.2.3 Ekstraksi kata dasar dan imbuhan kepemilikan

Tahap berikutnya dari preproses adalah ekstraksi kata dasar menggunakan inaNLP Stemmer. Selain itu juga dilakukan

ekstraksi imbuhan kepemilikan dan dilakukan konversi ke kata dasarnya. Implementasi kode diuraikan pada Gambar 4.4.

```
private String Stemmer(String text) {
    String result = "";
    IndonesianStemmer stemmer = new IndonesianStemmer();
    for (int i = 0, j = 0; i < text.length(); i = j + 1) {
        j = text.indexOf(" ", i);
        if (j == -1) {
            String word = text.substring(i, text.length());
            result += word;
            break;
        }
        String word = text.substring(i, j);
        word = stemmer.stem(word);
        if((stemmer.possessivepronounsuffix.equals("mu")) {
            word += " kamu";
        }
        else if((stemmer.possessivepronounsuffix.equals("nya")) {
            word += " dia";
        }
        else if((stemmer.possessivepronounsuffix.equals("ku")) {
            word += " aku";
        }
        result += word;
        result += " ";
    }
    return result;
}
```

Gambar 4.4 Kode penggunaan stemmer

4.2.3 Tahap Prediksi

Tahap prediksi adalah tahap sistem melakukan prediksi pada data latih yang telah diunggah menggunakan model klasifikasi *NaiveBayes*. Implementasi kode diuraikan dalam Gambar 4.5.

```
Instances performance = Result.getPredictionsAsInstances(result);
DataSink.write(valuesOutput, performance);
Writer writeOutput = new Writer();
String path = writeOutput.writePrediction(pathResult);
```

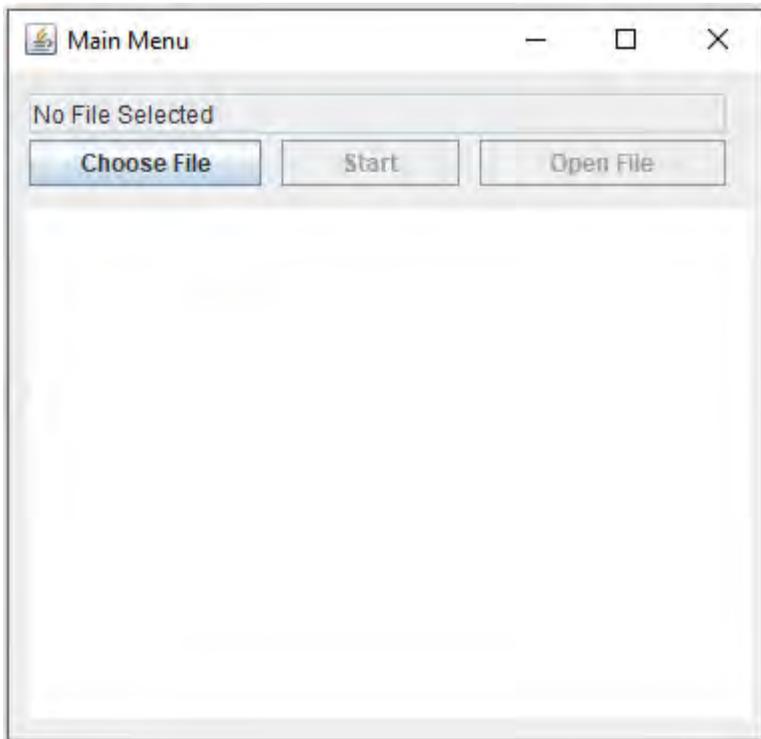
Gambar 4.5 Kode prediksi

4.3 Implementasi Antarmuka

Berikut merupakan antarmuka aplikasi desktop. Halaman utama memperlihatkan kebutuhan fungsional yang ditawarkan oleh sistem tersebut.

4.3.1 Antarmuka Kolom *Main Menu*

Berikut merupakan tampilan antarmuka kolom *Main Menu* pada halaman utama situs. Kolom *Pilih File* berfungsi untuk mencari File untuk Data Testing dan Kolom *Start* berfungsi untuk melakukan proses prediksi seperti terlihat pada Gambar 4.2.



Gambar 4.6 Antarmuka perangkat lunak

[Halaman ini sengaja dikosongkan]

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas hasil dan pembahasan pada aplikasi yang dikembangkan. Pada bab ini akan dijelaskan tentang data yang digunakan, hasil yang didapatkan dari penggunaan perangkat lunak dan uji coba yang dilakukan pada perangkat lunak yang telah dikerjakan untuk menguji apakah fungsionalitas aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i3-3217U @1.80GHz, RAM 4GB dengan Sistem Operasi Windows 10 Professional
 - b. Memori (RAM): 4 GB
 - c. Tipe sistem: Windows 10 Home
2. Perangkat lunak
 - a. Sistem operasi: Windows 10 Home
 - b. Perangkat pengembang: Java JDK 8

5.2 Data Uji Coba

Data uji yang digunakan terdiri dari delapan puluh data *tweet* yang diambil secara acak melalui fitur pencarian *hashtag*. Isi dari Data Uji Coba diuraikan pada Tabel 5.1.

Tabel 5.1 Tabel yang berisi data uji coba

No	Tweet
1	jarang nongol tugas numpuk
2	Tampaknya hari ini hari yang d penuh quiz waduh

3	Mendengar orang tua bertengkar de aku kecil smp skrng 30thn rasanya melelahkan
4	Galau Tugas Akhir
5	Ketika mata dan tugas tidak mau bekerjasama
6	Giliran ada niat, dosen gak masuk. Giliran gak niat, dosennya masuk. Maunya apa sih?
7	Dikerjar Deadline dan mentok di deadline. Moga aja tidak berjodoh dengan deadline yah
8	skripsi itu ga penting
9	Baru juga 2 minggu kuliah, badan udah harus ditemelin koyo karena pegel-pegel bikin tugas dan masuk angin
10	Liburan kemana ya?
11	Ini beneran malam ni begadang lagi? Padahal badan udah pegel
12	di saat dosen ngasih tugas yg melebihi kemampuan mahasiswanya?
13	Mudah mengucapkan selamat tinggal kepada dia tapi yg tersulit adalah melupakan kenangan indah bersama dia hiks
14	jenuh, Boring, dan butuh piknik.
15	Aku tidak perlu uang, yang aku perlu Keluarga yang utuh.
16	Politik dosen, materi UAS dari buku ringkasan yang disusunnya kemudian mahasiswa dsuruh beli.
17	Amburadul kotanya bahagia gubernurnya
18	kerja lagi, lembur lagi. lelah.
19	Orang2 mah nontonnya debat lah gue mah masih sibuk ngerjain tugas kuliah
20	Ini dosen memang beneran ya. Pengumuman nilai paling telat dan nggak acuh sama mahasiswa yg protes nilai
21	Punya adik satu susah bgt dikasih tau, semaunya aja bertindak
22	Kuliah kerasa makin berat menjelang semester pra-thesis.. Pngen ngamuk terus gigit2 panda beijing boleh ga ya?

23	Aku sayang ibu ku.. selamat hari ibu
24	Iri Melihat Mereka yg Bisa Rayain Lebaran nanti Bareng Keluarga yg lengkap. Syukuri jika Kamu masih bisa berkumpul dengan Keluarga.
25	Nunggu 1 jam Masuk 15 menit Baru mulai presentasi Pamit, salam, lalu keluar. Memang dosen
26	Kebut terus laporan
27	Astagfirulloh. Suhu badan sampe 40 derajat.
28	Akhir semester mau berakhri, tpi tugas belum juga beres malah makin hari makin nambah.
29	Semester 2 udah selesai ! Super pusing gila ! Gausah lanjut semester 3 lanjut kawin aja deh
30	Gmna kalo mahasiswa yang PHPin dosen?
31	Kelas Malem Sial
32	Sore ini bete banget nyetirnya. Nggak konsen. Serba salah. Ambil puter balik gagal ga sukses biasanya
33	Hp boleh gede tapi paketan gaada
34	Sampe kapan selesai kuliah sm dosen ini
35	Kok tambah cakep ketika pake kemeja putih, kopiah, dan sarung
36	Tugas numpuk banyak aduhh kapan kelar matematika sangat rumit bagi saya tapi saya harus belajar. belajar itu memang sulit awalnya
37	apakah masih pantas dipanggil ayah? sdangkan seorang ayah tidak pernah mau tau tentang kemauan & kehidupan anaknya
38	Hari ini kesel bangt udah nyasar ban mobil kempes dan tempat tujuan sdh tutup alamak padahal sdh disiapkan semuanya

39	Berduka cita buat temanku padahal siang bicara sama aku selamat jalan ya meskipun Kita tidak se kelas tetapi Kamu teman dekatku
40	Lagi bete-betanya eh, ada yang bikin tambah bete.
41	Ya Allah, hapuskan jarak antara aku dgn jodohku, karena kini ku mulai tersadar bahwa aku membutuhkannya,
42	Selamat Tahun Baru 2018 Ganks. Semoga selalu menjadi tahun yang penuh berkah & penuh semangat baru untuk kita semua.
43	Dosen untung mahasiswa buntung
44	Sudah berada pada titik yg paling jenuh
45	Tidak semua orang Amerika setuju dengan Trump. Jadi untuk apa boikot semua produk Amerika
46	Pas dateng ke dosen buat protes nilai: Pngen dari C jadi E?
47	Di kejar deadline tugas
48	Apakah kamu ga ngerti juga ha ??Aku ini slalu mikirin kamu!!!
49	Panas...lelah meningkat... Pusing berkelanjutan
50	Kerja mah tiap hari juga semangat ,yang enggak bikin semangat mah kamu ujug2 jadian sama si kakait bandros itu
51	Hari ini politik semua, ohh politik Teori Politik dan Pemikiran politik indonesia.
52	Windows 8 di layar sentuh dan windows 8 di layar laptop itu beda ya ternyata.
53	jenuh seharian di kamar
54	Datang, baca slide, ngasih tugas, pergi.. Dosen jaman sekarang
55	Bosen dirumah
56	Kamu tau aku belum tidur jam segini karena apa? Karena kamu!
57	Ngantuk Berat hari ini !!!
58	Pngen punya pacar

59	Udah kaya ngerjain Skripsi aja ampe begadang kaya gini. Pusing
60	Tugas numpuk !!! Kantung mata udah kaya kantung belanjaan. Mata udah mirip kaya Panda
61	Biaya kuliah tidak sesuai dengan fasilitas yang ada!
62	6 jt perbulan dengan fasilitas Kondisi panas, signal susah!!
63	Paracetamol 500mg dosis 4 kali sehari. Demam kapan kau turun.
64	Ketika revisi ditolak tanpa ditimbang, draft dilarang tanpa ada alasan ,hanya ada 4 kata... Gelot wae njo pak dosen?
65	Ciri dosen gaji buta: slide dengan warna monoton, bicara datar, tidak punya pengalaman di luar kampus
66	susah.. nanti cari ttd dosen sama2 ee ..ngerjain tugas aja udah kayak gini, gmn entar pas tesis..???
67	Mudah mengucapkan selamat tinggal kepada dia, tapi yg tersulit adalah melupakan kenangan indah bersama dia
68	Wifi kampus cacat gr! Percuma bayar kuliah mahal-mahal, koneksi wi-fi aja kagak connect
69	politik itu sebenarnya apa sih?
70	Mengapa aku yang harus selalu mengalah
71	deadline menumpuk semua tugas...
72	Lagi bosan banget nih suntuk
73	Mau pergi ke masa tahun 1992 sampe 2015 ketika ayah masih satu rumah dan gak poligami ninggalin aku,ibu dan adik aku, setegar tegarnya seorang anak pasti akan goyah ketika orang tuanya bercerai
74	tugas sampai pagi, tidur sebentar, lanjut tugas lagi.
75	smoga menjadi keluarga yang samawah yo, sampai kakek nenek, dan smoga cpt di beri momongan

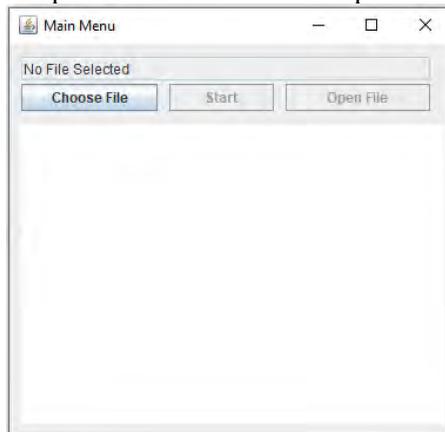
76	Ulah dari dosen killer, mahasiswa tertekan selama perkuliahan, aturan penilaian tetap jadi misteri
77	Tugas makalah banyak bgt Deadline pke alternatif terakhir, sistem kebut semalam
78	Udah sakit punggung, mungkin waktunya istirahat dulu. #deadline
79	Waktunya tidur malah tugas, waktunya tugas malah tidur dan akhirnya keteteran juga ama tugas siap2 aja besok presentasi hmm
80	Ngantri bayar kuliah sama kaya ngantri beli tiket disenayan panjang banget

5.3 Skenario Pengujian

Scenario pengujian berikut yang dilakukan dalam tugas akhir memiliki beberapa tahapan pengujian seperti pada subbab berikut ini.

5.3.1 Membuka aplikasi

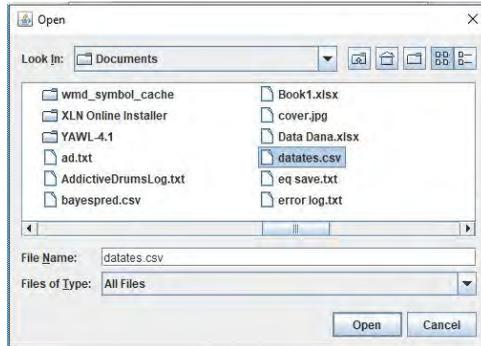
Aktor membuka aplikasi dan sistem menampilkan halaman utama.



Gambar 5.1 Antarmuka halaman utama

5.3.2 Memilih file data latihan

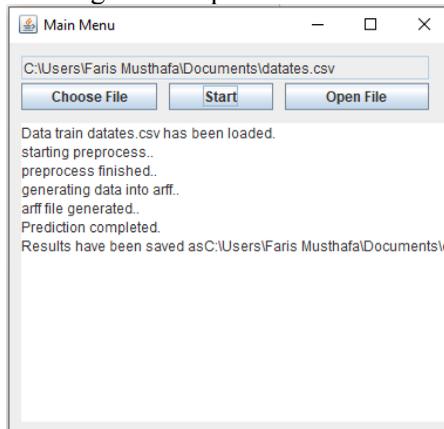
Aktor menekan tombol *choose file* untuk memilih file data latihan. Sistem menampilkan *filechooser* dan aktor memilih file data latihan.



Gambar 5.2 Antarmuka *filechooser*

5.3.3 Melakukan prediksi pada data latihan

Aktor menekan tombol *start* untuk melakukan prediksi. Sistem akan menghasilkan prediksi dalam bentuk file *.csv*



Gambar 5.3 Antarmuka dengan notifikasi sukses prediksi

5.3.4 Membuka file hasil prediksi

Aktor menekan tombol *open file* untuk membuka file hasil prediksi. Sistem akan membuka *file (.csv)* dengan menggunakan *default program* pembacaan *file (.csv)*.

```

1 "jarang nongol tugas numpuk", keluhan subjek
2 "Tampaknya hari ini hari yang d penuh quiz waduh", keluhan personal, keluhan subjek
3 "Mendengar orang tua bertengkar de aku kecil smp skrng 30thn rasanya melelahkan", keluhan relasi
4 "Galau Tugas Akhir", keluhan subjek
5 "Ketika mata dan tugas tidak mau bekerjasama", keluhan subjek
6 "Giliran ada niat, dosen gak masuk. Giliran gak niat, dosennya masuk. Maunya apa sih?", keluhan relasi, keluhan institusi
7 "Dikerjar Deadline dan mentok di deadline. Moga aja tidak berjodoh dengan deadline yah", keluhan relasi
8 "skripsi itu ga penting", keluhan subjek, keluhan relasi
9 "Baru juga 2 minggu kuliah, badan udah harus ditemelin koyo karena pegel-pegel bikin tugas dan masuk angin", keluhan subjek
10 "Liburan kemana ya?", keluhan subjek
11 "Ini beneran malam ni begadang lagi? Padahal badan udah pegel", keluhan personal
12 "di saat dosen ngasih tugas yg melebihi

```

Gambar 5.4 Isi file (.csv) hasil prediksi sistem yang dibuka dengan menggunakan sublime text

5.4 Hasil uji coba

Tabel 5.2 Tabel berisi hasil deteksi

Tweet	Personal	Subjek	Relasi	Institusi
jarang nongol tugas numpuk	-	+	-	-
Tampaknya hari ini hari yang d penuh quiz waduh	+	+	-	-
Mendengar orang tua bertengkar de aku kecil smp skrng 30thn rasanya melelahkan	-	-	+	-
Galau Tugas Akhir	-	+	-	-

Ketika mata dan tugas tidak mau bekerjasama	-	+	-	-
Giliran ada niat, dosen gak masuk. Giliran gak niat, dosennya masuk. Maunya apa sih?	-	-	+	+
Dikerjar Deadline dan mentok di deadline. Moga aja tidak berjodoh dengan deadline yah	-	-	+	-
skripsi itu ga penting	-	+	-	-
Baru juga 2 minggu kuliah, badan udah harus ditemelin koyo karena pegel-pegel bikin tugas dan masuk angin	-	+	-	-
Liburan kemana ya?	-	+	-	-
Ini beneran malam ni begadang lagi? Padahal badan udah pegel	+	-	-	-
di saat dosen ngasih tugas yg melebihi kemampuan mahasiswanya?	-	+	-	-
Mudah mengucapkan selamat tinggal kepada dia tapi yg tersulit adalah melupakan kenangan indah bersama dia hiks	-	-	+	-
jenuh, Boring, dan butuh piknik.	+	-	-	-
Kuis lalu praktikum lalu bikin laporan lalu jalan ama sayanku capeknyaa	-	-	+	-
Aku tidak perlu uang, yang aku perlu Keluarga yang utuh.	-	-	+	-

Politik dosen, materi UAS dari buku ringkasan yang disusunnya kemudian mahasiswa dsuruh beli.	-	-	-	+
Amburadul kotanya bahagia gubenuhnya	+	-	-	-
kerja lagi, lembur lagi. lelah.	-	+	-	-
Orang2 mah nontonnya debat lah gue mah masih sibuk ngerjain tugas kuliah	-	+	-	-
Ini dosen memang beneran ya. Pengumuman nilai paling telat dan nggak acuh sama mahasiswa yg protes nilai	-	-	+	+
Punya adik satu susah bgt dikasih tau, semaunya aja bertindak	-	-	+	-
Kuliah kerasa makin berat menjelang semester pra-thesis.. Pengen ngamuk terus gigit2 panda beijing boleh ga ya?	-	-	+	-
Aku sayang ibu ku.. selamat hari ibu	-	-	+	-
Iri Melihat Mereka yg Bisa Rayain Lebaran nanti Bareng Keluarga yg lengkap. Syukuri jika Kamu masih bisa berkumpul dengan Keluarga.	-	-	+	-
Nunggu 1 jam Masuk 15 menit Baru mulai presentasi Pamit, salam, lalu keluar. Memang dosen	-	-	-	+
Kebut terus laporan	-	+	-	-
Astagfirulloh. Suhu badan sampe 40 derajat.	-	-	+	-

Akhir semester mau berakhri, tpi tugas belum juga beres malah makin hari makin nambah.	-	+	-	-
Deadline ngga ada habisnya laporan, progress, rapat koordinasi event mane cowo aku ngjak jalan mulu lagi aghhhh mau tidur ajaaa	-	-	+	-
Semester 2 udah selesai ! Super pusing gila ! Gausah lanjut semester 3 lanjut kawin aja deh	+	+	-	-
Gmna kalo mahasiswa yang PHPin dosen?	-	-	-	+
Kelas Malem Sial	-	+	-	-
Sore ini bete banget nyetirnya. Nggak konsen. Serba salah. Ambil puter balik gagal ga sesukses biasanya	-	-	+	-
Hp boleh gede tapi paketan gaada	-	-	-	-
Sampe kapan selesai kuliah sm dosen ini	-	-	-	-
Kok tambah cakep ketika pake kemeja putih, kopiah, dan sarung	-	-	-	-
Tugas numpuk banyak aduhh kapan kelar matematika sangat rumit bagi saya tapi saya harus belajar. belajar itu memang sulit awalnya	-	+	-	-

apakah masih pantas dipanggil ayah? sdangkan seorang ayah tidak pernah mau tau tentang kemauan & kehidupan anaknya	-	-	+	-
Hari ini kesel bangt udah nyasar ban mobil kempes dan tempat tujuan sdh tutup alamak padahal sdh disiapkan semuanya	-	+	-	-
Berduka cita buat temanku padahal siang bicara sama aku selamat jalan ya meskipun Kita tidak se kelas tetapi Kamu teman dekatku	-	+	+	-
Lagi bete-betanya eh, ada yang bikin tambah bete.	+	-	+	-
Ya Allah, hapuskan jarak antara aku dgn jodohku, karena kini ku mulai tersadar bahwa aku membutuhkannya,	-	-	+	-
Selamat Tahun Baru 2018 Ganks. Semoga selalu menjadi tahun yang penuh berkah & penuh semangat baru untuk kita semua.	+	-	-	-
Gilee susah2 belajar ehh remed lagi mane pas bgt ama deadline laporan ayolah cpt weekend biar bisa jln ama sayangku	-	+	+	-
Dosen untung mahasiswa buntung	-	-	-	+
Sudah berada pada titik yg paling jenuh	+	-	-	-
Tidak semua orang Amerika setuju dengan Trump. Jadi untuk apa boikot semua produk Amerika	-	-	+	-

Pas dateng ke dosen buat protes nilai: Pngen dari C jadi E?	-	-	-	+
Di kejar deadline tugas	-	+	-	-
Apakah kamu ga ngerti juga ha ??Aku ini slalu mikirin kamu!!!	-	-	+	-
Panas...lelah meningkat... Pusing berkelanjutan	+	-	-	-
Kerja mah tiap hari juga semangat, yang enggak bikin semangat mah kamu ujug2 jadian sama si kakait bandros itu	-	+	+	-
Hari ini politik semua, ohh politik Teori Politik dan Pemikiran politik indonesia.	+	+	-	-
Windows 8 di layar sentuh dan windows 8 di layar laptop itu beda ya ternyata.	-	-	-	-
jenuh seharian di kamar	+	-	-	-
Datang, baca slide, ngasih tugas, pergi.. Dosen jaman sekarang	-	+	-	-
Bosen dirumah	+	-	-	-
Kamu tau aku belum tidur jam segini karena apa? Karena kamu!	-	-	+	-
Yaampun dosenku ngasih tugas ga mikir dah baru aja tugas dikasih eh tambahan lagi mana wifi di kampus suka bermasalah terus cowokku ga bantuin lagi	-	+	+	+
Ngantuk Berat hari ini !!!	+	+	-	-
Pngen punya pacar	-	-	+	-

Udah kaya ngerjain Skripsi aja ampe begadang kaya gini. Pusing	+	+	-	-
Tugas numpuk !!! Kantung mata udah kaya kantung belanjaan. Mata udah mirip kaya Panda	-	+	-	-
Biaya kuliah tidak sesuai dengan fasilitas yang ada!	-	-	+	+
6 jt perbulan dengan fasilitas Kondisi panas, signal susah!!	-	-	-	+
Paracetamol 500mg dosis 4 kali sehari. Demam kapan kau turun.	+	+	-	-
Ketika revisi ditolak tanpa ditimbang, draft dilarang tanpa ada alasan ,hanya ada 4 kata... Gelot wae njo pak dosen?	-	-	-	+
Ciri dosen gaji buta: slide dengan warna monoton, bicara datar, tidak punya pengalaman di luar kampus	-	-	+	+
susah.. nanti cari ttd dosen sama2 ee ..ngerjain tugas aja udah kayak gini, gmn entar pas tesis..???	-	-	-	+
Mudah mengucapkan selamat tinggal kepada dia, tapi yg tersulit adalah melupakan kenangan indah bersama dia	-	-	+	-
Wifi kampus cacat gr! Percuma bayar kuliah mahal-mahal, koneksi wi-fi aja kagak connect	-	-	+	+
politik itu sebenarnya apa sih?	-	-	-	-
Mengapa aku yang harus selalu menang	-	-	+	-

What should i choose? Laporan? Laper? Or love?	-	-	-	-
deadline menumpuk semua tugas...	-	+	-	-
Lagi bosan banget nih suntuk	+	-	-	-
Mau pergi ke masa tahun 1992 sampe 2015 ketika ayah masih satu rumah dan gak poligami ninggalin aku,ibu dan adik aku, setegar tegarnya seorang anak pasti akan goyah ketika orang tuanya bercerai	-	-	+	-
tugas sampai pagi, tidur sebentar, lanjut tugas lagi.	-	+	-	-
smoga menjadi keluarga yang samawah yo, sampai kakek nenek, dan smoga cpt di beri momongan	-	-	+	-
Ulah dari dosen killer, mahasiswa tertekan selama perkuliahan, aturan penilaian tetap jadi misteri	-	-	-	+
Tugas makalah banyak bgt Deadline pke alternatif terakhir, sistem kebut semalam	-	+	-	-
Udah sakit punggung, mungkin waktunya istirahat dulu. #deadline	+	-	-	-
Waktunya tidur malah tugas, waktunya tugas malah tidur dan akhirnya keteteran juga ama tugas siap2 aja besok presentasi hmm	-	+	-	-
Ngantri bayar kuliah sama kaya ngantri beli tiket disenayan panjang banget	-	-	+	-

Tabel 5.3 Tabel legenda hasil deteksi

Simbol	Penjelasan
+	Keluhan terdeteksi dan benar
-	Keluhan tidak terdeteksi dan benar
+	Keluhan terdeteksi dan salah
-	Keluhan tidak terdeteksi dan salah

Tabel 5.4 Tabel statistik dari uji coba

Akurasi keseluruhan		68.75%
Deteksi benar		55
Deteksi meleset		25
akurasi per label	personal	76.47%
	subjek	75.00%
	relasi	80.00%
	inststitusi	80.00%
	bukan keluhan	23.08%

Tabel 5.2 Merupakan hasil representasi dari hasil deteksi model. Tanda (+) merupakan hasil deteksi yang bernilai positif dan tanda (-) merupakan hasil deteksi negatif. Sedangkan hasil deteksi yang sesuai dengan nilai sebenarnya (deteksi akurat) ditandai dengan simbol berwarna hijau dan hasil deteksi yang meleset dari nilai sebenarnya diberi warna merah.

Berdasarkan statistik pada tabel 5.3, uji coba berhasil dilakukan dengan nilai akurasi sebesar 68.75% dengan rincian 55 deteksi bernilai benar dan 25 deteksi bernilai salah dari 80 data uji. Akurasi yang didapat dari masing-masing label sebesar 76.47% untuk deteksi keluhan personal, 75% untuk deteksi keluhan subjek, 80% untuk deteksi keluhan relasi, 80% untuk deteksi keluhan institusi, dan 23.08% untuk deteksi non keluhan.

5.5 Evaluasi Pengujian

Berdasarkan hasil percobaan, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan. Berikut merupakan rangkuman hasil pengujian fungsionalitas program yang berhasil dan bekerja sesuai dengan harapan.

ID	Nama	Hasil Uji Coba
UJ-UC-001	Memilih file data uji	Berhasil
UJ-UC-002	Melakukan prediksi	Berhasil
UJ-UC-003	Melihat hasil prediksi	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Model dibuat dengan melakukan prapemrosesan data latih menggunakan *inaNLP Formalizer*, *inaNLP Stemmer*, *inaNLP POS Tagger*, dan *word to word matching*. Data yang telah di praproses kemudian dilabelisasi dan dilatih menggunakan transformasi multilabel *binary relevance* dan *classifier Naïve Bayes*. Hasil latih disimpan dalam bentuk *file model*.
2. *Word to word matching* dengan Kamus Besar Bahasa Indonesia pada data latih digunakan untuk mengatasi kendala pada *inaNLP POS Tagger* yang tidak bisa membedakan kata ganti benda dan kata yang tidak baku.
3. Berdasarkan pengujian yang telah dilakukan, *classifier Naïve Bayes* menghasilkan akurasi tertinggi dan *running time* terendah dibandingkan dengan metode *classifier Simple Logistic*, *KStar*, *Decision Table*, dan *j48*.
4. Pemodelan deteksi multilabel keluhan menggunakan *classifier Naïve Bayes* pada delapan puluh data uji coba terhadap label keluhan personal, subjek, relasi, dan institusi telah berhasil dengan masing-masing akurasi bernilai 76.47%, 75%, 80%, dan 80%.
5. Perangkat lunak antarmuka deteksi multilabel keluhan mahasiswa berhasil dibuat dengan tampilan yang jelas dan

sederhana yang terdiri dari tiga tahapan penggunaan: unggah data *tweet*, mulai melakukan deteksi, dan buka hasil deteksi.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Memperbaiki proses POS Tagger inaNLP agar bisa melakukan klasifikasi kategori kata secara lebih akurat. Dengan bekerjanya POS Tagger inaNLP proses *word to word matching* bisa dihilangkan karena menggunakan banyak *resource*.
2. Meminimalisir penggunaan basis data relasional dan memindahkan semua penyimpanan data ke format yang lebih sederhana seperti file (.csv) untuk mengurangi beban penggunaan *resource* pada proses baca-tulis data.
3. Menambahkan kata-kata hashtag sehingga lebih banyak data latih yang tersaring yang berimbas pada naiknya akurasi deteksi sistem.
4. Menambahkan pengembangan metode semantik kalimat dengan hipotesis dapat memperbaiki hasil deteksi multilabelisasi dengan menggunakan model yang dibangun sekarang.

DAFTAR PUSTAKA

- [1] P. Meedeche, N. Jam-on, and T. Boongoen, "Prediction of Student Dropout Using Personal Profile and DataMining Approach," in *Intelligent and Evolutionary Systems*, vol. 5, Springer International Publishing Switzerland, 2016, pp. 143–155.
- [2] R. Chen, "Institutional Characteristics and College Student Dropout Risks: A Multilevel Event History Analysis," *Res. High. Educ.*, vol. 53, no. 5, pp. 487–505, 2012.
- [3] K. Kori, M. Pedaste, E. Tõnisson, H. Altin, and R. Rantsus, "First-year dropout in ICT studies," in *2015 IEEE Global Engineering Education Conference (EDUCON)*, 2015, no. March, pp. 444–452.
- [4] M. A. Yehuala, "Application of Data mining Techniques for student success and failure prediction," *Int. J. Sci. Technol. Res.*, vol. 4, no. 4, pp. 3–6, 2015.
- [5] A. Zhang and C. L. Aasheim, "Academic Success Factors : An IT Student Perspective," *J. Inf. Technol. Educ. Res.*, vol. 10, no. 1, pp. 309–331, 2011.
- [6] X. Chen, M. Vorvoreanu, and K. P. C. Madhavan, "Mining Social Media Data for Understanding Students' Learning Experiences," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 246–259, 2014.
- [7] A. R. Naradhipa and A. Purwarianti, "Sentiment classification for Indonesian message in social media," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–4.
- [8] Valdis Krebs, "Social Network Analysis: An Introduction," *Social Network Analysis: An Introduction*. [Online]. Available: <http://www.orgnet.com/sna.html>. [Accessed: 02-Sep- 2017].
- [9] J. Bayer, H. Bydzovská, J. Géryk, T. Obsivac, and L. Popelinský, "Predicting drop-out from social behaviour of students," in *Proceedings of the 5th International Conference on Educational Data Mining*, 2012, no. Dm, pp. 103–109.
- [10] DIKTI, "Pengelolaan Perguruan Tinggi Swasta." [Online]. Available: <http://dikti.go.id/blog/2014/09/24/pengelolaan-perguruan-tinggi-swasta>. [Accessed: 01- Feb-2015].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Muhammad Faris Musthafa atau yang biasa disapa dengan nama Faris lahir di Jakarta pada tanggal 17 Januari 1995. Memiliki seorang kakak laki-laki dan telah menempuh pendidikan di TK Sampaghita, SD Negeri Rawa Barat 05 Pagi (2001-2007), SMP Negeri 13 Jakarta (2007-2010), SMA Negeri 55 Jakarta (2010-2013) dan saat ini sedang menempuh pendidikan Sarjana di Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan tahun 2013.

Memiliki hobi menonton film dan tertarik dengan dunia tersebut. Terlibat aktif dalam organisasi kemahasiswaan serta kepanitiaan selama perkuliahan, antara lain staff Dalam Negeri di Himpunan Mahasiswa Teknik Computer-Informatika ITS 2014/2015, Staf NST (National Seminar and Technology) Schematics 2014, Staf Keamanan dan Perizinan acara Schematics 2015. Untuk komunikasi, dapat dihubungi melalui surel: m.farismusthafa@gmail.com