



**PENGUKURAN INDEKS KEBAHAGIAAN MENGGUNAKAN
ANALISIS SENTIMEN DI MEDIA SOSIAL DENGAN
METODE *NAIVE BAYES CLASSIFIER***

SKRIPSI

Oleh:

**Vananda Rahadika
NIM 132410101079**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2017**



**PENGUKURAN INDEKS KEBAHAGIAAN MENGGUNAKAN
ANALISIS SENTIMEN DI MEDIA SOSIAL DENGAN
METODE *NAIVE BAYES CLASSIFIER***

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas
Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh:

**Vananda Rahadika
NIM 132410101079**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2017**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda tercinta Soevrieni Eka Boediarti dan Ayahanda alm. Bambang Djatmiko Cahyono;
3. Saudara perempuan Vidya Elza Loriza;
4. Keluarga besar terutama alm. Bambang Siswanto Cahyono, Nanik Nurfati, RR Widiartik, Suwari;
5. Guru – guru baik dari pendidikan formal maupun informal.
6. Almamater Program Studi Sistem Informasi Universitas Jember.

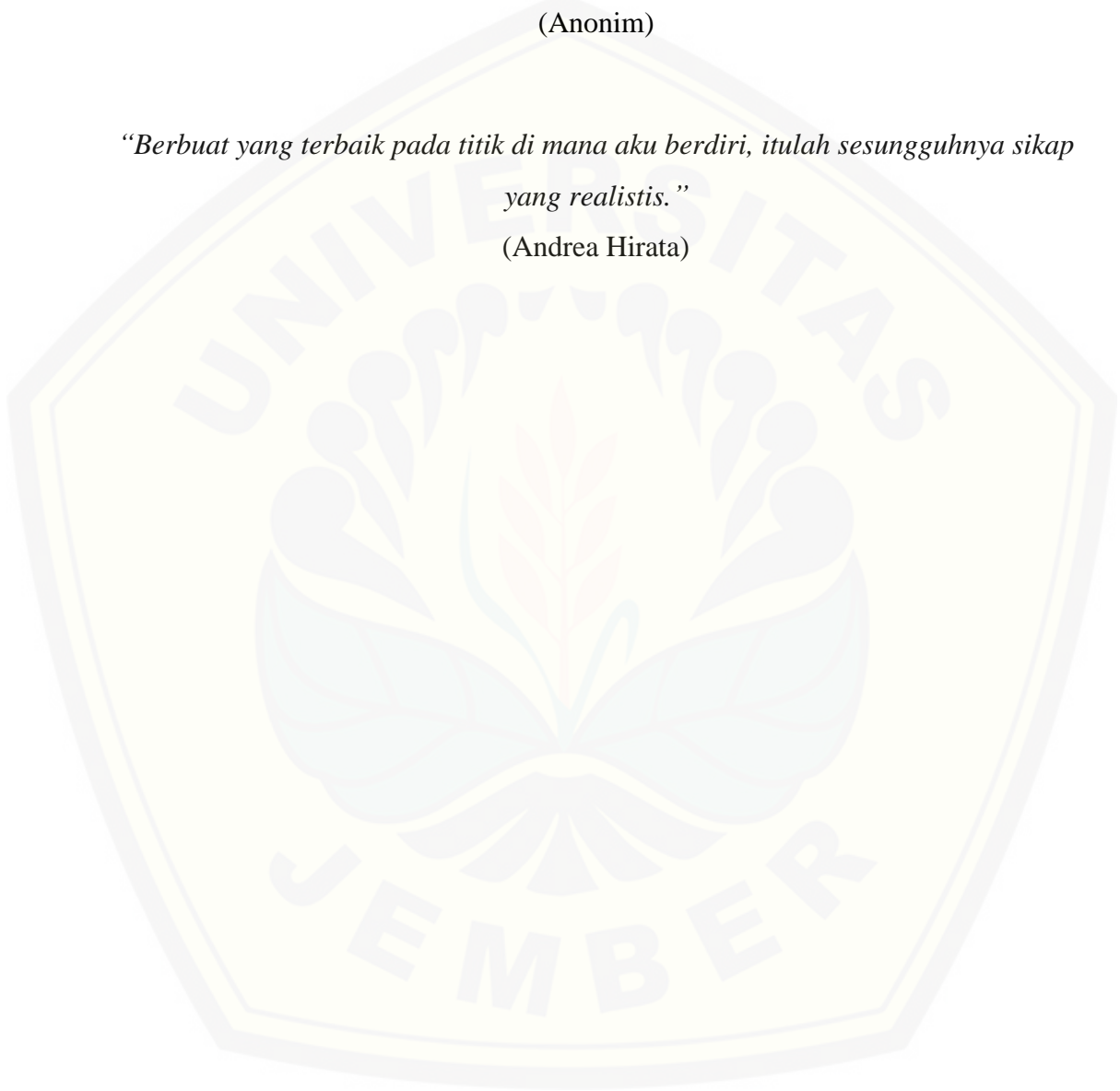
MOTO

“Sekali gagal bukan berarti selamanya, sedangkan berhasil itu soal seberapa gigih kamu mau berusaha”

(Anonim)

“Berbuat yang terbaik pada titik di mana aku berdiri, itulah sesungguhnya sikap yang realistis.”

(Andrea Hirata)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Vananda Rahadika

NIM : 132410101079

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Pengukuran Indeks Kebahagiaan Menggunakan Analisis Sentimen Di Media Sosial Dengan Metode *Naive Bayes Classifier*” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 21 Juli 2017

Yang menyatakan,

Vananda Rahadika

NIM 132410101079

SKRIPSI

**PENGUKURAN INDEKS KEBAHAGIAAN MENGGUNAKAN
ANALISIS SENTIMEN DI MEDIA SOSIAL DENGAN
METODE *NAÏVE BAYES CLASSIFIER***

Oleh :

Vananda Rahadika

NIM 132410101079

Pembimbing

Dosen Pembimbing Utama : Nelly Oktavia Adiwijaya S.Si., MT.

Dosen Pembimbing Pendamping : Oktalia Juwita, S.Kom., M.MT

PENGESAHAN PEMBIMBING

Skripsi berjudul “Pengukuran Indeks Kebahagiaan Menggunakan Analisis Sentimen Di Media Sosial Dengan Metode *Naive Bayes Classifier*”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 21 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Nelly Oktavia Adiwijaya S.Si., MT

Oktalia Juwita, S.Kom., M.MT

NIP 198410242009122008

NIP 198110202014042001

PENGESAHAN PENGUJI

Skripsi berjudul “Pengukuran Indeks Kebahagiaan Menggunakan Analisis Sentimen Di Media Sosial Dengan Metode *Naive Bayes Classifier*”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 21 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Drs. Antonius C.P., M.App.Sc., Ph.D.

Fajrin Nurman Arifin, S.T., M.Eng.

NIP. 196909281993021001

NIP. 198511282015041002

Mengesahkan

Ketua Program Studi

Prof. Drs. Slamim, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001

RINGKASAN

Pengukuran Indeks Kebahagiaan Menggunakan Analisis Sentimen Di Media Sosial Dengan Metode *Naive Bayes Classifier*; Vananda Rahadika, 132410101079 2017, 208 halaman; Program Studi Sistem Informasi Universitas Jember.

Kata “kebahagiaan” seringkali menjadi sesuatu yang kabur dan sangat sulit diukur. Apabila seseorang ditanya “seberapa bahagiakah anda?”, maka jawaban yang muncul sangat beragam, karena setiap orang memiliki cara yang berbeda dalam memaknai arti kebahagiaan. Beberapa ahli di dunia, lebih mendefinisikan kebahagiaan sebagai “kesejahteraan subjektif” agar tidak terjadi ambiguitas makna. Sulitnya mengukur tingkat kebahagiaan, bukan berarti kebahagiaan tidak bisa diukur.

Dalam metodologi pengukuran indeks kebahagiaan yang diterbitkan oleh BPS Indonesia, kebahagiaan diukur dalam indeks komposit menggunakan suatu survei yang mencakup 10 domain/variabel. Di satu sisi, penetrasi internet Indonesia menunjukkan angka yang tidak bisa diabaikan. Berdasarkan data yang didapat bahwa hampir semua pengguna internet Indonesia memiliki media sosial. Salah satu media sosial yang banyak di akses adalah *twitter* yang menempati peringkat ke-5 (7.2 juta jiwa) di antara media sosial lainnya.

Berdasarkan data tersebut, maka bisa dibuat sebuah pendekatan lain sebagai alternatif untuk pengukuran indeks kebahagiaan salah satunya yaitu pendekatan pengukuran indeks kebahagiaan melalui sentimen yang ada di media sosial *twitter*. Manfaat yang didapatkan dibandingkan pengukuran indeks kebahagiaan secara konvensional seperti pengurangan biaya, pengurangan waktu dan juga pengurangan tenaga yang dikeluarkan karena pengukuran tidak memerlukan survei dan data dapat diambil secara mudah dan cepat melalui sentimen di media sosial. Dalam hal ini, penelitian ini ingin memberikan kontribusi terhadap kemajuan pendekatan ini dengan menghitung indeks kebahagiaan tersebut menggunakan metode *Naive Bayes Classifier*. Penggunaan *Naive Bayes Classifier*

dipilih karena metode klasifikasi ini cukup banyak dipakai dalam melakukan sentimen analisis dan *text mining*. Hasil dari penelitian ini diharapkan mampu memberikan komparasi dan menambah literatur dalam pengembangan pendekatan indeks kebahagiaan melalui analisis sentimen.



PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengukuran Indeks Kebahagiaan Menggunakan Analisis Sentimen Di Media Sosial Dengan Metode *Naive Bayes Classifier*”, Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Nelly Oktavia A, S.Si, MT selaku dosen pembimbing utama dan Oktalia Juwita, S.Kom., M.MT selaku dosen pembimbing anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Drs. Antonius C.P., M.App.Sc., Ph.D. selaku penguji utama dan Fajrin Nurman Arifin, S.T., M.Eng selaku penguji anggota yang telah berkenan untuk menguji skripsi ini dan memberikan masukan serta saran untuk pengembangan diri penulis dan skripsi ini;
4. Seluruh bapak dan ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ibunda Soevrieni Eka Boediarti, Ayahanda alm. Bambang Djatmiko Cahyo, dan keluarga besar yang selalu mendukung dan mendoakan;
6. Diah Aisiyah yang telah membantu dan memberikan motivasi;
7. Para sahabat Sholihatun Nikmah, Muhammad Ilham Fauzi, Savira Oktari, Sinta Eka Fitriyanti, Anisa Dzulkarnain, Antonius Halim Febrianto dan Yusuf Eka Sayogana yang telah membantu, memotivasi dan menemani;
8. Keluarga KKN 040 gelombang II tahun 2015/2016 Vina Wahyu Paramita, Hendra Budianto, Putri Karuniawati, Nurul Aini, Efa Uswatun Khasanah, Arman, Fitri Wulan Andriyani, Nia Martasari, dan Aulia Syafar untuk semua keceriaan, pengalaman, kasih, serta semangat;

9. Keluarga INTENTION atas kekeluargaan yang selalu hangat dan menenangkan;
10. Teman-Teman Program Studi Sistem Informasi di semua angkatan;
11. Semua pihak yang tidak dapat disebutkan satu persatu;

Dengan harapan bahwa penelitian ini nantinya terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 21 Juli 2017

Penulis



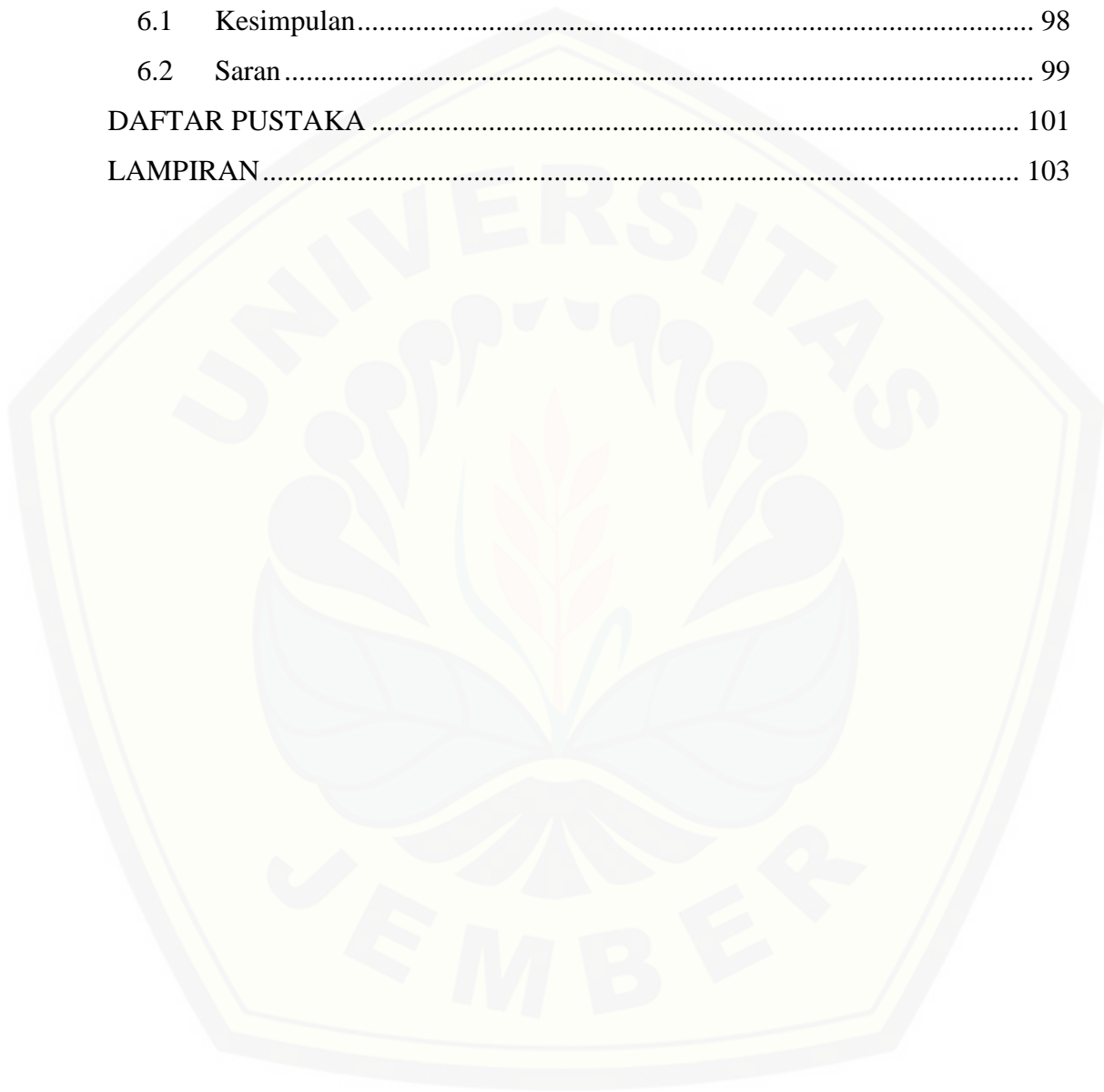
DAFTAR ISI

PERSEMBAHAN	iii
MOTO	iv
PERNYATAAN	v
SKRIPSI	vi
PENGESAHAN PEMBIMBING	vii
PENGESAHAN PENGUJI	viii
RINGKASAN	ix
PRAKATA	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xviii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Text Mining	6
2.2 Indeks Kebahagiaan	8
2.3 Analisis Sentimen	8
2.4 Naive Bayes Classifier	9
2.5 Algoritma Enhanced Confix-stripping	12
2.6 Twitter API	15
2.7 Uji Performansi	16
BAB 3. METODOLOGI PENELITIAN	18
3.1 Metode Penelitian	18
3.2 Metode Pengumpulan Data	18
3.3 Metode Klasifikasi	18

3.3.1 Analisis Kebutuhan (<i>Requirement</i>).....	19
3.3.2 Desain Sistem.....	19
3.3.3 Implementasi	20
3.3.4 <i>Testing</i> dan Evaluasi	20
3.3.5 Pemeliharaan	20
3.4 Gambaran Algoritma Sistem	21
BAB 4. PERANCANGAN SISTEM	25
4.1 Deskripsi Umum Sistem.....	25
4.2 Analisis Kebutuhan	25
4.2.1 Studi Literatur	25
4.2.2 Kebutuhan fungsional	26
4.2.3 Kebutuhan Non Fungsional	27
4.3 Desain Sistem	28
4.3.1 <i>Business Process</i>	28
4.3.2 <i>Use Case Diagram</i>	29
4.3.3 Skenario	32
4.3.4 <i>Sequence Diagram</i>	38
4.3.5 Activity Diagram.....	42
4.3.6 <i>Class diagram</i>	46
4.3.7 Entity Relationship Diagram (ERD).....	46
4.4 Penulisan Kode Program	47
4.4.1 Kode Program <i>Login</i>	47
4.4.2 Kode Program Registrasi Akun	47
4.4.3 Kode Program Mengelola <i>User</i>	47
4.4.4 Kode Program Mengelola <i>Tweet</i>	48
4.4.5 Kode Program Mengelola Data <i>Training</i>	48
4.4.6 Kode Program Melihat <i>Text Mining Data Training</i>	48
4.4.7 Kode Program Mengelola Data <i>Testing</i>	49
4.4.8 Kode Program Melihat <i>Text Mining Data Testing</i>	49
4.4.9 Kode Program Melihat Klasifikasi dan Uji Performansi Data <i>Testing</i>	49
4.4.10 Kode Program Mengelola <i>Stopword</i>	50

4.4.11 Kode Program Mengelola Ontologi	50
4.4.12 Kode Program Melihat Indeks Kebahagiaan Data <i>Testing</i>	51
4.4.13 Kode Program Mengelola Data Indeks Kebahagiaan Harian	51
4.4.14 Kode Program Melihat <i>FAQ</i>	51
4.4.15 Kode Program <i>Logout</i>	51
4.5 Pengujian Sistem	52
4.5.1 Pengujian White Box	52
4.5.2 Black Box	55
BAB 5. HASIL DAN PEMBAHASAN	64
5.1 Hasil Penerapan <i>Crawling</i> di Media Sosial <i>Twitter</i>	64
5.2 Hasil Dataset	65
5.3 Hasil Text Preprocessing	66
5.4 Hasil Feature Generation	68
5.5 Hasil Klasifikasi Domain	71
5.6 Hasil Klasifikasi Sentimen	73
5.7 Perhitungan Manual Naïve Bayes	76
5.8 Uji Performansi	78
5.9 Hasil Perhitungan Indeks Kebahagiaan	79
5.10 Visualisasi	81
5.10.1 Tampilan Halaman <i>Index</i>	81
5.10.2 Tampilan Halaman Kelebihan Sistem	82
5.10.3 Tampilan Halaman Indeks Kebahagiaan	82
5.10.4 Tampilan Halaman <i>Login</i>	83
5.10.5 Tampilan Halaman Utama Admin	84
5.10.6 Tampilan Halaman <i>Tweet</i>	85
5.10.7 Tampilan Halaman Data <i>Training</i>	86
5.10.8 Tampilan Halaman Data <i>Testing</i>	87
5.10.9 Tampilan Halaman <i>Stopword</i>	87
5.10.10 Tampilan Halaman Ontologi	89
5.10.11 Tampilan Halaman Indeks Kebahagiaan Harian	91
5.10.12 Tampilan Halaman <i>User</i>	93

5.10.13 Tampilan Halaman Utama Ahli Bahasa.....	94
5.10.14 Tampilan Halaman <i>FAQ</i>	95
5.11 Pembahasan Sistem dan Hasil	95
BAB 6. PENUTUP	98
6.1 Kesimpulan.....	98
6.2 Saran.....	99
DAFTAR PUSTAKA	101
LAMPIRAN.....	103



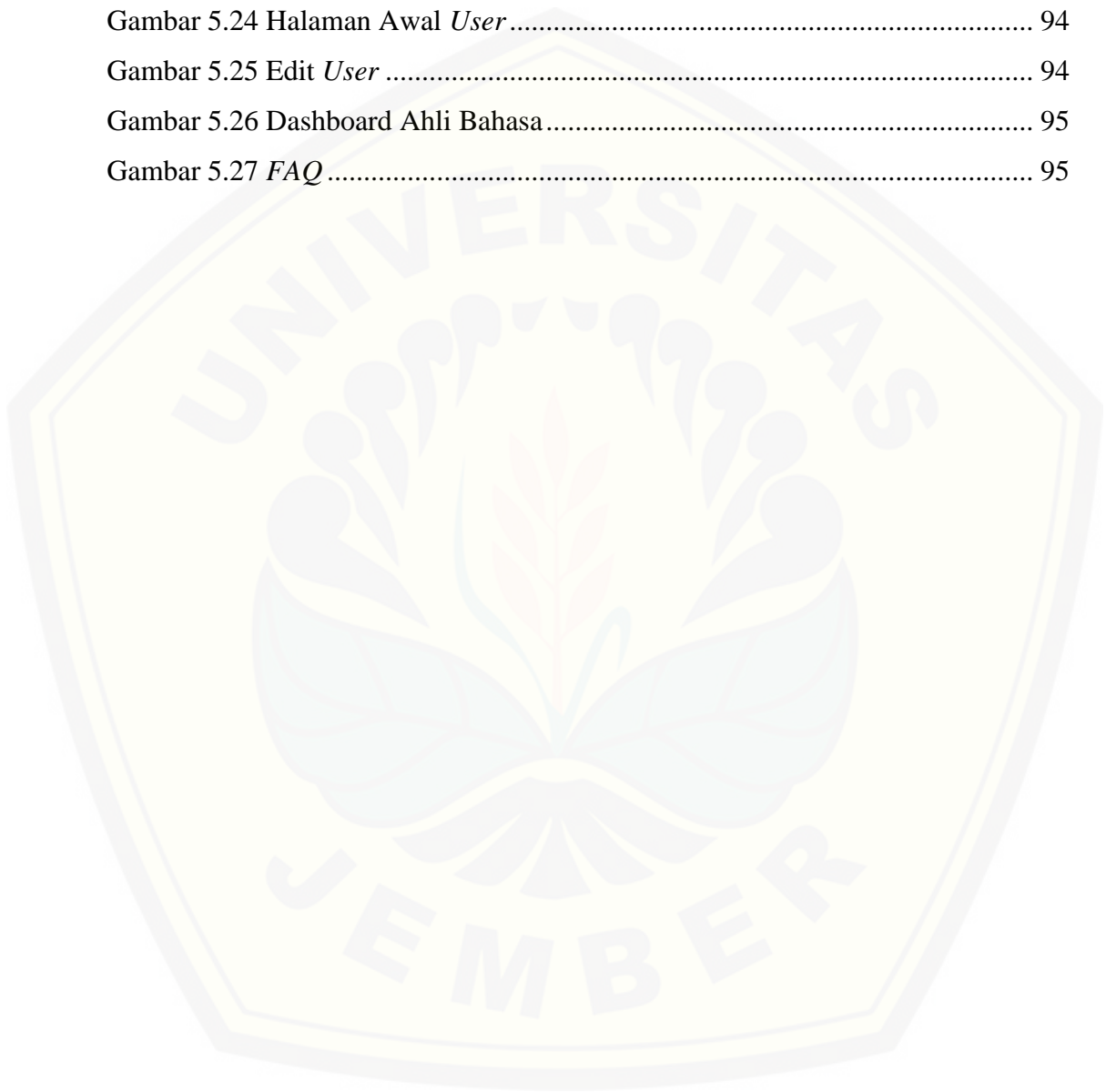
DAFTAR TABEL

Tabel 2.1 Aturan pemenggalan dalam kata ECS	13
Tabel 2.2 Matriks <i>contingency</i> kelas prediksi dan aktual	16
Tabel 3.1 Penjelasan Alur Sistem	21
Tabel 4.1 Definisi Aktor	30
Tabel 4.2 Definisi <i>Usecase</i>	30
Tabel 4.3 Skenario Klasifikasi dan Uji Performansi Data <i>Testing</i>	33
Tabel 4.4 <i>Test case</i> dari <i>function</i> klasifikasi()	53
Tabel 4.5 Hasil pengujian <i>black box</i> pada fitur hasil peramalan	55
Tabel 5.1 Contoh <i>Data Training</i>	65
Tabel 5.2 Hasil <i>Text Preprocessing</i>	67
Tabel 5.3 Hasil <i>Stemming</i>	68
Tabel 5.4 Hasil Penyaringan Kata Dasar	69
Tabel 5.5 Hasil <i>Stopword Removal</i>	70
Tabel 5.6 Contoh Hasil Klasifikasi Domain	72
Tabel 5.7 Data <i>Testing</i> Domain Salah Prediksi	73
Tabel 5.8 Contoh Hasil Klasifikasi Sentimen	74
Tabel 5.9 Data <i>Testing</i> Sentimen Salah Prediksi	75
Tabel 5.10 Perhitungan Probabilitas Kata Pakde	76
Tabel 5.11 Perhitungan Probabilitas Kata Gaji	77
Tabel 5.12 Perhitungan Probabilitas Kata Terima	77
Tabel 5.13 Perhitungan <i>Prior Probability</i>	77
Tabel 5.14 Perhitungan Probabilitas Status	77
Tabel 5.15 Uji Performansi Kelas Sentimen Rata-Rata	78
Tabel 5.16 Uji Performansi Kelas Kategori Rata-Rata	79
Tabel 5.17 Daftar Kata Ontologi	80
Tabel 5.18 <i>Tweet</i> Untuk Indeks Kebahagiaan	80
Tabel 5.19 Hasil Perhitungan Indeks Kebahagiaan Harian	81

DAFTAR GAMBAR

Gambar 3.1 <i>Model waterfall</i>	19
Gambar 3.2 <i>Flowchart</i> Alur Sistem	21
Gambar 4.1 <i>Business Process</i>	28
Gambar 4.2 <i>Use Case Diagram</i>	29
Gambar 4.3 <i>Sequence Diagram</i> Melihat Klasifikasi dan Uji Performansi Data ..	41
Gambar 4.4 <i>Activity Diagram</i> Melihat Klasifikasi	44
Gambar 4.5 <i>Class diagram</i>	46
Gambar 4.6 <i>Entity Relationship Diagram (ERD)</i>	47
Gambar 4.7 Algoritma <i>function klasifikasi()</i>	50
Gambar 4.8 <i>Flowgraph function</i> klasifikasi.....	52
Gambar 5.1 <i>Flowchart Crawling</i>	64
Gambar 5.2 Halaman <i>Index</i>	82
Gambar 5.3 Halaman Kelebihan Sistem	82
Gambar 5.4 Halaman Indeks Kebahagiaan	83
Gambar 5.5 Halaman <i>Login</i>	84
Gambar 5.6 Halaman Buat Akun	84
Gambar 5.7 Tampilan Halaman Utama Admin	85
Gambar 5.8 Tampilan Halaman <i>Tweet</i>	85
Gambar 5.9 Grafik Sentimen Data Dan Kategori Data.....	86
Gambar 5.10 Daftar Data <i>Training</i>	86
Gambar 5.11 Daftar Data <i>Testing</i>	87
Gambar 5.12 Halaman <i>Stopword</i> Awal	88
Gambar 5.13 Input <i>Stopword</i>	88
Gambar 5.14 Edit <i>Stopword</i>	89
Gambar 5.15 Hapus <i>Stopword</i>	89
Gambar 5.16 Halaman Ontologi Awal.....	90
Gambar 5.17 Input Ontologi	90
Gambar 5.18 Edit Ontologi.....	91
Gambar 5.19 Hapus Ontologi	91

Gambar 5.20 Halaman Awal Indeks Kebahagiaan	92
Gambar 5.21 Grafik Indeks Kebahagiaan.....	92
Gambar 5.22 Tombol Klasifikasi Hari Ini	93
Gambar 5.23 Indeks Kebahagiaan Hari ini	93
Gambar 5.24 Halaman Awal <i>User</i>	94
Gambar 5.25 Edit <i>User</i>	94
Gambar 5.26 Dashboard Ahli Bahasa.....	95
Gambar 5.27 <i>FAQ</i>	95



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, indeks kebahagiaan menjadi sesuatu yang diminati oleh beberapa negara di dunia di dalam mengukur tingkat kemakmuran dan kemampuan sosio-ekonomi masyarakat. Hal ini menggeser paradigma pengukuran sebelumnya yang melihat tingkat kemakmuran berdasarkan Produk Domestik Bruto/*Gross Domestic Product* (PDB/GDP). Pergeseran paradigma ini terjadi karena ekonomi yang menjadi indikator dalam pengukuran GDP hanya menjadi salah satu sarana dari tujuan akhir kemakmuran, yaitu kebahagiaan (Bismantara, 2005).

Kata kebahagiaan seringkali menjadi sesuatu yang kabur dan sangat sulit diukur. Apabila seseorang ditanya “seberapa bahagiakah anda?”, maka jawaban yang muncul sangat beragam, karena setiap orang memiliki cara yang berbeda dalam memaknai arti kebahagiaan. Beberapa ahli di dunia, lebih mendefinisikan kebahagiaan sebagai “kesejahteraan subjektif” agar tidak terjadi ambiguitas makna. Sulitnya mengukur tingkat kebahagiaan, bukan berarti kebahagiaan tidak bisa diukur. Beberapa pendekatan telah dilakukan berbagai lembaga untuk mengukur tingkat kebahagiaan, salah satunya pendekatan yang dibuat oleh BPS (Badan Pusat Statistik) Indonesia. Dalam metodologi pengukuran indeks kebahagiaan yang diterbitkan oleh BPS Indonesia, kebahagiaan diukur dalam indeks komposit menggunakan suatu survei yang mencakup 10 domain/variabel yang mampu merefleksikan tingkat kebahagiaan masyarakat, antara lain: (1) pekerjaan, (2) pendapatan rumah tangga, (3) kondisi rumah dan aset, (4) pendidikan, (5) kesehatan, (6) keharmonisan keluarga, (7) hubungan sosial, (8) ketersediaan waktu luang, (9) kondisi lingkungan, dan (10) kondisi keamanan. (Statistik, 2015)

Di satu sisi, penetrasi internet Indonesia menunjukkan angka yang tidak bisa diabaikan. Berdasarkan data yang didapat oleh APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2016, jumlah pengguna internet di Indonesia mencapai 132,7 juta jiwa dari 256,2 juta penduduk Indonesia, dimana media

sosial mendapatkan peringkat tertinggi untuk jenis konten yang sering diakses (129.2 juta jiwa) yang artinya bahwa hampir semua pengguna internet Indonesia memiliki media sosial. Salah satu media sosial yang banyak di akses adalah *twitter* yang menempati peringkat ke-5 (7.2 juta jiwa) di antara media sosial lainnya (APJII, 2016).

Berdasarkan data angka penetrasi tersebut, maka bisa dibuat sebuah pendekatan lain sebagai alternatif untuk pengukuran indeks kebahagiaan. Ponilan dalam jurnalnya yang berjudul Pengukuran *Happiness Index* Masyarakat Kota Bandung pada Media Sosial *Twitter* Menggunakan Pendekatan Ontologi *Top-Down Hierarchy*, membuat sebuah pendekatan pengukuran indeks kebahagiaan melalui sentimen yang ada di media sosial. Pendekatan yang dilakukan dengan mengklasifikasikan sentimen positif dan negatif di sebuah daerah (dalam penelitian ini adalah Bandung) kemudian menghitung *score*-nya. Apabila lebih besar *score* sentimen positif daripada negatif maka dapat disimpulkan masyarakat pada daerah tersebut bahagia dan sebaliknya (Rahayu Ponilan, Herdiani, & Selviandro, 2016).

Penelitian lainnya dilakukan oleh Anisa Herdiani dalam jurnalnya yang berjudul Pengukuran Indeks Kebahagiaan Masyarakat Kota Bandung dari Jejaring Sosial *Twitter* Menggunakan Ontologi dengan Paradigma *Bottom-Up*. Anisa Herdiani menggunakan pendekatan yang sama dengan Ponilan, hanya saja berbeda dalam metode pembangunan ontologi yang dipakai dalam pencarian data *twitter*. Kesimpulan dari dua penelitian tersebut menunjukkan bahwa pendekatan tersebut dapat dipakai dalam mengukur indeks kebahagiaan (Herdiani, Selviandro, & Fitra, 2016).

Dengan adanya pendekatan ini, banyak manfaat yang ditimbulkan dibandingkan pengukuran indeks kebahagiaan secara konvensional seperti pengurangan biaya, pengurangan waktu dan juga pengurangan tenaga yang dikeluarkan karena pengukuran tidak memerlukan survei dan data dapat diambil secara mudah dan cepat melalui sentimen di media sosial. Dalam hal ini, penelitian ini ingin memberikan kontribusi terhadap kemajuan pendekatan ini dengan menghitung indeks kebahagiaan tersebut menggunakan metode *Naive*

Bayes Classifier. Penggunaan *Naive Bayes Classifier* dipilih karena metode klasifikasi ini cukup banyak dipakai dalam melakukan sentimen analisis dan *text mining*, seperti penggunaan *Naive Bayes Classifier* untuk melakukan analisis sentimen terhadap provider telekomunikasi di Indonesia (Manalu, 2014). Objek penelitian pada penelitian ini yaitu menggunakan data dari media sosial *twitter*. Hal ini dikarenakan banyaknya pengguna *twitter* di Indonesia berdasarkan hasil survei APJII. Hasil dari penelitian ini diharapkan mampu memberikan komparasi dan menambah literatur dalam pengembangan pendekatan indeks kebahagiaan melalui analisis sentimen.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang, terdapat beberapa permasalahan yang harus diselesaikan dalam penelitian ini yaitu:

1. Bagaimana menerapkan *crawling* di media sosial *twitter* untuk mengambil *dataset*?
2. Bagaimana menerapkan metode *Naive Bayes Classifier* dalam mengklasifikasikan data *tweet*?
3. Bagaimana uji performansi klasifikasi dalam penerapan *Naive Bayes Classifier*?
4. Bagaimana mengukur indeks kebahagiaan berdasarkan hasil klasifikasi?

1.3 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan sistem, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data diambil menggunakan *Search API Twitter* dengan *API Key* penulis.
2. Data *tweet* yang diambil menggunakan bahasa indonesia yang benar.
3. Satu *tweet* hanya dapat dikelompokkan ke dalam satu kategori dan satu sentiment.
4. *Keyword Query* yang dipakai sesuai dengan kata di ontologi.

5. Algoritma yang digunakan dalam pengklasifikasian adalah *Naive Bayes Classifier* tanpa membandingkannya dengan algoritma lain.
6. Menggunakan Algoritma *Enhanced Confix-Stripping* (ECS) dalam melakukan *stemming*.
7. Dalam menjalankan sistem yang dibangun, apabila *Search API Twitter* yang disediakan *down* maka proses *crawling* juga berhenti.

1.4 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Mengetahui cara penerapan *crawling* untuk mengambil *dataset*.
2. Mengetahui cara penerapan metode *Naive Bayes Classifier* dalam mengklasifikasikan data *tweet*.
3. Mengetahui uji performansi klasifikasi dalam penerapan *Naive Bayes Classifier*.
4. Mengetahui cara mengukur indeks kebahagiaan berdasarkan hasil klasifikasi.
5. Merancang sistem pengukur indeks kebahagiaan menggunakan analisis sentimen di media sosial dengan metode *Naive Bayes Classifier*.

1.5 Sistematika Penulisan

Sistematika penulisan dijelaskan sebagai berikut:

1. Pendahuluan
Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan terkait penelitian yang dilakukan.
2. Tinjauan Pustaka
Bab ini menjelaskan tentang kajian-kajian teori terkait dengan penelitian yang dapat mendukung penelitian yang dilakukan.

3. Metodologi Penelitian

Bab ini menjelaskan tentang jenis penelitian, pengembangan sistem dan pengujian metode metode yang digunakan..

4. Pengembangan Sistem

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem yang dibangun.

5. Hasil dan Pembahasan

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dan manfaat dari penelitian ini seperti apa yang telah ditentukan pada awal penelitian.

6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

2.1 Text Mining

Text Mining dapat didefinisikan sebagai suatu proses menggali informasi yang baru dan tidak diketahui sebelumnya oleh komputer dengan mengekstraksi sumber-sumber berupa data teks tak terstruktur (*unstructured*). Dibandingkan dengan *data mining*, *text mining* sangat sulit untuk diuraikan dengan pendekatan algoritma, namun karena dalam budaya modern, data teks adalah perantara yang paling umum untuk pertukaran informasi maka penggunaan *text mining* menjadi penting (Yanti, 2015).

Informasi yang diperoleh dari *text mining* biasanya melalui peramalan pola dan kecenderungan melalui sarana seperti pembelajaran pola statistik. Dalam prosesnya, *text mining* biasanya melibatkan proses penataan teks input (seperti *parsing* dan penambahan fitur linguistik lainnya), menentukan pola dalam data terstruktur dan akhirnya mengevaluasi dan menginterpretasi *output*. *Text mining* yang berkualitas tinggi mengacu pada kombinasi relevansi, kebaruan dan *interestingness*. Secara umum, *text mining* terbagi menjadi tiga tahapan yaitu: pemrosesan awal terhadap suatu teks (*text preprocessing*), transformasi teks ke dalam bentuk antara (*text Transformation/feature generation*) dan penemuan pola (*pattern discovery*).

2.2.1 Text Preprocessing

Tahapan awal dari *text mining* adalah *text preprocessing*. Tahapan ini bertujuan untuk mempersiapkan data agar dapat digunakan dalam proses selanjutnya. Dalam tahapan ini dilakukan penganalisaan dari segi sintaktik, dimana data teks yang ada diubah menjadi *toLowerCase* (semua karakter menjadi huruf kecil) kemudian dilakukan *Tokenizing* yaitu proses penguraian deskripsi yang semula berupa kalimat-kalimat menjadi kata/*token* dan menghilangkan delimiter-delimiter seperti titik, koma dan tanda baca lainnya (Manalu, 2014).

2.2.2 Text Transformation/Feature Generation

Hasil yang diperoleh dari tahap *text Preprocessing* digunakan pada tahap *Text Transformation/Feature Generation* melalui proses transformasi. Proses

transformasi dilakukan dengan mengurangi jumlah kata-kata yang ada dengan penghilangan *stopword* dan juga dengan mengubah kata-kata ke dalam bentuk dasarnya (*stemming*). *Stopword* adalah kosakata yang bukan merupakan ciri (kata unik) dari suatu dokumen. Sebelum proses *stopword removal* dilakukan, harus dibuat daftar *stopword* (*stoplist*). Jika termasuk di dalam *stoplist* maka kata-kata tersebut dihapus dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi dianggap sebagai kata-kata yang mencirikan isi dari suatu dokumen atau *keywords*. Dengan menghilangkan *stopword* maka beban kerja sistem dapat berkurang karena sistem hanya memperhitungkan kata-kata yang dianggap penting (Manalu, 2014). Dalam penelitian, *stopword* yang digunakan berasal dari Tala.

Selanjutnya adalah proses *stemming*. *Stemming* adalah proses mereduksi kata ke dalam bentuk dasarnya. Menurut Tala *stemming* adalah suatu proses menyediakan suatu pemetaan antara berbagai kata dengan morfologi yang beda walaupun satu bentuk dasar (*stem*). Tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik kata depan (prefiks), kata akhir (sufiks) atau keduanya (confiks). *Stemming* dilakukan karena pada dasarnya kata yang memiliki bentuk dasar sama seharusnya memiliki kedekatan arti. Jika setiap kata diproses tanpa *stemming*, maka satu macam kata dasar disimpan dengan berbagai macam bentuk sesuai imbuhanannya. Hal ini sangat berbeda apabila kita menerapkan proses *stemming* di dalamnya, sehingga *database* yang terpakai lebih efisien (Manalu, 2014). Pada penelitian ini proses *stemming* menggunakan algoritma *Enhanced Confix-Stripping* (ECS) yang merupakan penyempurnaan dari algoritma Nazief dan Adriani.

2.2.3 Pattern Discovery

Tahap penemuan pola atau *pattern discovery* adalah tahap terpenting dari seluruh proses *text mining*. Tahap ini berusaha menemukan pola atau pengetahuan dari keseluruhan teks. Terdapat dua teknik pembelajaran dalam *pattern discovery*, yaitu *unsupervised* dan *supervised learning*. *Supervised learning* adalah teknik pembelajaran dimana terdapat label atau nama kelas pada data latih dan data baru diklasifikasikan berdasarkan data latih. Sedangkan *unsupervised learning* adalah

teknik pembelajaran yang tidak terdapat label atau nama kelas pada data latih, data latih dikelompokkan berdasarkan ukuran kemiripan pada suatu kelas. Pada penelitian ini digunakan teknik pembelajaran *supervised learning* yaitu *Naïve Bayes Classifier* dengan menggunakan data latih dari korpus (kumpulan kata) yang telah diklasifikasikan oleh ahli bahasa (Yanti, 2015).

2.2 Indeks Kebahagiaan

Penelitian mengenai pengukuran indeks kebahagiaan dilakukan oleh Ika Rahayu Ponilan dengan judul jurnal Pengukuran *Index Happiness* Masyarakat Kota Bandung pada Media Sosial *Twitter* Menggunakan Pendekatan Ontologi *Top-Down Hierarchy*. Pada jurnal tersebut dilakukan pengukuran indeks kebahagiaan menggunakan media sosial *twitter* masyarakat Bandung memakai pendekatan Ontologi dan *Support Vector Machine* (SVM). Hasilnya, penelitiannya mampu menghitung indeks kebahagiaan kota Bandung dengan nilai *precision* 64,52%, *recall* 70,12% dan *f1score* 67,20%. Ketidaksesuaian yang terjadi dikarenakan: bobot *term* ontologi yang sama di beberapa kelas, keterbatasan *term* di ontologi, proses *lemimization* yang tidak sempurna dan adanya frasa yang tidak dikenali (Rahayu Ponilan et al., 2016).

Berdasarkan Metodologi Pengukuran Indeks Kebahagiaan yang diterbitkan oleh BPS Indonesia, Indeks kebahagiaan merupakan indeks komposit yang disusun oleh tingkat kepuasan terhadap 10 aspek kehidupan yang esensial. Kesepuluh aspek tersebut secara substansi dan bersama-sama merefleksikan tingkat kebahagiaan yang meliputi kepuasan terhadap: 1) kesehatan, 2) pendidikan, 3) pekerjaan, 4) pendapatan rumah tangga, 5) keharmonisan keluarga, 6) ketersediaan waktu luang, 7) hubungan sosial, 8) kondisi rumah dan aset, 9) keadaan lingkungan, dan 10) kondisi keamanan. (BPS Indonesia, 2016)

2.3 Analisis Sentimen

Analisis Sentimen adalah proses memahami, mengekstrak, dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Proses utama dalam melakukan analisis sentimen yaitu mengelompokkan teks yang terdapat dalam sebuah kalimat atau

dokumen kemudian menentukan polaritas pendapat yang dikemukakan apakah bersifat positif, negatif atau netral (Nurzahputra & Muslim, 2016).

Berdasarkan sumber datanya, analisis sentimen dapat dibedakan menjadi dua kategori, yaitu:

1. *Coarse Grained Sentiment Analysis*

Pada jenis ini, sentimen analisis dilakukan pada tingkat dokumen. *Coarse Grained Sentiment Analysis* menganggap bahwa seluruh isi dokumen sebagai sebuah sentimen positif dan sentimen negatif.

2. *Fined Grained Sentiment Analysis*

Fined Grained Sentiment Analysis adalah *Sentiment Analysis* yang dilakukan pada tingkat kalimat. Pada jenis ini dikatakan bahwa setiap kalimat memiliki sentiment yang berbeda walaupun berada dalam satu dokumen.

Pada penelitian ini analisis sentimen dilakukan pada level dokumen. Dengan asumsi bahwa status sosial media yang didapat merupakan bentuk dari sebuah dokumen. Sosial media yang digunakan dalam penelitian adalah *twitter*.

2.4 Naive Bayes Classifier

Penelitian tentang penggunaan *Naive Bayes Classifier* dalam melakukan klasifikasi sentimen telah banyak dilakukan, salah satunya dilakukan oleh Boy Utomo Manalu dengan skripsi berjudul *Analisis Sentimen Pada Twitter Menggunakan Text Mining*. Dalam skripsi ini dilakukan penelitian tentang penggunaan *Naive Bayes Classifier* untuk melakukan analisis sentiment terhadap provider telekomunikasi di Indonesia. Dari percobaan yang dilakukan, *Naive Bayes Classifier* mampu melakukan klasifikasi sentimen dengan tingkat akurasi 87% pada dataset 1000 dengan data *training* 300 pada masing-masing sentimen (Manalu, 2014).

Penelitian tentang perbandingan antara penggunaan *Naive Bayes Classifier* dan *Support Vector Machine* juga banyak dilakukan, salah satunya dilakukan oleh Ahmad Fathan Hidayatullah dalam Seminar Nasional Aplikasi Teknologi Informasi 2016 yang membahas tentang penerapan *text mining* dalam klasifikasi judul skripsi. Dalam penelitian ini dilakukan *text mining* dalam melakukan klasifikasi judul skripsi dengan dua metode yaitu *Naive Bayes* dan *SVM*. Secara

keseluruhan, hasil perolehan *f1score* pada algoritma *Naïve Bayes* memberikan hasil yang lebih tinggi dibandingkan *SVM* (Hidayatullah, 2016).

Naïve Bayes Classifier (NBC) adalah *classifier* probablistik sederhana yang menerapkan teorema *Bayes* dengan asumsi independensi yang kuat (naïf). Dengan kata lain, NBC mengasumsikan bahwa kehadiran (atau ketiadaan) fitur tertentu dari suatu kelas tidak berhubungan dengan kehadiran (atau ketiadaan) fitur lainnya (Kaku, 2014).

Prediksi *Naïve Bayes* didasarkan pada teorema *Bayes* dengan *formula* untuk klasifikasi sebagai berikut:

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)} \dots\dots\dots(2.6)$$

Sedangkan *Naïve Bayes* dengan fitur kontinu memiliki *formula* :

$$P(X|Y) = \frac{1}{\sqrt{2\pi} \sigma} \exp \frac{-(x-\mu)^2}{2\sigma^2} \dots\dots\dots(2.7)$$

Keterangan :

$P(Y|X)$ = probabilitas data dengan atribut X pada kelas Y (*Posterior Probability*)

$P(Y)$ = probabilitas awal kelas Y (*Prior Probability*)

σ = standar deviasi

μ = *mean* atau nilai rata-rata dari atribut dengan fitur kontinu

$\prod_{i=1}^q P(X_i|Y)$ = Probabilitas independen kelas Y dari semua fitur dalam vektor X (*Likelihood*)

Dalam penelitian ini yang menjadi data uji adalah status (yang telah diubah menjadi bentuk *term*) yang beredar di sosial media, sedangkan untuk *data training* menggunakan ahli bahasa untuk mengklasifikan secara manual sebuah status. Dengan menggunakan *NBC*, setiap *term* status direpresentasikan dengan pasangan atribut “X1, X2, X3,...Xn” dimana X1 adalah kata pertama, X2 adalah kata kedua, dan seterusnya. Sedangkan Y adalah himpunan kategori sentimen dan

juga kategori domain. Klasifikasi ditentukan dengan mencari probabilitas tertinggi dari semua kategori status yang diujikan (Y_{map}) sehingga persamaannya seperti berikut:

$$Y_{map} = \frac{P(X_1, X_2, X_3, \dots, X_n | Y_j) P(Y_j)}{P(X_1, X_2, X_3, \dots, X_n)} \dots\dots\dots(2.8)$$

Karena nilai $P(X_1, X_2, X_3, \dots, X_n)$ bernilai konstan untuk semua kategori (Y_j) sehingga persamaan dapat ditulis

$$Y_{map} = P(X_1, X_2, X_3, \dots, X_n | Y_j) P(Y_j) \dots\dots\dots(2.9)$$

Persamaan di atas dapat disederhanakan menjadi:

$$Y_{map} = \prod_{i=1}^n P(X_i | Y_j) P(Y_j) \dots\dots\dots(2.10)$$

Keterangan :

- Y_j = kategori klasifikasi atau class
- $P(x_i | Y_j)$ = probabilitas x_i pada kategori Y_j
- $P(Y_j)$ = probabilitas *prior* dari kategori (Y_j)

Untuk $P(Y_j)$ dan $P(x_i | Y_j)$ dihitung pada saat pelatihan dimana persamaannya adalah sebagai berikut :

$$P(Y_j) = \frac{|Y_j|}{|Y_{total}|} \dots\dots\dots(2.11)$$

$$P(x_i | Y_j) = \frac{n_k + 1}{n + |kosakata|} \dots\dots\dots(2.12)$$

Keterangan

- $|Y_j|$ = jumlah dokumen setiap kategori j
- $|Y_{total}|$ = jumlah dokumen dari semua kategori
- n_k = jumlah frekuensi kemunculan setiap kata
- n = jumlah frekuensi kemunculan kata dari setiap kategori
- $|kosakata|$ = jumlah semua kata dari semua kategori

2.5 Algoritma Enhanced Confix-stripping

Algoritma *Enhanced Confix-Stripping* (ECS) adalah algoritma *stemming* yang merupakan penyempurnaan dari Algoritma *Confix Stripping* dan algoritma Nazief-Adriani (1996). *Stemming* yang ada pada algoritma ECS ini dibuat berdasarkan aturan morfologi yang ada pada bahasa Indonesia, dengan menggunakan kamus kata dasar dan mendukung penambahan kata dasar yang tidak ada dalam kamus (Tahitoe & Purwitasari, 2010). Aturan morfologi Bahasa Indonesia mengelompokkan imbuhan ke dalam beberapa kategori seperti berikut :

1. *Inflection Suffixes*, adalah kumpulan akhiran yang tidak merubah kata dasar. Dibagi menjadi dua:
 - a. *Particle* (P), yaitu “-lah”, “-kah”, “-tah” dan “-pun”
 - b. *Possessive Pronoun* (PP) atau kata ganti kepunyaan yaitu “-ku”, “-mu” dan “-nya”
2. *Derivation Suffixes*, adalah kumpulan akhiran yang secara angung ditambahkan pada kata dasar, yaitu “-i”, “-kan” dan “-an”.
3. *Derivation Prefixes* (DP) yakni kumpulan awalan pada kata murni atau yang sudah mendapat 2 awalan. *Termasuk* :
 - a. Awalan bermorfologi, seperti “me-”, “be-”, “pe-” dan “te-”
 - b. Awalan tidak bermorfologi, seperti “di-”, “ke-” dan “se-”

Berdasarkan pengklasifikasian imbuhan-imbuhan di atas, maka kata di Indonesia dapat dimodelkan sebagai berikut :

$$[DP + [DP + [DP +]]] \text{ Kata Dasar } [[+ DS] [+PP] [+P]]$$

Pada algoritma ECS terdapat beberapa aturan sebagai berikut:

- a. Ada kombinasi (konfiks) yang tidak diperbolehkan, yaitu imbuhan “be-i”, “di-an”, “ke-i”, “ke-kan”, “me-an”, “se-i”, “se-kan” dan “te-an”
- b. Jika suatu kata hanya terdiri dari dua huruf maka proses *stemming* tidak dilakukan
- c. Penambahan suatu awalan tertentu dapat mengubah bentuk kata dasar, yang dinamakan morfologi. Sebagai contoh, awalan me- dapat berubah menjadi “meng-”, “men-” dll. Dalam penambahan awalan yang bermorfologi, ECS

menggunakan suatu aturan pemenggalan kata, yang dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2.1 Aturan pemenggalan dalam kata ECS

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V... be-rV...
2	berCAP...	ber-CAP... dimana C!=’r’ & P!=’er’
3	berCAerV...	ber-CaerV...dimana C!=’r’
4	belajar...	bel-ajar
5	beC1erC2...	be-C1erC2...dimana C1!={’r’ ’l’}
6	terV...	ter-V... te-rV...
7	terCerV...	ter-CerV...dimana C!=’r’
8	terCP...	ter-CP...dimana C!=’r’ dan P!=’er’
9	teC1erC2...	te-C1erC2...dimana C1!=’r’
10	me{l r w y}V...	me-{l r w y}V...
11	mem{b f v}...	mem-{b f v}...
12	mempe...	mem-pe...
13	mem{rV V}...	me-m{rV V}... me-p{rV V}...
14	men{c dj s z}...	men-{c dj s z}...
15	menV...	me-nV... me-p{rV V}...
16	meng{g h q k}...	meng-{g h q k}
17	mengV...	meng-V... meng-kV... (mengV-...jika V=’e’)
18	menyV...	meny-Sv
19	mempA...	mem-pA...dimana A!=’e’
20	pe{w y}V...	pe-{w y}V...
21	perV...	per-V... pe-rV...
23	perCAP...	per-CAP...dimana C!=’r’ dan P!=’er’
24	perCAerV...	per-CAerV...dimana C!=’r’
25	pem{b f V}...	pem-{b f V}...

Dilanjutkan

Lanjutan

Aturan	Format Kata	Pemenggalan
26	$\text{pem}\{rV V\}\dots$	$\text{pe-m}\{rV V\}\dots \text{pe-p}\{rV V\}\dots$
27	$\text{pen}\{c dj z\}\dots$	$\text{pen-}\{c dj z\}\dots$
28	$\text{pen}V\dots$	$\text{pe-n}V\dots \text{pe-t}V\dots$
29	$\text{peng}C\dots$	$\text{peng-C}\dots$
30	$\text{peng}V\dots$	$\text{peng-V}\dots \text{peng-k}V\dots (\text{peng}V\dots \text{jika } V='e')$
31	$\text{peny}V\dots$	$\text{peny-s}V\dots$
32	$\text{pel}V\dots$	$\text{pe-l}V\dots$ kecuali pelajar yang menghasilkan “ajar”
33	$\text{peCer}V\dots$	$\text{per-er}V\dots$ dimana $C! = \{r w y l m n\}$
34	$\text{peCP}\dots$	$\text{pe-CP}\dots$ dimana $C! = \{r w y l m n\}$ dan $P! = 'er'$
35	$\text{ter}C_1\text{er}C_2\dots$	$\text{ter-C}_1\text{er}C_2\dots$ dimana $C_1! = 'r'$
36	$\text{pe}C_1\text{er}C_2\dots$	$\text{pe-C}_1\text{er}C_2\dots$ dimana $C_1! = \{r w y l m n\}$

Sumber : (Tahitoe & Purwitasari, 2010)

Langkah-langkah algoritma ECS dapat diuraikan seperti berikut (Tahitoe & Purwitasari, 2010):

- Lakukan pengecekan terhadap kata ke kamus kata dasar, jika kata ditemukan berarti *stemming* selesai dan kata merupakan kata dasar. Setelah *stemming* selesai, lakukan pengecekan hasil prosesnya ke kamus kata dasar. Jika kata dasar ditemukan, berarti kata tersebut sudah berbentuk kata dasar. Jika tidak ditemukan, maka dilanjutkan ke proses berikutnya.
- Melakukan proses *rulePrecedence* dimana proses ini menentukan apakah memulai dari penghilangan akhiran terlebih dahulu atau awalan terlebih dahulu. Apabila awalan terlebih dahulu maka proses berikut dilakukan sesuai nomor (5-3-4), sedangkan jika dilakukan penghilangan akhiran terlebih dahulu, maka urutan proses selanjutnya adalah (3-4-5).
- Hilangkan *inflectional suffixes*. Dimulai dari *particle* (P) kemudian *possessive pronoun* (PP).
- Hilangkan *derivation suffixes*.

- e. Hilangkan *derivation prefixes*.
- a. Langkah 5 berhenti jika :
 - 1) Terjadi kombinasi awalan dan akhiran yang terlarang
 - 2) Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
 - 3) Tiga awalan telah dihilangkan
 - b. Identifikasi tipe awalan dan akhiran. Awalan terdiri dari 2 tipe:
 - 1) Tidak bermorfologi, maka awalannya langsung dihilangkan
 - 2) Bermorfologi, gunakan aturan pada tabel untuk mendapatkan hasil pemenggalan yang tepat
 - c. Cari kata yang telah dihilangkan awalannya ini dalam kamus kata dasar. Apabila tidak ditemukan maka ulangi langkah 5, apabila ditemukan maka proses dihentikan.
 - d. Apabila sampai langkah ke 5, kata dasar belum ditemukan. Maka dilakukan proses *recoding* berdasarkan tabel 1. *Recoding* dilakukan dengan menambahkan karakter *recoding* setelah tanda hubung ('-') dan terkadang berada sebelum tanda kurung. Sebagai contoh : menulis. Dalam tabel 2.1, menulis masuk ke dalam aturan 15, dimana kata dasar yang dihasilkan adalah "nulis". Karena tidak valid, maka dilakukan proses *recoding* sehingga kata dasar yang didapat menjadi "tulis".
 - e. Jika semua langkah yang dilakukan tidak menemukan kata dasar, maka inputan awal dijadikan sebagai kata dasar.

2.6 Twitter API

Twitter API yaitu sebuah aplikasi yang diciptakan oleh pihak *twitter* agar mempermudah pihak *developer* lain untuk mengakses informasi web *twitter* tersebut dengan ketentuan dan syarat yang berlaku seperti yang terdapat pada <https://dev.twitter.com/oauth>. Untuk dapat menggunakan *Twitter API*, pengguna diwajibkan *login* di alamat <https://dev.twitter.com> untuk mendapatkan 4 buah *key* berupa *consumer key*, *consumer secret*, *access token* dan *access token secret* yang

akan digunakan sebagai syarat *authentication* untuk dapat mengakses data *twitter* (Sussolaikah & Alwi, 2016). Ada beberapa jenis *Twitter API*, diantaranya:

1. *Twitter REST API*

Terdiri dari *Twitter REST* dan *Twitter Search*. *Twitter REST API* memberikan *core data* dan *core twitter objects*. *Twitter Search API* berfungsi berfungsi untuk mencari kata kunci tertentu dalam *tweet*, menemukan *tweet* referensi pengguna tertentu, atau mencari *tweets* dari pengguna tertentu. Dengan *API* ini pengguna dapat memiliki akses ke data tren.

2. *Twitter Streaming API*

API ini biasa digunakan untuk penggalian data karena melalui *API* ini informasi bisa didapatkan secara *realtime* dengan volume yang sangat tinggi

2.7 Uji Performansi

Pengujian klasifikasi diukur dengan uji performansi yang meliputi *precision*, *recall* dan *f1score*. (Rahayu Ponilan et al., 2016) Untuk mempermudah pemahaman mengenai uji performansi dapat dilihat pada Tabel 3.1.

Tabel 2.2 Matriks *contingency* kelas prediksi dan aktual

		Predicted Class	
		Class = Yes	Class = No
Actual Class	Class = Yes	TP	FN
	Class = No	FP	TN

Sumber : (Rahayu Ponilan et al., 2016)

Keterangan :

- TP(*True Positive*) adalah kelas yang diprediksi yes, dan ternyata faktanya yes
- TN(*True Negative*) adalah kelas yang diprediksi no, dan ternyata faktanya no
- FP(*False Positive*) adalah kelas yang diprediksi yes, dan ternyata faktanya no
- FN(*False Negative*) adalah kelas yang diprediksi no, dan ternyata faktanya yes

Precision adalah rasio jumlah ketepatan prediksi suatu kelas terhadap jumlah total prediksi yang diklasifikasikan dalam kelas tersebut. Rumus mencari *precision* dapat ditulis sebagai berikut

$$precision (P) = \frac{TP}{TP+FP} \dots \dots \dots (3.1)$$

Recall adalah rasio jumlah ketepatan prediksi suatu kelas terhadap jumlah total fakta yang diklasifikasikan dalam kelas tersebut. Rumus mencari *recall* dapat ditulis sebagai berikut

$$recall (P) = \frac{TP}{TP+FN} \dots \dots \dots (3.2)$$

Untuk menggabungkan keduanya maka digunakan perhitungan *f1score*. Rumus mencari *f1score* dapat ditulis sebagai berikut

$$f1_{score} = \frac{2*precision*recall}{precision+recall} \dots \dots \dots (3.3)$$

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, pengembangan sistem dan pengujian metode *Naive Bayes Classifier*.

3.1 Metode Penelitian

Metode penelitian yang dilakukan merupakan penelitian pengembangan. Penelitian pengembangan bertujuan untuk mengembangkan sebuah produk tertentu dan menguji keefektifan produk tersebut. Pada penelitian ini, produk yang dikembangkan adalah sebuah sistem pengukuran indeks kebahagiaan menggunakan sentiment di media sosial.

3.2 Metode Pengumpulan Data

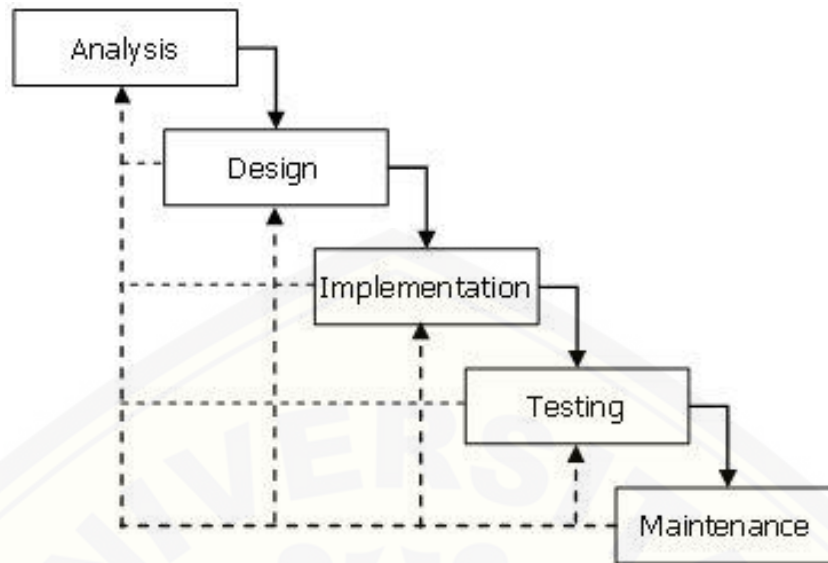
Data yang diperoleh untuk melakukan penelitian didapat langsung dari *tweet* menggunakan *Twitter Search API* dengan *API Key* dari penulis.

3.3 Metode Klasifikasi

Klasifikasi sentimen dan kategori menggunakan *Naive Bayes Classifier*. *Naive Bayes Classifier* dipakai karena beberapa penelitian terdahulu telah menggunakan *Naive Bayes Classifier* dalam melakukan klasifikasi sentimen. Klasifikasi manual dilakukan oleh mahasiswa Program Studi Bahasa Indonesia Fakultas Sastra Universitas Jember dengan melihat semantik dari data *tweet*.

3.1 Metode Pengembangan Software

Metode pengembangan software dalam penelitian ini menggunakan *Software Development Life Cycle (SDLC) waterfall*. Model *SDLC Waterfall* adalah proses pengembangan perangkat lunak sekuensial di mana perkembangan terlihat seperti mengalir semakin ke bawah (mirip dengan air terjun) melalui daftar tahapan yang harus dijalankan agar berhasil membangun sebuah perangkat lunak komputer (Bassil, 2012).



Gambar 3.1 Model waterfall

3.3.1 Analisis Kebutuhan (*Requirement*)

Tahapan awal dalam perancangan dan pengembangan sistem ini adalah analisis kebutuhan-kebutuhan tentang bagaimana sistem pengukur indeks kebahagiaan ini dibangun. Pada tahapan ini, peneliti melakukan beberapa studi literatur terkait algoritma yang dipakai, metode *crawling* data dan studi literatur lainnya yang dibutuhkan untuk pengembangan sistem.

3.3.2 Desain Sistem

Tahap desain sistem yang dibangun menggunakan *Unified Modeling Language* (UML) yang mendukung konsep pemodelan *programming* berbasis objek. Pada tahap ini diperoleh dokumentasi antara lain:

1. *Business Process* digunakan untuk mendefinisikan aktifitas dan proses.
2. *Use Case Diagram* digunakan untuk mendefinisikan fungsional sistem.
3. *Scenario* digunakan untuk menjelaskan fitur sistem.
4. *Sequence Diagram* digunakan untuk menunjukkan rangkaian pesan yang sikirim antar object juga interaksi antar object.
5. *Class diagram* digunakan untuk menggambar struktur statis class dalam sistem.

6. *Entity Realtionship Diagram* digunakan untuk menunjukkan relasi antar object.

3.3.3 Implementasi

Tahap implementasi adalah tahap yang mengkonversi desain sistem yang telah dibuat ke dalam bentuk kode program. Sistem yang dibangun ditulis dengan bahasa PHP (*Page Hypertext Pre-processor*) dan menggunakan PHPStorm sebagai IDE-nya. Sistem ini menggunakan *local server* dan *database* yang terdapat pada aplikasi XAMPP.

3.3.4 *Testing* dan Evaluasi

Tahap selanjutnya adalah tahap *testing*. Pada penelitian ini dilakukan tiga metode pengujian sistem yaitu:

1 *White box testing*

White box testing Merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program-program yang ada. Pengujian ini, dilakukan oleh *developer*. Jika ada modul yang menghasilkan *output* yang tidak sesuai maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu dicek dan diperbaiki, kemudian di *compile* ulang.

2 *Black box testing*

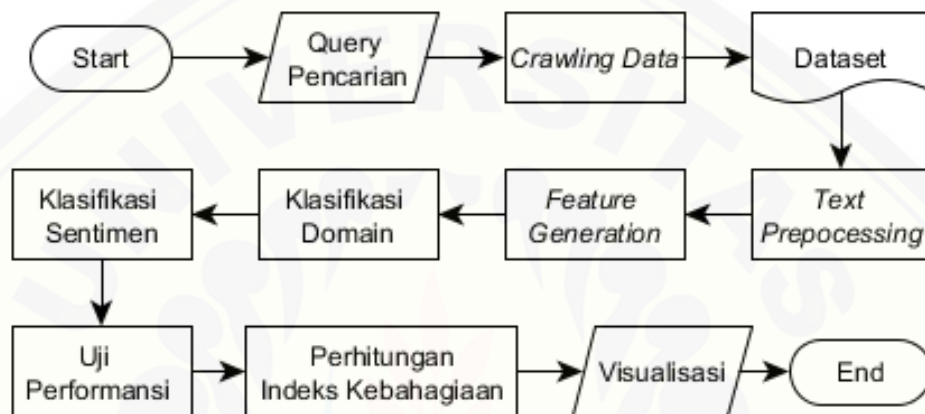
Pengujian *black box* melibatkan pengguna/*user*, dimana hanya memperhatikan fungsionalitas yang berkaitan dengan masukan/keluaran (I/O) apakah sesuai dengan sistem yang dijalankan. Pengujian dilakukan oleh beberapa mahasiswa yang diambil secara acak, menggunakan kuesioner.

3.3.5 Pemeliharaan

Pemeliharaan diadakan untuk mengatasi masalah pada sistem dilain waktu ketika aplikasi sudah dapat digunakan oleh *user*. Selama *user* menemui bug pada sistem, maka *user* langsung konfirmasi kepada *developer* untuk segera ditangani.

3.4 Gambaran Algoritma Sistem

Sistem pengukur indeks kebahagiaan yang dibangun adalah sebuah sistem berbasis web yang dapat mengukur indeks kebahagiaan suatu daerah dengan memanfaatkan analisis sentimen pada media sosial. Algoritma yang diterapkan dalam sistem tersebut dapat dilihat sebagai pada Gambar 3.2 dengan penjelasan alur sistem pada Tabel 3.1.



Gambar 3.2 Flowchart Alur Sistem

Tabel 3.1 Penjelasan Alur Sistem

No	Tahap	Input	Proses	Output
1.	Query Pencarian	-	Pada tahap ini pencarian dimasukkan, contoh kata “teman” untuk mencari dataset yang berhubungan dengan hubungan sosial.	Query dataset
2.	Crawling Data	Query, Twitter Search API	Crawling data dilakukan dengan menggunakan Twitter Search API dan dicocokkan dengan query yang telah diinputkan sebelumnya	Dataset

Dilanjutkan

Lanjutan

No	Tahap	Input	Proses	Output
3.	<i>Text Preprocessing</i>	<i>Dataset</i>	<ol style="list-style-type: none"> 1. Pengambilan satu <i>dataset</i>/konten/status. Pengambilan ini nantinya dilakukan pengulangan, hingga seluruh <i>dataset</i> berhasil dianalisa. 2. Pengubahan <i>dataset</i> yang ada menjadi <i>lowercase</i> 3. Penghilangan tanda baca titik, koma, dan tanda baca lainnya yang tidak diperlukan 	<i>Dataset</i> yang telah menjadi <i>lowercase</i> dan dihilangkan tanda bacanya
4.	<i>Feature Generation</i>	<i>Dataset</i> yang telah di <i>lowercase</i> dan dihilangkan tanda bacanya	<ol style="list-style-type: none"> 1. <i>Stemming</i> Penghilangan imbuhan, baik prefiks (kata depan), sufiks (kata akhir), konfiks. <i>Stemming</i> dilakukan agar tidak terjadi duplikasi dalam <i>database</i>. Algoritma <i>stemming</i> yang dipakai adalah algoritma <i>enhanced confix-stripping</i> 2. Penghilangan kata yang bukan kata dasar. Kata-yang telah di-<i>stemming</i>, selanjutnya dilakukan pengecekan kata dengan kamus kata dasar, apabila kata tidak ditemukan sebagai kata dasar maka kata dihilangkan karena dianggap tidak menggunakan bahasa indonesia yang benar. 	<i>Term</i> yang telah dilakukan <i>stemming</i> , penghilangan kata dasar dan penghilangan <i>stopword</i>

Dilanjutkan

Lanjutan

No	Tahap	Input	Proses	Output
			3. Penghilangan <i>stopword</i> atau kata-kata yang tidak mengandung sentimen seperti “di”, “pada”, “oleh”. Daftar <i>stopword</i> diinputkan dalam <i>stoplist</i> . 4. Pemisahan menjadi <i>term</i>	
5.	Klasifikasi Domain	<i>Term</i> yang telah dilakukan <i>stemming</i> , penghilangan kata dasar dan penghilangan <i>stopword</i>	Penghitungan klasifikasi domain/parameter dari <i>term</i> yang telah terbentuk. Klasifikasi menggunakan <i>Naive Bayes Classifier</i> .	Hasil klasifikasi domain
6.	Klasifikasi Sentimen	<i>Term</i> yang telah dilakukan <i>stemming</i> , penghilangan kata dasar dan penghilangan <i>stopword</i>	Penghitungan klasifikasi sentimen dari <i>term</i> yang telah terbentuk. Klasifikasi menggunakan <i>Naive Bayes Classifier</i> .	Hasil klasifikasi sentimen
7.	Uji Performansi	Hasil klasifikasi sentimen dan klasifikasi domain	Hasil klasifikasi diuji performansinya.	Uji Performansi Klasifikasi
8.	Perhitungan Indeks Kebahagiaan	Hasil klasifikasi sentimen dan klasifikasi domain	Tahap ini dilakukan penghitungan indeks kebahagiaan menggunakan perbandingan jumlah sentimen positif dengan jumlah sentiment negatif.	Indeks Kebahagiaan

Dilanjutkan

Lanjutan

No	Tahap	Input	Proses	Output
9.	Visualisasi	Indeks Kebahagiaan, Hasil Klasifikasi Sentimen, Hasil Klasifikasi Domain, <i>Dataset</i>	Semua data dan penghitungan yang telah dilakukan divisualisasikan dalam bentuk statistik dan grafik.	Visualisasi data



BAB 4. PERANCANGAN SISTEM

Bab 4 sistem menjelaskan tentang perancangan sistem pengukur indeks kebahagiaan menggunakan sentimen di media sosial menggunakan SDLC Waterfall . Perancangan sistem ini terdiri dari analisis kebutuhan sistem, *business process*, *usecase diagram*, *scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan *entity relationship diagram* (ERD).

4.1 Deskripsi Umum Sistem

Sistem pengukur indeks kebahagiaan menggunakan sentimen di media sosial merupakan sebuah sistem informasi berbasis website yang mampu mengukur indeks kebahagiaan menggunakan sentimen di media sosial dengan metode *Naive Bayes Classifier*. Data yang digunakan berasal dari data *tweet* pada *twitter* menggunakan *Twitter Search API* yang digunakan sebagai *dataset*. Setiap *dataset* diklasifikasikan sentimen dan kategorinya. Hasil dari klasifikasi tersebut dilakukan uji performansi dengan menghitung *recall*, *precision* dan *f1score*. Indeks kebahagiaan didapat dengan membandingkan jumlah sentimen positif dan negatif hasil dari klasifikasi sentimen, yang kemudian di visualisasi dalam bentuk grafik agar mudah dibaca.

4.2 Analisis Kebutuhan

Tahap ini merupakan tahap dalam menentukan kebutuhan-kebutuhan apa saja yang dapat dilakukan oleh sistem. Kebutuhan sistem dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Analisis kebutuhan sistem didapatkan dengan penganalisaan pada jurnal yang memiliki objek penelitian yang sama dan studi terhadap algoritma yang dipakai.

4.2.1 Studi Literatur

Studi literatur yang digunakan berasal dari jurnal, buku dan penelitian terdahulu. Studi literatur dilakukan terhadap algoritma *Naive Bayes Classifier*, penerapan *text mining*, penghitungan indeks kebahagiaan, penggunaan ontologi dan penggunaan *sentiword* sehingga didapatkan kebutuhan fungsional dan non fungsional yang sesuai dan dapat digunakan dalam melakukan penghitungan

indeks kebahagiaan menggunakan analisis sentimen di media sosial. Melalui studi literatur yang telah dilakukan, didapatkan hasil berupa data dan informasi yang selanjutnya diolah untuk menganalisis kebutuhan fungsional dan non fungsional.

4.2.2 Kebutuhan fungsional

1. Sistem menggunakan fitur *login* untuk mengautentikasi hak akses pengguna sistem. Kebutuhan ini merupakan kebutuhan yang umum dipakai dalam membuat sebuah sistem.
2. Sistem dapat menyimpan data *user* pada registrasi *user*. Kebutuhan ini merupakan kebutuhan yang umum dipakai dalam membuat sebuah sistem.
3. Sistem dapat mengelola data *user* yang meliputi *view*, *insert*, *edit* dan *delete*. Kebutuhan ini merupakan kebutuhan yang umum dipakai dalam membuat sebuah sistem.
4. Sistem dapat melakukan *crawling* data *twitter*. Kebutuhan ini dimunculkan agar sistem dapat memiliki dataset yang digunakan di dalam melakukan penelitian.
5. Sistem dapat melakukan klasifikasi sentimen dan kategori secara manual. Kebutuhan ini didapatkan setelah melakukan studi literatur terhadap metode *Naïve Bayes Classifier*, dimana untuk melakukan pembelajaran pada data training dibutuhkan klasifikasi secara manual. Selain itu, kebutuhan ini digunakan dalam melakukan uji performansi, dimana data testing yang ingin diuji, terlebih dahulu diketahui kelas yang seharusnya didapatkan dari pengklasifikasian manual.
6. Sistem dapat melakukan klasifikasi sentimen dan kategori secara otomatis menggunakan metode *Naive Bayes Classifier*. Kebutuhan ini didapat setelah melakukan studi literatur terhadap metode *Naive Bayes Classifier*, dimana *NBC* merupakan metode yang dipakai untuk melakukan pengklasifikasian yang dibutuhkan untuk mendapatkan indeks kebahagiaan.
7. Sistem dapat menampilkan indeks kebahagiaan sesuai dengan hasil klasifikasi. Kebutuhan ini didapatkan karena merupakan tujuan dari sistem yaitu menghasilkan indeks kebahagiaan.

8. Sistem dapat mengelola data stopword yang meliputi *view*, *insert*, *edit* dan *delete*. Kebutuhan ini didapatkan setelah melakukan studi literatur pada *stopword*. *Stopword* adalah kata-kata yang tidak mengandung sentimen seperti “di”, “pada”, “oleh”. Dalam penelitian ini *stopword* berasal dari Tala, namun agar sistem mampu berjalan dinamis, dimunculkan kebutuhan sistem untuk dapat mengelola data *stopword*.
9. Sistem dapat mengolah data ontologi/*keyword* yang meliputi *view*, *insert*, *edit* dan *delete*. Kebutuhan ini didapat setelah melakukan studi literatur pada ontologi. Untuk mendapatkan indeks kebahagiaan yang dapat dihitung secara harian, dibutuhkan *keyword* yang dipakai untuk membatasi lingkup pencarian. *Keyword* yang dipakai didapat dengan mempelajari arti kata tersebut kemudian memberikan pernyataan bahwa kata tersebut termasuk ke dalam lingkup pencarian. Agar sistem dapat berjalan dinamis, *keyword* dapat dilakukan perubahan sehingga dimunculkan kebutuhan ini.

4.2.3 Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang merupakan sekumpulan karakteristik dan properti pada sistem

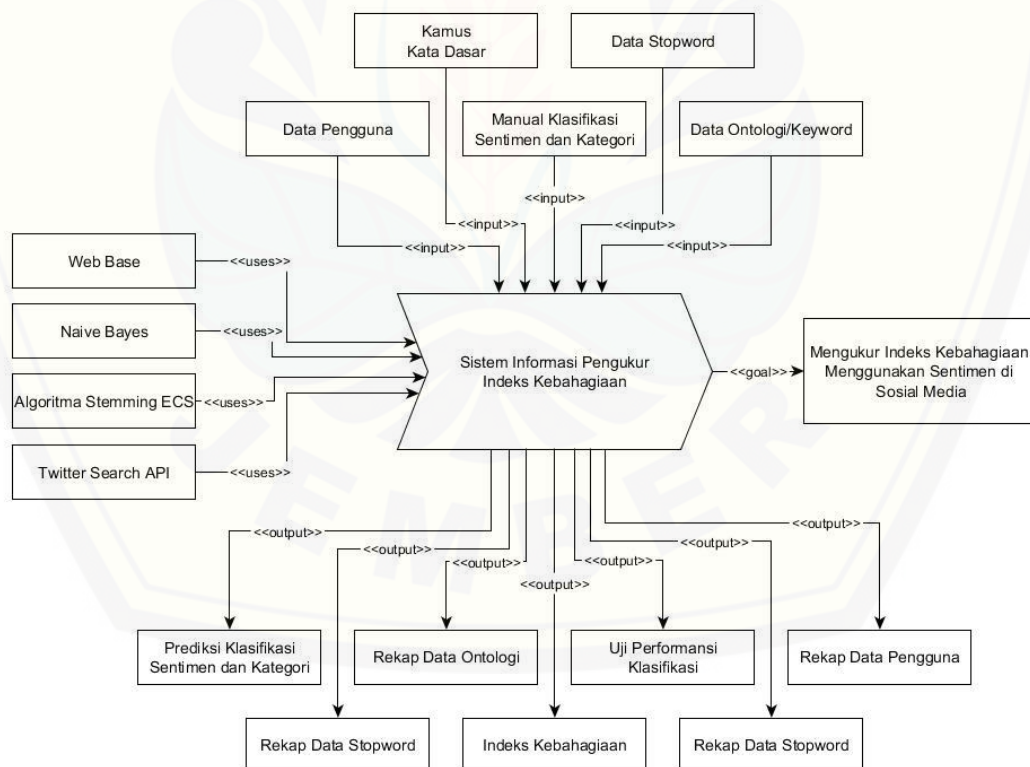
1. Sistem menggunakan autentikasi berupa *username*, *email address* dan *password*. Kebutuhan ini merupakan kebutuhan yang umum dipakai dalam membuat sebuah sistem dan merupakan *default* dari *framework* laravel.
2. Sistem menggunakan *Search API Twitter* untuk *crawling* data. Kebutuhan ini didapatkan setelah melakukan studi literatur tentang *Twitter API* dan *Search API Twitter* merupakan *API* yang gampang digunakan dibandingkan dengan yang lain.
3. Sistem berbasis website. Website dipakai agar sistem dapat diakses dimanapun dan informasi indeks kebahagiaan dapat dengan mudah diterima.
4. Tampilan website yang *responsive*. Kebutuhan ini bertujuan untuk memudahkan pengguna dalam mengakses aplikasi di semua komputer.

4.3 Desain Sistem

Desain sistem yang dibuat meliputi *bussines process*, *use case diagram*, *use case scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan *entity relationship diagram* (ERD).

4.3.1 Business Process

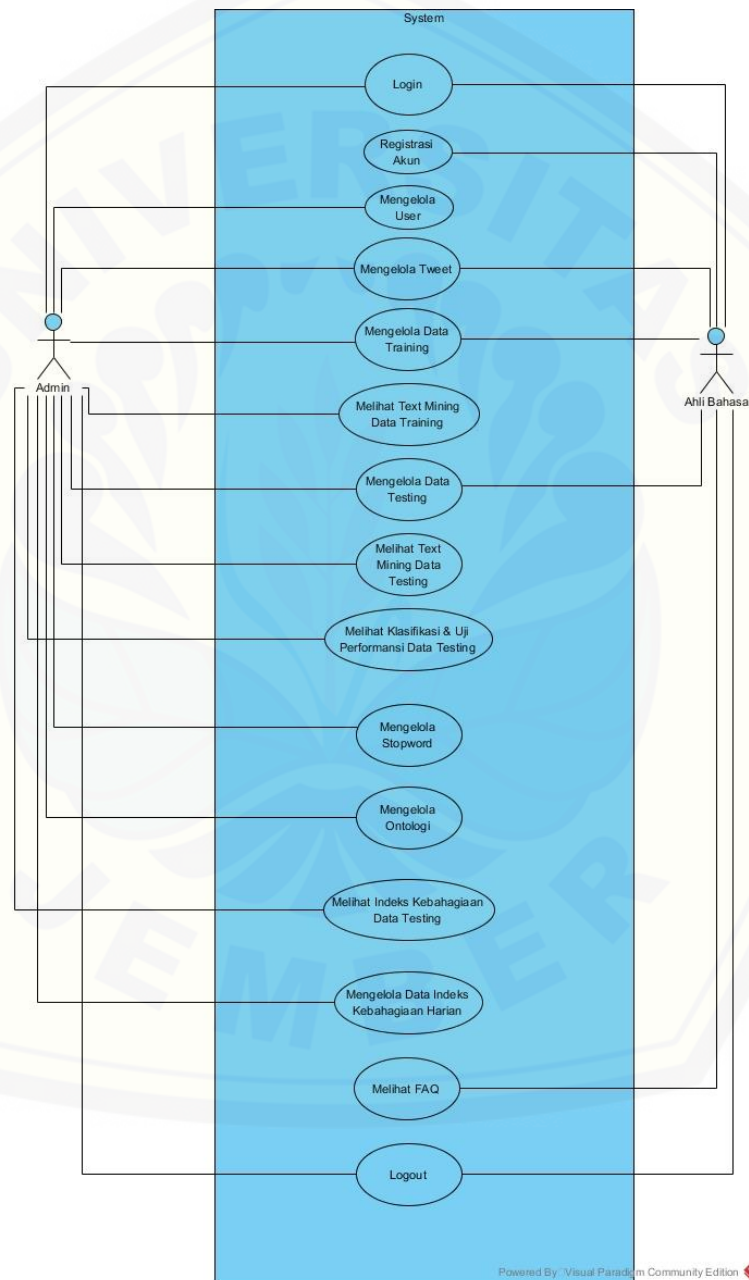
Business Process merupakan suatu kumpulan dari aktivitas atau pekerjaan terstruktur yang saling berhubungan untuk menghasilkan suatu produk. Ada beberapa komponen yang terdapat dalam *business process* diantaranya data yang menjadi masukan sistem (*input*), data masukan yang telah diolah sehingga menghasilkan data keluaran (*output*), media yang digunakan (*uses*), serta tujuan yang ingin dicapai (*goal*). *Business process* pada sistem ini dapat dilihat pada gambar 4.1.



Gambar 4.1 *Business Process*

4.3.2 Use Case Diagram

Use Case Diagram merupakan gambaran fungsional dari sistem yang dapat menunjukkan fitur-fitur yang disediakan oleh sistem serta aktor yang dapat mengakses fitur tersebut. *Use Case Diagram* pada sistem ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Use Case Diagram

Gambar 4.2 menunjukkan *use case diagram* sistem pengukur indeks kebahagiaan menggunakan analisis sentimen di media sosial. *Use case* sistem pengukur indeks kebahagiaan ini mempunyai penjabar berupa tabel definisi aktor yang menggambarkan tugas-tugas aktor dalam mengoperasikan sistem tersebut dapat dilihat pada Tabel 4.1 Definisi Aktor. Penjelasan lainnya yaitu disebut definisi *use case* yang menggambarkan fungsionalitas dari setiap *usecase* dapat dilihat pada Tabel 4.2 Definisi *Use Case*.

1. Definisi Aktor

Definisi aktor yaitu penjelasan mengenai aktor-aktor sebagai pengguna sistem pengukur indeks kebahagiaan menggunakan analisa sentimen media sosial yang dibangun. Terdapat 2 Aktor seperti yang dijelaskan pada Tabel 4.1

Tabel 4.1 Definisi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor Admin pada sistem ini memiliki hak akses untuk mengelola <i>user</i> , mengelola <i>tweet</i> , mengelola <i>training</i> , melihat <i>text mining</i> data <i>training</i> , mengelola data <i>testing</i> , melihat <i>text mining</i> data <i>testing</i> , melihat klasifikasi & uji performansi data <i>testing</i> , melihat indeks kebahagiaan data <i>testing</i> , mengelola <i>stopword</i> , mengelola ontologi, mengelola data indeks kebahagiaan harian.
2.	Ahli Bahasa	Aktor ahli bahasa pada sistem ini memiliki peran untuk dapat mengelola data <i>training</i> , mengelola data <i>testing</i> , melihat <i>faq</i> , mengelola <i>tweet</i> .

2. Definisi Usecase

Definisi *Usecase* yaitu penjelasan tentang fitur-fitur yang tersedia dalam sistem pengukur indeks kebahagiaan menggunakan analisa sentimen media sosial. Terdapat 15 *usecase* dijelaskan pada Tabel 4.2.

Tabel 4.2 Definisi *Usecase*

No.	<i>Usecase</i>	Deskripsi
1.	<i>Login</i>	Menggambarkan proses autentifikasi <i>user</i> yang digunakan untuk masuk ke sistem.
2.	Registrasi Akun	Menggambarkan proses pendaftaran akun <i>user</i> baru

Dilanjutkan

Lanjutan

No.	Usecase	Deskripsi
3.	Mengelola <i>User</i>	Menggambarkan proses pengelolaan data <i>user</i> meliputi melihat melihat, mengedit dan menghapus <i>user</i> .
4.	Mengelola <i>Tweet</i>	Menggambarkan proses pengelolaan data <i>tweet</i> meliputi melihat melihat, mengedit dan menghapus <i>tweet</i> .
5.	Mengelola <i>Data Training</i>	Menggambarkan proses pengelolaan <i>data training</i> meliputi melihat menambahkan, melihat, mengedit dan menghapus <i>data training</i> .
6.	Melihat <i>Text Mining Data Training</i>	Menggambarkan proses untuk melihat <i>teks mining data training</i> .
7.	Mengelola <i>Data Testing</i>	Menggambarkan proses pengelolaan <i>data testing</i> meliputi melihat menambahkan, melihat, mengedit dan menghapus <i>data testing</i> .
8.	Melihat <i>Text Mining Data Testing</i>	Menggambarkan proses untuk melihat <i>teks mining data testing</i> .
9.	Melihat <i>Klasifikasi & Uji Performansi Data Testing</i>	Menggambarkan proses untuk melihat klasifikasi & uji performansi <i>data testing</i> .
10.	Melihat <i>Indeks Kebahagiaan Data Testing</i>	Menggambarkan proses untuk melihat indeks kebahagiaan <i>data testing</i> .
11.	Mengelola <i>Stopword</i>	Menggambarkan proses pengelolaan data <i>stopword</i> meliputi melihat menambahkan, melihat, mengedit dan menghapus <i>stopword</i> .
12.	Mengelola <i>Ontologi</i>	Menggambarkan proses pengelolaan data ontologi meliputi melihat menambahkan, melihat, mengedit dan menghapus ontologi.
13.	Mengelola <i>Data Indeks Kebahagiaan Harian</i>	Menggambarkan proses pengelolaan data indeks kebahagiaan harian meliputi melihat menambahkan, melihat, mengedit dan menghapus data indeks kebahagiaan harian.
14.	Melihat <i>FAQ</i>	Menggambarkan untuk melihat <i>FAQ</i>
15.	<i>Logout</i>	Menggambarkan proses keluar dari sistem.

4.3.3 Skenario

Skenario merupakan penjabaran alur kerja sistem yang terdapat pada *usecase diagram*. Skenario sistem pengukuran indeks kebahagiaan menggunakan sentimen di media sosial adalah sebagai berikut:

4.3.3.1 Skenario *Login*

Skenario *login* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor masuk ke dalam sistem. Hanya aktor yang memiliki *email* dan *password* yang dapat mengakses sistem. Terdapat dua *user* dalam sistem ini yaitu admin dan ahli bahasa. Skenario *login* dapat dilihat pada lampiran A.1.

4.3.3.2 Skenario Registrasi Akun

Skenario registrasi akun merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melakukan registrasi. aktor dapat memasukkan *username*, *e-mail*, dan *password* yang telah dimiliki sebelumnya. Skenario registrasi akun dapat dilihat pada lampiran A.2.

4.3.3.3 Skenario *User*

Skenario mengolah *user* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengolah *user*. Aktor dapat melihat, mengedit, dan menghapus data *user*. Skenario mengelola *user* dapat dilihat pada lampiran A.3..

4.3.3.4 Skenario Mengelola *Tweet*

Skenario mengelola *tweet* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola *tweet*. Aktor dapat melakukan proses *tweet* dengan memilih menu *tweet*. Skenario mengelola *tweet* dapat dilihat pada lampiran A.4.

4.3.3.5 Skenario Mengelola Data *Training*

Skenario mengelola data *training* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data *training*. Aktor dapat melihat, mengedit, dan menghapus data *training* dengan memilih menu *training* tabel. Skenario mengelola data *training* dapat dilihat pada lampiran A.5.

4.3.3.6 Skenario Melihat *Text Mining* Data *Training*

Skenario melihat *text mining* data *training* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat *text mining* data *training*. Aktor dapat melihat *text mining* data *training* dengan memilih menu data *training text mining*. Skenario melihat *text mining* data *training* dapat dilihat pada lampiran A.6.

4.3.3.7 Skenario Mengelola Data *Testing*

Skenario mengelola data *testing* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data *testing*. Aktor dapat melihat, mengedit dan menghapus data *testing* dengan memilih menu data *testing* tabel. Skenario mengelola data *testing* dapat dilihat pada lampiran A.7.

4.3.3.8 Skenario Melihat *Text Mining* Data *Testing*

Skenario melihat *text mining* data *testing* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat *text mining* data *testing*. Aktor dapat melihat *text mining* data *testing* dengan memilih menu data *testing text mining*. Skenario melihat *text mining* data *testing* dapat dilihat pada lampiran A.8..

4.3.3.9 Skenario Melihat Klasifikasi dan Uji Performansi Data *Testing*

Skenario melihat klasifikasi dan uji performansi data *testing* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat klasifikasi dan uji performansi data *testing*. Aktor dapat melihat klasifikasi dan uji performansi dengan memilih menu *testing* klasifikasi. Klasifikasi dan uji performansi dapat ditampilkan dengan menampilkan halaman data tabs klasifikasi kategori dan tabs uji performansi sentimen. Skenario melihat klasifikasi dan uji performansi data *testing* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Skenario Klasifikasi dan Uji Performansi Data *Testing*

No. Usecase	UC09
Nama Usecase	Melihat Klasifikasi & Uji Performansi Data <i>Testing</i>

Dilanjutkan

Lanjutan

Aktor	Admin
PreKondisi	Admin memilih menu <i>Testing</i> Klasifikasi
PostKondisi	Admin berhasil melihat Klasifikasi & Uji Performansi Data <i>Testing</i>

Aliran normal

Lihat data klasifikasi

1. Klik menu <i>Testing</i> Klasifikasi	2. Menampilkan halaman data <i>Testing</i> Klasifikasi yang berisi: <ul style="list-style-type: none"> - <i>Dropdown Show</i> - Tabs Klasifikasi Kategori - Tabs Uji Performansi Sentimen - Tabs Uji Performansi Kategori - Tabs Indeks Kebahagiaan Tabel yang berisi <ul style="list-style-type: none"> - No - <i>Tweet</i> - Probabilitas Positif - Probabilitas Negatif - Probabilitas Netral - Klasifikasi Sentimen
--	---

Aliran normal

Lihat data Klasifikasi Kategori

1. Klik Tabs Klasifikasi Kategori	2. Menampilkan halaman data Tabs Klasifikasi Kategori yang berisi: <ul style="list-style-type: none"> - <i>Dropdown Show</i>
--	---

Dilanjutkan

Lanjutan

	<ul style="list-style-type: none"> - Tabs Klasifikasi Sentimen - Tabs Uji Performansi Sentimen - Tabs Uji Performansi Kategori - Tabs Indeks Kebahagiaan <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Tweet</i> - Prob Pekerjaan
--	--

Aliran normal**Lihat data Uji Performansi Sentimen**

1. Klik Tabs Uji Performansi Sentimen	
	<p>2. Menampilkan halaman data Tabs Uji Performansi Sentimen yang berisi:</p> <ul style="list-style-type: none"> - <i>Dropdown Show</i> - Tabs Klasifikasi Sentimen - Tabs Klasifikasi Kategori - Tabs Uji Performansi Kategori - Tabs Indeks Kebahagiaan <p>Tabel Average yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai Rata-rata <p>Tabel <i>Recall</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Recall</i>

Dilanjutkan

Lanjutan

	<p>Tabel <i>Precision</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Precision</i> <p>Tabel <i>F1 score</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>F1 score</i> <p>Tabel <i>Support</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Support</i>
Aliran normal	
Lihat data Uji Performansi Kategori	
<p>1. Klik Tabs Uji Performansi Kategori</p>	<p>2. Menampilkan halaman data <i>Testing</i> Uji Performansi Kategori yang berisi:</p> <ul style="list-style-type: none"> - <i>Dropdown Show</i> - Tabs Klasifikasi Sentimen - Tabs Klasifikasi Kategori - Tabs Uji Performansi Sentimen - Tabs Indeks Kebahagiaan <p>Tabel <i>Average</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai Rata-rata

Dilanjutkan

Lanjutan

	<p>Tabel <i>Recall</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Recall</i> <p>Tabel <i>Precision</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Precision</i> <p>Tabel <i>F1 score</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>F1 score</i> <p>Tabel <i>Support</i> yang berisi</p> <ul style="list-style-type: none"> - No - <i>Class</i> - Nilai <i>Support</i>
--	---

4.3.3.10 Skenario Mengelola *Stopword*

Skenario mengelola *stopword* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola *stopword*. Aktor dapat menambahkan, mengedit, menghapus, melihat data *stopword* dengan memilih menu *stopword*. Skenario mengelola *stopword* dapat dilihat pada lampiran A.9..

4.3.3.11 Skenario Mengelola Ontologi

Skenario mengelola ontologi merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola ontologi. Aktor dapat menambahkan, mengedit, menghapus, melihat data ontologi dengan memilih menu ontologi. Skenario mengelola ontologi dapat dilihat pada lampiran A.10.

4.3.3.12 Skenario Melihat Indeks Kebahagiaan Data *Testing*

Skenario melihat indeks kebahagiaan data *testing* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat indeks kebahagiaan data *testing*. Aktor dapat melihat indeks kebahagiaan data *testing*. Skenario melihat indeks kebahagiaan data *testing* dapat dilihat pada lampiran A.11.

4.3.3.13 Skenario Mengelola Data Indeks Kebahagiaan Harian

Skenario mengelola data indeks kebahagiaan harian merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data indeks kebahagiaan harian. Aktor dapat menambah, melihat, dan menghapus indeks kebahagiaan harian dengan memilih menu indeks kebahagiaan. Skenario mengelola data indeks kebahagiaan harian dapat dilihat pada lampiran A.12.

4.3.3.14 Skenario Melihat *FAQ*

Skenario melihat *FAQ* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat *FAQ*. Aktor dapat melihat data *FAQ* tabel dengan memilih menu *FAQ*. Skenario melihat *FAQ* dapat dilihat pada lampiran A.13.

4.3.3.15 Skenario *Logout*

Skenario *logout* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melakukan *logout*. Aktor dapat keluar dari sistem dengan memilih tombol keluar. Skenario *logout* dapat dilihat pada lampiran A.14.

4.3.4 *Sequence Diagram*

Sequence Diagram merupakan diagram yang menggambarkan interaksi antar kelas yang dilakukan sesuai dengan alurnya masing-masing. *Sequence Diagram* merupakan desain perancangan yang mendekati proses pengkodean, maka dari itu di dalam *Sequence Diagram* bahasa yang dituliskan merupakan bahasa pengkodean, dimana bahasa pengkodean tersebut menjelaskan jalannya alur dalam fitur secara berurutan. *Sequence Diagram* dalam sistem ini antara lain sebagai berikut:

4.3.4.1 *Sequence Diagram Login*

Sequence Diagram login terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *landingpage*, *login*, *home* dengan *Controller LoginController* dan *model User*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.1.

4.3.4.2 *Sequence Diagram Registrasi Akun*

Sequence Diagram Registrasi Akun terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *landingpage*, *register* dengan *Controller RegisterController* dan *model user*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.2.

4.3.4.3 *Sequence Diagram Mengelola User*

Sequence Diagram Mengelola User terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *user*, *edit* dengan *Controller UserController* dan *model user*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.3.

4.3.4.4 *Sequence Diagram Mengelola Tweet*

Sequence Diagram Mengelola Tweet terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *tweet* dengan *Controller TweetController* dan *model Tweet*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.4.

4.3.4.5 *Sequence Diagram Mengelola Data Training*

Sequence Diagram Mengelola Data Training terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *datatraining*, *edit* dengan *Controller DataTrainingController* dan *model*

Tweet serta *Classification*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.5.

4.3.4.6 *Sequence Diagram* Melihat *Text Mining Data Training*

Sequence Diagram Melihat *Text Mining Data Training* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *textPreprocessing* dengan *Controller DataTrainingController* dan *model Tweet*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.6.

4.3.4.7 *Sequence Diagram* Mengelola *Data Testing*

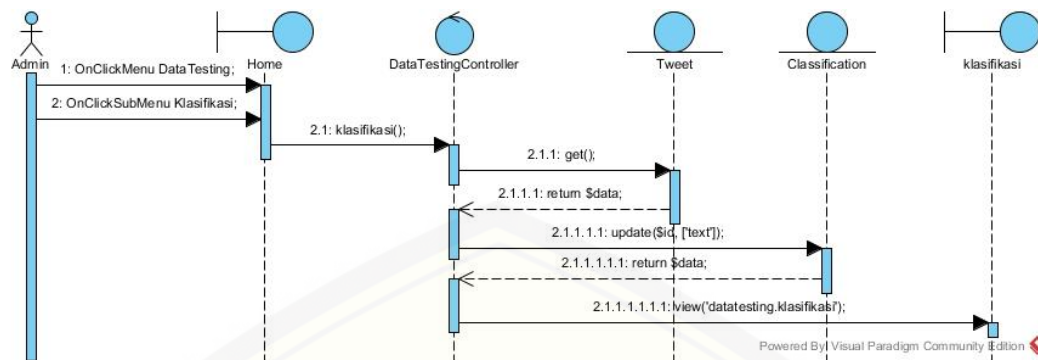
Sequence Diagram Mengelola *Data Testing* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *datatesting*, *edit* dengan *Controller DataTestingController* dan *model Tweet* serta *Classification*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.7.

4.3.4.8 *Sequence Diagram* Melihat *Text Mining data Testing*

Sequence Diagram Melihat *Text Mining data Testing* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *textPreprocessing* dengan *Controller DataTestingController* dan *model Tweet*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.8.

4.3.4.9 *Sequence Diagram* Melihat *Klasifikasi dan Uji Performansi Data*

Sequence Diagram Melihat *Klasifikasi dan Uji Performansi Data* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *klasifikasi* dengan *Controller DataTestingController* dan *model Tweet* dan *Classification*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.3



Gambar 4.3 *Sequence Diagram* Melihat Klasifikasi dan Uji Performansi Data

4.3.4.10 *Sequence Diagram* Mengelola *Stopword*

Sequence Diagram Mengelola *Stopword* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *stopwordg* dengan *Controller StopwordController* dan *model Stopword*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.9.

4.3.4.11 *Sequence Diagram* Mengelola Ontologi

Sequence Diagram Mengelola Ontologi terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *ontology* dengan *Controller OntologyController* dan *model Ontology*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.10.

4.3.4.12 *Sequence Diagram* Melihat Indeks Kebahagiaan Data *Testing*

Sequence Diagram Melihat Indeks Kebahagiaan Data *Testing* terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *indekskebahagiaan* dengan *Controller DataTestingController* dan *model classification*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.11.

4.3.4.13 *Sequence Diagram* Mengelola Data Indeks Kebahagiaan Harian

Sequence Diagram Mengelola Data Indeks Kebahagiaan Harian terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *streaming* dengan *Controller StreamingController* dan *model* *Happiness* serta *Tweet*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.12.

4.3.4.14 *Sequence Diagram* Melihat FAQ

Sequence Diagram Melihat FAQ terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling berhubungan adalah kelas tampilan *Home*, *FAQ* dengan *Controller FAQController*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.13.

4.3.4.15 *Sequence Diagram* Logout

Sequence Diagram melihat transaksi keluar terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling terhubung adalah *controller* *AuthController* dan *model* *User*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.14.

4.3.5 Activity Diagram

Activity Diagram merupakan diagram yang memodelkan proses-proses yang terjadi pada system. *Activity Diagram* dalam sistem ini antara lain sebagai berikut:

4.3.5.1 *Activity Diagram* Login

Activity Diagram login menggambarkan tentang proses yang terjadi saat aktor masuk ke dalam sistem. Aktor yang dapat masuk ke dalam sistem adalah admin dan ahli bahasa yang memiliki *email* dan *password*. *Activity Diagram* login dapat dilihat pada lampiran C.1.

4.3.5.2 *Activity Diagram* Registrasi Akun

Activity Diagram registrasi akun menggambarkan tentang proses yang

terjadi saat aktor melakukan registrasi. Aktor dapat memasukkan *username*, *e-mail*, dan *password* yang telah dimiliki sebelumnya. *Activity Diagram* registrasi akun dapat dilihat pada lampiran C.2.

4.3.5.3 *Activity Diagram* Mengelola *User*

Activity Diagram mengelola *user* menggambarkan tentang proses yang terjadi saat aktor mengolah *user*. Aktor dapat melihat, mengedit, dan menghapus data *user*. *Activity Diagram* mengelola *user* dapat dilihat pada lampiran C.3.

4.3.5.4 *Activity Diagram* Mengelola *Tweet*

Activity Diagram mengelola *tweet* menggambarkan tentang proses yang terjadi saat aktor mengolah *tweet*. Aktor dapat melakukan proses *tweet* dengan memilih menu *tweet*. *Activity Diagram* mengelola *tweet* dapat dilihat pada lampiran C.4.

4.3.5.5 *Activity Diagram* Mengelola Data *Training*

Activity Diagram mengelola data *training* menggambarkan tentang proses yang terjadi saat aktor mengelola data *training*. Aktor dapat melihat, mengedit, dan menghapus data *training* dengan memilih menu *training* tabel. *Activity Diagram* mengelola data *training* dapat dilihat pada lampiran C.5.

4.3.5.6 *Activity Diagram* Melihat *Text Mining* Data *Training*

Activity Diagram melihat *text mining* data *training* menggambarkan tentang proses yang terjadi saat aktor melihat *text mining* data *training*. Aktor dapat melihat *text mining* data *training* dengan memilih menu data *training text mining*. *Activity Diagram* melihat *text mining* data *training* dapat dilihat pada lampiran C.6.

4.3.5.7 *Activity Diagram* Mengelola Data *Testing*

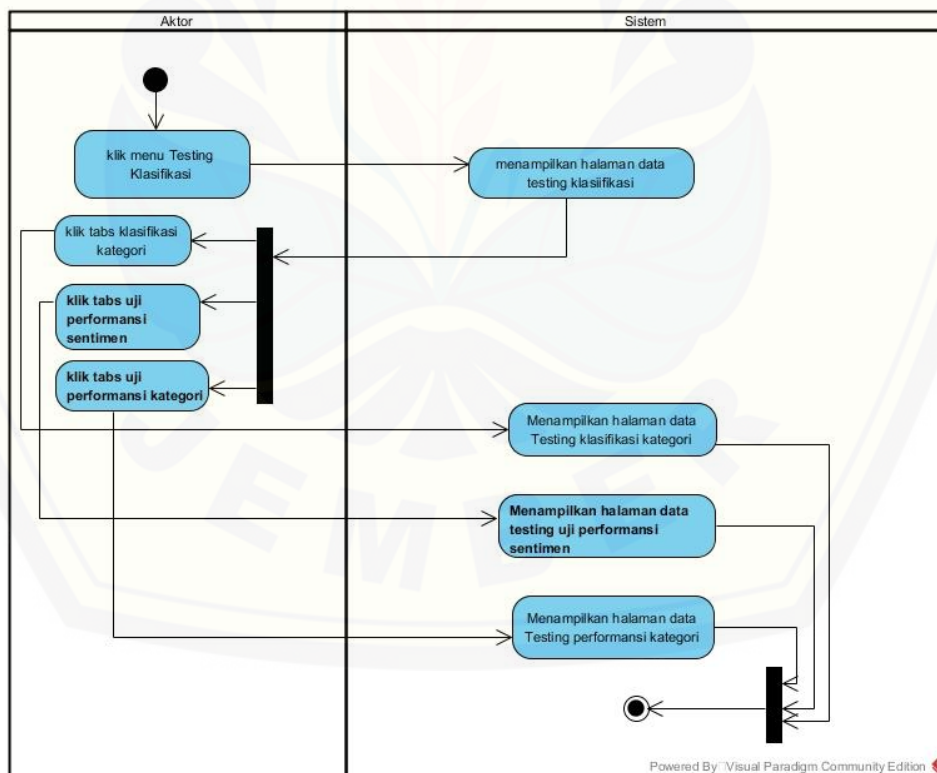
Activity Diagram mengelola data *testing* menggambarkan tentang proses yang terjadi saat aktor mengelola data *testing*. Aktor dapat melihat, mengedit dan menghapus data *testing* dengan memilih menu data *testing* tabel. *Activity Diagram* mengelola data *testing* dapat dilihat pada lampiran C.7.

4.3.5.8 Activity Diagram Melihat Text Mining data Testing

Activity Diagram melihat *text mining* data *testing* menggambarkan tentang proses yang terjadi saat aktor melihat *text mining* data *testing*. Aktor dapat melihat *text mining* data *testing* dengan memilih menu data *testing text mining*. Activity Diagram melihat *text mining* data *testing* dapat dilihat pada lampiran C.8.

4.3.5.9 Activity Diagram Melihat Klasifikasi dan Uji Performansi Data

Activity Diagram melihat klasifikasi dan uji performansi data *testing* menggambarkan tentang proses yang terjadi saat aktor melihat klasifikasi dan uji performansi data *testing*. Aktor dapat melihat klasifikasi dan uji performansi dengan memilih menu *testing* klasifikasi. Klasifikasi dan uji performansi dapat ditampilkan dengan menampilkan halaman data tabs klasifikasi kategori dan tabs uji performansi sentimen. Activity Diagram melihat klasifikasi dan uji performansi data *testing* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Activity Diagram Melihat Klasifikasi dan Uji Performansi Data Testing

4.3.5.10 *Activity Diagram* Mengelola *Stopword*

Activity Diagram mengelola *stopword* menggambarkan tentang proses yang terjadi saat aktor mengelola *stopword*. Aktor dapat menambahkan, mengedit, menghapus, melihat data *stopword* dengan memilih menu *stopword*. *Activity Diagram* mengelola *stopword* dapat dilihat pada lampiran C.9.

4.3.5.11 *Activity Diagram* Mengelola Ontologi

Activity Diagram mengelola ontologi menggambarkan tentang proses yang terjadi saat aktor mengelola ontologi. Aktor dapat menambahkan, mengedit, menghapus, melihat data ontologi dengan memilih menu ontologi. *Activity Diagram* mengelola ontologi dapat dilihat pada lampiran C.10.

4.3.5.12 *Activity Diagram* Melihat Indeks Kebahagiaan Data *Testing*

Activity Diagram melihat indeks kebahagiaan data *testing* menggambarkan tentang proses yang terjadi saat aktor melihat indeks kebahagiaan data *testing*. Aktor dapat melihat indeks kebahagiaan data *testing*. *Activity Diagram* melihat indeks kebahagiaan data *testing* dapat dilihat pada lampiran C.11.

4.3.5.13 *Activity Diagram* Mengelola Data Indeks Kebahagiaan Harian

Activity Diagram mengelola data indeks kebahagiaan harian menggambarkan tentang proses yang terjadi saat aktor mengelola data indeks kebahagiaan harian. Aktor dapat menambah, melihat, dan menghapus indeks kebahagiaan harian dengan memilih menu indeks kebahagiaan. *Activity Diagram* mengelola data indeks kebahagiaan harian dapat dilihat pada lampiran C.12.

4.3.5.14 *Activity Diagram* Melihat *FAQ*

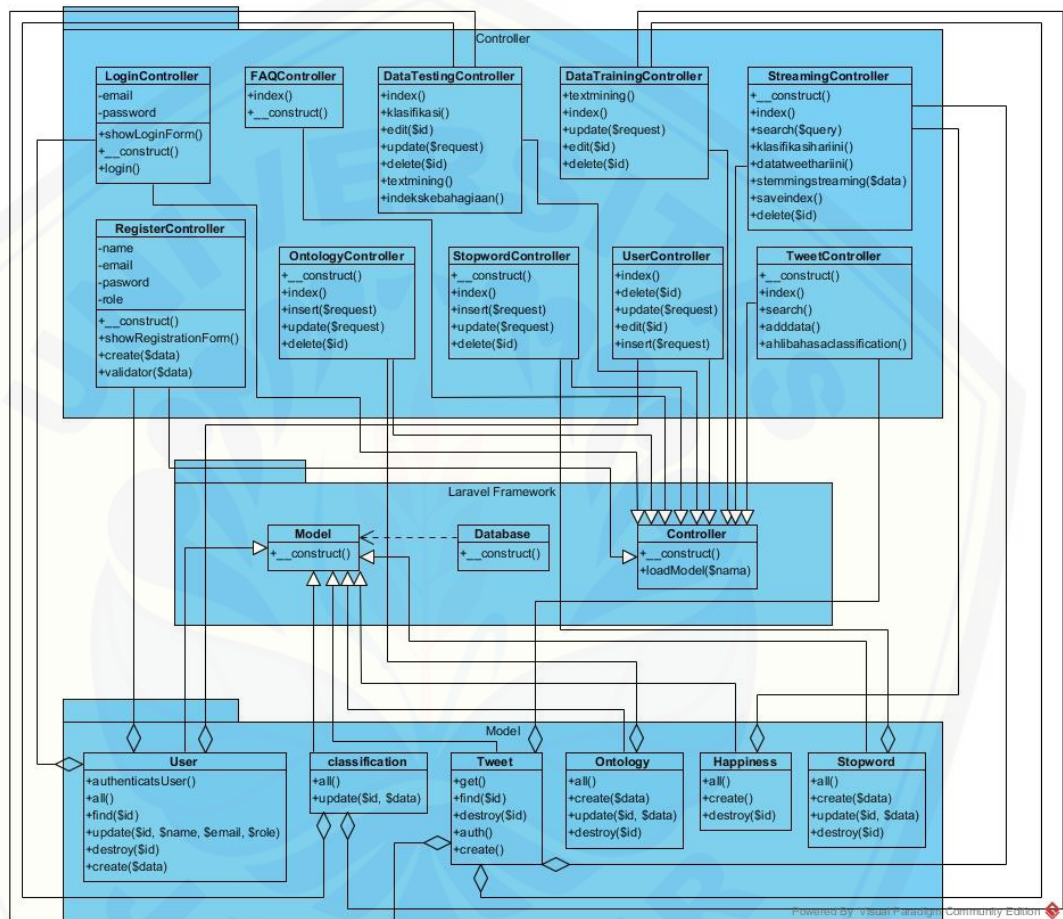
Activity Diagram melihat *FAQ* menggambarkan tentang proses yang terjadi saat aktor melihat *FAQ*. Aktor dapat melihat data *FAQ* tabel dengan memilih menu *FAQ*. *Activity Diagram* melihat *FAQ* dapat dilihat pada lampiran C.13.

4.3.5.15 *Activity Diagram* *Logout*

Activity Diagram *logout* menggambarkan tentang proses yang terjadi saat aktor melakukan *logout*. Aktor dapat keluar dari sistem dengan memilih tombol keluar. *Activity Diagram* *logout* dapat dilihat pada lampiran C.14.

4.3.6 Class diagram

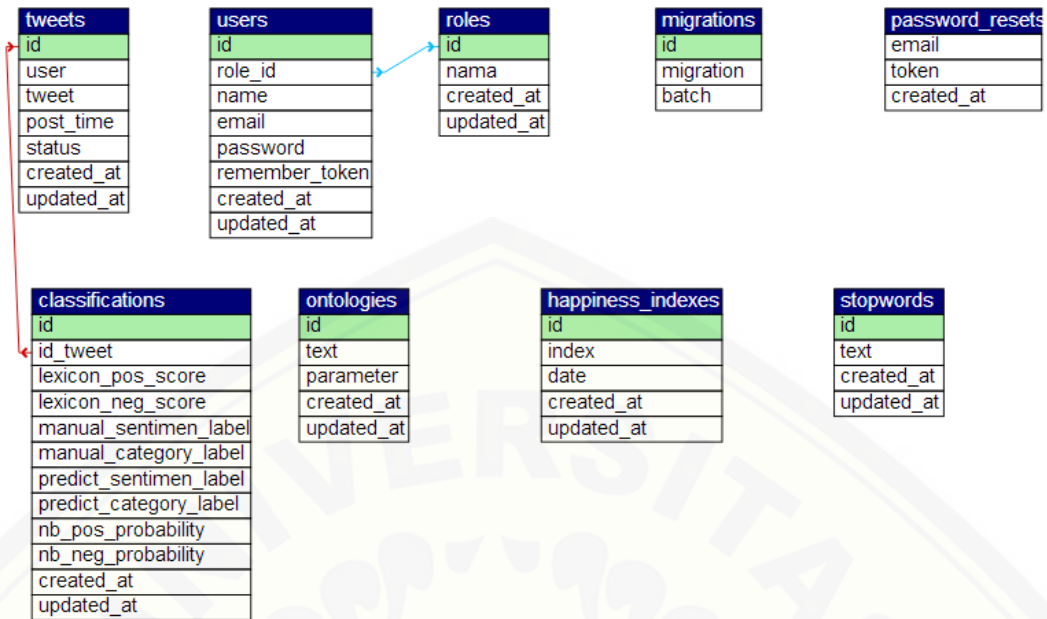
Class diagram merupakan diagram untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya digunakan. Class diagram yang digunakan sistem pengukuran indeks kebahagiaan menggunakan analisis sentimen di media sosial dapat dilihat pada Gambar 4.5.



Gambar 4.5 Class diagram

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan gambaran hubungan antar data dalam basis data yang memiliki relasi. ERD yang digunakan sistem pengukuran indeks kebahagiaan menggunakan analisis sentimen di media sosial dapat dilihat pada gambar 4.6.



Gambar 4.6 Entity Relationship Diagram (ERD)

4.4 Penulisan Kode Program

Penulisan kode program merupakan implementasi dari desain sistem yang telah dirancang sebelumnya.

4.4.1 Kode Program *Login*

Kode program *login* terletak pada *controller LoginController*, *model User*, dan *view page-login*. Terdapat pengecekan *username* dan *password* yang diinputkan *user* pada saat proses *login*. Jika *username* dan *password* tidak sesuai dengan yang terdapat di *database*, maka *user* tidak dapat masuk ke dalam sistem. Penulisan kode program *login* dapat dilihat pada lampiran D.1.

4.4.2 Kode Program Registrasi Akun

Kode program registrasi akun terletak pada *registerController* dan *view register*. Menggunakan *model User*. Terdapat *function* yaitu bisa mendaftarkan *user* bar. Penulisan registrasi akun dapat dilihat pada lampiran D.2.

4.4.3 Kode Program Mengelola *User*

Kode program mengelola *user* terletak pada *UserController*, *view user.index*, dan *view user.edit*. Menggunakan *model Role* dan *User*. Terdapat

function *index()* yang digunakan untuk menampilkan halaman *user*, *delete()* untuk menghapus data *user*, *edit()* untuk mengedit data *user*, *edit()* untuk menampilkan halaman edit, dan *insert()* untuk membuat *user* baru. Penulisan kode program mengelola *user* dapat dilihat pada lampiran D.3.

4.4.4 Kode Program Mengelola *Tweet*

Kode program mengelola *tweet* terletak pada *TweetController*, *view* *crawling.tweet*. Menggunakan *model* *Classification* dan *Tweet*. Terdapat *function* *index()* untuk menampilkan halaman *tweet*, *search()* untuk mencari data *tweet*, dan *adddata()* untuk menambahkan *tweet* yang diingkan ke dalam *database*. Penulisan kode program mengelola *tweet* dapat dilihat pada lampiran D.4.

4.4.5 Kode Program Mengelola Data *Training*

Kode program mengelola data *training* terletak pada *DataTrainingController*, *view* *datatraining.edit*. Menggunakan *model* *Classification* dan *Tweet*. Terdapat *function* *index()* untuk menampilkan halaman data *training*, *edit()* untuk mengedit data *training*, *edit()* untuk menampilkan *form* edit, *delete()* untuk menghapus data *training*, *manual()* untuk menampilkan halaman pelabelan manual, dan *labelling()* untuk melakukan pelabelan manual. Penulisan kode program mengelola data *training* dapat dilihat pada lampiran D.5.

4.4.6 Kode Program Melihat *Text Mining* Data *Training*

Kode program melihat *text mining* data *training* terletak pada *DataTrainingController*, *view* *datatraining.textpreprocessing*. yang memiliki *function* *textmining()* untuk melakukan *text mining*. Dan terletak pada *TextMiningController* yang memiliki *frequencyngram()* untuk mengetahui jumlah frekuensi *unigram* yang terbentuk, *ngram()* untuk mengubah data *training* menjadi bentuk *unigram*, *stemming()* untuk mengubah data *training* menjadi kata dasar, *stopwordremoval()* untuk menghapus *stopword*, *preprocessing()* untuk melakukan *text preprocessing*, *getKataDasar()* untuk mengambil daftar kata dasar dalam dokumen, *removeBukanKataDasar()* untuk menghapus kata yang bukan kata dasar, dan *cekCorpus()* untuk mengecek apakah kata ada dalam kamus kata dasar

atau tidak. Penulisan kode program melihat *text mining data training* dapat dilihat pada lampiran D.6.

4.4.7 Kode Program Mengelola Data *Testing*

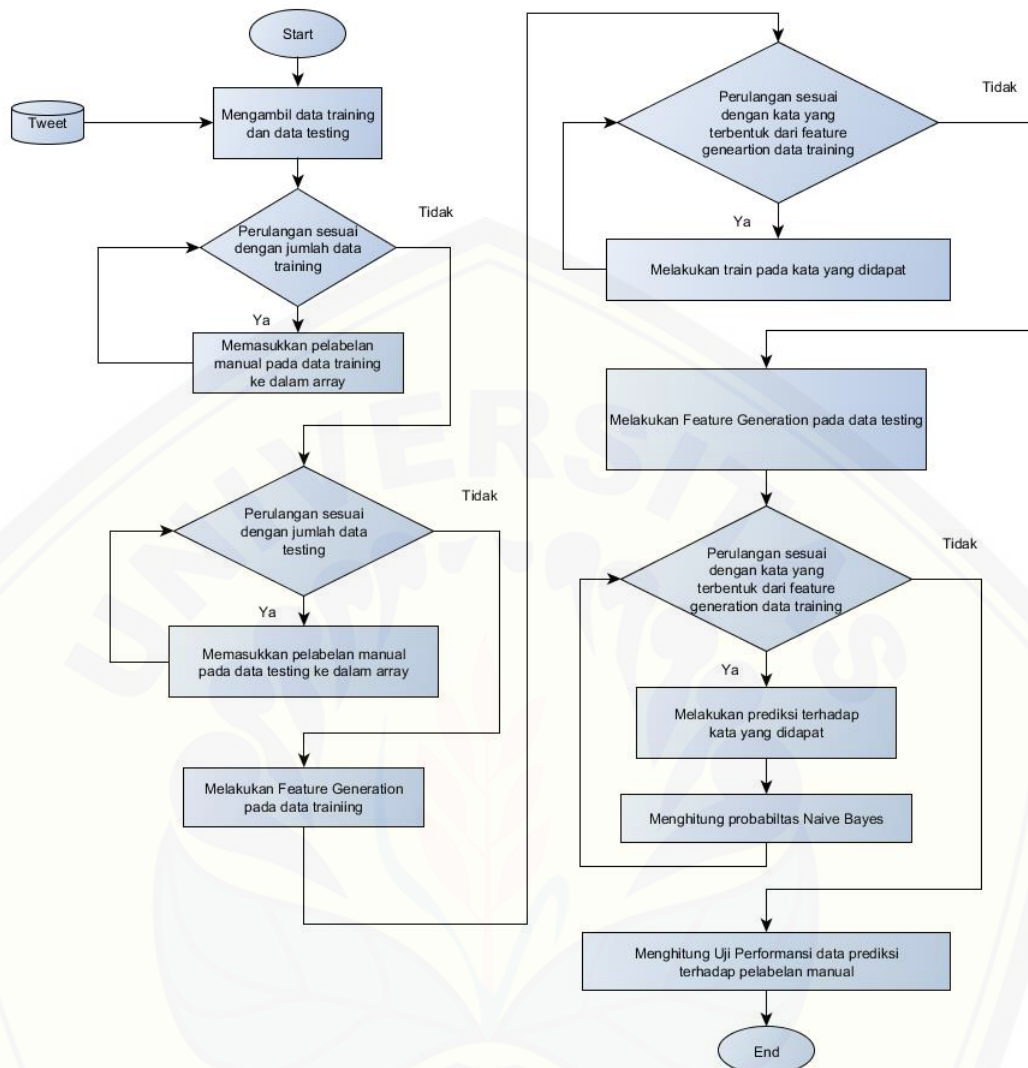
Kode program mengelola data *testing* terletak pada *DataTestingController*, *view* *datatesting.index*. Menggunakan *model* *Classification* dan *Tweet*. Terdapat *function* *index()* untuk menampilkan halaman data *testing*, *delete()* untuk menghapus data *testing*, *edit()* untuk mengedit data *testing*, *edit()* untuk mengedit data *testing*, dan *labelling()* untuk melakukan pelabelan manual. Penulisan kode program mengelola data *testing* dapat dilihat pada lampiran D.7.

4.4.8 Kode Program Melihat *Text Mining Data Testing*

Kode program melihat *text mining data testing* terletak pada *DataTestingController*, *view* *datatesting.textpreprocessing*, menggunakan *model* *Classification* dan *Tweet*, terdapat *function* *textmining()* untuk menampilkan halaman *text mining data testing*. Dan terletak pada *TextMiningController*, menggunakan *model* *Classification* dan *Tweet*, terdapat *function* *frequencygram()* untuk mengetahui jumlah frekuensi *unigram* yang terbentuk, *ngram()* untuk mengubah data *training* menjadi bentuk *ngram*, *stemming()* untuk mengubah data *training* menjadi kata dasar, *stopwordremoval()* untuk menghapus *stopword*, *preprocessing()* untuk melakukan *text preprocessing*, *getKataDasar()* untuk mengambil daftar kata dasar dalam dokumen, *removeBukanKataDasar()* untuk menghapus kata yang bukan kata dasar, dan *cekCorpus()* untuk mengecek apakah kata ada dalam kamus kata dasar atau tidak. Penulisan kode program melihat *text mining data testing* dapat dilihat pada lampiran D.8.

4.4.9 Kode Program Melihat Klasifikasi dan Uji Performansi Data *Testing*

Kode program melihat klasifikasi dan uji performansi data *testing* terletak pada *DataTestingController*. Menggunakan *model* *Classification* dan *Tweet*. Terdapat *function* *klasifikasi()* untuk melakukan klasifikasi. Penulisan kode program melihat klasifikasi dan uji performansi data *testing* dapat dilihat pada lampiran D.9. Algoritma melihat klasifikasi dan uji performansi data *testing* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Algoritma *function* klasifikasi()

4.4.10 Kode Program Mengelola *Stopword*

Kode program mengelola *stopword* terletak pada *StopwordController*. Menggunakan *model Stopword*. Terdapat *index()* untuk menampilkan halaman *stopword*, *insert()* untuk membuat *stopword* baru, *edit()* untuk mengedit *stopword*, dan *delete()* untuk menghapus *stopword*. Penulisan kode program mengelola *stopword* dapat dilihat pada lampiran D.10.

4.4.11 Kode Program Mengelola Ontologi

Kode program mengelola ontologi terletak pada *OntologyController*. Menggunakan *model Ontology*. Terdapat *index()* untuk menampilkan halaman

ontologi, *insert()* untuk membuat ontologi baru, *edit()* untuk mengedit ontologi, dan *delete()* untuk menghapus ontologi. Penulisan kode program mengelola ontologi dapat dilihat pada lampiran D.11.

4.4.12 Kode Program Melihat Indeks Kebahagiaan Data *Testing*

Kode program melihat indeks kebahagiaan data *testing* terletak pada *DataTestingController*, dan *view* *datatesting.indekskebahagiaan*. Menggunakan *model* *Classification* dan *Tweet*. Terdapat *function* *indekskebahagiaan()* untuk mengecek indeks kebahagiaan pada data *testing*. Penulisan kode program melihat indeks kebahagiaan data *testing* dapat dilihat pada lampiran D.12.

4.4.13 Kode Program Mengelola Data Indeks Kebahagiaan Harian

Kode program mengelola data indeks kebahagiaan harian terletak pada *StreamingController*. Terdapat *function* *index()* untuk menampilkan halaman indeks kebahagiaan, *search()* untuk mencari *tweet*, *klasifikasihariini()* untuk mengklasifikasikan indeks kebahagiaan hari ini, *stemmingstreaming()* untuk melakukan *stemming*, *datatweethariini()* untuk menampilkan *tweet* hari ini, *saveindex()* untuk menyimpan indeks kebahagiaan, dan *delete()* untuk menghapus indeks kebahagiaan. Penulisan kode program mengelola data indeks dapat dilihat pada lampiran D.13.

4.4.14 Kode Program Melihat *FAQ*

Kode program melihat *FAQ* terletak pada *FAQController* *view* *faq.index*. Terdapat *function* *index()* untuk melihat *FAQ*. Penulisan kode melihat *FAQ* dapat dilihat pada lampiran D.14.

4.4.15 Kode Program *Logout*

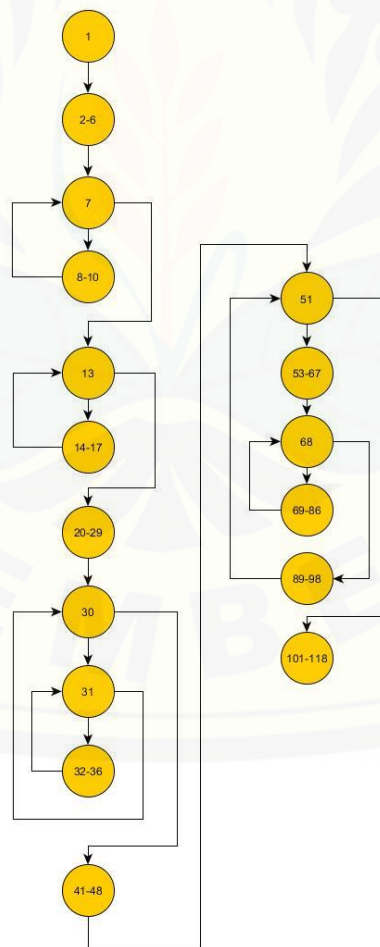
Kode program *logout* terletak pada *controller* *LoginController*. *Logout* digunakan *user* untuk keluar dari sistem. Penulisan kode program *login* dapat dilihat pada lampiran D.15.

4.5 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian whitebox terlebih dahulu kemudian dilanjutkan dengan pengujian blackbox.

4.5.1 Pengujian White Box

Pengujian *white box* pada sistem pengukur indeks kebahagiaan ini dengan cara menggambar *flowgraph*, menghitung *cyclometric complexity* (CC), dan membuat tabel pengujian *test case*. Pengujian ini menggunakan fitur klasifikasi dan uji performansi yang terdapat pada *function* klasifikasi() di dalam *controller* *DataTestingController*. Listing program *function* klasifikasi() dapat dilihat pada lampiran D.9. *Flowgraph* *function* klasifikasi() dapat dilihat pada Gambar 4.8



Gambar 4.8 *Flowgraph* *function* klasifikasi

Berdasarkan *flowgraph* yang telah dibuat, maka perhitungan *cyclometric complexity* adalah sebagai berikut

$$\begin{aligned} \text{Cyclomatic Complexity (CC)} &= E - N + 2 \\ &= 22 - 17 + 2 \\ &= 7 \end{aligned}$$

Jalur 1 : 1 – 2 – 6 – 7 – 8 – 10 – 7

Jalur 2 : 1 – 2 – 6 – 7 – 13 – 14 – 17 – 13

Jalur 3 : 1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 31 – 30

Jalur 4 : 1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 31 – 32 – 36 – 31

Jalur 5 : 1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 53 – 67 – 68 – 89 – 98 – 51

Jalur 6 : 1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 53 – 67 – 68 – 69 – 86 – 68 – 89 – 98 – 51

Jalur 7 : 1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 101 – 118

Diketahui bahwa *cyclomatic complexity function index()* berjumlah 7 yang berarti sistem yang dibuat sederhana dan mudah dipahami. Maka *test case* dari *function* *klasifikasi()* dapat dilihat pada Tabel 4.4. di bawah ini.

Tabel 4.4 *Test case* dari *function* *klasifikasi()*

<i>Test case 1</i>	Selama data memiliki panjang sama dengan data <i>training</i> , maka data <i>training</i> masuk ke dalam <i>array</i>
Target yang diharapkan	Data <i>training</i> masuk ke dalam <i>array</i>
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 8 – 10 – 7
<i>Test case 2</i>	Selama data memiliki panjang sama dengan data <i>testing</i> , maka data <i>testing</i> masuk ke dalam <i>array</i>
Target yang diharapkan	Data <i>testing</i> masuk ke dalam <i>array</i>
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 14 – 17 – 13

Dilanjutkan

Lanjutan

<i>Test case 3</i>	Selama data memiliki panjang sama dengan hasil <i>unigram</i> pada data <i>training</i> , maka masuk ke perulangan kedua
Target yang diharapkan	Dapat mengeksekusi perulangan kedua
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 31 – 30
<i>Test case 4</i>	Selama data memiliki panjang sama dengan hasil dari <i>array</i> yang dihasilkan pada perulangan pertama, maka <i>training</i> dilakukan pada sentimen dan kategori
Target yang diharapkan	Dapat melakukan <i>training</i> pada <i>unigram</i> data train
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 31 – 32 – 36 – 31
<i>Test case 5</i>	Selama data memiliki panjang sama dengan hasil <i>unigram</i> pada data <i>training</i> , maka pendeklarasian variabel prediksi dilakukan dan masuk ke perulangan kedua
Target yang diharapkan	Dapat mendeklarasikan variabel prediksi dan masuk ke perulangan kedua
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 53 – 67 – 68 – 89 – 98 – 51
<i>Test case 6</i>	Selama data memiliki panjang sama dengan hasil dari <i>array</i> yang dihasilkan pada perulangan pertama, maka prediksi dilakukan
Target yang diharapkan	Dapat melakukan prediksi
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 53 – 67 – 68 – 69 – 86 – 68 – 89 – 98 – 51
<i>Test case 7</i>	Menampilkan nilai prediksi dan uji performansi
Target yang diharapkan	Menampilkan nilai prediksi dan uji performansi
Hasil pengujian	Benar
<i>Path/Jalur</i>	1 – 2 – 6 – 7 – 13 – 20 – 29 – 30 – 41 – 48 – 51 – 101 – 118

4.5.2 Black Box

Pengujian *black box* berfungsi untuk menguji sistem dari sisi fungsionalitas sistem dengan tujuan mengetahui apakah fitur-fitur, inputan dan keluaran sistem sesuai dengan kebutuhan pengguna. Fitur-fitur yang diujikan mencakup fitur melihat, menambah, mengubah dan menghapus data serta keluaran yang diharapkan dari sistem saat fitur tersebut dijalankan. Hasil pengujian *black box* pada fitur hasil peramalan dapat dilihat pada Tabel 4.4.

Tabel 4.5 Hasil pengujian *black box* pada fitur hasil peramalan

No.	Fitur	Aksi	Hasil	Keterangan
1.	<i>Login</i>	Mengisi <i>form</i> E-mail lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan <i>user</i> masuk ke dalam sistem sesuai hak akses pengguna	Berhasil
		Klik tombol <i>login</i> namun E-mail atau password kosong	Menampilkan <i>span</i> “harap isi bidang ini”	Berhasil
		Klik tombol <i>login</i> namun E-mail kosong	Menampilkan <i>span</i> “harap isi bidang ini”	Berhasil
		Klik tombol <i>login</i> namun password kosong	Menampilkan <i>span</i> “harap isi bidang ini”	Berhasil
		Klik tombol <i>login</i> namun E-mail atau password salah	Menampilkan <i>span</i> “ <i>These credentials do not match our records.</i> ”	Berhasil
2.	Registrasi Akun	Klik tombol daftar sebagai ahli bahasa	Menampilkan halaman yang berisi <i>form</i> registrasi	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
		Masukkan <i>Username, E-mail, Password</i> dan Konfirmasi <i>Password</i> pada halaman registrasi lalu klik sign up	Registrasi berhasil dan sudah terdaftar sebagai ahli bahasa	Berhasil
		Klik Sign Up namun <i>form</i> ada yang kosong	Menampilkan span”harap isi bidang ini”	Berhasil
3.	Mengelola <i>User</i>	Klik menu data <i>user</i>	Menampilkan halaman data <i>user</i>	Berhasil
		Klik Tombol Edit pada halaman data <i>user</i>	Menampilkan halaman <i>form</i> edit data <i>user</i> , tombol <i>save</i>	
		klik <i>save</i> pada halaman <i>Form</i> Edit Data <i>User</i> dengan sesuai.	mengubah data ke <i>database</i> dan menampilkan halaman data <i>user</i>	Berhasil
		Klik Hapus pada halaman data <i>user</i>	Data di hapus dari <i>database</i> dan menampilkan halaman data <i>user</i>	Berhasil
4.	Mengelola <i>Tweet</i>	Klik menu <i>tweet</i>	menampilkan <i>tweet</i> yang ada pada halaman <i>tweet</i> dan <i>form</i> inputan <i>tweet</i>	Berhasil
		Mengisikan query dan jumlah <i>tweet</i> lalu klik tombol search	Menampilkan <i>tweet</i> yang sedang dicari pada kolom query	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
		Centang <i>tweet</i> yang diinginkan lalu Klik Tombol Add -> Jadikan Data <i>Training</i>	Menampilkan hasil data <i>training tweet</i> yang diinginkan	Berhasil
		Centang <i>tweet</i> yang diinginkan lalu Klik Tombol Add -> Jadikan Data <i>Testing</i>	Menampilkan hasil data <i>testing tweet</i> yang diinginkan	Berhasil
5.	Mengelola Data <i>Training</i>	Klik menu <i>Training</i> sub menu Tabel	Menampilkan halaman data <i>Training</i> Tabel dan tombol edit serta hapus pada data <i>Training</i>	Berhasil
		Klik Tombol edit pada halaman <i>training</i> tabel	Menampilkan <i>Form</i> Edit <i>Training</i> Tabel dan tombol <i>save</i>	Berhasil
		Mengubah data <i>Training</i> dan sesuai lalu Klik tombol <i>Save</i>	Megubah data ke <i>database</i> dan menampilkan halaman <i>Training</i> Tabel	Berhasil
		Klik Hapus pada <i>Training</i> Tabel	Menghapus data <i>Training</i> dan menampilkan halaman <i>Training</i> Tabel	Berhasil
6.	Melihat <i>Text Mining</i> Data <i>Training</i>	Klik menu <i>Training Text Preprocessing</i>	Menampilkan halaman <i>training preprocessing</i> dan tabs	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
			- Tabs <i>Text</i> Tranformation - Tabs <i>Text</i> <i>Unigram</i> - Tabs <i>Text Unigram</i> <i>Frequency Total</i>	
		Klik Tabs <i>Text</i> <i>Transformation</i>	Menampilkan data halaman <i>text</i> <i>Transformation</i>	Berhasil
		Klik Tabs <i>Text</i> <i>Unigram</i>	Menampilkan data halaman <i>Text</i> <i>Unigram</i>	Berhasil
		Klik Tabs <i>Unigram</i> <i>Frequency Total</i>	Menampilkan data Halaman <i>Unigram</i> <i>Frequency Total</i>	Berhasil
7.	Mengelola <i>Testing</i> Tabel	Klik menu <i>Testing</i> sub menu Tabel	Menampilkan halaman data <i>Testing</i> Tabel dan tombol edit serta hapus pada data <i>Testing</i>	Berhasil
		Klik Tombol edit pada halaman <i>testing</i> tabel	Menampilkan <i>Form</i> Edit <i>Testing</i> Tabel dan tombol <i>save</i>	Berhasil
		Mengubah data <i>Testing</i> dan sesuai lalu Klik tombol <i>Save</i>	Mengubah data ke <i>database</i> dan menampilkan halaman <i>Testing</i> Tabel	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
		Klik Hapus pada <i>Testing</i> Tabel	Menghapus data <i>Testing</i> dan menampilkan halaman <i>Testing</i> Tabel	Berhasil
8.	Melihat <i>Text Mining</i> Data <i>Testing</i>	Klik menu <i>Testing Text Preprocessing</i>	Menampilkan halaman <i>Testing Preprocessing</i> dan tabs: <ul style="list-style-type: none"> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram</i> - Tabs <i>Text Unigram Frequency Total</i> 	Berhasil
		Klik Tabs <i>Text Transformation</i>	Menampilkan data <i>testing</i> halaman <i>text Transformation</i>	Berhasil
		Klik Tabs <i>Text Unigram</i>	Menampilkan data <i>testing</i> halaman <i>Text Unigram</i>	Berhasil
		Klik Tabs <i>Unigram Frequency Total</i>	Menampilkan data <i>testing</i> Halaman <i>Unigram Frequency Total</i>	Berhasil
9.	Melihat Klasifikasi & Uji Performansi Data <i>Testing</i>	Klik menu <i>Testing</i> Sub Menu Klasifikasi	Menampilkan halaman data <i>Testing</i> Klasifikasi dan yang berisi beberapa tabs:	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
			<ul style="list-style-type: none"> - Tabs Klasifikasi Kategori - Tabs Uji Performansi Sentimen - Tabs Uji Performansi Kategori 	
		Klik Tabs Klasifikasi Kategori	Menampilkan halaman data <i>testing</i> Klasifikasi Kategori	Berhasil
		Klik Tabs Uji Performansi Sentimen	Menampilkan halaman data <i>testing</i> Uji Performansi Sentimen	Berhasil
		Klik Tabs Uji Performansi Kategori	Menampilkan halaman data <i>testing</i> Uji Performansi Kategori	Berhasil
10.	Mengelola <i>Stopword</i>	Klik menu <i>stopword</i>	Menampilkan halaman data <i>stopword</i> dan tombol input, edit dan hapus	Berhasil
		Klik Tombol Tambah	Menampilkan modal berupa <i>form</i> tambah <i>stopword</i> dan tombol <i>save</i>	Berhasiil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
		Mengisikan <i>form</i> input <i>stopword</i> dan terpenuhi semua lalu klik <i>save</i>	Menambahkan data <i>stopword</i> ke <i>database</i> dan menampilkan halaman <i>stopword</i>	Berhasil
		Mengisikan <i>form</i> input <i>stopword</i> dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	Berhasil
		Klik tombol edit	Menampilkan modal berupa <i>form</i> edit <i>stopword</i> dan tombol <i>save</i>	Berhasil
		Mengubah data <i>stopword</i> dan tidak kosong lalu klik <i>save</i>	Mengubah data dari <i>database</i> dan menampilkan halaman <i>stopword</i>	Berhasil
		Mengubah data <i>stopword</i> dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	Berhasil
		Klik Hapus	Menghapus data <i>stopword</i> dari <i>database</i> dan menampilkan halaman <i>stopword</i>	Berhasil
11.	Mengelola Ontologi	Klik menu ontologi	Menampilkan halaman ontologi dan tombol input, edit dan hapus	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
		Klik Tombol Tambah	Menampilkan modal berupa <i>form</i> tambah ontologi dan tombol <i>save</i>	Berhasil
		Mengisikan <i>form</i> input ontologi dan terpenuhi semua lalu klik <i>save</i>	Menambahkan data ontologi ke <i>database</i> dan menampilkan halaman ontologi	Berhasil
		Mengisikan input ontologi dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	Berhasil
		Klik tombol edit	Menampilkan modal berupa <i>form</i> edit ontologi dan tombol <i>save</i>	Berhasil
		Mengubah data ontologi dan tidak kosong lalu klik <i>save</i>	Mengubah data dari <i>database</i> dan menampilkan halaman ontologi	Berhasil
		Mengubah data ontologi dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	Berhasil
		Klik Hapus	Menghapus data ontologi dari <i>database</i> dan menampilkan halaman ontology	Berhasil
12.	Melihat Indeks Kebahagiaan Data <i>Testing</i>	Klik menu Data <i>Testing</i> Indeks Kebahagiaan	Menampilkan halaman Data <i>Testing</i> Indeks Kebahagiaan	Berhasil

Dilanjutkan

Lanjutan

No.	Fitur	Aksi	Hasil	Keterangan
13.	Mengelola Data Indeks Kebahagiaan Harian	Klik menu Indeks Kebahagiaan	Menampilkan data halaman Indeks Kebahagiaan dan tombol Klasifikas hari ini serta tombol hapus	Berhasil
		Klik Klasifikasi Hari Ini	Menampilkan halaman Indeks Kebahagiaan Hari Ini dan tombol <i>save</i>	Berhasil
		Klik <i>save</i> pada Klasifikasi Hari ini	menyimpan ke <i>database</i> dan Menampilkan halaman indeks Kebahagiaan	Berhasil
		Klik Hapus pada Indeks Kebahagiaan	menghapus data dari <i>database</i> dan menampilkan haaman indeks kebahagiaan	Berhasil
14.	Melihat <i>FAQ</i>	klik Menu <i>FAQ</i>	Menampilkan halaman <i>FAQ</i>	Berhasil
15.	<i>Logout</i>	Klik <i>logout</i> pada dahsboard <i>user</i>	Keluar dari halaman <i>user</i> kembali ke halama sebelum <i>login</i>	Berhasil

BAB 6. PENUTUP

6.1 Kesimpulan

Berdasarkan analisis dan pengujian yang dilakukan pada bab sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut :

1. *Crawling* data digunakan untuk mendapatkan *dataset* yang berada di sosial media *Twitter* menggunakan *Twitter Search API*. Namun dalam penggunaan *Twitter Search API* memiliki beberapa keterbatasan seperti *tweet* tidak menampilkan keseluruhan *tweet* yang ada pada *twitter*, eksekusi pencarian hanya bisa dilakukan 100 kali dalam satu jam, dan hanya menampilkan *tweet* dalam kurun waktu tujuh hari terakhir. *Crawling* data dilakukan dengan memasukkan *query* dan jumlah *tweet* yang diinginkan. Kemudian proses *crawling* data akan dijalankan oleh *Twitter Search API* yang disesuaikan dengan masukan yang diinputkan. Setelah *dataset* yang diinginkan terkumpul, *dataset* akan disimpan ke dalam *database*.
2. Penerapan metode *Naive Bayes Classifier* dalam mengklasifikasikan data *tweet* dengan cara data *training* yang digunakan berasal dari klasifikasi manual oleh ahli bahasa. Dalam penelitian ini, ahli bahasa tersebut merupakan mahasiswa Jurusan Linguistik Bahasa Indonesia Fakultas Ilmu Budaya Universitas Jember. Klasifikasi melihat dari sisi semantik sebuah *tweet*. Data *training* yang dihasilkan berjumlah 900 *tweet* dengan pelabelan yang merata baik itu pelabelan sentimen maupun pelabelan kategori.
3. Pengujian klasifikasi diukur dengan uji performansi *precision*, *recall*, dan *f1 score*. Hasil dari uji performansi kelas sentimen rata-rata memiliki nilai *precision* sebesar 0.70650820599887, nilai *recall* sebesar 0.67530589543938, dan *f1 score* sebesar 0.67708253161959. sedangkan pada hasil uji performansi kelas kategori rata-rata memiliki nilai *precision* sebesar 0.6731746031746, nilai *recall* sebesar 0.6731746031746, dan *f1 score* sebesar 0.63077568708643. Hasil dari uji performansi tersebut menunjukkan bahwa klasifikasi yang dilakukan cukup baik dan sistem

mampu melakukan klasifikasi sentimen dan kategori pada data tweet secara otomatis. Dalam penelitian ini, uji performansi menghasilkan rentang 0.6-0.7 dikarenakan data *training* yang dipakai hanya berjumlah 900 *tweet*, semakin banyak data training yang dipakai maka hasil uji performansi akan meningkat.

4. Pengukuran indeks kebahagiaan ditentukan setelah kelas sentimen pada data *tweet* didapatkan. Dalam perhitungan indeks kebahagiaan, satu data *tweet* mewakili satu sentimen. Indeks kebahagiaan dihitung dengan perbandingan *tweet* positif dan negatif. Data yang digunakan untuk menghitung indeks kebahagiaan harian adalah 2000 *tweet* setiap perhitungan, kurang lebih 0.018% dari data populasi. Hasil dari perhitungan indeks kebahagiaan harian pada penelitian ini yang dilakukan dalam waktu satu minggu yaitu sebesar 0.46564885496183 pada hari Sabtu tanggal 8 Juli 2017, 0.4436146377894 pada hari Minggu, tanggal 9 Juli 2017, 0.38550501156515 pada hari Senin tanggal 10 Juli 2017, 0.40909090909091 pada hari Selasa tanggal 11 Juli 2017, 0.42654364797729 pada hari Rabu tanggal 12 Juli 2017, 0.46573426573427 pada hari Kamis tanggal 13 Juli 2017, dan 0.45211581291759 pada hari Jum'at tanggal 14 Juli 2017.

6.2 Saran

Penulis menyarankan pengembangan penelitian lebih lanjut sistem pengukuran indeks kebahagiaan sebagai berikut:

1. Implementasi kode sebaiknya menggunakan bahasa pemrograman selain *PHP*, seperti *python*, agar tidak lama dalam mengeksekusi sebuah program.
2. Pada penelitian ini, *dataset* hasil proses *crawling* yang didapat masih belum sesuai dengan yang diharapkan. Maka dari itu, diharapkan pada penelitian selanjutnya memiliki penambahan dan fitur seleksi penyaringan data agar *dataset* yang didapatkan lebih sesuai dengan yang diharapkan.

3. Untuk mendapatkan akurasi yang lebih baik diharapkan menambahkan data *training*. Selain itu, menambah proses pengolahan data, contohnya menggunakan *POS Tagging* atau *n-gram*,
4. Penelitian selanjutnya dapat menggunakan Facebook, Youtube, Instagram dan sosial media lainnya untuk menjangkau data lebih banyak.



DAFTAR PUSTAKA

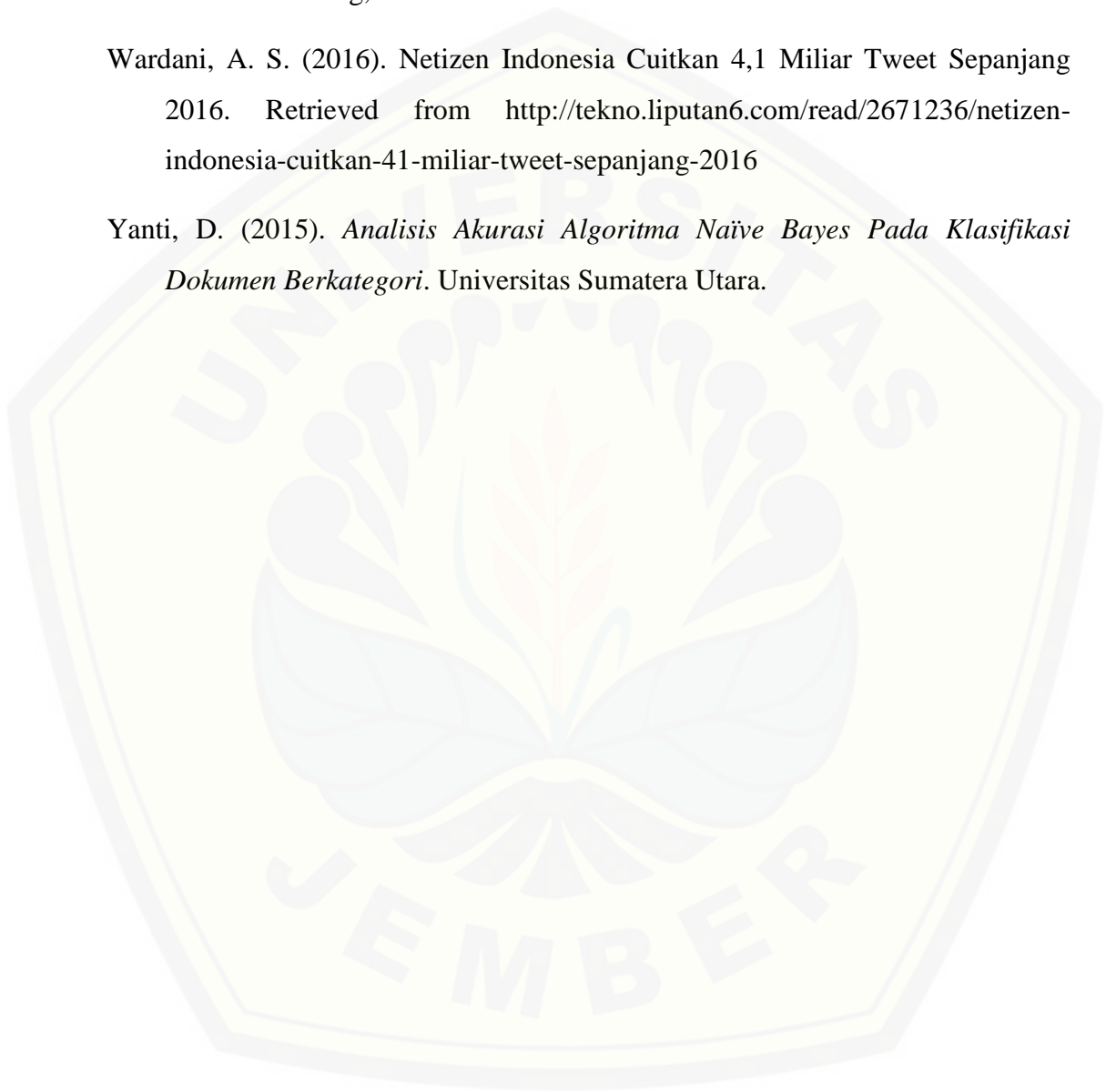
- APJII. (2016). *Penetrasi dan Perilaku Pengguna Internet di Indonesia*. Jakarta.
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology*, 2(5), 2049–3444.
- Bismantara. (2005). *Economic Of Happiness : Kajian Teoritis Bagi Pengukuran Alternatif Kesejahteraan Negara*.
- Herdiani, A., Selviandro, N., & Fitra, D. I. (2016). Pengukuran Indeks Kebahagiaan Masyarakat Kota Bandung dari Jejaring Sosial Twitter Menggunakan Ontologi dengan Paradigma Bottom-up. *E-Proceeding of Engineering*, 3(2), 3596–3602.
- Hidayatullah, A. F. (2016). Penerapan Text Mining dalam Klasifikasi Judul Skripsi, 33–36.
- Kaku, R. (2014). *Penerapan Metode Naive Bayes untuk Klasifikasi Jagung Produktif di Gorontalo*. Universitas Negeri Gorontalo.
- Manalu, B. U. (2014). Analisis Sentimen Pada Twitter Menggunakan Text Mining. *Teknologi Informasi Fak. ILKOM UNSUT*.
- Nurzahputra, A., & Muslim, M. A. (2016). Analisis Sentimen pada Opini Mahasiswa Menggunakan Natural Language Processing, (Seminar Nasional Ilmu Komputer (SNIK 2016)), 114–118.
- Rahayu Ponilan, I., Herdiani, A., & Selviandro, N. (2016). Pengukuran Happiness Index Masyarakat Kota Bandung pada Media Sosial Twitter Menggunakan Pendekatan Ontologi Top-Down Hierarchy. *Indosc 2016*, 17–22.
- Statistik, B. P. (2015). *Indeks Kebahagiaan Indonesia Tahun 2014*.
- Sussolaikah, K., & Alwi, A. (2016). Sentimen Analysis Terhadap Acara Televisi Mata Najwa Berdasarkan Opini Masyarakat Pada Microblogging Twitter,

(November).

Tahitoe, A. D., & Purwitasari, D. (2010). Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming, 1–15.

Wardani, A. S. (2016). Netizen Indonesia Cuitkan 4,1 Miliar Tweet Sepanjang 2016. Retrieved from <http://tekno.liputan6.com/read/2671236/netizen-indonesia-cuitkan-41-miliar-tweet-sepanjang-2016>

Yanti, D. (2015). *Analisis Akurasi Algoritma Naïve Bayes Pada Klasifikasi Dokumen Berkategori*. Universitas Sumatera Utara.



LAMPIRAN

LAMPIRAN A. SKENARIO

A.1. Skenario *Login*

No. Usecase	UC01
Nama Usecase	<i>Login</i>
Aktor	Admin
PreKondisi	Admin telah memiliki <i>email</i> dan <i>password</i> untuk <i>login</i> sebagai admin
PostKondisi	Admin berhasil <i>login</i>
Aliran normal	
Aksi Aktor	Reaksi Sistem
1. Masukkan email dan password	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan halaman <i>dashboard</i> admin
Alur alternatif	
Email address dan Password Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”
	5. Menampilkan halaman awal sistem
Alur alternatif	
Password Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”
	5. Menampilkan halaman awal sistem
Alur alternatif	
Email Kosong	
2. Klik tombol <i>login</i>	

	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”
	5. Menampilkan halaman awal sistem
Alur alternatif <i>Email</i> atau <i>password</i> salah	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “ <i>These credentials do not match our records.</i> ”
	5. Menampilkan halaman awal sistem

No. Usecase	UC01
Nama Usecase	<i>Login</i>
Aktor	Ahli Bahasa
PreKondisi	Ahli Bahasa telah memiliki <i>email</i> dan <i>password</i> untuk <i>login</i> sebagai Ahli Bahasa
PostKondisi	Ahli Bahasa berhasil <i>login</i>
Aliran normal	
Aksi Aktor	Reaksi Sistem
1. Masukkan email dan password	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan halaman <i>dashboard</i> Ahli Bahasa
Alur alternatif Email address dan Password Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”

	5. Menampilkan halaman awal sistem
Alur alternatif Password Kosong	
2. Klik tombol login	
	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”
	5. Menampilkan halaman awal sistem
Alur alternatif Email Kosong	
2. Klik tombol login	
	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”
	5. Menampilkan halaman awal sistem
Alur alternatif Email atau password salah	
2. Klik tombol login	
	3. Sistem mengecek <i>Email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “ <i>These credentials do not match our records.</i> ”
	5. Menampilkan halaman awal sistem

A.2. Skenario Registrasi Akun

No. Usecase	UC02
Nama Usecase	Registrasi Akun
Aktor	Ahli Bahasa
PreKondisi	Ahli Bahasa belum melakukan proses Registrasi
PostKondisi	Ahli Bahasa berhasil Registrasi
Aliran normal	
Aksi Aktor	Reaksi Sistem

1. Masukkan <i>Username, E-mail, Password</i> dan Konfirmasi <i>Password</i> pada halaman registrasi	
2. Klik tombol <i>Sign Up</i>	
	3. Sistem mengecek inputan pada <i>text field</i> yang disediakan
	4. Menampilkan halaman <i>dashboard</i> Ahli Bahasa
Alur alternatif Text Field Kosong	
2. Klik tombol <i>Sign Up</i>	
	3. Sistem mengecek <i>username, email, password</i> dan konfirmas <i>password</i>
	4. Menampilkan <i>alert</i> “harap isi bidang ini”

A.3. Skenario Mengelola *User*

No. Usecase	UC03
Nama Usecase	Mengelola <i>User</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>User</i>
PostKondisi	Admin berhasil melihat, mengedit dan menghapus data <i>User</i>
Aliran normal Lihat data user	
1. Klik menu data <i>user</i>	
	2. Menampilkan halaman data <i>user</i> dengan tabel yang berisi: <ul style="list-style-type: none"> - No - <i>Username</i> - Email - Role

	<ul style="list-style-type: none"> - Action (tombol Edit dan Hapus) Dropdwon Show Tombol Search
Aliran normal Ubah Data <i>User</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu data <i>User</i>	
	<p>2. Menampilkan halaman data <i>user</i> dengan tabel yang berisi:</p> <ul style="list-style-type: none"> - No - <i>Username</i> - Email - Role - Action (tombol Edit dan Hapus) Dropdwon Show Tombol Search
3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit <i>user</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Username</i> - Email - Role Tombol <i>Save</i> Tombol <i>Cancel</i>
5. Mengubah data <i>user</i>	
6. Klik tombol submit	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>

Aliran normal Lihat data <i>user</i>	
1. Klik menu data <i>user</i>	
	2. Menampilkan halaman data <i>user</i> dengan tabel yang berisi: <ul style="list-style-type: none"> - No - <i>Username</i> - Email - Role - Action (tombol Edit dan Hapus) Dropdwon Show Tombol Search
3. Klik tombol hapus	
	4. Menghapus data <i>user</i> dari <i>database</i>
	5. Menampilkan halaman data <i>user</i>

A4. Skenario Mengelola *Tweet*

No. Usecase	UC04
Nama Usecase	Mengelola <i>Tweet</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Tweet</i>
PostKondisi	Admin berhasil melakukan proses <i>tweet</i>
Aliran normal Lihat data <i>tweet</i>	
1. Klik menu <i>Tweet</i>	
	2. Menampilkan halaman <i>Tweet</i> yang berisi: <ul style="list-style-type: none"> - Field Query (berupa inputan)

	<ul style="list-style-type: none"> - Field Jumlah <i>Tweet</i> (berupa inputan) - Tombol Search <li style="padding-left: 40px;">Tabel berisi - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
Aliran normal	
Input Data <i>Training</i> dari <i>Tweet</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Tweet</i>	
	<p>2. Menampilkan halaman <i>Tweet</i> dengan yang berisi:</p> <ul style="list-style-type: none"> - Field Query (berupa inputan) - Field Jumlah <i>Tweet</i> (berupa inputan) - Tombol Search <li style="padding-left: 40px;">Tabel berisi - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
3. Masukkan Query dan Jumlah <i>tweet</i> yang diisikan	
4. Klik Tombol Search	

	5. Menampilkan <i>tweet</i> yang sedang dicari pada kolom query
6. Centang <i>tweet</i> yang diinginkan	
7. Klik Tombol Add -> Jadikan Data Training	
Aliran normal Input Data Testing dari Tweet	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Tweet</i>	
	2. Menampilkan halaman <i>Tweet</i> dengan yang berisi: <ul style="list-style-type: none"> - Field Query (berupa inputan) - Field Jumlah <i>Tweet</i> (berupa inputan) - Tombol Search <li style="padding-left: 40px;">Tabel berisi - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
3. Masukkan Query dan Jumlah <i>tweet</i> yang diisikan	
4. Klik Tombol Search	
	5. Menampilkan <i>tweet</i> yang sedang dicari pada kolom query
6. Centang <i>tweet</i> yang diinginkan	
7. Klik Tombol Add -> Jadikan	

Data Testing	
---------------------	--

No. Usecase	UC04
Nama Usecase	Mengelola <i>Tweet</i>
Aktor	Ahli Bahasa
PreKondisi	Ahli Bahasa memilih menu <i>Tweet</i>
PostKondisi	Ahli Bahasa berhasil melakukan proses <i>tweet</i>
Aliran normal	
Lihat data <i>tweet</i>	
1. Klik menu <i>Tweet</i>	
	<p>2. Menampilkan halaman <i>Tweet</i> yang berisi:</p> <ul style="list-style-type: none"> - Field Query (berupa inputan) - Field Jumlah <i>Tweet</i> (berupa inputan) - Tombol Search <p style="padding-left: 40px;">Tabel berisi</p> <ul style="list-style-type: none"> - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
Aliran normal	
Input Data <i>Training</i> dari <i>Tweet</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Tweet</i>	
	2. Menampilkan halaman <i>Tweet</i>

	<p>dengan yang berisi:</p> <ul style="list-style-type: none"> - Field Query (berupa inputan) - Field Jumlah <i>Tweet</i> (berupa inputan) - Tombol Search <p>Tabel berisi</p> <ul style="list-style-type: none"> - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
3. Masukkan Query dan Jumlah <i>tweet</i> yang diisikan	
4. Klik Tombol Search	
	5. Menampilkan <i>tweet</i> yang sedang dicari pada kolom query
6. Centang <i>tweet</i> yang diinginkan	
7. Klik Tombol Add -> Jadikan <i>Data Training</i>	
Aliran normal	
Input Data <i>Testing</i> dari <i>Tweet</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Tweet</i>	
	<p>2. Menampilkan halaman <i>Tweet</i> dengan yang berisi:</p> <ul style="list-style-type: none"> - Field Query (berupa inputan) - Field Jumlah <i>Tweet</i> (berupa inputan)

	<ul style="list-style-type: none"> - Tombol Search <li style="padding-left: 20px;">Tabel berisi - No - <i>User</i> - <i>Tweet</i> - Waktu Post - Tombol Tambah
3. Masukkan Query dan Jumlah <i>tweet</i> yang diisikan	
4. Klik Tombol Search	
	5. Menampilkan <i>tweet</i> yang sedang dicari pada kolom query
6. Centang <i>tweet</i> yang diinginkan	
7. Klik Tombol Add -> Jadikan <i>Data Testing</i>	

A.5. Skenario Mengelola Data *Training*

No. Usecase	UC05
Nama Usecase	Mengelola Data <i>Training</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Training</i> Tabel
PostKondisi	Admin berhasil melihat, mengedit dan menghapus data <i>training</i>
Aliran normal	
Lihat Data <i>Training</i>	
1. Klik menu <i>Training</i> Tabel	
	2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:

	<ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search - Tabel yang berisi <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time - Action (tombol Edit dan Hapus)
<p>Aliran normal</p> <p>Mengedit Data <i>Training</i></p>	
<p>1. Klik menu <i>Training</i> Tabel</p>	<p>2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search - Tabel yang berisi <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time

3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit Data <i>Training</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> - Sentimen - Kategori <p>Tombol <i>Save</i> Tombol <i>Close</i></p>
5. Mengubah data <i>Training</i>	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal	
Hapus Data <i>Training</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Training</i> Tabel	
	<p>2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i>

	- Post Time
3. Klik tombol hapus	
	4. Menghapus data <i>Training</i> dari <i>database</i>
	5. Menampilkan halaman data <i>Training</i>

No. Usecase	UC05
Nama Usecase	Mengelola Data <i>Training</i>
Aktor	Ahli Bahasa
PreKondisi	Ahli Bahasa memilih menu <i>Training</i> Tabel
PostKondisi	Ahli Bahasa berhasil melihat, mengedit dan menghapus data <i>training</i>
Aliran normal	
Lihat Data <i>Training</i>	
1. Klik menu <i>Training</i> Tabel	
	2. Menampilkan halaman data <i>Training</i> Tabel yang berisi: <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search Tabel yang berisi - No - <i>Username</i>

	<ul style="list-style-type: none"> - <i>Tweet</i> - Post Time - Action (tombol Edit dan Hapus)
Aliran normal Mengedit Data <i>Training</i>	
1. Klik menu <i>Training</i> Tabel	
	<p>2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time
3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit <i>user</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> - Sentimen - Kategori <p style="padding-left: 40px;">Tombol <i>Save</i></p> <p style="padding-left: 40px;">Tombol <i>Close</i></p>
5. Mengubah data <i>Training</i>	

6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal Hapus Data <i>Training</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Training</i> Tabel	
	<p>2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time
3. Klik tombol hapus	
	4. Menghapus data <i>Training</i> dari <i>database</i>
	5. Menampilkan halaman data <i>Training</i>
No. Usecase	UC05
Nama Usecase	Melihat <i>Training</i> Tabel

Aktor	Ahli bahasa
PreKondisi	Ahli bahasa memilih menu <i>Training</i> Tabel
PostKondisi	Ahli bahasa berhasil melihat data <i>Training</i> Tabel
Aliran normal	
Lihat data <i>Training</i> Tabel	
1. Klik menu <i>Data Training</i>	
	<p>2. Menampilkan halaman data <i>Training</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time

A.6. Skenario Melihat *Text Mining* Data *Training*

No. Usecase	UC06
Nama Usecase	Melihat <i>Text Mining</i> Data <i>Training</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Data Training</i> <i>Text Mining</i>

PostKondisi	Admin berhasil melihat <i>text mining</i> Data <i>Training</i>
Aliran normal Lihat data <i>Text Preprocessing</i>	
1. Klik menu <i>Training Text Preprocessing</i>	
	<p>2. Menampilkan halaman data <i>Training Preprocessing</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Tranformation</i> - Tabs <i>Text Unigram</i> - Tabs <i>Text Unigram Frequency</i> <p>Total</p> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Data Awal - Hasi <i>Preprocessing</i>
Aliran normal Lihat data <i>Text Transformation</i>	
1. Klik Tabs <i>Text Transformation</i>	
	<p>2. Menampilkan halaman data <i>Training Preprocessing</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Unigram</i> - Tabs <i>Text Unigram Frequency</i> <p>Total</p> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Hasil <i>Preprocessing</i>

	<ul style="list-style-type: none"> - <i>Stemming</i> - <i>Stopword Removal</i>
Aliran normal Lihat data <i>Unigram</i>	
1. Klik Tabs <i>Unigram</i>	
	<p>2. Menampilkan halaman data <i>Training Unigram</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram Frequency</i> <p style="padding-left: 40px;">Total</p> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Hasil <i>Stopword Removal</i> - <i>Unigram</i>
Aliran normal Lihat data Frekuensi <i>Unigram</i>	
1. Klik Tabs <i>Unigram Frequency</i>	
	<p>2. Menampilkan halaman data <i>Training Unigram Frequency</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram</i> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Unigram</i> - Frekuensi

A.7. Skenario Mengelola Data *Testing*

No. Usecase	UC07
Nama Usecase	Mengelola Data <i>Testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu Data <i>Testing</i> Tabel
PostKondisi	Admin berhasil melihat, mengedit dan menghapus data <i>Testing</i>
Aliran normal	
Lihat Data <i>Testing</i>	
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Username - Tweet - Post Time - Action (tombol Edit dan Hapus)
Aliran normal	
Mengedit Data <i>Testing</i>	
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p>

	<ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search - Tabel yang berisi <ul style="list-style-type: none"> - No - Username - Tweet - Post Time - Action (tombol Edit dan Hapus)
3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit Data <i>Testing</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> - Sentimen - Kategori <p>Tombol <i>Save</i> Tombol <i>Close</i></p>
5. Mengubah data <i>Testing</i>	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal	
Hapus Data <i>Testing</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Testing</i> Tabel	

	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time - Action (tombol Edit dan Hapus)
3. Klik tombol hapus	
	4. Menghapus data <i>Testing</i> dari <i>database</i>
	5. Menampilkan halaman data <i>Testing</i>

No. Usecase	UC07
Nama Usecase	Mengelola Data <i>Testing</i>
Aktor	Ahli Bahasa
PreKondisi	Ahli Bahasa memilih menu Data <i>Testing</i> Tabel
PostKondisi	Ahli Bahasa berhasil melihat, mengedit dan menghapus data <i>Testing</i>
Aliran normal	
Lihat Data <i>Testing</i>	

<p>1. Klik menu <i>Testing</i> Tabel</p>	
	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i> - <i>Tweet</i> - Post Time - Action (tombol Edit dan Hapus)
<p>Aliran normal Mengedit Data <i>Testing</i></p>	
<p>1. Klik menu <i>Testing</i> Tabel</p>	
	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Username</i>

	<ul style="list-style-type: none"> - <i>Tweet</i> - Post Time
3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit Data <i>Testing</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> - Sentimen - Kategori <p>Tombol <i>Save</i> Tombol <i>Close</i></p>
5. Mengubah data <i>Testing</i>	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal	
Hapus Data <i>Testing</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman data <i>Testing</i> Tabel yang berisi:</p> <ul style="list-style-type: none"> - Chart berbentuk Pie yang berisi Sentimen Data dan Kategori Data - Dropdown Show data yang ditampilkan - Tombol Search <p>Tabel yang berisi</p>

	<ul style="list-style-type: none"> - No - Username - Tweet - Post Time
3. Klik tombol hapus	
	4. Menghapus data <i>Testing</i> dari <i>database</i>
	5. Menampilkan halaman data <i>Testing</i>

A.8. Skenario Melihat *Text Mining* data *Testing*

No. Usecase	UC08
Nama Usecase	Melihat <i>Text Mining</i> Data <i>Testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu Data <i>Testing Text Mining</i>
PostKondisi	Admin berhasil melihat <i>text mining</i> Data <i>Testing</i>
Aliran normal	
Lihat data <i>Text Preprocessing</i>	
1. Klik menu <i>Testing Text Preprocessing</i>	
	2. Menampilkan halaman data <i>Testing Preprocessing</i> yang berisi: <ul style="list-style-type: none"> - Tabs <i>Text Tranformation</i> - Tabs <i>Text Unigram</i> - Tabs <i>Text Unigram Frequency</i>

	<p>Total</p> <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Data Awal - Hasi <i>Preprocessing</i>
<p>Aliran normal</p> <p>Lihat data <i>Text Transformation</i></p>	
1. Klik Tabs <i>Text Transformation</i>	
	<p>2. Menampilkan halaman data <i>Testing Preprocessing</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Unigram</i> - Tabs <i>Text Unigram Frequency</i> <p>Total</p> <p>Tabel yang berisi</p> <ul style="list-style-type: none"> - No - Hasil <i>Preprocessing</i> - <i>Stemming</i> - <i>Stopword Removal</i>
<p>Aliran normal</p> <p>Lihat data <i>Unigram</i></p>	
1. Klik Tabs <i>Unigram</i>	
	<p>2. Menampilkan halaman data <i>Testing Unigram</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram Frequency</i> <p>Total</p> <p>Tabel yang berisi</p>

	<ul style="list-style-type: none"> - No - Hasil <i>Stopword</i> Removal - <i>Unigram</i>
Aliran normal Lihat data Frekuensi <i>Unigram</i>	
1. Klik Tabs <i>Unigram</i> Frequency	
	<p>2. Menampilkan halaman data <i>Testing Unigram Frequency</i> yang berisi:</p> <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram</i> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No - <i>Unigram</i> - Frekuensi

A.9. Skenario Mengelola *Stopword*

No. Usecase	UC10
Nama Usecase	Mengelola <i>Stopword</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Stopword</i>
PostKondisi	Admin berhasil menambahkan, mengedit, menghapus, Melihat data <i>Stopword</i>
Aliran normal Tambah/input data <i>Stopword</i>	
Aksi Aktor	Reaksi Sistem

1. Klik menu data <i>Stopword</i>	
	<p>2. Menampilkan halaman data <i>Stopword</i>, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - <i>Stopword</i> - Action (edit dan hapus)
3. Klik tombol tambah	
	<p>4. Menampilkan modal berupa <i>form</i> Input <i>stopword</i> yang berisi:</p> <ul style="list-style-type: none"> - Field <i>Stopword</i> - Tombol Close dan <i>Save</i>
5. Mengisi <i>form</i> input	
6. Klik tombol <i>Save</i>	
	7. Menambahkan data
	8. Menampilkan halaman data <i>Stopword</i>
Alur Alternatif	
Data Input Yang Kosong	
6. Klik tombol <i>save</i>	
	7. Menampilkan span " <i>harap isi bidang ini</i> "
Aliran normal	
Lihat data <i>Stopword</i>	
1. Klik menu data <i>Stopword</i>	
	2. Menampilkan halaman data

	<p><i>Stopword</i>, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - <i>Stopword</i> <p>Action (edit dan hapus)</p>
<p>Aliran normal Ubah data <i>Stopword</i></p>	
Aksi Aktor	Reaksi Sistem
1. Klik menu data <i>Stopword</i>	
	<p>2. Menampilkan halaman data <i>Stopword</i>, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - <i>Stopword</i> <p>Action (edit dan hapus)</p>
3. Klik tombol edit	
	<p>4. Menampilkan Modal <i>form</i> Edit <i>Stopword</i> yang berisi:</p> <ul style="list-style-type: none"> - <i>Stopword</i> - Tombol Close dan <i>Save</i>
5. Mengubah data <i>Stopword</i>	
6. Klik tombol <i>save</i>	
	<p>7. Mengubah data</p>

	8. Menampilkan halaman data <i>Stopword</i>
Alur Alternatif Data Ubah <i>Stopword</i> Ada Yang Kosong	
6. Klik tombol <i>save</i>	
	7. Menampilkan span “ <i>harap isi bidang ini</i> ”
Aliran normal Hapus data <i>Stopword</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu data <i>Stopword</i>	
	2. Menampilkan halaman data <i>Stopword</i> , meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi: - no - <i>Stopword</i> Action (edit dan hapus)
3. Klik tombol hapus	
	4. Menghapus data
	5. Menampilkan halaman data <i>Stopword</i>

A.10. Skenario Mengelola Ontologi

No. Usecase	UC11
Nama Usecase	Mengelola Ontologi
Aktor	Admin

PreKondisi	Admin memilih menu Ontologi
PostKondisi	Admin berhasil menambahkan, mengedit, menghapus, Melihat data Ontologi
Aliran normal	
Tambah/input data Ontologi	
Aksi Aktor	Reaksi Sistem
1. Klik menu data Ontologi	
	<p>2. Menampilkan halaman data Ontologi, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - Kata - Parameter - Action (edit dan hapus)
3. Klik tombol tambah	
	<p>4. Menampilkan modal berupa <i>form</i> Input ontologi yang berisi:</p> <ul style="list-style-type: none"> - Field Ontologi - Tombol Close dan <i>Save</i>
5. Mengisi <i>form</i> input	
6. Klik tombol <i>Save</i>	
	7. Menambahkan data
	8. Menampilkan halaman data Ontologi
Alur Alternatif	
Data Input Yang Kosong	

6. Klik tombol <i>save</i>	
	7. Menampilkan span “ <i>harap isi bidang ini</i> ”
Aliran normal	
Lihat data Ontologi	
1. Klik menu data Ontologi	
	<p>2. Menampilkan halaman data Ontologi, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - Kata - Parameter <p>Action (edit dan hapus)</p>
Aliran normal	
Ubah data Ontologi	
Aksi Aktor	Reaksi Sistem
1. Klik menu data Ontologi	
	<p>2. Menampilkan halaman data Ontologi, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - Kata - Parameter <p>Action (edit dan hapus)</p>
3. Klik tombol edit	
	4. Menampilkan Modal <i>form</i> Edit Ontologi yang berisi:

	<ul style="list-style-type: none"> - Ontologi - Tombol Close dan <i>Save</i>
5. Mengubah data Ontologi	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data Ontologi
Alur Alternatif	
Data Ubah Ontologi Ada Yang Kosong	
6. Klik tombol <i>save</i>	
	7. Menampilkan span " <i>harap isi bidang ini</i> "
Aliran normal	
Hapus data Ontologi	
Aksi Aktor	Reaksi Sistem
1. Klik menu data Ontologi	
	<p>2. Menampilkan halaman data Ontologi, meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi:</p> <ul style="list-style-type: none"> - no - Kata - Parameter <p>Action (edit dan hapus)</p>
3. Klik tombol hapus	
	4. Menghapus data

	5. Menampilkan halaman data Ontologi
--	--------------------------------------

A.11. Melihat Indeks Kebahagiaan Data *Testing*

No. Usecase	UC12
Nama Usecase	Melihat Indeks Kebahagiaan Data <i>Testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu Data <i>Testing</i> Indeks Kebahagiaan
PostKondisi	Admin berhasil melihat Indeks Kebahagiaan Data <i>Testing</i>
Aliran normal	
Lihat data Indeks Kebahagiaan Data <i>Testing</i>	
1. Klik menu Data <i>Testing</i> Indeks Kebahagiaan	
	2. Menampilkan halaman Data <i>Testing</i> Indeks Kebahagiaan yang berisi: <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Indeks Kebahagiaan

A.12. Mengelola Data Indeks Kebahagia Harian

No. Usecase	UC13
Nama Usecase	Mengelola Data Indeks Kebahagiaan Harian
Aktor	Admin
PreKondisi	Admin memilih menu Indeks

	Kebahagiaan
PostKondisi	Admin berhasil menambah, melihat dan menghapus indeks kebahagiaan harian
Aliran normal	
Lihat Data Indeks Kebahagiaan Harian	
1. Klik menu	
	<p>2. Menampilkan halaman Indeks Kebahagiaan yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Indeks Kebahagiaan <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - Indeks - Tanggal - Action (tombol Hapus) <p style="padding-left: 40px;">Tombol Klasifikasi Hari ini</p>
Aliran normal	
Menambah Indeks Kebahagiaan	
1. Klik menu	
	<p>2. Menampilkan halaman Data <i>Testing</i> Indeks Kebahagiaan yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Indeks Kebahagiaan <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - Indeks - Tanggal - Action (tombol Hapus) <p style="padding-left: 40px;">Tombol Klasifikasi Hari ini</p>

3. Klik Tombol Klasifikasi Hari Ini	
	<p>4. Menampilkan halaman Indeks Kebahagiaan Hari Ini dengan tabel yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> - Sentimen Positif - <i>Tweet</i> Positif - <i>Tweet</i> Negatif - Indeks Kebahagiaan - Tanggal
5. Klik Save	
	6. Kembali ke halaman Indeks Kebahagiaan
Aliran normal Menghapus Indeks Kebahagiaan	
1. Klik menu	
	<p>2. Menampilkan halaman Data <i>Testing</i> Indeks Kebahagiaan yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Indeks Kebahagiaan - Tabel yang berisi <ul style="list-style-type: none"> - Indeks - Tanggal - Action (tombol Hapus) <p>Tombol Klasifikasi Hari ini</p>
3. Klik Tombol Hapus	
	4. Menghapus data indeks kebahagiaan dari <i>database</i>

	5. Menampilkan halaman Indeks Kebahagiaan
--	---

A.13. Melihat *FAQ*

No. Usecase	UC14
Nama Usecase	Melihat <i>FAQ</i>
Aktor	Ahli bahasa
PreKondisi	Ahli bahasa memilih menu <i>FAQ</i>
PostKondisi	Ahli bahasa berhasil melihat data <i>FAQ</i> Tabel
Aliran normal Lihat Halaman Fax	
1. Klik menu Fax	
	2. Menampilkan halaman <i>FAQ</i> Tabel yang berisi: <ul style="list-style-type: none"> - Penjelasan Fungsi Ahli bahasa - Penjelasan Cara Melakukan Klasifikasi

A.14. *Logout*

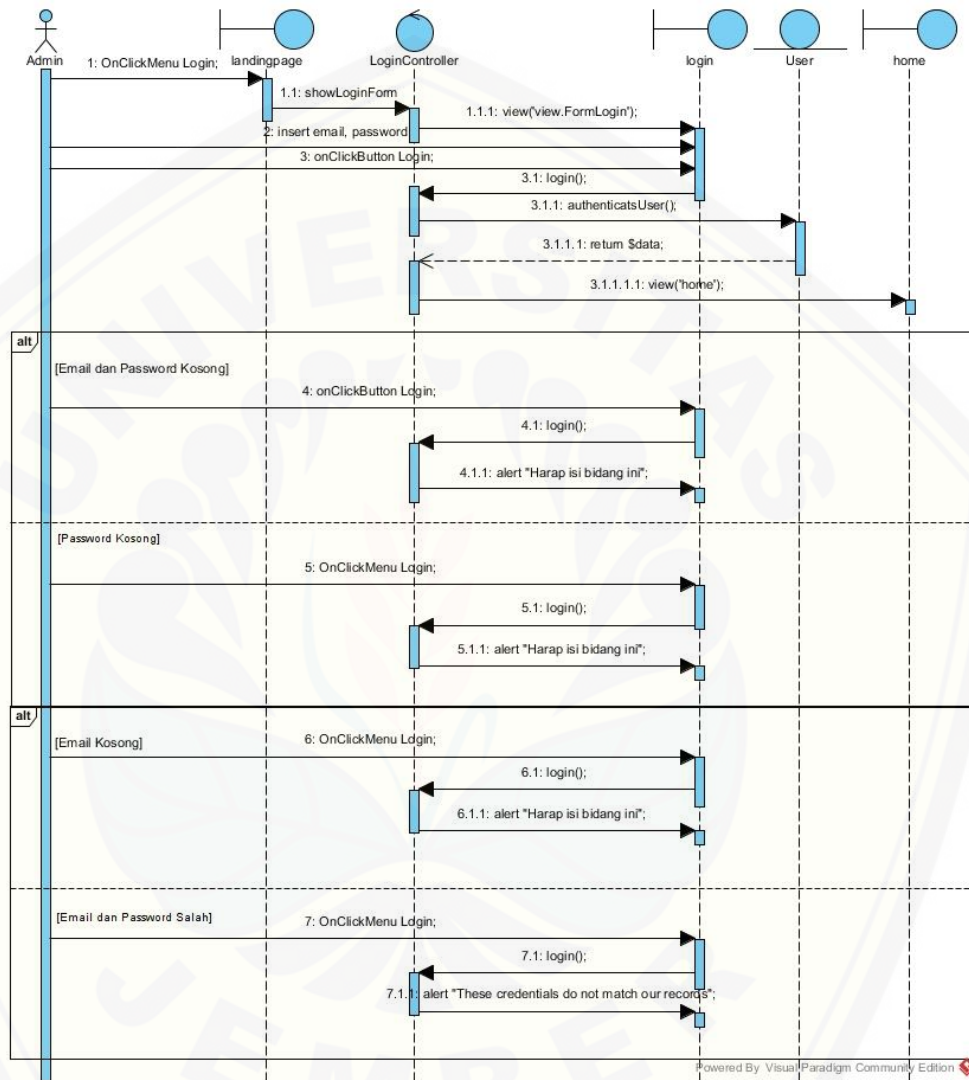
No. Usecase	UC15
Nama Usecase	<i>Logout</i>
Aktor	Admin
Prekondisi	Admin Memilih Tombol keluar
Postkondisi	Admin Berhasil Keluar
Aliran normal	

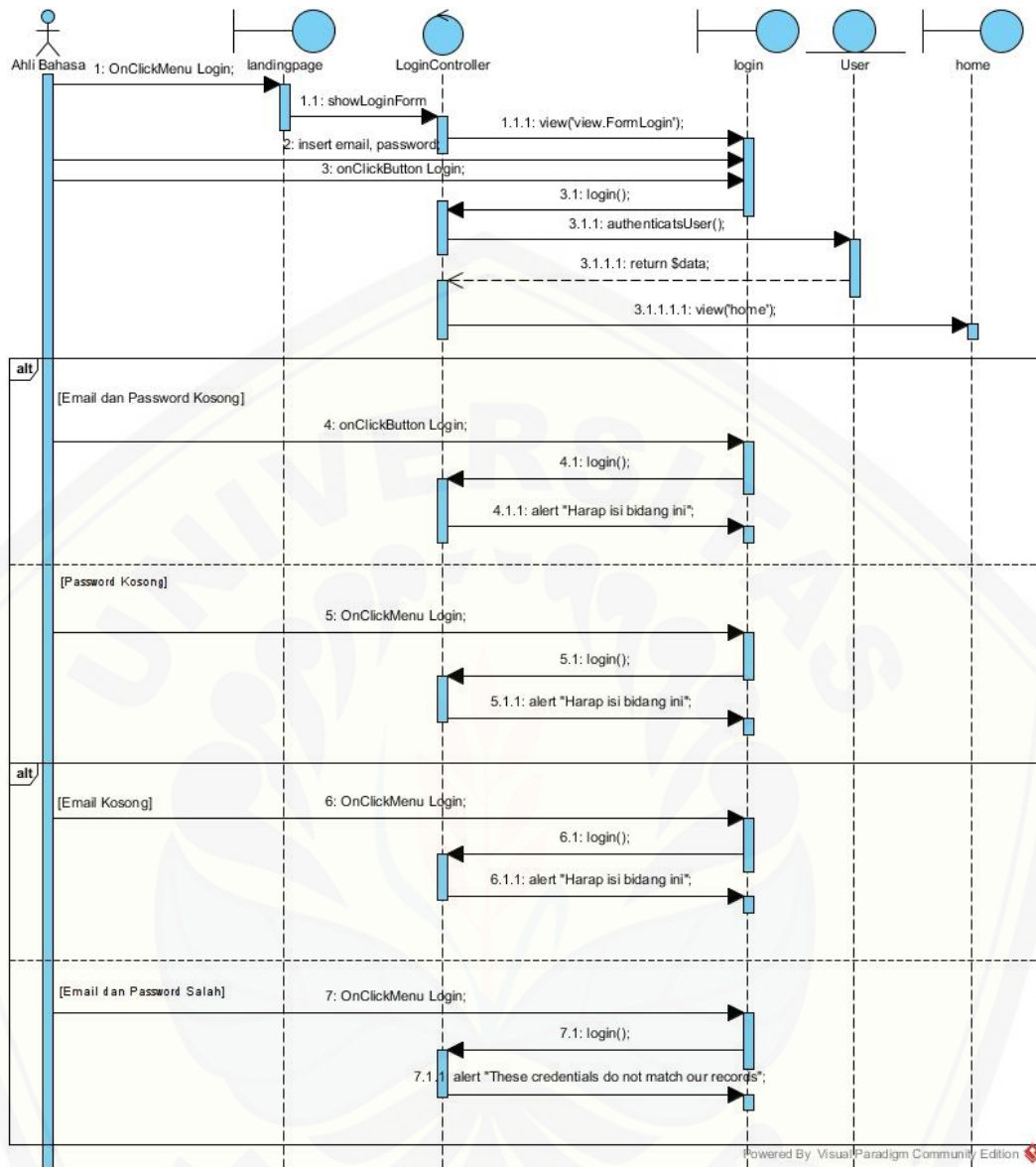
Keluar	
Aksi Aktor	Reaksi Sistem
1. Klik tombol keluar	
	2. Menampilkan halaman sistem sebelum <i>login</i>

No. Usecase	UC15
Nama Usecase	<i>Logout</i>
Aktor	Ahli Bahasa
Prekondisi	Ahli Bahasa Memilih Tombol keluar
Postkondisi	Ahli Bahasa Berhasil Keluar
Aliran normal	
Keluar	
Aksi Aktor	Reaksi Sistem
1. Klik tombol keluar	
	2. Menampilkan halaman sistem sebelum <i>login</i>

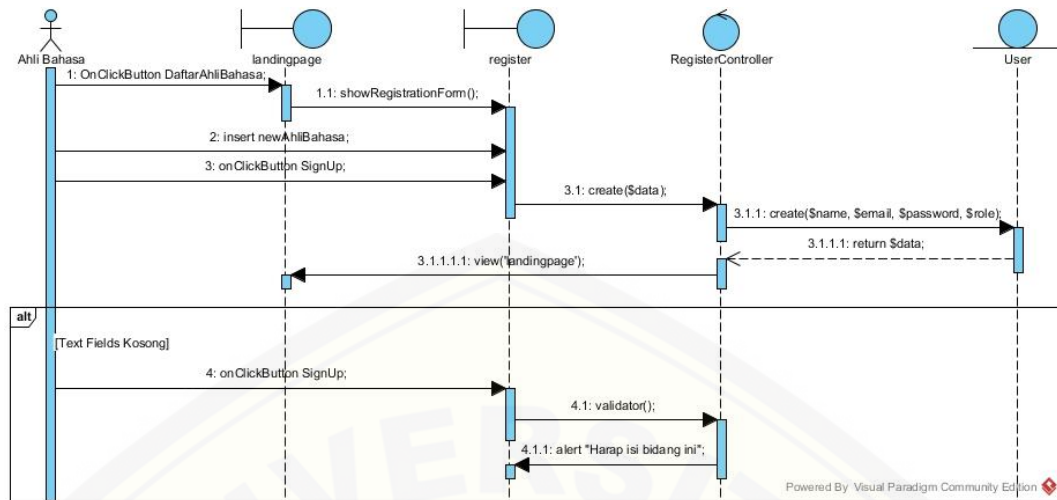
LAMPIRAN B. SEQUENCE DIAGRAM

B.1. Sequence Diagram Login

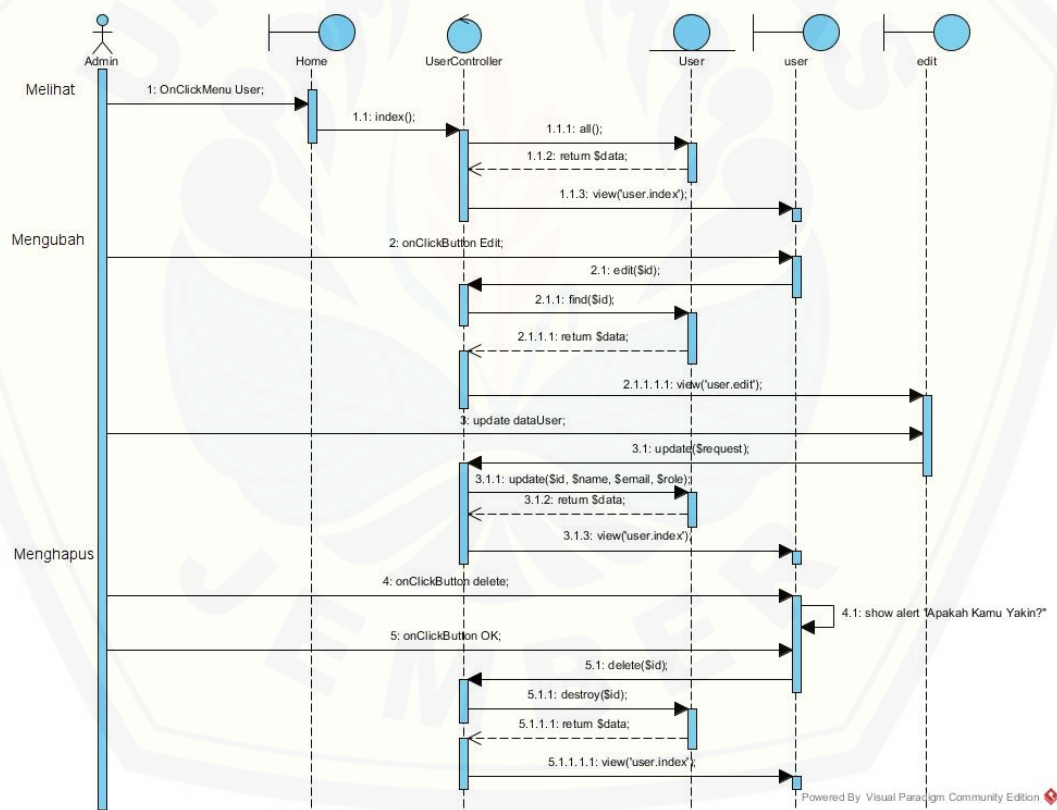




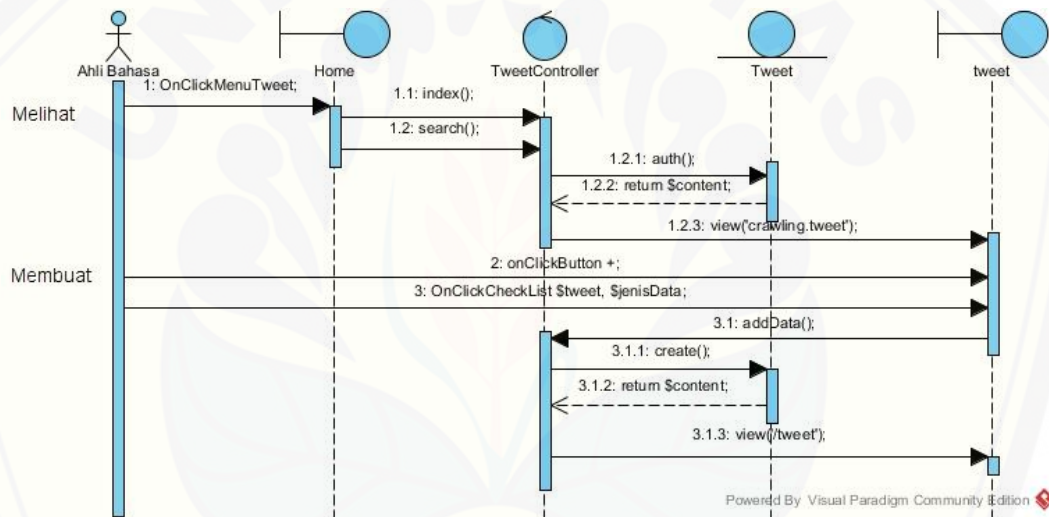
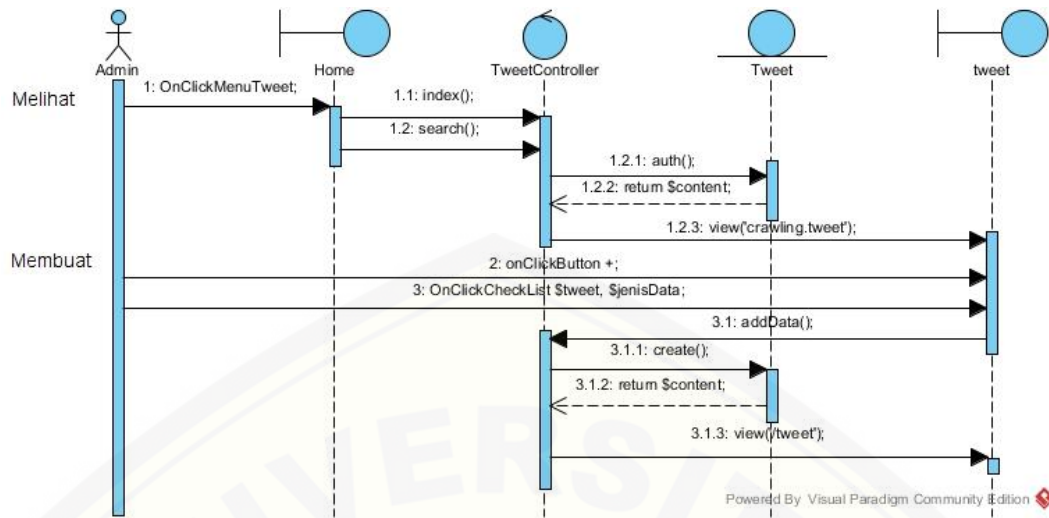
B.2. Sequence Diagram Registrasi Akun



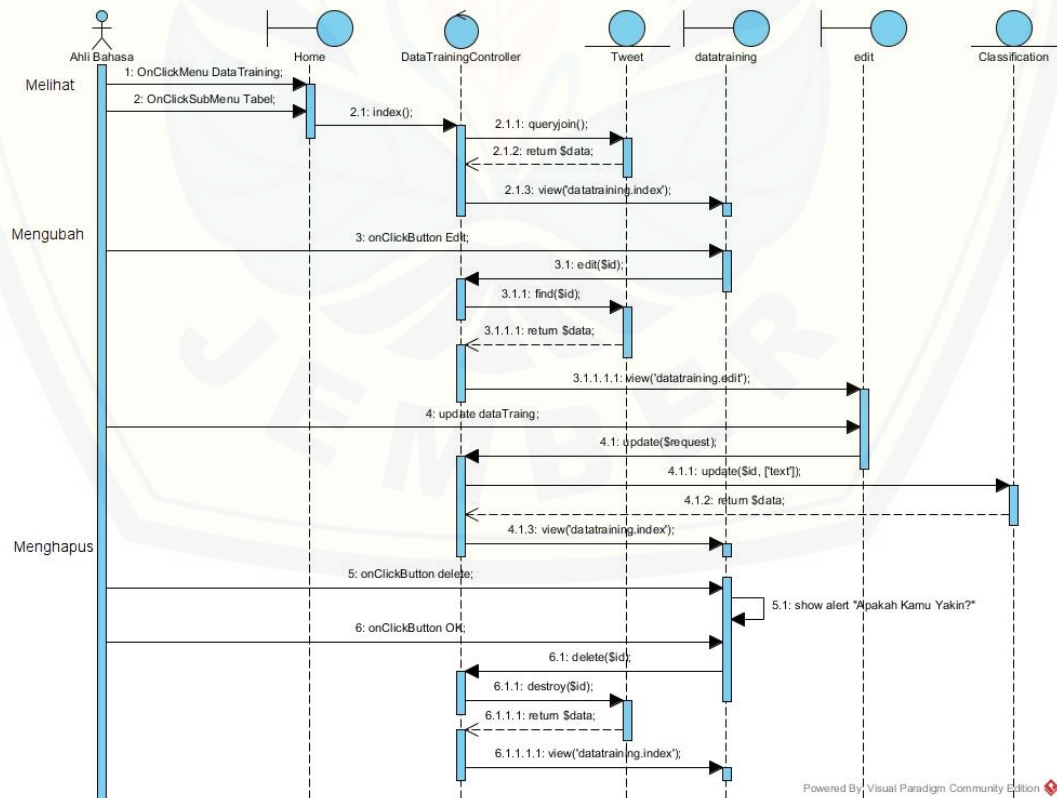
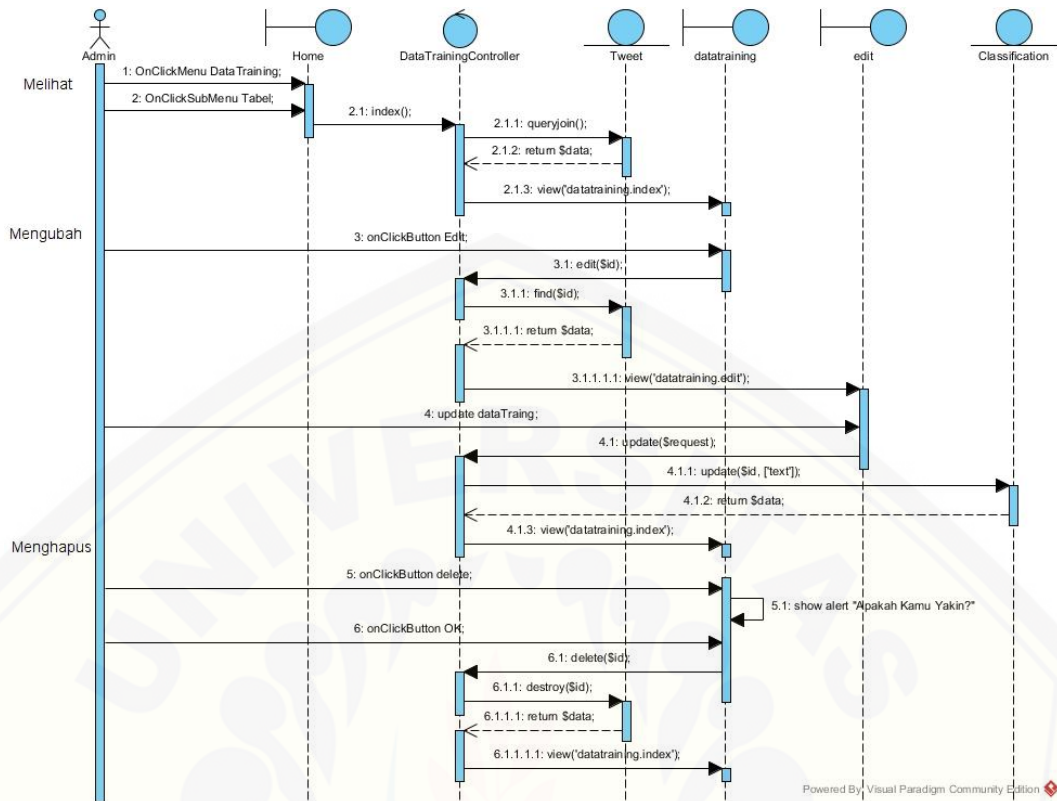
B.3. Sequence Diagram Mengelola User



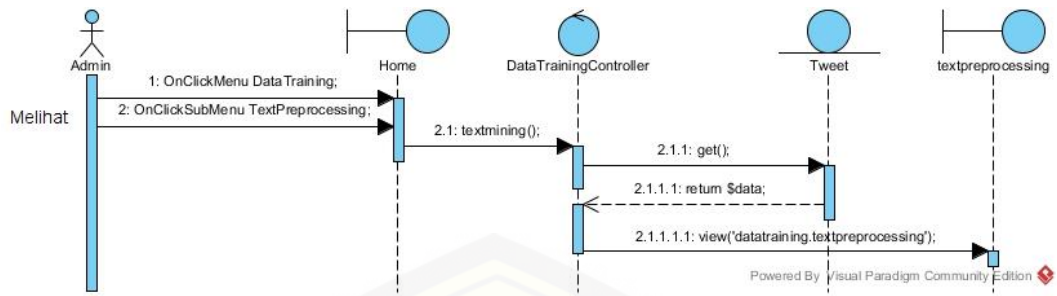
B.4. Sequence Diagram Mengelola Tweet



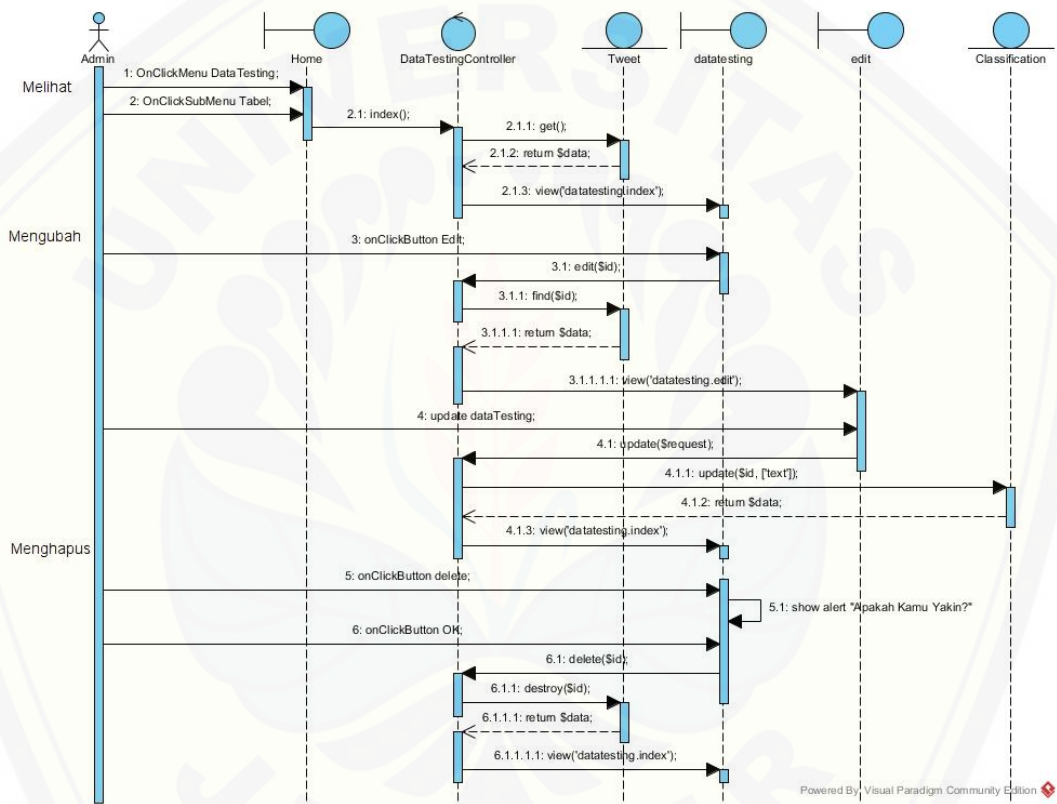
B.5. Sequence Diagram Mengelola Data Training

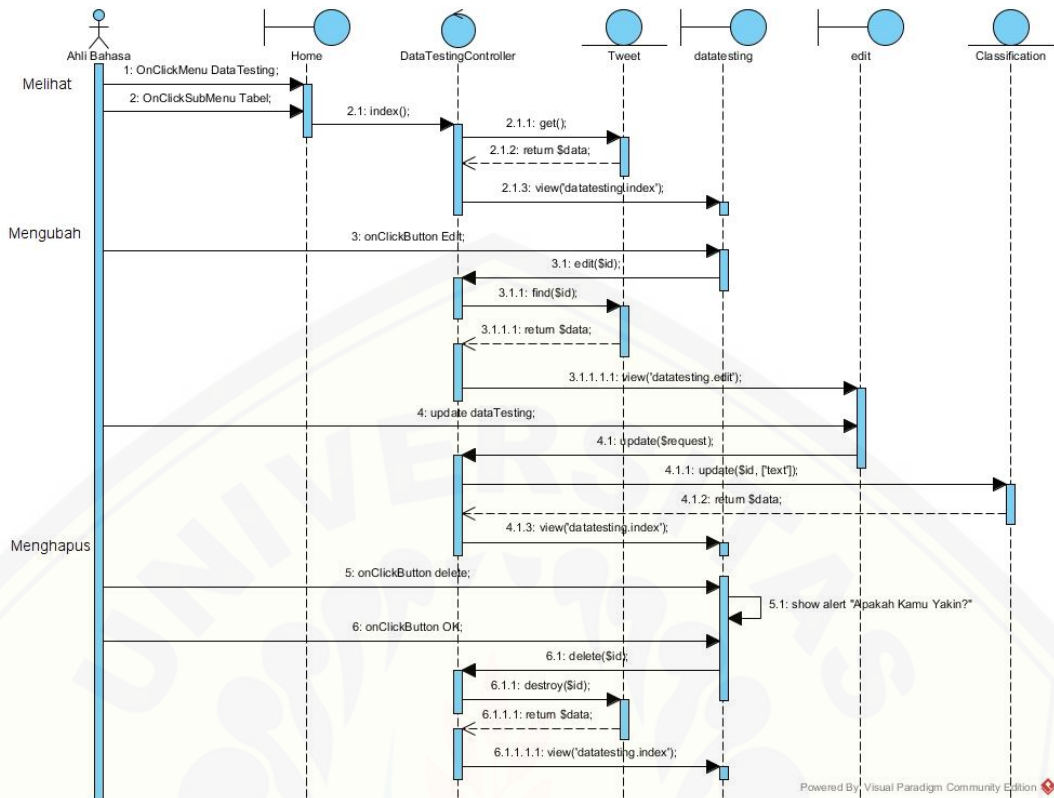


B.6. Sequence Diagram Melihat Text Mining Data Training

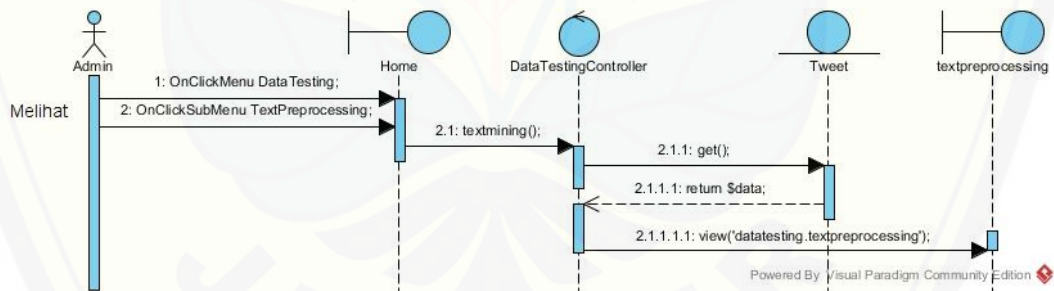


B.7. Sequence Diagram Mengelola Data Testing

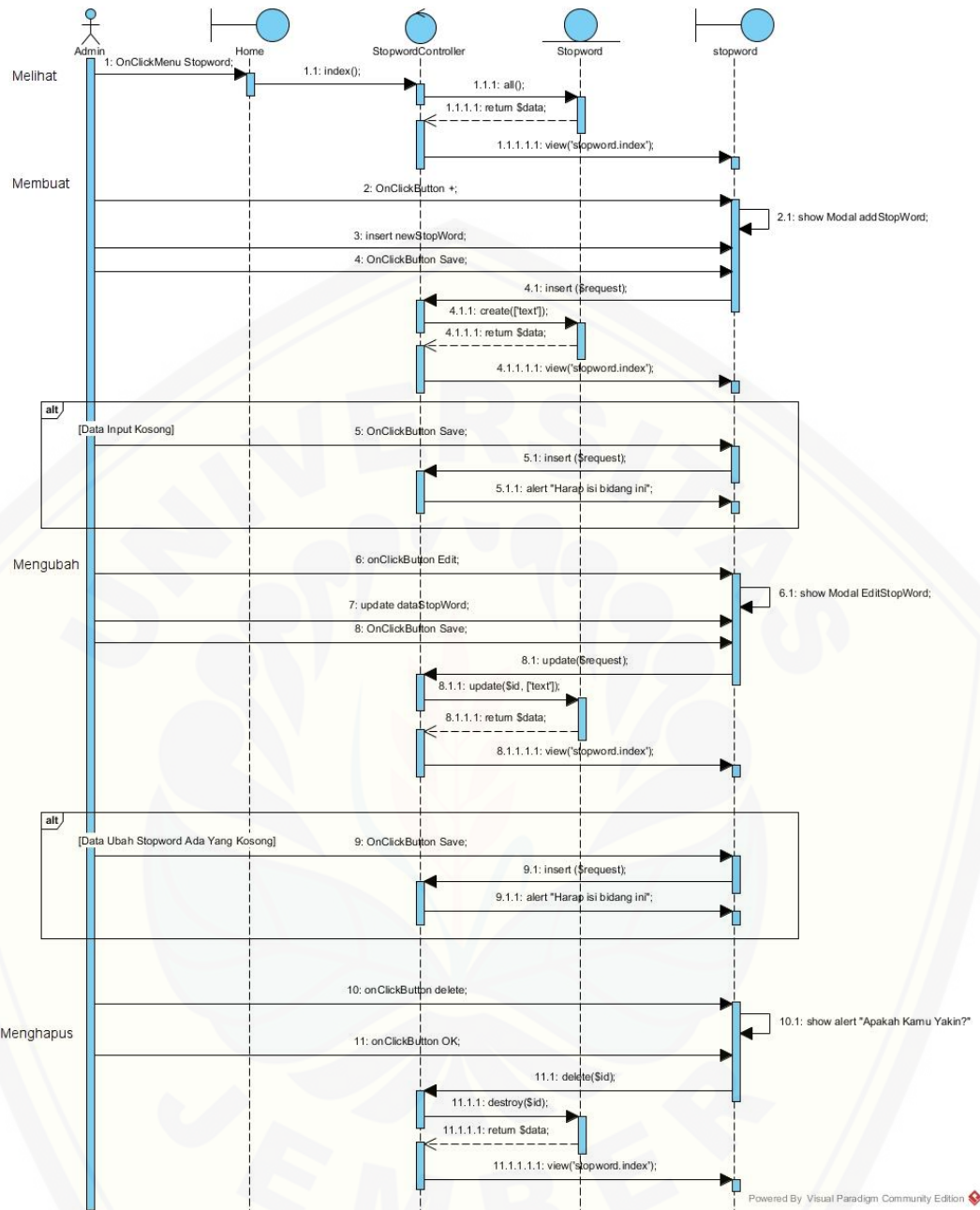




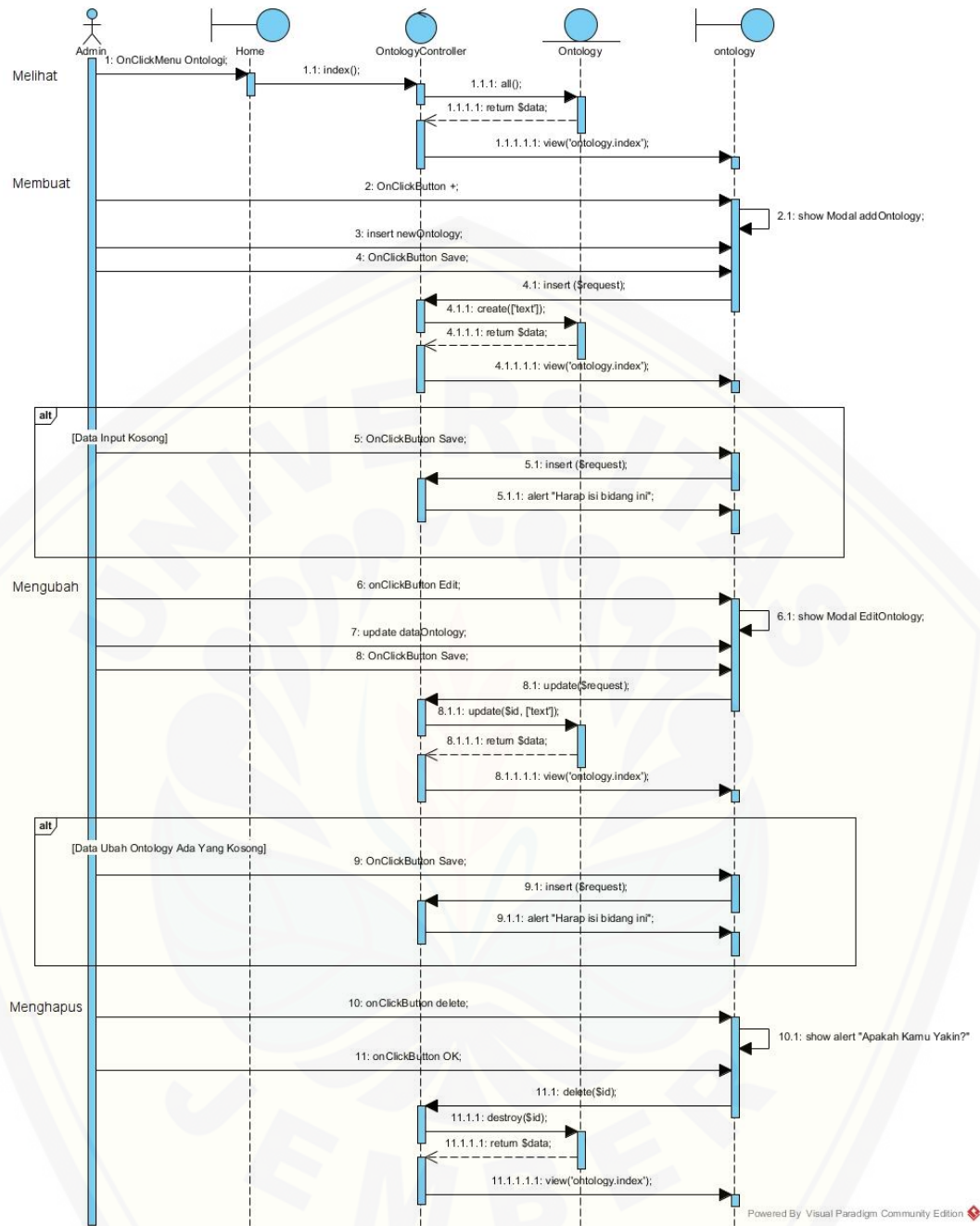
B.8. Sequence Diagram Melihat Text Mining data Testing



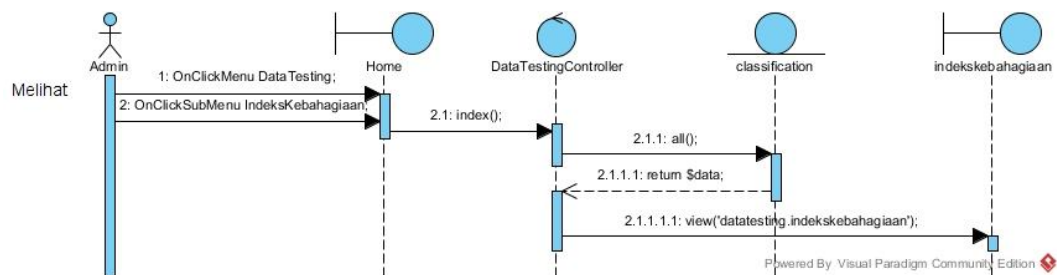
B.9. Sequence Diagram Mengelola Stopword



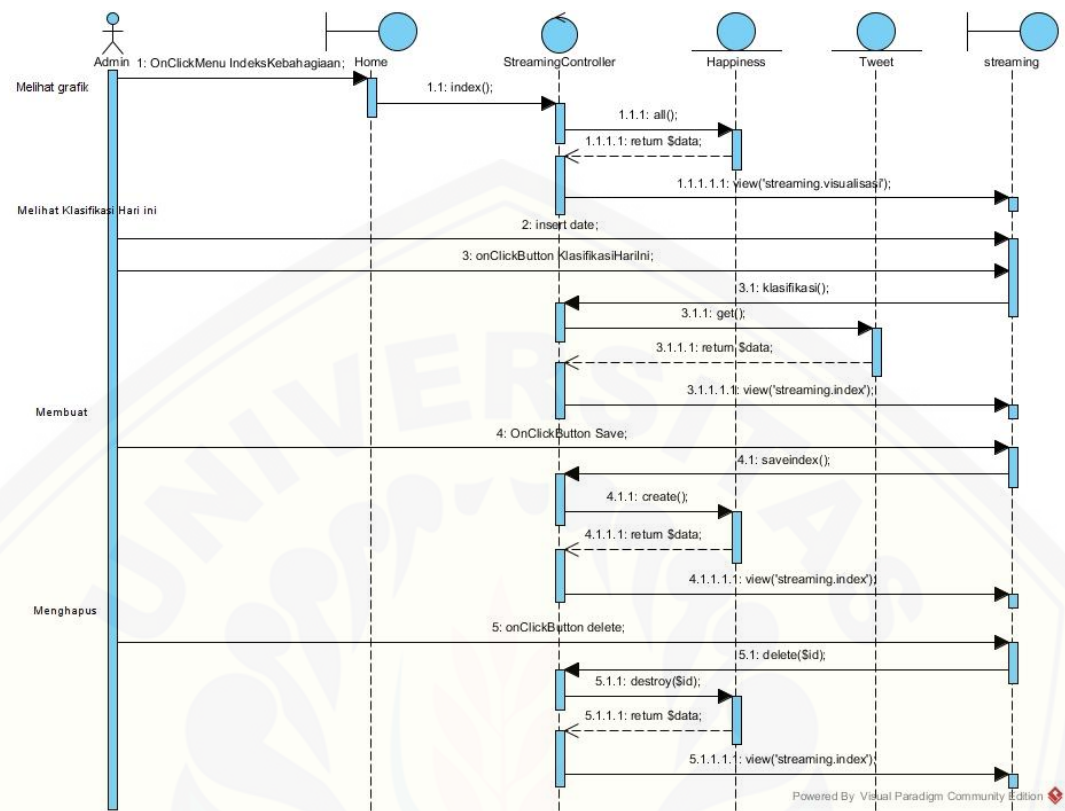
B.10. Sequence Diagram Mengelola Ontologi



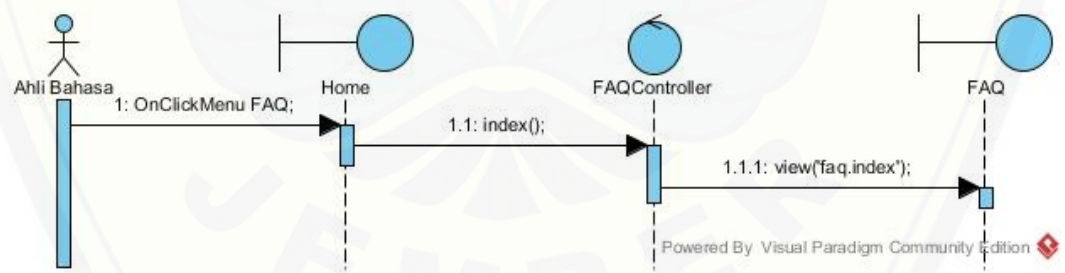
B.11. Sequence Diagram Melihat Indeks Kebahagiaan Data Testing



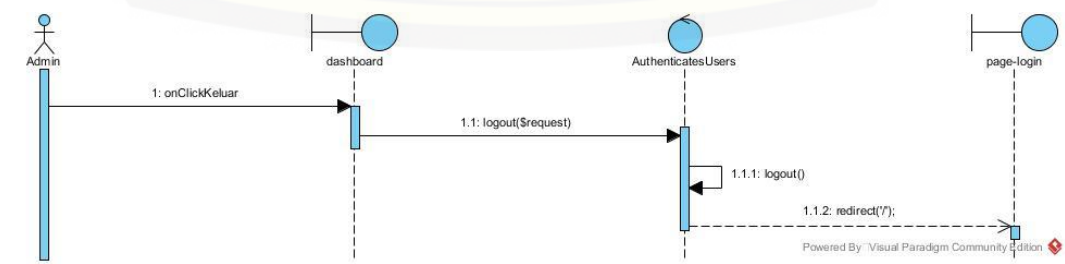
B.12. Sequence Diagram Mengelola Data Indeks Kebahagiaan Harian



B.13. Sequence Diagram Melihat FAQ

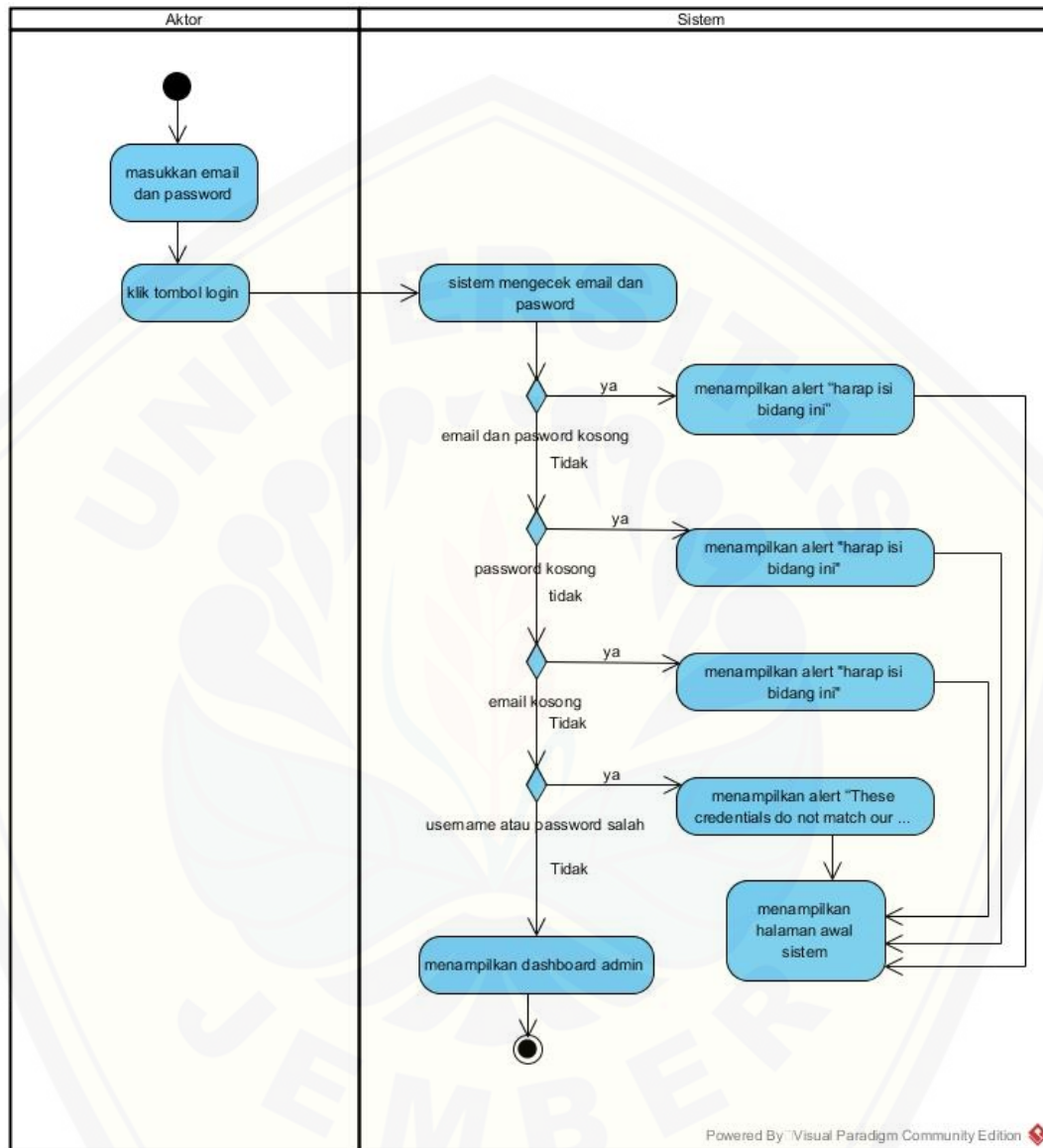


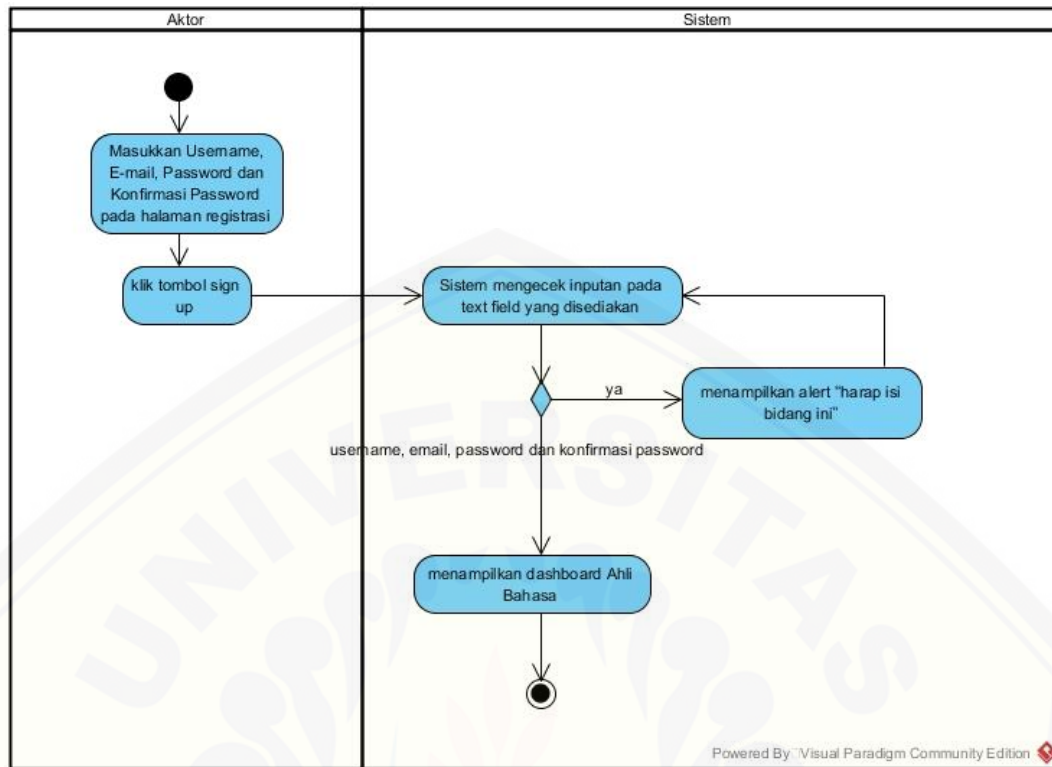
B.14. Sequence Diagram Logout

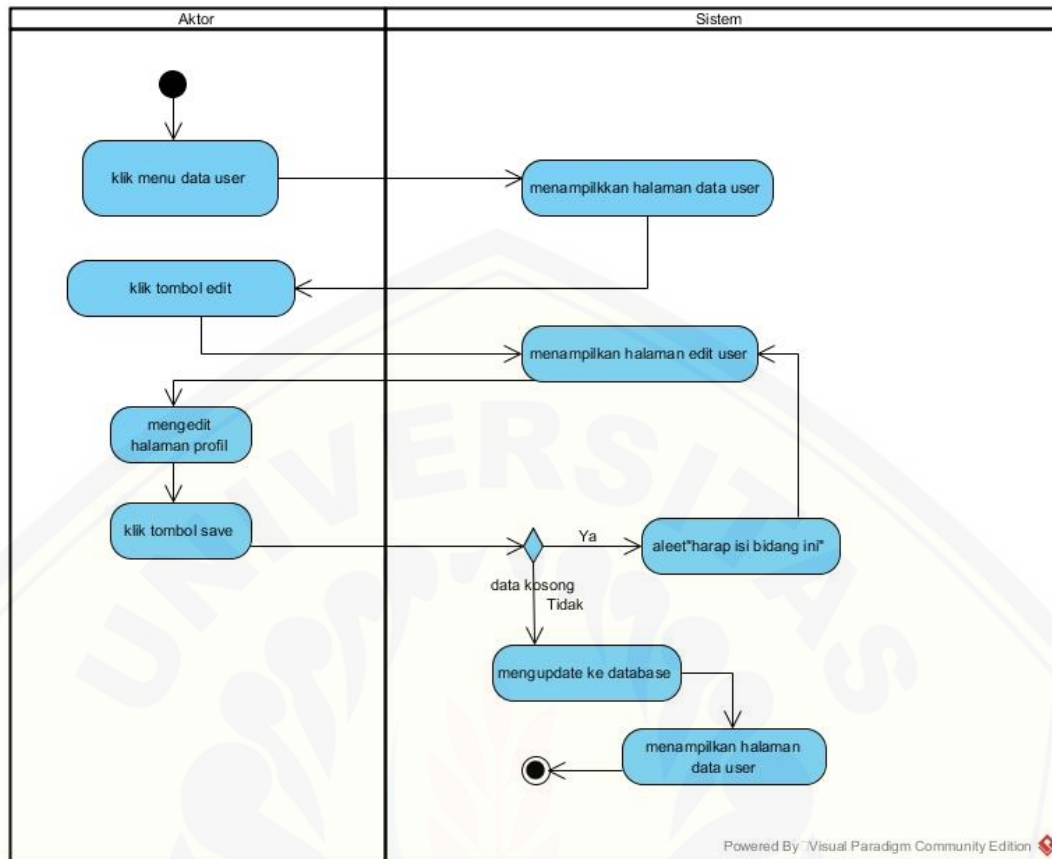


LAMPIRAN C. ACTIVITY DIAGRAM

C.1. Activity Diagram Login

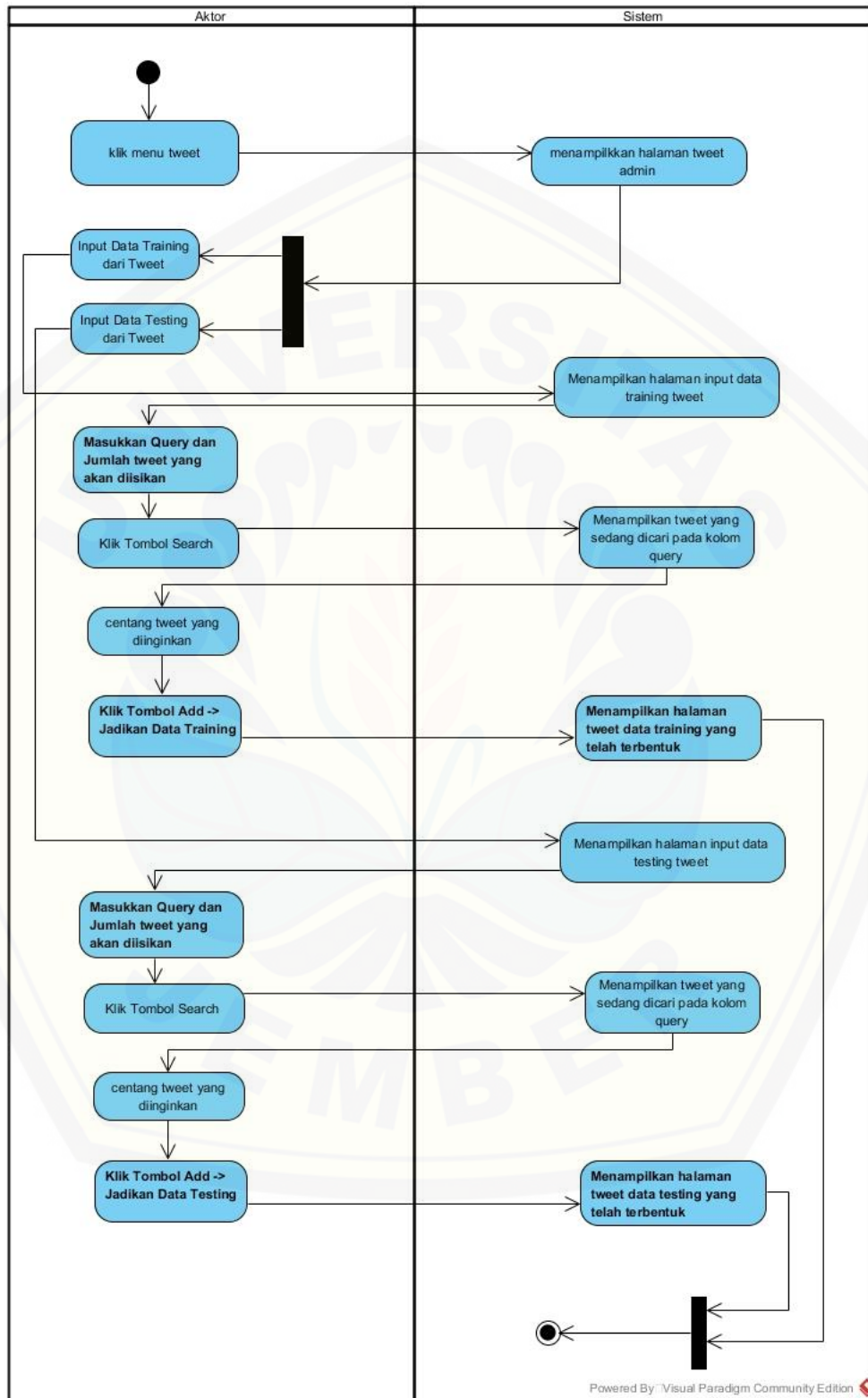




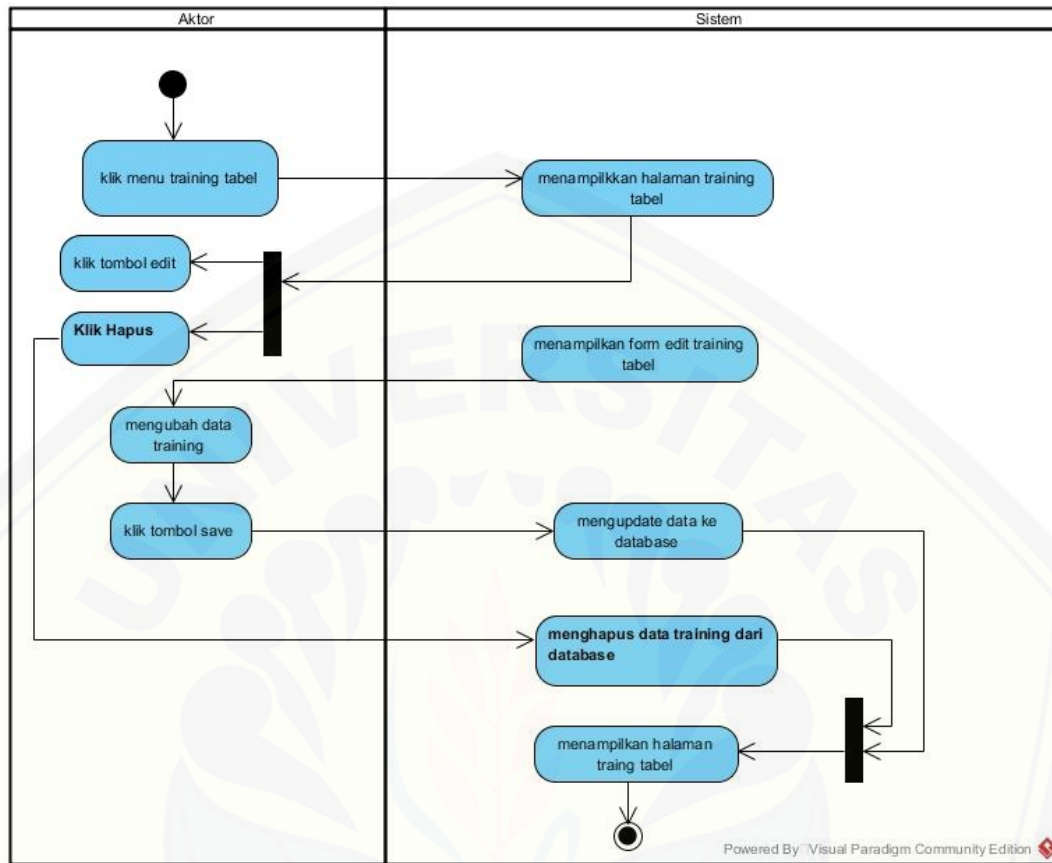


C

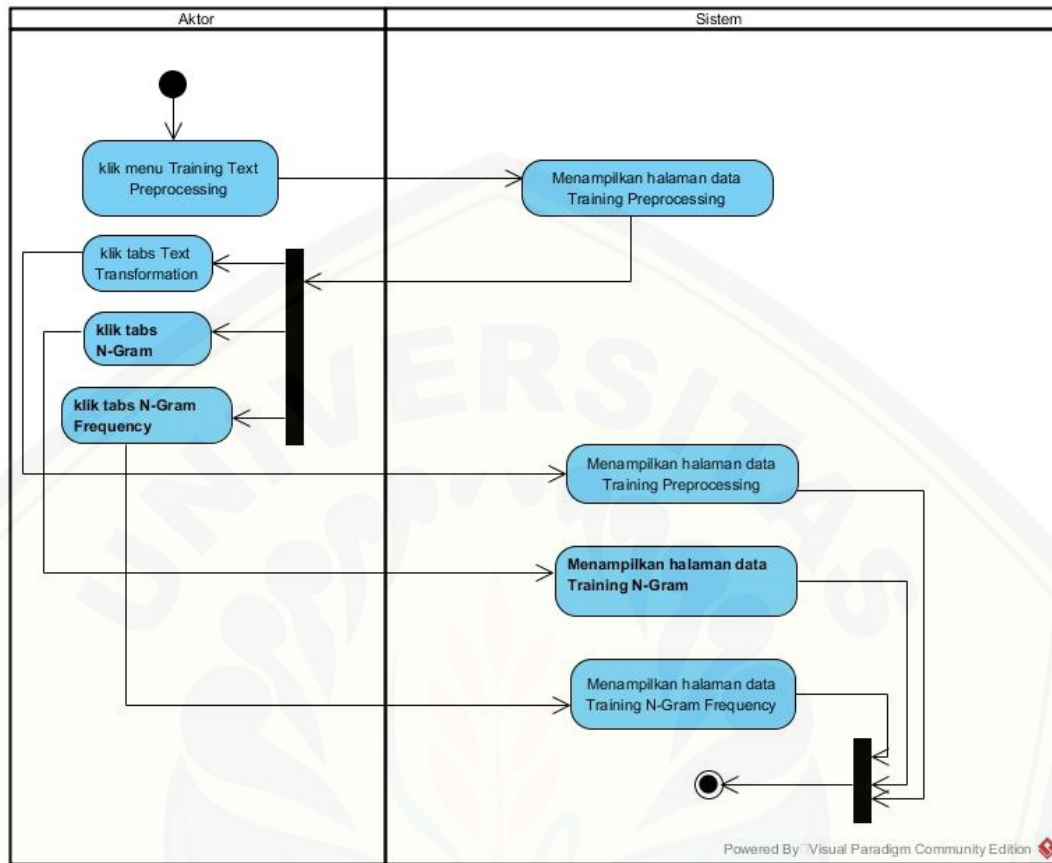
4. Activity Diagram Mengelola Tweet



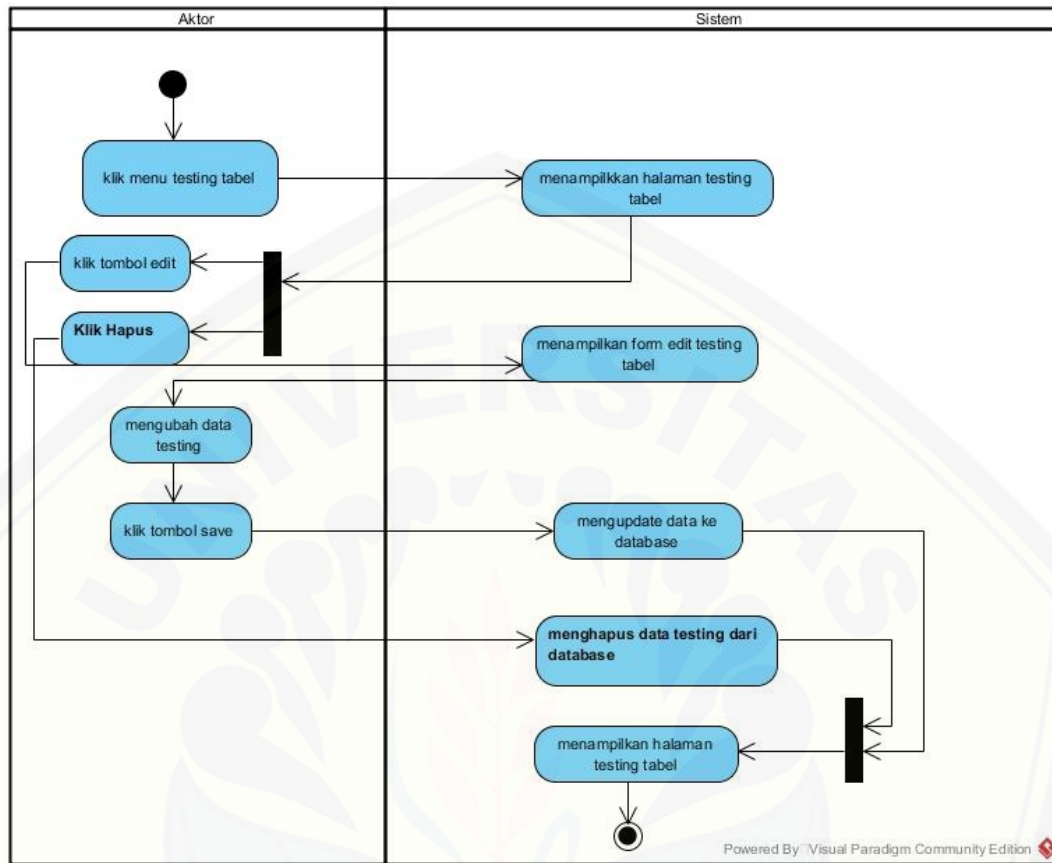
C.5. Activity Diagram Mengelola Data Training



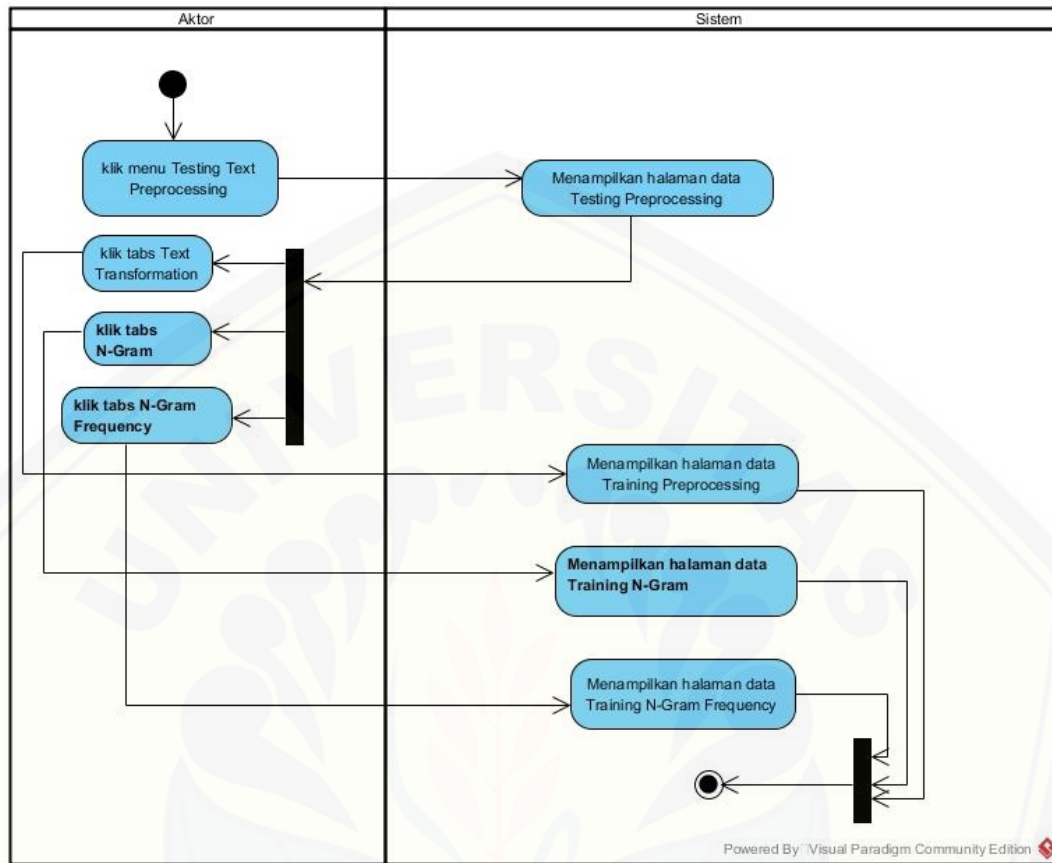
C.6. Activity Diagram Melihat Text Mining Data Training



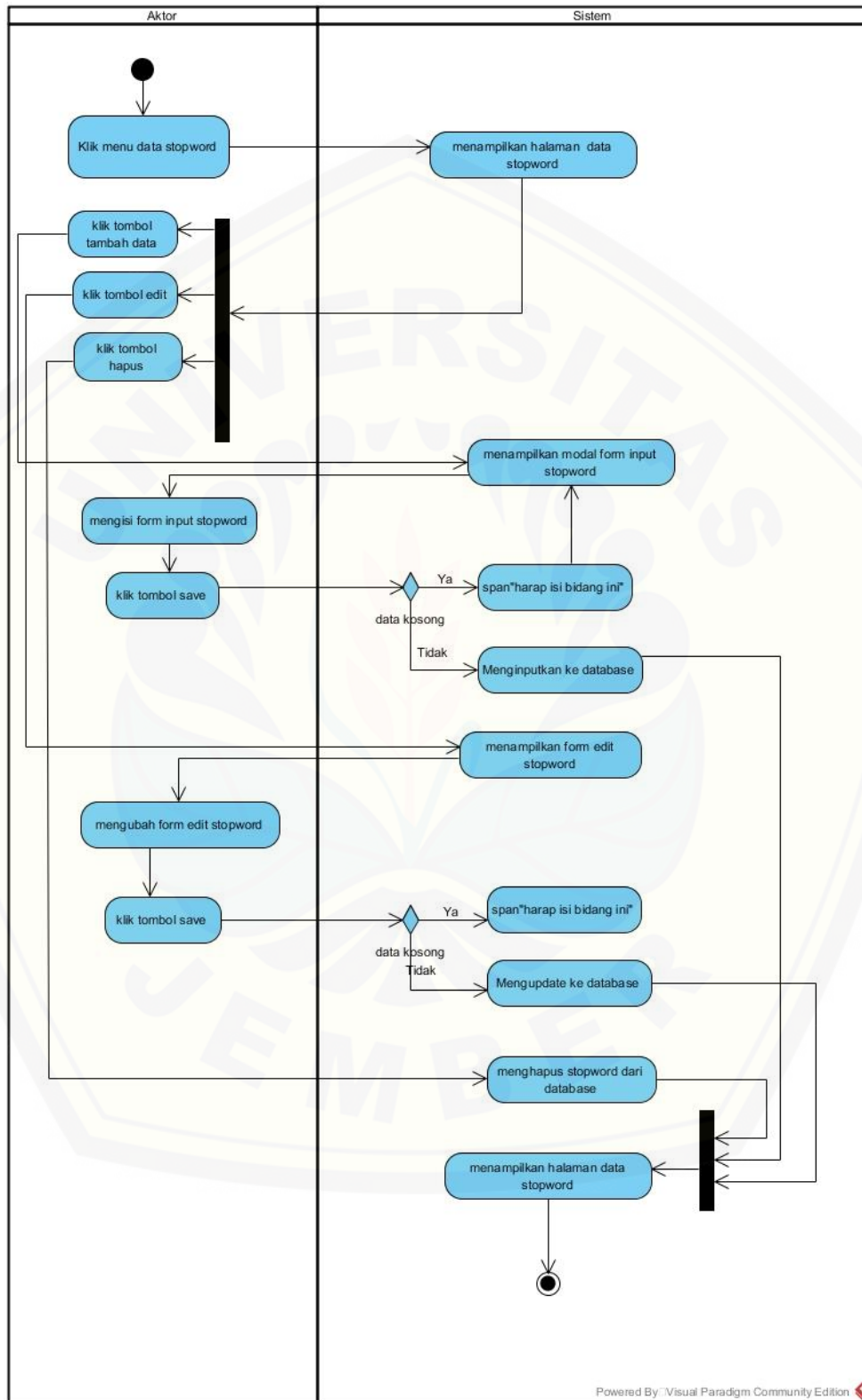
C.7. Activity Diagram Mengelola Data Testing



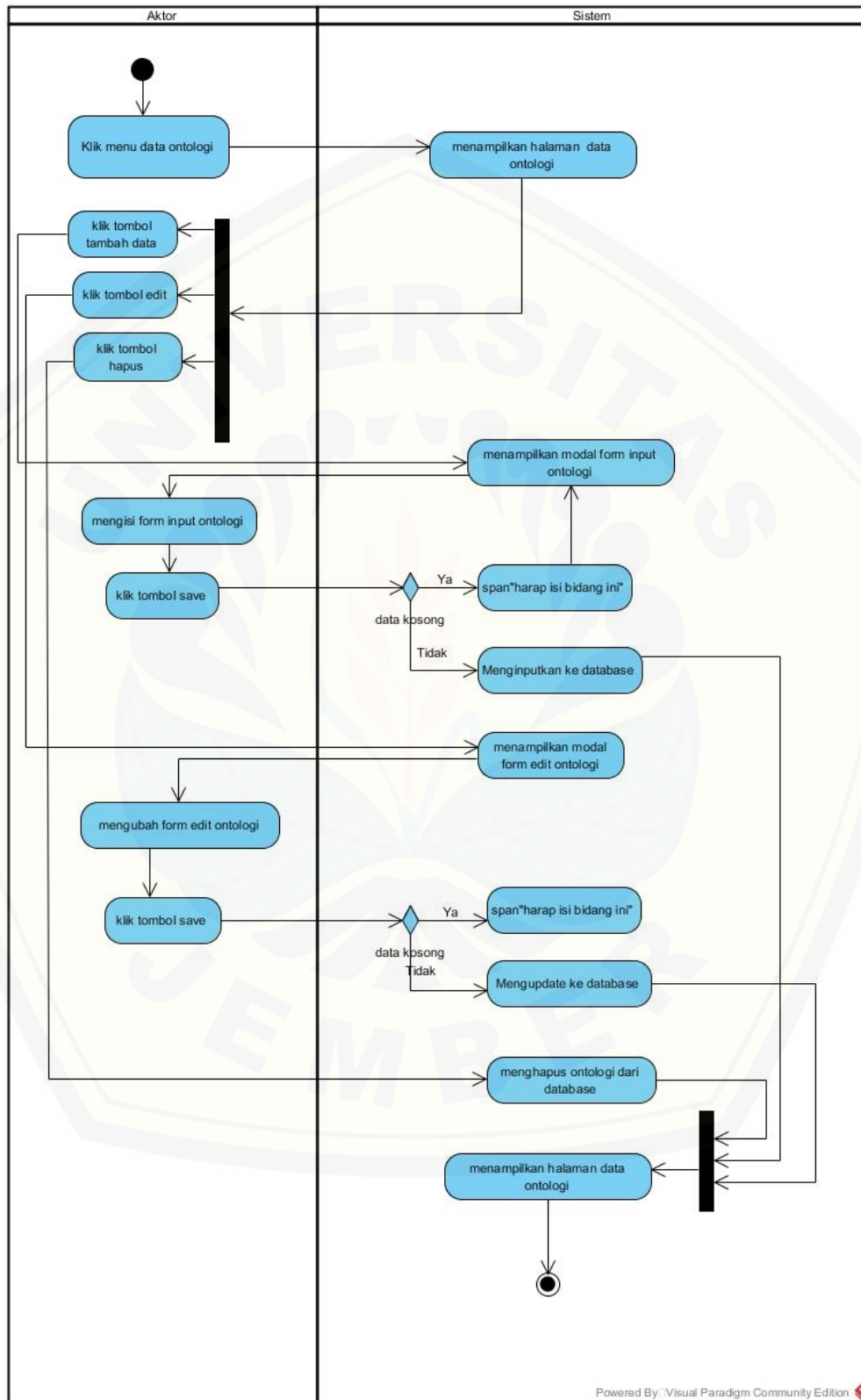
C.8. Activity Diagram Melihat Text Mining data Testing



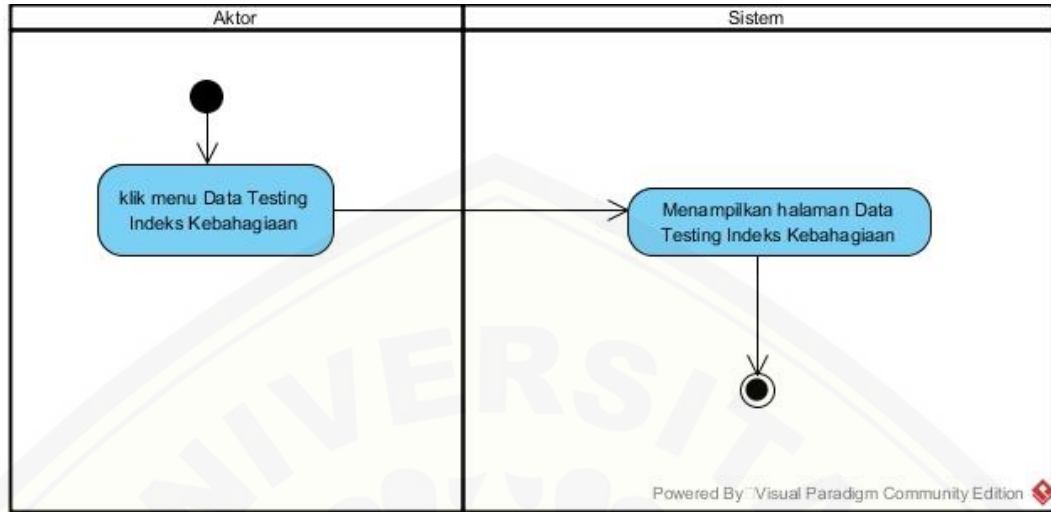
C.9. Activity Diagram Mengelola Stopword



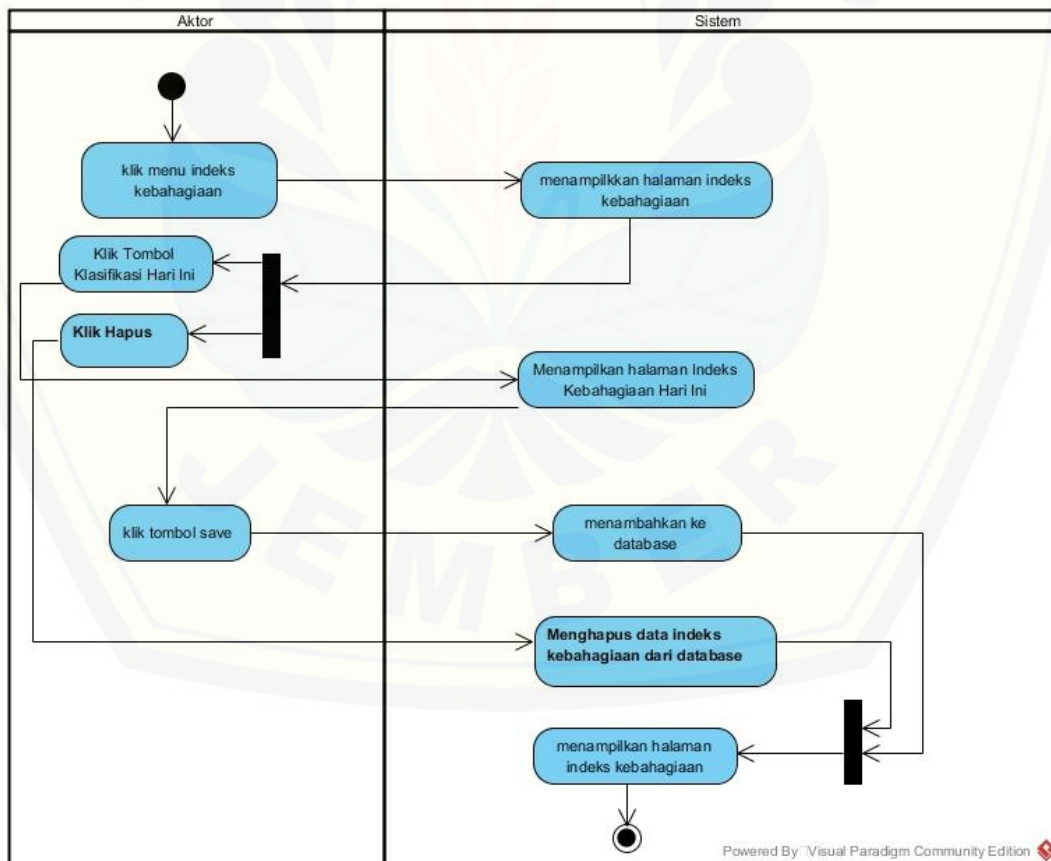
C.10. Activity Diagram Mengelola Ontologi



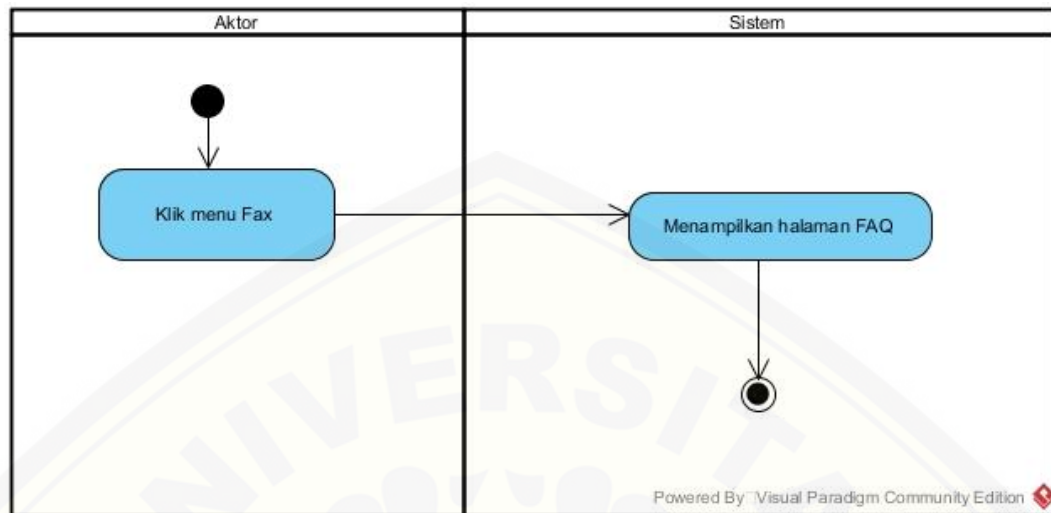
C.11. Activity Diagram Melihat Indeks Kebahagiaan Data Testing



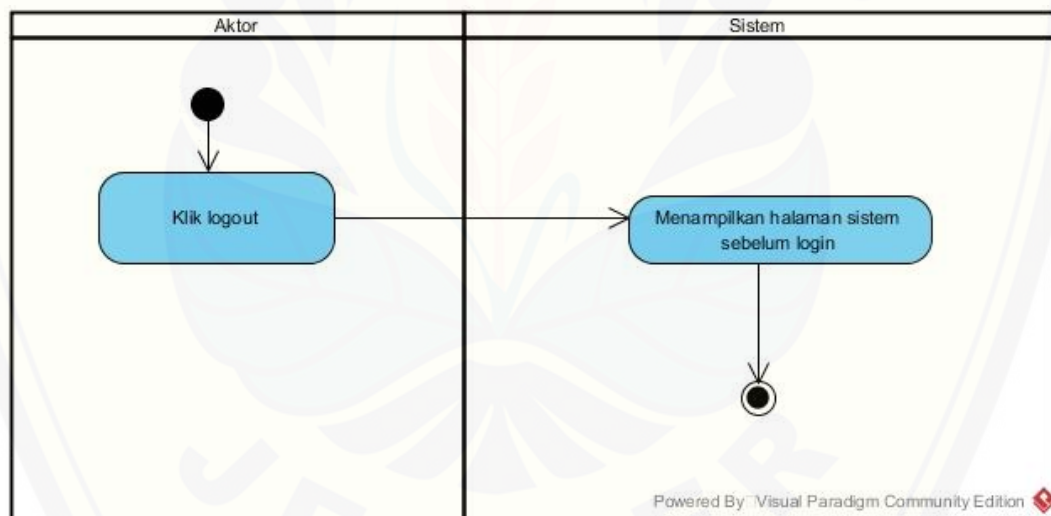
C.12. Activity Diagram Mengelola Data Indeks Kebahagiaan Harian



C.13. Activity Diagram Melihat FAQ



C.14. Activity Diagram Logout



LAMPIRAN D. KODE PROGRAM

D.1. Kode Program *Login*

```
public function login(Request $request)
{
    $this->validateLogin($request);

    // If the class is using the ThrottlesLogins trait, we can
    automatically throttle
    // the login attempts for this application. We'll key this by
    the username and
    // the IP address of the client making these requests into
    this application.
    if ($this->hasTooManyLoginAttempts($request)) {
        $this->fireLockoutEvent($request);

        return $this->sendLockoutResponse($request);
    }

    if ($this->attemptLogin($request)) {
        return $this->sendLoginResponse($request);
    }

    // If the login attempt was unsuccessful we will increment the
    number of attempts
    // to login and redirect the user back to the login form. Of
    course, when this
    // user surpasses their maximum number of attempts they will
    get locked out.
    $this->incrementLoginAttempts($request);

    return $this->sendFailedLoginResponse($request);
}

protected function validateLogin(Request $request)
{
    $this->validate($request, [
        $this->username() => 'required', 'password' => 'required',
    ]);
}
```

```
}

/**
 * Attempt to log the user into the application.
 *
 * @param \Illuminate\Http\Request $request
 * @return bool
 */
protected function attemptLogin(Request $request)
{
    return $this->guard()->attempt(
        $this->credentials($request), $request->has('remember')
    );
}

protected function credentials(Request $request)
{
    return $request->only($this->username(), 'password');
}

protected function sendLoginResponse(Request $request)
{
    $request->session()->regenerate();

    $this->clearLoginAttempts($request);

    return $this->authenticated($request, $this->guard()->user())
        ?: redirect()->intended($this->redirectPath());
}

protected function sendFailedLoginResponse(Request $request)
{
    $errors = [$this->username() => trans('auth.failed')];

    if ($request->expectsJson()) {
        return response()->json($errors, 422);
    }

    return redirect()->back()
        ->withInput($request->only($this->username(), 'remember'))
        ->withErrors($errors);
}

/**
 * Get the login username to be used by the controller.
 *
 * @return string
 */
public function username()
{
    return 'email';
}
```

D.2. Kode Program Registrasi Akun (*RegisterController.php*)

```
<?php

namespace App\Http\Controllers\Auth;

use App\User;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
    /**
     |-----
     | Register Controller
     |-----
     |
     | This controller handles the registration of new users as
     | well as their
     | validation and creation. By default this controller uses a
     | trait to
     | provide this functionality without requiring any additional
     | code.
     |
     */

    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => 'required|string|max:255',
        ]);
    }
}
```

```

        'email' =>
'required|string|email|max:255|unique:users',
        'password' => 'required|string|min:6|confirmed',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
        'role_id'=>2
    ]);
}

public function showRegistrationForm()
{
    if (view()->exists('auth.authenticate')) {
        return view('auth.authenticate');
    }
    return view('auth.register2');
}
}
}

```

D.3. Kode Program Mengelola *User* (*UserController.php*)

```

<?php

namespace App\Http\Controllers;

use App\Role;
use App\User;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    {
        $user=User::all();
        return view('user.index', ['user'=>$user]);
    }
    function delete($id){
        User::destroy($id);
        return Redirect::to('/datatraining/tabel');
    }
    public function edit(Request $request){

```

```

        User::where('id', $request->id)->edit([
            'name'=>$request->username,
            'email'=>$request->email,
            'role_id'=>$request->role,
        ]);

        return Redirect::to('/user');
    }
    public function edit($id) {
        $user=User::find($id);
        $role=Role::all();
        return view('user.edit', ['user'=>$user, 'role'=>$role]);
    }
    public function insert(Request $request)
    {
        $this->validate($request, [
            'ontology'=>'required'
        ]);
        User::create([
            'text'=>$request->ontology,
            'parameter'=>$request->parameter
        ]);
        return Redirect::to('/ontology');
    }
}

```

D.4. Kode Program Mengelola *Tweet* (*TweetController.php*)

```

<?php

namespace App\Http\Controllers;
define('CONSUMER_KEY', '0Y9KZFaZaUI67BQ5RjBcPPhSN');
define('CONSUMER_SECRET',
'ey1LZH8xYegJ0h9EIB0m2nsD18shVcJHZtxlNfcHKdzTR5tbyK');
define('ACCESS_TOKEN', '3014251646-
J0LadQ2pyEAs0xLwnhwxu5nv27gZL8of0NAPzGB');
define('ACCESS_TOKEN_SECRET',
'R06seEsYZdaPy2LkpHz54JLhLxBx0snXg2AdpPuaBipdY');

use Abraham\TwitterOAuth\TwitterOAuth;
use App\Classification;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Input;
use Illuminate\Support\Facades\Redirect;

class TweetController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
}

```



```

public function index()
{
    $tweet=$this->search();
    return view('crawling.tweet', ['tweet'=>$tweet]);
}

function search()
{
    $limit =
(isset($_GET['jumlahtweet']))?$_GET['jumlahtweet']:100;
    $max_id = null;
    $count=100;
    $contents = array();
    $connection = new TwitterOAuth(CONSUMER_KEY,
CONSUMER SECRET, ACCESS TOKEN, ACCESS TOKEN SECRET);
    for ($i = 0; $i < $limit; $i += $count) {

        $content = $connection->get('search/tweets', array(
            "q" =>
(isset($_GET['query']))?$_GET['query']:"jember",
            "count"=>$limit,
            'max_id'=>$max_id,
            'lang'=>'id'
        ));
        $contents[] = $content;
        if(count($content->statuses)>0)
            $max_id=($content->statuses[count($content->statuses)-1]->id_str);
    }
    return $contents;
}

public function adddata() {
    $tweet = Input::get('tweet');
    $jenis=Input::get('jenis');
    if (is_array($tweet)){
        foreach ($tweet as $tw){
            $tweetarray=explode(";", $tw);
            $datatraining=Tweet::create([
                'user'=>$tweetarray[1],
                'tweet'=>$tweetarray[0],
                'post_time'=>"$tweetarray[2]",
                'status'=>$jenis
            ]);

            $lexicon=Classification::create([
                'id_tweet'=>$datatraining->id,
                'lexicon_pos_score'=>0,
                'lexicon_neg_score'=>0,
                'manual_sentimen_label'=>null,
                'manual_category_label'=>null,
                'nb_pos_probability'=>0,
                'nb_neg_probability'=>0
            ]);
        }
    }
}

```

```

        return Redirect::to('/tweet');
    }
}

```

D.5. Kode Program Mengelola Data *Training* (*DataTrainingController.php*)

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Classification;
use App\Tweet;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
class DataTrainingController extends TextMiningController
{
    public function index()
    {
        $datatraining=Tweet::where('status',1)->get();
        $distribusisentimen=DB::table('classifications')->
>join('tweets', 'id_tweet', '=', 'tweets.id')
        ->select('manual_sentimen_label',DB::raw('count(*) as
total'))
        ->where('status',1)
        ->groupBy('manual_sentimen_label')->get();
        $distribusicategory=DB::table('classifications')->
>join('tweets', 'id_tweet', '=', 'tweets.id')
        ->select('manual_category_label',DB::raw('count(*) as
total'))
        ->where('status',1)
        ->groupBy('manual_category_label')->get();
        return
view('datatraining.index', ['datatraining'=>$datatraining, 'distribu
sisentimen'=>$distribusisentimen, 'distribusicategory'=>$distribusi
category]);
    }

    public function edit(Request $request){
        Classification::where('id_tweet', $request->id)->edit([
            'manual_sentimen_label'=>$request->
>manual_sentimen_label,
            'manual_category_label'=>$request->
>manual_category_label
        ]);
        return Redirect::to('/datatraining/tabel');
    }

    public function edit($id){
        $datatraining=Tweet::find($id);
        return

```

```

view('datatraining.edit', ['datatraining'=>$datatraining]);
}
function delete($id){
    Tweet::destroy($id);
    return Redirect::to('/datatraining/tabel');
}
public function manual(){
    $datatraining=Tweet::where('status',1)->get();
    return
view('datatraining.manual', ['datatraining'=>$datatraining]);
}
public function labelling(Request $request){

    Classification::find($request->id)->edit([
        'manual_sentimen_label'=>$request->sentimen,
        'manual_category_label'=>$request->kategori
    ]);
    return Redirect::to('/datatraining/manual');
}

```

D.6. Kode Program Melihat *Text Mining* Data Training

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Classification;
use App\Tweet;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
class DataTrainingController extends TextMiningController
{
public function textmining()
{
    $datatraining=Tweet::where('status',1)->get();
    $hasilpreprocessing=$this->preprocessing($datatraining);
    $hasilstemming=$this->stemming($datatraining);
    $hasilbukancorpus=$this->cekCorpus($hasilstemming);
    $hasilstopwordremoval=$this->
>stopwordremoval($hasilbukancorpus);
    $hasilngram=$this->ngram($hasilstopwordremoval);

    $b=array();
    foreach ($hasilngram as $hn){
        $b=$b+$hn;
    }
    $hasilfrequencyngram=$this->frequencyngram($b);

    return
view('datatraining.textpreprocessing', ['datatraining'=>$datatraining]);
}

```

```

ng, 'hasilpreprocessing'=>$hasilpreprocessing,

'hasilstemming'=>$hasilstemming, 'hasilstopwordremoval'=>$hasilstop
wordremoval, 'hasilngram'=>$hasilngram,

'hasilbukancorpus'=>$hasilbukancorpus, 'hasilfrequencyngram'=>$hasi
lfrequencyngram]);
}
)

```

```

<?php

namespace App\Http\Controllers;

use App\Classification;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;
use Sastrawi\Dictionary\ArrayDictionary;
use Phpml\Classification\NaiveBayes;

class TextMiningController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function frequencyngram($ngram) {
        $y = new \Ngram\Tool\Ngram\Frequency();
        $z=$y->get($ngram);
        $output=array();
        $i=0;
        for ($i=0;$i<count($z);$i++) {
            $output[$i][0]=key($z);
            $output[$i][1]=$z[key($z)];

            next($z);
        }
        return $output;
    }
    public function ngram($datatraining, $n=3) {
        $output=array();
        $i=0;

        foreach ($datatraining as $dt) {
            $w = new \Ngram\Frequency\Letter($dt);
            $j=0;
            foreach ($w->extract($n) as $ye) {
                $output[$i][$j]=$ye;
                $j++;
            }
            $i++;
        }
        return $output;
    }
}

```

```
}  
public function stemming($datatraining) {  
    $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();  
    $stemmer = $stemmerFactory->createStemmer();  
    $output=array();  
    // stem  
    $i=0;  
    foreach ($datatraining as $dt) {  
        $kw[$i]=$dt->tweet;  
        $i++;  
    }  
    $i=0;  
    foreach ($kw as $k) {  
        $output[$i]=$stemmer->stem($k);  
        $i++;  
    }  
    return($output);  
}  
public function lexiconing($datatraining) {  
    $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();  
    $stemmer = $stemmerFactory->createStemmer();  
    $output=array();  
    // stem  
    $i=0;  
    foreach ($datatraining as $dt) {  
        $kw[$i]=$dt->tweet;  
        $i++;  
    }  
    $i=0;  
    foreach ($kw as $k) {  
        $output[$i]=$stemmer->stem($k);  
        $i++;  
    }  
    return($output);  
}  
public function stopwordremoval($datatraining) {  
    $stopwordFactory = new  
\Sastrawi\StopwordRemover\StopwordRemoverFactory();  
    $stopword=$stopwordFactory->createStopwordRemover();  
    $output=array();  
    $i=0;  
    foreach ($datatraining as $dt) {  
        $output[$i]=$stopword->remove($dt);  
        if($output[$i]=='') {  
            $output[$i]=' ' ;  
        }  
        $i++;  
    }  
    return $output;  
}  
  
public function preprocessing($datatraining) {  
    $normalizer=new \Sastrawi\Stemmer\FILTER\TextNormalizer();
```

```

        $output=array();
        $i=0;
        foreach ($datatraining as $dt){
            $kw[$i]=$dt->tweet;
            $i++;
        }
        $i=0;
        foreach ($kw as $k){
            $output[$i]=$normalizer->normalizeText($k);
            $i++;
        }
        return $output;
    }
    public function getKataDasar()
    {
        $dictionaryFile=public_path().'/kata-dasar.txt';
        return explode("\n", file_get_contents($dictionaryFile));
    }
    public function removeBukanKataDasar($text)
    {
        $kataDasar = $this->getKataDasar();

        $dictionary = new ArrayDictionary($kataDasar);

        $words = explode(' ', $text);

        foreach ($words as $i => $word) {
            if (!$dictionary->contains($word)) {
                unset($words[$i]);
            }
        }

        return implode(' ', $words);
    }
    public function cekCorpus($datatraining){
        $output=array();
        $i=0;
        foreach ($datatraining as $dt){
            $output[$i]=$this->removeBukanKataDasar($dt);
            $i++;
        }
        return $output;
    }
}
}

```

D.7. Kode Program Mengelola Data *Testing*

```

<?php

namespace App\Http\Controllers;

use App\Classification;

```

```

use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
use Sastrawi\Dictionary\ArrayDictionary;

class DataTestingController extends TextMiningController
{
    public function index()
    {
        $datatesting=Tweet::where('status',2)->get();
        return
view('datatesting.index', ['datatesting'=>$datatesting]);
    }
    function delete($id){
        Tweet::destroy($id);
        return Redirect::to('/datatraining/tabel');
    }
    public function edit(Request $request){
        Classification::where('id_tweet', $request->id)->edit([
            'manual_sentimen_label'=>$request-
>manual_sentimen_label,
            'manual_category_label'=>$request-
>manual_category_label
        ]);
        return Redirect::to('/datatesting/tabel');
    }
    public function edit($id){
        $datatesting=Tweet::find($id);
        return
view('datatesting.edit', ['datatesting'=>$datatesting]);
    }

    public function labelling(Request $request){

        Classification::find($request->id)->edit([
            'manual_sentimen_label'=>$request->sentimen,
            'manual_category_label'=>$request->kategori
        ]);
        return Redirect::to('/datatesting/manual');
    }
}

```

D.8. Kode Program Melihat *Text Mining* Data Testing

```

<?php

namespace App\Http\Controllers;

use App\Classification;
use App\Tweet;

```

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
use Sastrawi\Dictionary\ArrayDictionary;

class DataTestingController extends TextMiningController
{
    public function textmining()
    {
        $datatraining=Tweet::where('status',2)->get();
        $hasilpreprocessing=$this->preprocessing($datatraining);
        $hasilstemming=$this->stemming($datatraining);
        $hasilbukancorpus=$this->cekCorpus($hasilstemming);
        $hasilstopwordremoval=$this->stopwordremoval($hasilbukancorpus);
        $hasilngram=$this->ngram($hasilstopwordremoval);
        $b=array();
        foreach ($hasilngram as $hn){
            $b=$b+$hn;
        }
        $hasilfrequencygram=$this->frequencygram($b);

        return
        view('datatesting.textpreprocessing', ['datatraining'=>$datatraining,
        'hasilpreprocessing'=>$hasilpreprocessing,
        'hasilstemming'=>$hasilstemming, 'hasilstopwordremoval'=>$hasilstopwordremoval, 'hasilngram'=>$hasilngram,
        'hasilbukancorpus'=>$hasilbukancorpus, 'hasilfrequencygram'=>$hasilfrequencygram]);
    }
}

```

```

<?php

namespace App\Http\Controllers;

use App\Classification;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;
use Sastrawi\Dictionary\ArrayDictionary;
use Phpml\Classification\NaiveBayes;

class TextMiningController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
}

```



```
public function frequencyngram($ngram) {
    $y = new \Ngram\Tool\Ngram\Frequency();
    $z=$y->get($ngram);
    $output=array();
    $i=0;
    for ($i=0;$i<count($z);$i++) {
        $output[$i][0]=key($z);
        $output[$i][1]=$z[key($z)];

        next($z);
    }
    return $output;
}
public function ngram($datatraining,$n=3) {
    $output=array();
    $i=0;

    foreach ($datatraining as $dt) {
        $w = new \Ngram\Frequency\Letter($dt);
        $j=0;
        foreach ($w->extract($n) as $ye) {
            $output[$i][$j]=$ye;
            $j++;
        }
        $i++;
    }
    return $output;
}
public function stemming($datatraining) {
    $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();
    $output=array();
    // stem
    $i=0;
    foreach ($datatraining as $dt) {
        $kw[$i]=$dt->tweet;
        $i++;
    }
    $i=0;
    foreach ($kw as $k) {
        $output[$i]=$stemmer->stem($k);
        $i++;
    }
    return($output);
}
public function lexiconing($datatraining) {
    $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();
    $output=array();
    // stem
    $i=0;
    foreach ($datatraining as $dt) {
        $kw[$i]=$dt->tweet;
        $i++;
    }
    $i=0;
```

```

        foreach ($kw as $k) {
            $output[$i]=$stemmer->stem($k);
            $i++;
        }
        return($output);
    }
    public function stopwordsremoval($datatraining) {
        $stopwordFactory = new
\Sastrawi\StopwordRemover\StopwordRemoverFactory();
        $stopword=$stopwordFactory->createStopwordRemover();
        $output=array();
        $i=0;
        foreach ($datatraining as $dt) {
            $output[$i]=$stopword->remove($dt);
            if($output[$i]=='') {
                $output[$i]=' ';
            }

            $i++;
        }
        return $output;
    }

    public function preprocessing($datatraining) {
        $normalizer=new \Sastrawi\Stemmer\Filter\TextNormalizer();
        $output=array();
        $i=0;
        foreach ($datatraining as $dt) {
            $kw[$i]=$dt->tweet;
            $i++;
        }
        $i=0;
        foreach ($kw as $k) {
            $output[$i]=$normalizer->normalizeText($k);
            $i++;
        }
        return $output;
    }
    public function getKataDasar()
    {
        $dictionaryFile=public_path().'/kata-dasar.txt';
        return explode("\n", file_get_contents($dictionaryFile));
    }
    public function removeBukanKataDasar($text)
    {
        $kataDasar = $this->getKataDasar();

        $dictionary = new ArrayDictionary($kataDasar);

        $words = explode(' ', $text);

        foreach ($words as $i => $word) {
            if (!$dictionary->contains($word)) {

```

```

        unset($words[$i]);
    }
}

return implode(' ', $words);
}
public function cekCorpus($datatraining) {
    $output=array();
    $i=0;
    foreach ($datatraining as $dt) {
        $output[$i]=$this->removeBukanKataDasar($dt);
        $i++;
    }
    return $output;
}
}
}

```

D.9. Kode Program Klasifikasi Dan Uji Performansi

```

<?php

namespace App\Http\Controllers;

use App\Classification;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
use Sastrawi\Dictionary\ArrayDictionary;

class DataTestingController extends TextMiningController
{
    public function klasifikasi(Request $request) {
        $datatraining=Tweet::where('status',1)->get();
        $datatesting=Tweet::where('status',2)->get();
        $manual=array();
        $j=0;
        foreach($datatraining as $dt) {
            $manualsentimen[$j]=$dt->classification-
>manual_sentimen_label;
            $manualkategori[$j]=$dt->classification-
>manual_category_label;
            $j++;
        }
        $l=0;
        foreach ($datatesting as $dtesting) {
            $tweetidtesting[$l]=$dtesting->id;
            $manualsentimentesting[$l]=$dtesting-

```

```

>classification->manual_sentimen_label;
    $manualkategoritesting[$l]=$dtesting-
>classification->manual_category_label;
    $l++;
}

$hasilstemmingtraining=$this-
>stemming($datatraining);
$hasilbukancorpustraining=$this-
>cekCorpus($hasilstemmingtraining);
$hasilstopwordremovaltraining=$this-
>stopwordremoval($hasilbukancorpustraining);
$hasilngramtraining=$this-
>ngram($hasilstopwordremovaltraining);
$b=array();
$i=0;
$k=0;
$trainlabels="";
$classifiersentimen=new NaiveBayes();
$classifierkategori=new NaiveBayes();
foreach ($hasilngramtraining as $hnt){
    foreach ($hnt as $ss){
//        $train[$i]=$ss;
        $classifiersentimen-
>train([[ $ss ]], [$manualsentimen[$k]]); //menambah training
        $classifierkategori-
>train([[ $ss ]], [$manualkategori[$k]]); //menambah training
//        $trainlabels=$trainlabels.$manual[$k].",
";
        $i++;
    }
    $k++;
}

$databtesting=Tweet::where('status',2)->get();
$hasilstemmingtesting=$this->stemming($databtesting);
$hasilbukancorpustraining=$this-
>cekCorpus($hasilstemmingtesting);
$hasilstopwordremovaltesting=$this-
>stopwordremoval($hasilbukancorpustraining);

$hasilngramtesting=$this-
>ngram($hasilstopwordremovaltesting);

$i=0;
//    $positif=0;
//    $negatif=0;
foreach ($hasilngramtesting as $hnt){

    $klasifikasisentimen[$i]['positif']=0;
    $klasifikasisentimen[$i]['negatif']=0;

```

```

$klasifikasimentimen[$i]['netral']=0;
$klasifikasimentimen['hasil'][$i]='';
$klasifikasikategori[$i]['pekerjaan']=0;
$klasifikasikategori[$i]['pendapatan rumah
tangga']=0;
$klasifikasikategori[$i]['kondisi rumah dan
aset']=0;
$klasifikasikategori[$i]['pendidikan']=0;
$klasifikasikategori[$i]['kesehatan']=0;
$klasifikasikategori[$i]['keharmonisan
keluarga']=0;
$klasifikasikategori[$i]['hubungan sosial']=0;
$klasifikasikategori[$i]['ketersediaan waktu
luang']=0;
$klasifikasikategori[$i]['kondisi
lingkungan']=0;
$klasifikasikategori[$i]['kondisi keamanan']=0;
$klasifikasikategori[$i]['tidak terkategori']=0;
foreach ($hnt as $ngram) {
    $predictedsentimen = $classifiersentimen-
>predict([$ngram]);
    $predictedkategori = $classifierkategori-
>predict([$ngram]);

(isset($predictedsentimen['positif']))?$klasifikasimentimen[
$i]['positif']+=$predictedsentimen['positif']:$klasifikasise
ntimen[$i]['positif']=0;

(isset($predictedsentimen['negatif']))?$klasifikasimentimen[
$i]['negatif']+=$predictedsentimen['negatif']:$klasifikasise
ntimen[$i]['negatif']=0;

(isset($predictedsentimen['netral']))?$klasifikasimentimen[
$i]['netral']+=$predictedsentimen['netral']:$klasifikasimenti
men[$i]['netral']=0;

(isset($predictedkategori['pekerjaan']))?$klasifikasikategor
i[$i]['pekerjaan']+=$predictedkategori['pekerjaan']:$klasifi
kasikategori[$i]['pekerjaan']=0;
(isset($predictedkategori['pendapatan rumah
tangga']))?$klasifikasikategori[$i]['pendapatan rumah
tangga']+=$predictedkategori['pendapatan rumah
tangga']:$klasifikasikategori[$i]['pendapatan rumah
tangga']=0;
(isset($predictedkategori['kondisi rumah dan
aset']))?$klasifikasikategori[$i]['kondisi rumah dan
aset']+=$predictedkategori['kondisi rumah dan
aset']:$klasifikasikategori[$i]['kondisi rumah dan aset']=0;

(isset($predictedkategori['pendidikan']))?$klasifikasikatego

```

```

ri[$i]['pendidikan']+=$predictedkategori['pendidikan'];$klasifikasi
ifikasikategori[$i]['pendidikan']=0;

(isset($predictedkategori['kesehatan']))?$klasifikasikategori[$i]['kesehatan']+=$predictedkategori['kesehatan'];$klasifikasi
kasikategori[$i]['kesehatan']=0;
(isset($predictedkategori['keharmonisan
keluarga']))?$klasifikasikategori[$i]['keharmonisan
keluarga']+=$predictedkategori['keharmonisan
keluarga'];$klasifikasikategori[$i]['keharmonisan
keluarga']=0;
(isset($predictedkategori['hubungan
sosial']))?$klasifikasikategori[$i]['hubungan
sosial']+=$predictedkategori['hubungan
sosial'];$klasifikasikategori[$i]['hubungan sosial']=0;
(isset($predictedkategori['ketersediaan
waktu luang']))?$klasifikasikategori[$i]['ketersediaan waktu
luang']+=$predictedkategori['ketersediaan waktu
luang'];$klasifikasikategori[$i]['ketersediaan waktu
luang']=0;
(isset($predictedkategori['kondisi
lingkungan']))?$klasifikasikategori[$i]['kondisi
lingkungan']+=$predictedkategori['kondisi
lingkungan'];$klasifikasikategori[$i]['kondisi
lingkungan']=0;
(isset($predictedkategori['kondisi
keamanan']))?$klasifikasikategori[$i]['kondisi
keamanan']+=$predictedkategori['kondisi
keamanan'];$klasifikasikategori[$i]['kondisi keamanan']=0;
(isset($predictedkategori['tidak
terkategori']))?$klasifikasikategori[$i]['tidak
terkategori']+=$predictedkategori['tidak
terkategori'];$klasifikasikategori[$i]['tidak
terkategori']=0;
}

arsort($klasifikasisentimen[$i], SORT_NUMERIC);

$klasifikasisentimen['hasil'][$i]=key($klasifikasisentimen[$
i]);

arsort($klasifikasikategori[$i], SORT_NUMERIC);

$klasifikasikategori['hasil'][$i]=key($klasifikasikategori[$
i]);

Classification::where('id_tweet',
$tweetidtesting[$i])->edit([

'predict_sentimen_label'=>$klasifikasisentimen['hasil'][$i],
'predict_category_label'=>$klasifikasikategori['hasil'][$i]

```

```

        });
        $i++;
    }

    $reportsentimen = new
ClassificationReport($manualsentimentesting,
$klasifikasisentimen['hasil']);

    $reportresultsentimen['precision']=$reportsentimen-
>getPrecision();
    $reportresultsentimen['recall']=$reportsentimen-
>getRecall();
    $reportresultsentimen['flscore']=$reportsentimen-
>getFlscore();
    $reportresultsentimen['support']=$reportsentimen-
>getSupport();
    $reportresultsentimen['average']=$reportsentimen-
>getAverage();

    $reportkategori = new
ClassificationReport($manualkategoritesting,
$klasifikasikategori['hasil']);

    $reportresultkategori['precision']=$reportkategori-
>getPrecision();
    $reportresultkategori['recall']=$reportkategori-
>getRecall();
    $reportresultkategori['flscore']=$reportkategori-
>getFlscore();
    $reportresultkategori['support']=$reportkategori-
>getSupport();
    $reportresultkategori['average']=$reportkategori-
>getAverage();

    return
view('datatesting.klasifikasi', ['datatesting'=>$datatesting,
'klasifikasisentimen'=>$klasifikasisentimen, 'reportresultsen-
timen'=>$reportresultsentimen, 'klasifikasikategori'=>$klasif-
ikasikategori, 'reportresultkategori'=>$reportresultkategori]
);
}

```

D.10. Kode Program Mengelola *Stopword*

```

<?php

namespace App\Http\Controllers;

use App\Stopword;

```

```

use Illuminate\Support\Facades\Redirect;
use Illuminate\Http\Request;

class StopwordController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index()
    {
        $stopword=Stopword::all();
        return view('stopword.index', ['stopwords'=>$stopword]);
    }
    public function insert(Request $request)
    {
        $this->validate($request, [
            'stopword'=>'required'
        ]);
        Stopword::create([
            'text'=>$request->stopword
        ]);
        return Redirect::to('/stopword');
    }
    public function edit(Request $request){
        Stopword::find($request->id)->edit([
            'text'=>$request->stopword
        ]);
        return Redirect::to('/stopword');
    }
    function delete($id){
        Stopword::destroy($id);
        return Redirect::to('/stopword');
    }
}

```

D.11. Kode Program Mengelola Ontologi (OntologiController.php)

```

<?php

namespace App\Http\Controllers;

use App\Ontology;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

class OntologyController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index()
    {

```



```

        $ontology=Ontology::all();
        return view('ontology.index', ['ontology'=>$ontology]);
    }
    public function insert(Request $request)
    {
        $this->validate($request, [
            'ontology'=>'required'
        ]);
        Ontology::create([
            'text'=>$request->ontology,
            'parameter'=>$request->parameter
        ]);
        return Redirect::to('/ontology');
    }
    public function edit(Request $request){
        Ontology::find($request->id)->edit([
            'text'=>$request->ontology,
            'parameter'=>$request->parameter
        ]);
        return Redirect::to('/ontology');
    }
    function delete($id){
        Ontology::destroy($id);
        return Redirect::to('/ontology');
    }
}
}

```

D.12. Melihat Indeks Kebahagiaan Data *Testing*

```

<?php
namespace App\Http\Controllers;

use App\Classification;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Phpml\Metric\ClassificationReport;

use Phpml\Classification\NaiveBayes;
use Sastrawi\Dictionary\ArrayDictionary;

class DataTestingController extends TextMiningController
{
    public function indekskebahagiaan(){
        $datapositif=DB::table('classifications')-
        >join('tweets', 'tweets.id', '=', 'classifications.id_tweet')
        ->select(DB::raw('count(*) as total, DATE(post_time)
as tanggal'))
        ->where('status', 2)-
        >where('predict_sentimen_label', 'positif')-
        >groupBy(DB::raw('DATE(post_time)'))->get();
        $dataprediksi=DB::table('classifications')-

```

```

>join('tweets', 'tweets.id', '=', 'classifications.id_tweet')
    ->select(DB::raw('count(*) as total, DATE(post_time)
as tanggal'))
    ->where('status', 2) -
>groupBy(DB::raw('DATE(post_time)'))->get();
    $positif=array();
    $totalprediksi=array();
    $indekskebahagiaan=array();
    foreach ($datapositif as $dp) {
        $positif[$dp->tanggal]=$dp->total;
    }
    foreach ($dataprediksi as $dp) {
        if (!isset($positif[$dp->tanggal])) {
            $positif[$dp->tanggal]=0;
        }
        $totalprediksi[$dp->tanggal]=$dp->total;
        $indekskebahagiaan[$dp->tanggal]=$positif[$dp-
>tanggal]/$totalprediksi[$dp->tanggal];
    }
    return
view('datatesting.indekskebahagiaan', ['indekskebahagiaan'=>$indeks
kebahagiaan]);
}

```

D.13. Kode Program Mengelola Data Indeks Kebahagiaan Harian

```

<?php

namespace App\Http\Controllers;
define('CONSUMER_KEY', '0Y9KZFaZaUI67BQ5RjBcPPhSN'); //isikan
dengan CONSUMER KEY anda
define('CONSUMER_SECRET',
'ey1LZH8xYegJ0h9EIB0m2nsD18shVcJHZtxlNfcHKdzTR5tbyK'); //isikan
dengan CONSUMER KEY anda
define('ACCESS_TOKEN', '3014251646-
J0LadQ2pyEAs0xLwnhwXu5nv27gZL8of0NAPzGB'); //isikan dengan
CONSUMER KEY anda
define('ACCESS_TOKEN_SECRET',
'R06seEsYZdaPy2LkpHz54JLhLxBx0snXg2AdpPuaBipdY'); //isikan dengan
CONSUMER KEY anda
use Abraham\TwitterOAuth\TwitterOAuth;
use App\Happiness;
use App\Ontology;
use App\Tweet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Input;
use Illuminate\Support\Facades\Redirect;
use Phpml\Classification\NaiveBayes;
use Spatie\TwitterStreamingApi\PublicStream;

class StreamingController extends TextMiningController
{

```

```

public function __construct()
{
    $this->middleware('auth');
}
public function index()
{
    $happiness=Happiness::all();
    return
view('streaming.visualisasi', ['happiness'=>$happiness]);
}

public function search($query)
{
    $limit =
(isset($_GET['jumlahtweet']))?$_GET['jumlahtweet']:100;
    $max_id = null;
    $count=100;
    $contents = array();
    $connection = new TwitterOAuth(CONSUMER_KEY,
CONSUMER_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET);
    for ($i = 0; $i < $limit; $i += $count) {

        $content = $connection->get('search/tweets', array(
            "q" =>$query,
            "count"=>$limit,
            'max_id'=>$max_id,
            'lang'=>'id'));
        $contents[] = $content;
        if(count($content->statuses)>0)
            $max_id=($content->statuses[count($content->
statuses)-1]->id_str);
    }

    return $contents;
}
public function klasifikasihariini()
{
    $datatraining=Tweet::where('status',1)->get();
    $hasilstemmingtraining=$this->stemming($datatraining);
    $hasilbukancorpustraining=$this->cekCorpus($hasilstemmingtraining);
    $hasilstopwordremovaltraining=$this->stopwordremoval($hasilbukancorpustraining);
    $hasilngramtraining=$this->ngram($hasilstopwordremovaltraining);
    $j=0;
    foreach($datatraining as $dt) {
        $manualsentimen[$j]=$dt->classification->manual_sentimen_label;
        $j++;
    }
    $classifiersentimen=new NaiveBayes();
    $k=0;
    $i=0;
    foreach ($hasilngramtraining as $hnt) {
        foreach ($hnt as $ss) {

```

```

        $classifiersentimen-
>train([[ $ss]], [ $manualsentimen[ $k]]); //menambah training
        $i++;
    }
    $k++;
}

    $tweethariini=$this->datatweethariini();
    $hasilstemmingtesting=$this-
>stemmingstreaming($tweethariini);
    $hasilbukancorpustesting=$this-
>cekCorpus($hasilstemmingtesting);
    $hasilstopwordremovaltesting=$this-
>stopwordremoval($hasilbukancorpustesting);
    $hasilngramtesting=$this-
>ngram($hasilstopwordremovaltesting);

    $positif=0;
    $negatif=0;
    $i=0;
    foreach ($hasilngramtesting as $hnt) {

        $klasifikasisentimen[ $i] [ 'positif'] =0;
        $klasifikasisentimen[ $i] [ 'negatif'] =0;
        $klasifikasisentimen[ $i] [ 'netral'] =0;
        $klasifikasisentimen[ 'hasil'] [ $i] ="";
        foreach ($hnt as $ngram) {

            $predictedsentimen = $classifiersentimen-
>predict([ $ngram]);

            (isset($predictedsentimen[ 'positif']))? $klasifikasisentimen[ $i] [ 'p
ositif'] += $predictedsentimen[ 'positif'] : $klasifikasisentimen[ $i] [ '
positif'] =0;

            (isset($predictedsentimen[ 'negatif']))? $klasifikasisentimen[ $i] [ 'n
egatif'] += $predictedsentimen[ 'negatif'] : $klasifikasisentimen[ $i] [ '
negatif'] =0;

            (isset($predictedsentimen[ 'netral']))? $klasifikasisentimen[ $i] [ 'ne
tral'] += $predictedsentimen[ 'netral'] : $klasifikasisentimen[ $i] [ 'net
ral'] =0;
        }

        arsort($klasifikasisentimen[ $i], SORT_NUMERIC);

        $klasifikasisentimen[ 'hasil'] [ $i] =key($klasifikasisentimen[ $i]);

        if ($klasifikasisentimen[ 'hasil'] [ $i] == 'positif') {
            $positif +=1;
        } else
if ($klasifikasisentimen[ 'hasil'] [ $i] == 'negatif') {
            $negatif +=1;
        }
    }
}

```

```

        $i++;
    }
    //      dd($tweethariini);
    $indekskebahagiaan['positif']=$positif;
    $indekskebahagiaan['negatif']=$negatif;
    $indekskebahagiaan['hasil']=$positif/($positif+$negatif);
    return
view('streaming.index', ['tweethariini'=>$tweethariini, 'klasifikasi
sentimen'=>$klasifikasisentimen, 'indekskebahagiaan'=>$indekskebaha
giaan]);
}

public function stemmingstreaming($data) {
    $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();
    $output=array();
    // stem
    $i=0;
    foreach ($data as $dt) {
        $kw[$i]=$dt->text;
        $i++;
    }
    $i=0;
    foreach ($kw as $k) {
        $output[$i]=$stemmer->stem($k);
        $i++;
    }
    return($output);
}

public function datatweethariini() {
    $ontologi['kesehatan']=Ontology::where('parameter','kesehatan')->
get();
    $ontologi['pendidikan']=Ontology::where('parameter','pendidikan')->
get();
    $ontologi['kondisi
keamanan']=Ontology::where('parameter','kondisi keamanan')->get();
    $ontologi['kondisi
lingkungan']=Ontology::where('parameter','kondisi lingkungan')->
get();
    $ontologi['keharmonisan
keluarga']=Ontology::where('parameter','keharmonisan keluarga')->
get();
    //
    $ontologi['hubungan
sosial']=Ontology::where('parameter','hubungan sosial')->get();
    //
    $ontologi['kondisi rumah dan
aset']=Ontology::where('parameter','kondisi rumah dan aset')->
get();
    $ontologi['pekerjaan']=Ontology::where('parameter','pekerjaan')->
get();
    $ontologi['ketersediaan waktu

```

```

luang'=Ontology::where('parameter','ketersediaan waktu luang')->get();
    $ontologi['pendapatan rumah
tangga'=Ontology::where('parameter','pendapatan rumah tangga')->get();
    $j=0;

    foreach ($ontologi as $onto){
        $queryontologi[$j]=$onto[0]->text;
        for ($i=1;$i<count($onto);$i++){
            $queryontologi[$j] =$queryontologi[$j]. ' OR
'. $onto[$i]->text;
        }
        $j++;
    }
//
    $tweethariini=array();
    $results=array();
    foreach ($queryontologi as $query){
        $results[] = $this->search($query);
    }

    foreach($results as $result1){
        foreach($result1 as $result2){
            foreach($result2->statuses as $result3) {
                $tweethariini[]=$result3;
            }
        }
    }
    return $tweethariini;
}
public function saveindex(){
    $tweet=Happiness::create([
        'index'=>Input::get('indekskebahagiaan'),
        'date'=>Input::get('tanggal'),
    ]);
    return Redirect::to('/streaming');
}
function delete($id){
    Happiness::destroy($id);
    return Redirect::to('/streaming');
}
}
}

```

D.14. Kode Program Melihat *FAQ*

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class FAQController extends Controller

```

```
{  
    public function index()  
    {  
        return view('faq.index');  
    }  
}
```

D.15. Logout

```
public function logout(Request $request)  
{  
    $this->guard()->logout();  
    $request->session()->flush();  
    $request->session()->regenerate();  
    return redirect('/');  
}
```