



**IDENTIFIKASI PERSEPSI KONSUMEN KOPI DI MEDIA SOSIAL
TWITTER MENGGUNAKAN ANALISIS SENTIMEN DENGAN METODE
NAÏVE BAYES CLASSIFIER**

SKRIPSI

Oleh

Adjie Suryanendra

NIM 142410101063

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2018



**IDENTIFIKASI PERSEPSI KONSUMEN KOPI DI MEDIA SOSIAL
TWITTER MENGGUNAKAN ANALISIS SENTIMEN DENGAN METODE
NAÏVE BAYES CLASSIFIER**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

Adjie Suryanendra

NIM 142410101063

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2018

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 Enie Soerjandari dan Ayahanda Suyitno;
3. Saudari – saudaraku Elok Cahyaning Palupi dan Candra Adji Palupi;
4. Keluarga besar;
5. Sahabat – sahabatku dengan dukungan beserta doanya;
6. Guru – guruku baik dari pendidikan formal maupun informal;
7. Almamater Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

MOTTO

“Strive for perfection in everything you do. Take the best that exists and make it better. When it does not exist, design it.”

“Berusahalah untuk kesempurnaan dalam segala hal yang Anda lakukan. Ambil yang terbaik yang ada dan buatlah menjadi lebih baik. Ketika sesuatu tidak ada, rancanglah.”¹

Sir Frederick Henry Royce



¹ Slogan for the Rolls-Royce company, quoted in *Transforming the Organization* (1996) by Francis J. Guillard, p. 85

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Adjie Suryanendra

NIM : 142410101063

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Identifikasi Persepsi Konsumen Kopi Di Media Sosial *Twitter* Menggunakan Analisis Sentimen Dengan Metode *Naïve 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, 5 Juli 2018
Yang menyatakan,

Adjie Suryanendra
NIM 142410101063

SKRIPSI

**IDENTIFIKASI PERSEPSI KONSUMEN KOPI DI MEDIA SOSIAL
TWITTER MENGGUNAKAN ANALISIS SENTIMEN DENGAN METODE
NAÏVE BAYES CLASSIFIER**

Oleh

Adjie Suryanendra

NIM 142410101063

Pembimbing

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

Dosen Pembimbing Pendamping : Ifrina Nuritha, S.Kom., M.Kom.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Identifikasi Persepsi Konsumen Kopi Di Media Sosial *Twitter* Menggunakan Analisis Sentimen Dengan Metode *Naïve Bayes Classifier*”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I

Pembimbing II

Nelly Oktavia Adiwijaya, S.Si., MT.
NIP 19841024 200912 2 008

Ifrina Nuritha, S.Kom., M.Kom.
NRP 760016786

PENGESAHAN PENGUJI

Skripsi berjudul “Identifikasi Persepsi Konsumen Kopi Di Media Sosial *Twitter* Menggunakan Analisis Sentimen Dengan Metode *Naïve Bayes Classifier*”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Penguji I

Penguji II

Yanuar Nurdiansyah, ST., M.Cs.
NIP 19820101 201012 1 004

Beny Prasetyo, S.Kom., M.Kom.
NRP 760016852

Mengesahkan
Dekan Fakultas Ilmu Komputer

Prof. Drs. Slamin, M.Comp.Sc., Ph.D.
NIP 19670420 199201 1 001

RINGKASAN

Identifikasi Persepsi Konsumen Kopi Di Media Sosial *Twitter* Menggunakan Analisis Sentimen Dengan Metode *Naïve Bayes Classifier*; Adjie Suryanendra, 142410101063; 2018, 170 halaman; Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Kopi adalah sejenis minuman yang berasal dari proses pengolahan dan ekstraksi biji tanaman kopi. Kopi kemudian terus berkembang hingga saat ini menjadi salah satu minuman paling populer di dunia yang dikonsumsi oleh berbagai kalangan masyarakat. Tradisi minum kopi tidak hanya sekedar untuk melepas dahaga, tetapi juga untuk menemani aktivitas kehidupan masyarakat seperti rapat, pertemuan bisnis, reuni, kencan, dan lain-lain. Tren peminum kopi terus meningkat tajam dan secara tidak disadari, tradisi minum kopi ini telah menjadi bagian dari gaya hidup. Tetapi, tidak semua orang menyukai kopi dengan berbagai alasannya. Ada banyak faktor yang mendasari orang enggan minum kopi padahal sama sekali tidak pernah meminum kopi seperti rasanya yang pahit, penyebab *maag*, perut kembung, dan jantung berdebar-debar, membuat susah tidur dan sebagainya. Hal ini tentunya membuat persepsi orang terhadap kopi tersebut berbeda-beda.

Proyeksi permintaan kopi oleh Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian tahun 2015 menunjukkan permintaan kopi di Indonesia tahun 2015 sampai tahun 2019 mengalami peningkatan sebesar 5,09% per tahunnya dengan konsumsi mencapai 1,623 kg/kapita. Tetapi harga kopi dari tahun ke tahun terus mengalami peningkatan.

Perilaku mengunggah status dan kegiatan yang sedang dilakukan ke media sosial yang dimilikinya, termasuk mengunggah opini-opini dan ulasan terhadap produk atau jasa yang pernah dicoba membuat penetrasi internet di Indonesia kini tumbuh dari tahun ke tahun dengan media sosial sebagai predikat tertinggi untuk konten yang sering diakses. Salah satunya adalah *Twitter*.

Dari permasalahan di atas, maka diperlukan sebuah sistem yang mampu melakukan identifikasi persepsi konsumen kopi secara real time tanpa harus

melakukan survei di lapangan dengan menggunakan model klasifikasi yang akurat dengan pendekatan analisis opini di media sosial.

Penelitian ini menggunakan analisis sentimen dengan pendekatan *Text Mining* dan metode *Naïve Bayes Classifier* dalam melakukan klasifikasi. Penelitian ini dilaksanakan dalam 4 tahap penelitian yaitu studi literatur, pengumpulan data, perancangan model klasifikasi, dan pengembangan perangkat lunak. Model pengembangan perangkat lunak dari sistem ini mengacu pada model *waterfall*. Uji performansi klasifikasi pada sistem ini menggunakan 3 perhitungan yaitu *precision*, *recall*, dan *f1score*. Hasil uji performansi terhadap sistem menunjukkan bahwa sistem dapat melakukan klasifikasi dengan baik yaitu dengan nilai 0,6 hingga 0,8 untuk klasifikasi sentimen dan kategori jenis kopi.

Crawling data menggunakan *Standard Twitter Search API* yang memiliki beberapa keterbatasan. Hasil *crawling* data pada media sosial *twitter* menggunakan kata kunci menghasilkan 900 *data training* dan 100 *data testing* dengan variasi pelabelan sentimen dan kategori kopi oleh ahli bahasa. Data selanjutnya melalui tahap *text mining* berupa *text preprocessing* yang terdiri dari *case folding* dan *tokenizing*, diteruskan dengan *feature generation* yang meliputi *stemming* dan *stopword removal* hingga menghasilkan data yang bersih dan siap untuk diklasifikasikan dengan metode *Naïve Bayes Classifier*.

Hasil klasifikasi persepsi konsumen kopi di Indonesia dari hari ke hari mengalami peningkatan dengan puncak pada akhir pekan, yaitu antara hari jumat (0,83769230769231), sabtu (0,84615384615384), minggu (0,86666666666667) hingga senin (0,85714285714285). Hal ini merupakan peningkatan sesuai gaya hidup minum kopi yang ada di suatu wilayah. Tradisi minum kopi tidak sekedar untuk melepas dahaga, tetapi untuk menemani aktivitas lain.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Identifikasi Persepsi Konsumen Kopi Di Media Sosial *Twitter* Menggunakan Analisis Sentimen Dengan Metode *Naïve 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. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Nelly Oktavia A, S.Si., MT., selaku dosen pembimbing utama dan Ifrina Nuritha, S.Kom., M.Kom., selaku dosen pembimbing pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Yanuar Nurdiansyah, ST., M.Cs. selaku dosen pembimbing akademik sekaligus dosen penguji utama dan Beny Prasetyo, S.Kom., M.Kom. selaku penguji anggota yang telah berkenan untuk menguji skripsi ini dan memberikan masukan serta saran untuk pengembangan penulis dan skripsi ini;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Fakultas Ilmu Komputer Universitas Jember;
5. Ibunda Enie Soerjandari dan Ayahanda Suyitno yang selalu mendukung serta mendoakan dan menjadi orang tua yang luar biasa;
6. Saudariku Elok Cahyaning Palupi dan saudaraku Candra Adji Palupi yang telah memberikan dukungan semangat;
7. Keluarga besar yang telah mendukung dan mendoakan;
8. Sahabatku Muhammad Heikal Bagus Hartantyo, yang telah menjadi sahabat yang setia mendengar keluh kesah, memberi semangat, motivasi;

9. Teman-teman seperjuangan Moch. Rifqiy Rajihudin, Anas Hidayah, Muhammad Huda M., Muhammad Khoirul Umam yang telah bersedia meluangkan waktu untuk membantu dan memberikan dukungan semangat;
10. Keluarga SENSATION angkatan 2014 atas motivasi yang selalu mengalir;
11. Teman-teman Fakultas Ilmu Komputer di semua angkatan atas bantuan dan dukungannya;
12. Semua pihak yang tidak dapat disebutkan satu persatu.

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

Jember, 16 Juli 2018

Penulis

DAFTAR ISI

SKRIPSI.....	1
SKRIPSI.....	ii
PERSEMBAHAN.....	iii
MOTTO	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN	ix
PRAKATA.....	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR	xix
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu.....	6
2.2 Konsumen Kopi.....	7

2.3	Text Mining	8
2.3.1	<i>Text Preprocessing</i>	9
2.3.2	<i>Text Transformation/Feature Generation</i>	9
2.3.3	<i>Pattern Discovery</i>	10
2.4	Analisis Sentimen	13
2.5	Naïve Bayes Classifier	14
2.6	Algoritma Enhanced Confix-Stripping (ECS)	16
2.7	Twitter API	19
2.8	Uji Performansi	20
BAB 3. METODOLOGI PENELITIAN		22
3.1	Jenis Penelitian	22
3.2	Tempat dan Waktu Penelitian	22
3.3	Tahapan Penelitian	22
3.3.1	Studi Literatur	22
3.3.2	Pengumpulan Data	23
3.3.3	Perancangan Model Klasifikasi	23
3.3.4	Pengembangan Perangkat Lunak	23
3.4	Tahapan Algoritma Sistem	27
BAB 4. PERANCANGAN SISTEM		31
4.1	Deskripsi Umum Sistem	31
4.2	Analisis Kebutuhan	31
4.2.1	Studi Literatur	31
4.2.2	Kebutuhan Fungsional	32
4.2.3	Kebutuhan Non Fungsional	33
4.3	Desain Sistem	33

4.3.1 <i>Business Process</i>	33
4.3.2 <i>Use Case Diagram</i>	34
4.3.3 Skenario	38
4.3.4 <i>Sequence Diagram</i>	43
4.3.5 <i>Activity Diagram</i>	46
4.3.6 <i>Class Diagram</i>	48
4.3.7 <i>Entity Relationship Diagram (ERD)</i>	49
4.4 Pengujian Sistem	50
BAB 5. HASIL DAN PEMBAHASAN.....	61
5.1 Hasil <i>Crawling Data</i> pada <i>Twitter</i>	61
5.2 Hasil Dataset.....	63
5.3 Hasil <i>Text Preprocessing</i>	65
5.4 Hasil <i>Feature Generation</i>	67
5.5 Hasil Klasifikasi Sentimen	69
5.6 Hasil Klasifikasi Kategori	71
5.7 Perhitungan Manual Naïve Bayes Classifier.....	72
5.8 Uji Performansi	75
5.9 Perhitungan Persepsi Konsumen Kopi	76
5.10 Visualisasi.....	77
5.10.1 Halaman Beranda.....	77
5.10.2 Halaman <i>Login</i>	78
5.10.3 Tampilan Halaman Admin.....	79
5.10.4 Tampilan Halaman <i>Tweet</i>	80
5.10.5 Tampilan Halaman <i>Data Training</i>	81
5.10.6 Tampilan Halaman <i>Text Mining</i>	82

5.10.7 Tampilan Halaman <i>Data testing</i>	82
5.10.8 Tampilan Halaman <i>Stopword</i>	83
5.10.9 Tampilan Halaman Ontologi.....	84
5.10.10 Tampilan Halaman <i>User</i>	84
5.10.11 Tampilan Halaman Utama Ahli Bahasa.....	85
5.10.12 Tampilan Utama <i>FAQ</i>	86
BAB 6. PENUTUP	87
6.1 Kesimpulan.....	87
6.2 Saran.....	89
DAFTAR PUSTAKA	90
LAMPIRAN.....	93

DAFTAR TABEL

Tabel 2.1 Contoh Data <i>Tweet</i> Sentimen dan Non Sentimen.....	11
Tabel 2.2 Contoh Data <i>Tweet</i> Sentimen Positif dan Negatif	11
Tabel 2.3 Contoh Luaran Tahapan <i>Case Folding</i>	12
Tabel 2.4 Contoh Luaran Tahapan <i>Tokenizing</i>	12
Tabel 2.5 Contoh Luaran Tahapan <i>Filtering Stopwords</i>	12
Tabel 2.6 Contoh Luaran Tahapan <i>Stemming</i>	12
Tabel 2.7 Aturan Pemenggalan Kata Dalam <i>ECS</i>	17
Tabel 2.8 <i>API Twitter</i>	20
Tabel 2.9 Matriks <i>Contingency</i> Kelas Prediksi dan Aktual	21
Tabel 3.1 Penjelasan Alur Sistem	28
Tabel 4.1 Definisi Aktor	36
Tabel 4.2 Definisi <i>Usecase</i>	36
Tabel 4.3 Melihat Klasifikasi & Uji Performansi <i>Data Testing</i>	40
Tabel 4.4 Hasil Pengujian <i>Black Box</i>	51
Tabel 5.1 Tabel <i>API Twitter</i>	62
Tabel 5.2 Contoh <i>Data Training</i>	63
Tabel 5.3 Tabel Pelabelan Sentimen.....	64
Tabel 5.4 Tabel Pelabelan Kategori	64
Tabel 5.5 Contoh <i>Text Preprocessing</i>	66
Tabel 5.6 Contoh Hasil <i>Stemming</i>	67
Tabel 5.7 Contoh Hasil Penyaringan Kata Dasar.....	68
Tabel 5.8 Contoh Hasil <i>Stopword Removal</i>	69
Tabel 5.9 Contoh Hasil Klasifikasi Sentimen	69
Tabel 5.10 Contoh Hasil Klasifikasi Kategori	71
Tabel 5.11 Contoh Hasil Klasifikasi Kategori Salah Prediksi	72
Tabel 5.12 Perhitungan Probabilitas Kata mari	73
Tabel 5.13 Perhitungan Probabilitas Kata nikmat.....	73
Tabel 5.14 Perhitungan Probabilitas Kata kopi	74
Tabel 5.15 Perhitungan <i>Prior Probability</i>	74

Tabel 5.16 Perhitungan Probabilitas Status	74
Tabel 5.17 Uji Performansi Kelas Sentimen Rata-Rata.....	75
Tabel 5.18 Uji Performansi Kelas Kategori Rata-Rata.....	75
Tabel 5.19 Hasil Perhitungan Persepsi Konsumen Kopi Harian	76



DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian	22
Gambar 3.2 <i>SDLC Waterfall</i>	24
Gambar 3.3 <i>Flowchart</i> Alur Sistem	27
Gambar 4.1 <i>Business Process</i>	34
Gambar 4.2 <i>Use Case Diagram</i>	35
Gambar 4.3 <i>Sequence diagram</i> Melihat Klasifikasi dan Uji Performansi Data ...	45
Gambar 4.4 <i>Activity diagram</i> Melihat Klasifikasi dan Uji Performansi <i>Data Testing</i>	47
Gambar 4.5 <i>Class Diagram</i>	49
Gambar 4.6 <i>Entity Relationship Diagram (ERD)</i>	50
Gambar 5.1 <i>Flowchart Crawling</i>	61
Gambar 5.2 Tampilan Halaman Beranda.....	77
Gambar 5.3 Tampilan Halaman Kelebihan Sistem.....	78
Gambar 5.4 Tampilan Halaman Identifikasi Persepsi Konsumen Kopi Harian ...	78
Gambar 5.5 Tampilan Halaman <i>Login</i>	79
Gambar 5.6 Tampilan Halaman Membuat Akun Baru	79
Gambar 5.7 Tampilan Halaman Utama Admin	80
Gambar 5.8 Tampilan Halaman <i>Tweet</i>	81
Gambar 5.9 Tampilan Halaman <i>Data Training</i>	81
Gambar 5.10 Tampilan Halaman <i>Text Mining</i>	82
Gambar 5.11 Tampilan Halaman <i>Data Testing</i>	83
Gambar 5.12 Tampilan Halaman <i>Stopword</i>	83
Gambar 5.13 Tampilan Halaman Ontologi.....	84
Gambar 5.14 Tampilan Halaman <i>User</i>	85
Gambar 5.15 Tampilan Halaman Utama Ahli Bahasa.....	85
Gambar 5.16 Tampilan Halaman <i>FAQ</i>	86

BAB 1. PENDAHULUAN

Bab ini merupakan bagian awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan yang ingin dicapai, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Kopi adalah sejenis minuman yang berasal dari proses pengolahan dan ekstraksi biji tanaman kopi. Kata kopi sendiri berasal dari bahasa Arab *qahwah* yang berarti kekuatan, karena pada awalnya kopi digunakan sebagai makanan berenergi tinggi. Secara umum, terdapat dua jenis biji kopi, yaitu *arabika* (kualitas terbaik) dan *robusta*. Kopi kemudian terus berkembang hingga saat ini menjadi salah satu minuman paling populer di dunia yang dikonsumsi oleh berbagai kalangan masyarakat. Indonesia sendiri telah mampu memproduksi lebih dari 400 ribu ton kopi per tahunnya. Di samping rasa dan aromanya yang menarik, kopi juga dapat menurunkan risiko terkena penyakit kanker, *diabetes*, batu empedu dan berbagai penyakit jantung (*kardiovaskuler*) (Market Intelligence Kopi, 2015).

Tradisi minum kopi tidak hanya sekedar untuk melepas dahaga, tetapi juga untuk menemani aktivitas kehidupan masyarakat seperti rapat, pertemuan bisnis, reuni, kengan, dan lain-lain. Hal inilah yang membuat tren peminum kopi terus meningkat tajam dan secara tidak disadari, tradisi minum kopi ini telah menjadi bagian dari gaya hidup (Suisa & Febrilia, 2014). Tetapi, tidak semua orang menyukai kopi dengan berbagai alasannya. Ada banyak faktor yang mendasari orang enggan minum kopi padahal sama sekali tidak pernah meminum kopi seperti rasanya yang pahit, penyebab *maag*, perut kembung, dan jantung berdebar-debar, membuat susah tidur dan sebagainya. Hal ini tentunya membuat persepsi orang terhadap kopi tersebut berbeda-beda (Yuliandri, 2017).

Kandungan *kafein* yang punya daya rangsang terhadap peningkatan kinerja beberapa bagian susunan saraf pusat, minuman kopi pun menjadi jenis minuman yang sangat digemari oleh manusia di berbagai penjuru dunia. Fakta menunjukkan

bahwa kopi merupakan komoditas nomor dua yang paling banyak diperdagangkan setelah minyak bumi (Fityan & Sinaga, 2015). Selain itu, proyeksi permintaan kopi oleh Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian tahun 2015 menunjukkan permintaan kopi di Indonesia tahun 2015 sampai tahun 2019 mengalami peningkatan sebesar 5,09% per tahunnya dengan konsumsi mencapai 1,623 kg/kapita. Walaupun begitu, pada tahun 2007 rata-rata harga satu kilogram kopi arabika di Indonesia adalah Rp.10.850,- sementara ditahun yang sama harga kopi robusta mencapai Rp.10.013,- per kilogram. Harga ini kemudian meningkat pada tahun 2013 menjadi Rp.21.620,- per kilogram untuk kopi arabika dan Rp.16.341,- untuk kopi robusta (Kementerian Pertanian, 2015).

Penetrasi internet Indonesia menunjukkan peningkatan dari tahun ke tahun. Berdasarkan data yang didapat oleh APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2017, jumlah pengguna Internet di Indonesia mencapai 143,26 juta jiwa, dimana media sosial mendapatkan peringkat tertinggi untuk jenis konten yang sering diakses mencapai 87,13% yang artinya bahwa hampir semua pengguna internet Indonesia memiliki media sosial. Hal ini kemudian memunculkan perilaku mengunggah status dan kegiatan yang sedang dilakukan ke media sosial yang dimilikinya, termasuk mengunggah opini-opini dan ulasan terhadap produk atau jasa yang pernah dicoba. (Asosiasi Penyelenggara Jasa Internet Indonesia, 2017).

Permasalahan yang timbul dari data tersebut adalah permintaan kopi yang terus meningkat diikuti dengan berbagai macam persepsi yang ada di masyarakat. Maka dapat dibuat sebuah pendekatan untuk mengetahui persepsi konsumen kopi dengan menggunakan data analisis sentimen yang didapat dari media sosial di masyarakat melalui tren mengunggah status di media sosial. Dengan adanya pendekatan ini, maka didapatkan keuntungan berupa pengukuran persepsi konsumen kopi di wilayah tertentu secara langsung dan cepat tanpa harus memerlukan survei di lapangan.

Penelitian ini dilakukan untuk memberikan kontribusi dengan mengidentifikasi persepsi konsumen kopi yang bermacam-macam melalui

penggunaan metode *Naïve Bayes Classifier* pada klasifikasinya. Penggunaan *Naïve Bayes Classifier* karena metode klasifikasi ini cukup banyak dipakai dalam melakukan sentimen analisis dan *text mining*, salah satunya Boy Utomo Manalu yang melakukan analisis sentimen provider telekomunikasi menggunakan *Naïve Bayes Classifier*. Dari percobaan yang dilakukan, *Naïve Bayes Classifier* mampu melakukan klasifikasi sentimen dengan tingkat akurasi di atas 80% pada data training di atas 300 pada masing-masing sentimen (Manalu, 2014).

1.2 Rumusan Masalah

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

1. Bagaimana mengidentifikasi persepsi konsumen kopi di media sosial *Twitter* menggunakan analisis sentimen?
2. Bagaimana persepsi konsumen kopi di Indonesia?

1.3 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 *Text Mining* dengan metode *Naïve Bayes Classifier* untuk mengidentifikasi persepsi konsumen kopi di media sosial *Twitter*.
2. Mengetahui persepsi konsumen kopi di Indonesia.

1.4 Batasan Masalah

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

1. Algoritma yang digunakan dalam pengklasifikasian adalah *Naïve Bayes Classifier*.

2. Menggunakan metode *case folding*, *tokenisasi*, *stemming* dan *stopword removal* untuk seleksi data dalam *text mining*.
3. Menggunakan Algoritma *Enhanced Confix-Stripping (ECS)* dalam melakukan *stemming*.
4. Data yang diambil berasal dari *Twitter* berbahasa Indonesia baku dan berada di wilayah Indonesia.
5. Kata kunci yang digunakan dalam teknik *crawling* data *Twitter* adalah ‘kopi hitam’, ‘kopi robusta’, ‘kopi arabika’, ‘kopi aceh’, ‘kopi toraja’, ‘kopi luwak’, ‘kopi papua’, ‘kopi hijau’, ‘kopi sakit’, ‘kopi pahit’.

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. Perancangan Sistem

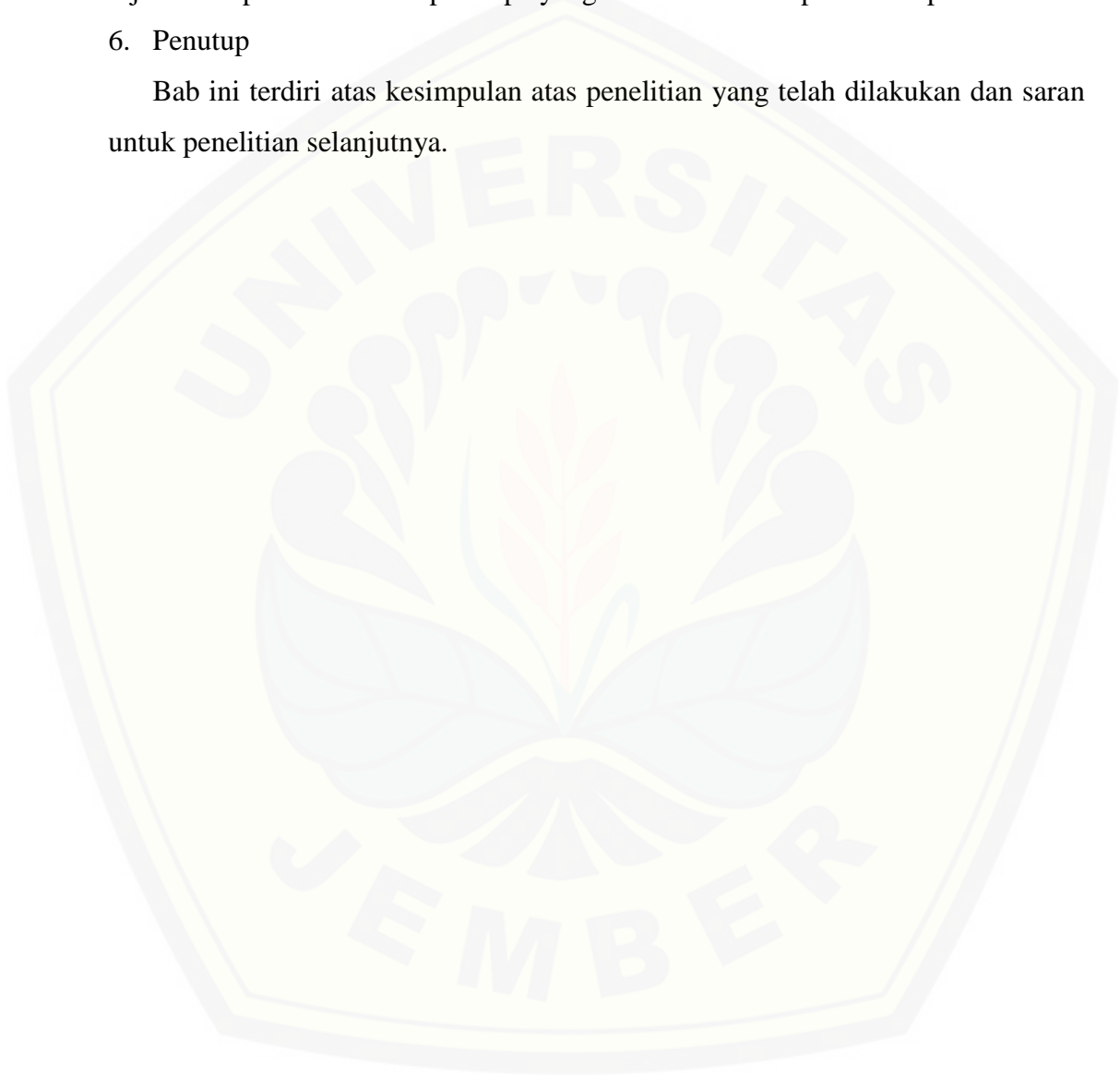
Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem yang dibangun.

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 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

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literatur dan jurnal.

2.1 Penelitian Terdahulu

Penelitian tentang sikap konsumen terhadap kopi dilakukan oleh Leo, Thomson, dan Salmiah melalui jurnal berjudul Analisis Sikap Konsumen Terhadap Kopi Bubuk Lokal dan Kopi Instan di Kota Pematangsiantar yang bertujuan untuk menjelaskan dan menganalisis sikap konsumen terhadap kopi bubuk lokal dan kopi bubuk instan secara *purposive* di Kota Pematangsiantar, Sumatera Utara. Hasil penelitian menunjukkan bahwa alasan utama konsumen mengkonsumsi kopi adalah kesegaran dengan pertimbangan awal berupa rasa dan kualitas kopinya. Konsumen menyatakan puas terhadap kopi yang dibeli dan dikonsumsi dan bersedia melakukan pembelian ulang meskipun harga produk kopi mengalami kenaikan. Selain itu, konsumen juga cenderung mengkonsumsi kopi instan dibandingkan kopi bubuk lokal (Tambunan, Sebayang, & Salmiah, 2018).

Penelitian tentang penggunaan *Naïve Bayes Classifier* dalam melakukan klasifikasi sentimen telah banyak dilakukan, salah satunya pernah dilakukan oleh Yanuar, Saiful, dan Rahmat (2018) dalam prosiding IOP Conference yang berjudul Sistem Analisis Sentimen Ulasan Film dalam Bahasa Indonesia dengan Metode *Naïve Bayes Classifier*. Data didapatkan dari situs Movienthusiast berupa 1201 *dataset* yang terdiri dari 783 sentimen positif dan 418 sentimen negatif. Hasil klasifikasi menghasilkan akurasi 88,37% dalam lima kali pengujian menggunakan *dataset* tersebut (Nurdiansyah, Bukhori, & Hidayat, 2018).

Penelitian lain dilakukan oleh Boy Utomo Manalu dengan skripsi berjudul Analisis Sentimen Pada *Twitter* Menggunakan *Text Mining*. Dalam skripsi ini dilakukan penelitian tentang penggunaan *Naïve Bayes Classifier* untuk melakukan

analisis sentimen terhadap provider telekomunikasi di Indonesia. Dari percobaan yang dilakukan, *Naïve Bayes Classifier* mampu melakukan klasifikasi sentimen dengan tingkat akurasi di atas 80% pada data training di atas 300 pada masing-masing sentimen (Manalu, 2014).

Penelitian tentang penggunaan *Text Mining* telah dilakukan, salah satunya dilakukan oleh Ahmad Fathan Hidayatullah dan Muhammad Rifqi Ma'arif 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 *Naïve Bayes Classifier* dan *Support Vector Machine (SVM)*. Secara keseluruhan, hasil perolehan *F-measure* pada algoritma *Naïve Bayes Classifier* memberikan hasil yang lebih tinggi dibandingkan *SVM* (Hidayatullah & Ma'arif, 2016).

2.2 Konsumen Kopi

Perilaku konsumen menurut Sudiyarto, dkk (2012) mencakup pemahaman terhadap tindakan yang langsung dilakukan konsumen dalam mendapatkan, mengkonsumsi, dan menghabiskan produk dan jasa, termasuk proses keputusan yang mendahului dan mengikuti tindakan tersebut. Hasil pengolahan bahan minuman yang bersumber dari sektor pertanian sangat diperlukan masyarakat untuk membantu dalam proses metabolisme tubuh, penghilang dahaga atau hanya sebagai bentuk kebiasaan. Salah satu jenis minuman yang populer di masyarakat adalah kopi (Sudiyarto, Widayati, & Kresna, 2012).

Kopi paling banyak dikonsumsi masyarakat setelah teh. Kopi dinikmati baik dingin maupun panas, selera minum kopi terbawa secara turun temurun hingga sekarang. Tingkat konsumsi kopi dalam negeri berdasarkan hasil survei LPEM UI tahun 1989 adalah sebesar 500 gram/kapita/tahun. Namun pada tahun 2011 tingkat konsumsi kopi di Indonesia telah mencapai 800 gram/kapita/tahun. Dengan demikian dalam kurun waktu 20 tahun peningkatan konsumsi kopi telah mencapai 300 gram/kapita/tahun. Masyarakat memiliki kebiasaan meminum kopi setiap

harinya. Berbagai kalangan status sosial menggemari minuman kopi dengan tujuan konsumsi yang berbeda-beda, antara lain untuk mencegah penyakit syaraf, menurunkan resiko kanker payudara, mencegah diabetes, dan berkhasiat merevitalisasi sel kulit baru dan menjaga kelembapan. Berdasarkan jenisnya, kopi olahan yang ada dipasaran dibedakan menjadi 2 jenis yaitu kopi tubruk dan kopi instan. Kopi tubruk adalah minuman kopi yang dibuat dengan mendidihkan biji kopi bersama dengan gula, saat diseduh minuman kopi tersebut juga meninggalkan ampas, sedangkan kopi instan dapat berupa campuran kopi, gula dan susu, yang melalui proses granulasi terlebih dahulu kemudian dikemas dengan bahan *aluminium foil*, toples, maupun botol. Dalam penyajiannya kopi instan tidak meninggalkan ampas (Sudiyarto, Widayati, & Kresna, 2012).

Tradisi minum kopi tidak hanya sekedar untuk melepas dahaga, tetapi juga untuk menemani aktivitas kehidupan masyarakat seperti rapat, pertemuan bisnis, reuni, kengan, dan lain-lain. Hal inilah yang membuat tren peminum kopi terus meningkat tajam dan secara tidak disadari, tradisi minum kopi ini telah menjadi bagian dari gaya hidup (Suisa & Febrilia, 2014). Tetapi, tidak semua orang menyukai kopi dengan berbagai alasannya. Ada banyak faktor yang mendasari orang enggan minum kopi padahal sama sekali tidak pernah meminum kopi seperti rasanya yang pahit, penyebab *maag*, perut kembung, dan jantung berdebar-debar, membuat susah tidur dan sebagainya. Hal ini tentunya membuat persepsi orang terhadap kopi tersebut berbeda-beda (Yuliandri, 2017).

2.3 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, 2013).

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 : pemrosesan awal teks (*text preprocessing*), transformasi teks ke dalam bentuk antara (*text transformation/feature generation*) dan penemuan pola (*pattern discovery*).

2.3.1 Text Preprocessing

Tahapan awal dari *Text Mining* adalah *text preprocessing*. Tahap ini akan dilakukan penganalisaan dari segi sintaktik, dimana data teks yang ada diubah menjadi *toLowerCase* (semua karakter menjadi huruf kecil) dengan menggunakan teknik *Case Folding*, kemudian dilakukan *Tokenizing* yaitu proses penguraian deskripsi yang semula berupa kalimat-kalimat menjadi kata/*token* dan menghilangkan *delimiter* seperti titik, koma dan tanda baca lainnya (Manalu, 2014).

2.3.2 Text Transformation/Feature Generation

Pada tahap ini hasil yang diperoleh dari tahap *text preprocessing* akan 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 akan 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* ini maka beban kerja sistem dapat berkurang karena sistem hanya akan memperhitungkan kata-kata yang dianggap penting (Manalu,

2014). Dalam penelitian ini *stopword* yang digunakan menggunakan *stopword* dari Tala (Tala, 1999).

Selanjutnya adalah proses *stemming*. *Stemming* merupakan proses mereduksi kata ke dalam bentuk dasarnya. *Stemming* menurut Yanuar, Fiqih, dan Firdaus (2018) adalah proses menggali informasi dengan mengubah kata – kata menjadi bentuk aslinya. *Stemming* dalam Bahasa Indonesia berbeda dari Bahasa Inggris karena Bahasa Indonesia memiliki awalan (*prefixes*), akhiran (*suffixes*), imbuhan (*infixes*) maupun gabungan di antara kata dasarnya (*confixes*) (Nurdiansyah, Muharrom, & Firdaus, 2018). Menurut Tala (1999), *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 (*prefix*), kata akhir (*suffix*) atau keduanya (*confix*). *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 akan disimpan dengan berbagai macam bentuk sesuai imbuhanannya. Hal ini sangat berbeda apabila kita menerapkan proses *stemming* di dalamnya, sehingga lebih efisien (Tala, 1999).

2.3.3 Pattern Discovery

Tahap penemuan pola atau *pattern discovery* adalah tahap terpenting dari seluruh proses *text mining*. Tahap ini bertujuan menemukan pola atau pengetahuan dari keseluruhan teks. Ada 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 *corpus* (kumpulan kata) yang telah diklasifikasikan (Yanti, 2013).

Berikut adalah beberapa contoh *Text Mining* yang dilakukan pada *tweet*. Data *tweet* yang telah terkumpul tersebut, dipilah menjadi dua bagian yaitu 1) *Tweet* yang bersifat sentimen, dan 2) *Tweet* yang tidak bersifat sentimen misalnya iklan maupun berita. Dua jenis *Tweet* tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1 Contoh Data *Tweet* Sentimen dan Non Sentimen

Jenis	<i>Tweet</i>
Sentimen	“Kemarin pas turun dari Sumbing, sempet beli bubuk kopi arabika asli Temanggung. Pas diseduh rasanya enak banget \ud83d\ude0b\u2615\u2764\ufe0f”
	“hujan dan secangkir kopi arabika dataran tinggi papua..... ah nikmat bos”
Non Sentimen	“Bukan Kopi Luwak, Ini Dia Kopi Termahal di Indonesia https://t.co/aDiwVvvgAD \nCek berita lainnya:\u2026 https://t.co/6TIEEc9bHY ”
	“Ingin kopi luwak yang enak? Klik saja https://t.co/tOE7ZnzMVq #Coffee”

Sumber: (Nuritha, 2017)

Data yang disimpan adalah *tweet* yang bersifat sentimen. Data *tweet* sentimen tersebut kemudian dikelompokkan menjadi dua kategori yaitu 1) sentimen positif, dan 2) Sentimen negatif. Data sentimen ini merupakan data mentah yang akan melalui tahap praproses sebelum dilakukan proses klasifikasi sentimen.

Tabel 2.2 Contoh Data *Tweet* Sentimen Positif dan Negatif

Kategori	<i>Tweet</i>
Positif	"@dr_tompi Pisang goreng mantap kayaknya nih sambil serudup kopi Gayo khas Aceh diracik dgn Vietnam Drip ulalala makyussss
Negatif	"Selaluuu setelah minum kopi perutku sakit, apakah ini tanda lambungku gakuat? \ud83e\udd14 tapi aku cinta kopi bgt\ud83d\ude41\ud83d\ude41\ud83d\ude41"

Sumber: (Nuritha, 2017)

Selanjutnya data *tweet* yang telah dikumpulkan diproses melalui tahapan praproses yang meliputi *case folding*, *tokenizing*, *stopword removing*, dan *stemming*.

Tabel 2.3 Contoh Luaran Tahapan *Case Folding*

<i>Tweet</i>	Luaran
"Selaluuu setelah minum kopi perutku sakit, apakah ini tanda lambungku gakuat? \ud83e\udd14 tapi aku cinta kopi bgt\ud83d\ude41\ud83d\ude41\ud83d\ude41"	selaluuu setelah minum kopi perutku sakit apakah ini tanda lambungku gakuat \ud83e\udd14 tapi aku cinta kopi bgt\ud83d\ude41\ud83d\ude41\ud83d\ude41

Sumber: (Nuritha, 2017)

Tabel 2.4 Contoh Luaran Tahapan *Tokenizing*

<i>Tweet</i>	Luaran
selaluuu setelah minum kopi perutku sakit apakah ini tanda lambungku gakuat \ud83e\udd14 tapi aku cinta kopi bgt\ud83d\ude41\ud83d\ude41\ud83d\ude41	selaluuu ; setelah ; minum ; kopi ; perutku ; sakit ; apakah ; ini ; tanda ; lambungku ; gakuat ; \ud83e\udd14 ; tapi ; aku ; cinta ; kopi ; bgt ; \ud83d\ude41\ud83d\ude41\ud83d\ude41

Sumber: (Nuritha, 2017)

Tabel 2.5 Contoh Luaran Tahapan *Filtering Stopwords*

<i>Tweet</i>	Luaran
selaluuu ; setelah ; minum ; kopi ; perutku ; sakit ; apakah ; ini ; tanda ; lambungku ; gakuat ; \ud83e\udd14 ; tapi ; aku ; cinta ; kopi ; bgt ; \ud83d\ude41\ud83d\ude41\ud83d\ude41	selaluuu ; kopi ; perutku ; sakit; tanda ; lambungku ; gakuat ; cinta ; kopi ; bgt

Sumber: (Nuritha, 2017)

Tabel 2.6 Contoh Luaran Tahapan *Stemming*

<i>Tweet</i>	Luaran
selaluuu ; kopi ; perutku ; sakit; tanda ; lambungku ; gakuat ; cinta ; kopi ; bgt	selaluuu ; kopi ; perut ; sakit; tanda ; lambung ; gakuat ; cinta ; kopi ; bgt

Sumber: (Nuritha, 2017)

2.4 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 (Fink, Chou, Kopecky, & Llorens, 2011).

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining* yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu (Liu, 2012).

Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif, negatif atau netral (Dehaff, 2010). *Sentiment analysis* juga dapat menyatakan perasaan emosional sedih, gembira, atau marah.

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

a. *Coarse Grained Sentiment Analysis*

Pada jenis ini, sentimen analisis dilakukan pada tingkat dokumen. *Coarse Grained Sentiment Analysis* menganggap bahwa seluruh isi dokumen sebagai sebuah sentiment positif dan sentiment negatif (Fink, Chou, Kopecky, & Llorens, 2011).

b. *Fined Grained Sentiment Analysis*

Fined Grained Sentiment Analysis adalah analisis sentimen yang dilakukan pada tingkat kalimat. Pada jenis ini dikatakan bahwa setiap kalimat memiliki sentiment yang berbeda walaupun berada dalam satu dokumen (Fink, Chou, Kopecky, & Llorens, 2011).

Pada penelitian ini analisis sentimen dilakukan pada level dokumen, dengan asumsi bahwa status media sosial yang didapat merupakan bentuk dari sebuah dokumen yang memiliki sebuah sentimen positif, netral, atau negatif saja.

2.5 Naïve Bayes Classifier

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, Mulyanto, & Rohandi, 2014).

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

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

Sedangkan *Naïve Bayes* dengan fitur kontinu memiliki formula :

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

Keterangan :

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

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

σ = deviasi standar

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 akan menjadi data uji adalah status (yang telah diubah menjadi bentuk *term*) yang beredar di media sosial, sedangkan untuk *data training* menggunakan responden untuk mengklasifikan secara manual sebuah

status dan *corpus* (kumpulan kata) berbahasa Indonesia yang telah dipastikan sentimennya dan kategori kopi. Dengan menggunakan *NBC*, setiap term status akan direpresentasikan dengan pasangan atribut “X1, X2, X3,...Xn” dimana X1 adalah *n-gram* pertama, X2 adalah *n-gram* kedua, dan seterusnya. Sedangkan Y adalah himpunan kategori sentimen dan juga kategori. Klasifikasi akan ditentukan dengan mencari probabilitas tertinggi dari semua kategori status yang diujikan (*Ymap*) sehingga persamaannya seperti berikut:

$$V_{map} = \frac{P(X1, X2, X3, \dots, Xn | Yj) P(Yj)}{P(X1, X2, X3, \dots, Xn)} \dots \dots \dots (3)$$

Karena nilai P(X1,X2,X3,...,Xn) bernilai konstan untuk semua kategori (Yj) sehingga persamaan dapat ditulis

$$V_{map} = P(X1, X2, X3, \dots, Xn | Yj) P(Yj) \dots \dots \dots (4)$$

Persamaan di atas dapat disederhanakan menjadi:

$$V_{map} = \prod_{i=1}^n P(Xi | Yj) P(Yj) \dots \dots \dots (5)$$

Keterangan :

- Y_j : Kategori klasifikasi atau class
- $P(x_i | Y_j)$: Probabilitas xi pada kategori Yj
- $P(Y_j)$: Probabilitas prior dari kategori (Yj)

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

$$P(Y_j) = \frac{|Y_j|}{|Y_{total}|} \dots \dots \dots (6)$$

$$P(x_i | Y_j) = \frac{n_k + 1}{n + |kosakata|} \dots \dots \dots (7)$$

- 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

2.6 Algoritma Enhanced Confix-Stripping (ECS)

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 :

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

Pada algoritma ECS terdapat beberapa aturan sebagai berikut:

- a. Ada kombinasi (*confix*) 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 2.7 di bawah ini.

Tabel 2.7 Aturan Pemenggalan Kata Dalam 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!=’r’
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 d j s z}...	men- {c d j 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}...
26	pem{rV V}...	pe-m {rV V}... pe-p {rV V}...
27	pen{c d j z}...	pen- {c d j z}...
28	penV...	pe-nV... pe-tV...
29	pengC...	peng-C...
30	pengV...	peng-V... peng-kV... (pengV-...jika V=’e’)
31	penyV...	peny-sV...
32	peIV...	pe-IV...kecuali pelajar yang menghasilkan “ajar”
33	peCerV...	per-erV...dimana C!={r w y l m n}
34	peCP...	pe-CP...dimana C!={r w y l m n} dan P!=’er’
35	terC1erC2...	ter-C1erC2...dimana C1!=’r’

Aturan	Format Kata	Pemenggalan
36	peC1erC2...	pe-C1erC2...dimana C1!={r w y l m n}

Sumber : (Tahitoe & Purwitasari, 2010)

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

- a. 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
- b. Melakukan proses *rulePrecedence* dimana proses ini akan 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)
- c. Hilangkan *inflectional suffixes*. Dimulai dari *particle* (P) kemudian *possessive pronoun* (PP).
- d. Hilangkan *derivation suffixes*.
- e. Hilangkan *derivation prefixes*.
 - 1) Langkah 5 berhenti jika :
 - a) Terjadi kombinasi awalan dan akhiran yang terlarang
 - b) Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya
 - c) Tiga awalan telah dihilangkan
 - 2) Identifikasi tipe awalan dan akhiran. Awalan terdiri dari 2 tipe:
 - a) Tidak bermorfologi, maka awalannya langsung dihilangkan
 - b) Bermorfologi, gunakan aturan pada tabel untuk mendapatkan hasil pemenggalan yang tepat

- 3) Cari kata yang telah dihilangkan awalnya ini dalam kamus kata dasar. Apabila tidak ditemukan maka ulangi langkah 5, apabila ditemukan maka proses dihentikan.
- 4) 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 1, menulis akan 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".

Jika semua langkah yang dilakukan tidak menemukan kata dasar, maka inputan awal dijadikan sebagai kata dasar.

2.7 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 masuk 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 tinggi.

Twitter menyediakan berbagai jenis *API Search* sesuai kebutuhan mulai dari versi gratis hingga berbayar. Tujuan dari masing-masing API digunakan sesuai kapasitas penggunaannya. Berikut adalah tabel perbandingan fitur yang terdapat pada masing-masing API.

Tabel 2.8 API Twitter

Kategori	Nama Produk	Garis Waktu	Kapabilitas Query	Titik Akhir	Kelengkapan Data
<i>Standard</i>	<i>Standard Search API</i>	7 hari terakhir	Operasi standar	Tidak tersedia	Terbatas
<i>Premium</i>	<i>Search Tweets: 30-day endpoint</i>	30 hari terakhir	Operasi penuh	Tersedia	Penuh
<i>Premium</i>	<i>Search Tweets: Full-archive endpoint</i>	<i>Tweet</i> awal 2006	Operasi penuh	Tersedia	Penuh
<i>Enterprise</i>	<i>30-day Search API</i>	30 hari terakhir	Operasi penuh	Sudah termasuk paket	Penuh
<i>Enterprise</i>	<i>Full-archive Search API</i>	<i>Tweet</i> awal 2006	Operasi penuh	Sudah termasuk paket	Penuh

Sumber: (Twitter, 2018)

2.8 Uji Performansi

Pengujian klasifikasi diukur dengan uji performansi yang meliputi *precision*, *recall* dan *f1score* (Ponilan, Herdiani, & Selviandro, 2016). Untuk mempermudah pemahaman mengenai uji performansi dapat dilihat pada Tabel 2.9.

Tabel 2.9 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 dipredeksi *yes*, dan ternyata faktanya *yes*
- TN(*True Negative*) adalah kelas yang dipredeksi *no*, dan ternyata faktanya *no*
- FP(*False Positive*) adalah kelas yang dipredeksi *yes*, dan ternyata faktanya *no*
- FN(*False Negative*) adalah kelas yang dipredeksi *no*, dan ternyata faktanya *yes*

Precision adalah rasio jumlah ketepatan prediksi suatu kelas terhadap jumlah total prediksi yang diklasifikasikan dalam kelas tersebut. *Precision* diartikan sebagai ketepatan atau kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. Rumus mencari *precision* dapat ditulis sebagai berikut

$$precision (P) = \frac{TP}{TP + FP} \dots \dots \dots (8)$$

Recall adalah rasio jumlah ketepatan prediksi suatu kelas terhadap jumlah total fakta yang diklasifikasikan dalam kelas tersebut. *Recall* diartikan sebagai tingkat keberhasilan sistem dalam menemukan kembali informasi yang relevan. Semakin rendah nilai *recall* maka efektivitas sistem dinyatakan kurang baik. Rumus mencari *recall* dapat ditulis sebagai berikut

$$recall (R) = \frac{TP}{TP + FN} \dots \dots \dots (9)$$

Untuk menggabungkan keduanya maka digunakan perhitungan *f1score*. *F1score* merupakan kombinasi rata-rata *harmonic* antara *precision* dengan *recall* yang berbanding lurus dengan nilai keduanya. Rumus mencari *f1score* dapat ditulis sebagai berikut

$$f1_{score} = \frac{2*precision*recall}{precision+recall} \dots \dots \dots (10)$$

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahapan sistematis yang dilakukan untuk menganalisa data guna menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang jenis, tempat beserta waktu penelitian dan tahapan dari penelitian.

3.1 Jenis Penelitian

Metode penelitian yang dilakukan pada penelitian ini menggunakan metode penelitian kuantitatif. Penelitian kuantitatif dilakukan pada tahap pengumpulan data dan proses klasifikasi. Penelitian ini menggunakan metode kuantitatif karena dalam penelitian ini menerapkan serta mengkaji teori serta rumus perhitungan yang sudah ada sebelumnya.

3.2 Tempat dan Waktu Penelitian

Tempat pelaksanaan penelitian adalah Laboratorium Basis Data Fakultas Ilmu Komputer Universitas Jember. Waktu penelitian dilakukan selama 4 bulan, dimulai pada bulan Maret 2018 sampai bulan Juni 2018.

3.3 Tahapan Penelitian

Tahapan penelitian merupakan urutan langkah penelitian yang dilakukan. Gambaran tahapan penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.3.1 Studi Literatur

Tahap awal dalam penelitian ini adalah studi literatur yang bertujuan untuk penyusunan dasar teori yang akan digunakan selama penelitian. Penelitian menggunakan referensi dalam buku, jurnal baik nasional maupun internasional, karya ilmiah, artikel dan penelitian – penelitian yang sudah dilakukan sebelumnya serta berbagai sumber lain yang mendukung dalam penelitian ini.

3.3.2 Pengumpulan Data

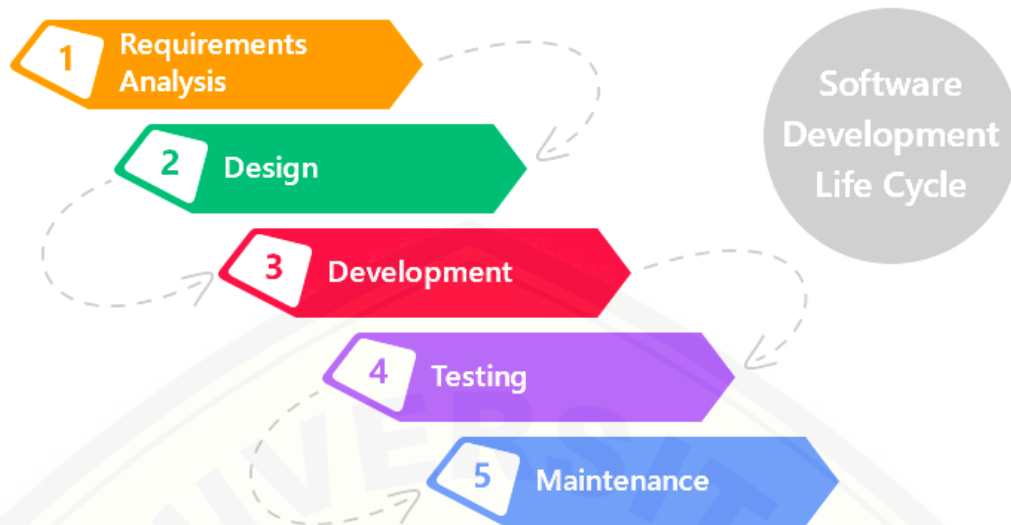
Tahap selanjutnya adalah pengumpulan data. Data yang diperoleh untuk melakukan penelitian didapat langsung dari cuitan dan status di media sosial *Twitter* berbahasa Indonesia baku dan berada di wilayah Indonesia sebagai obyek penelitian berdasarkan kata – kata kunci yang sudah ditetapkan sebelumnya.

3.3.3 Perancangan Model Klasifikasi

Klasifikasi sentimen dan kategori menggunakan *Naïve Bayes Classifier*. *Naïve Bayes Classifier* dipakai karena beberapa penelitian terdahulu telah menggunakan *Naïve Bayes Classifier* dalam melakukan klasifikasi sentimen. Model klasifikasi didapatkan dengan mengambil *dataset* yang berupa *data training* dan *data testing* dengan formulasi perbandingan tertentu untuk mendapatkan tingkat akurasi terbaiknya.

3.3.4 Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak 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 (Pressman, 2010). Alur metode pengembangan software dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 SDLC Waterfall

Sumber: (Pressman, 2010)

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahapan awal dalam perancangan dan pengembangan sistem ini adalah analisis kebutuhan-kebutuhan tentang bagaimana sistem identifikasi persepsi konsumen kopi 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.

2. Desain Sistem (*Design*)

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

- a) *Business Process* digunakan untuk mendefinisikan aktifitas dan proses.
- b) *Use Case Diagram* digunakan untuk mendefinisikan fungsional sistem.
- c) *Scenario* digunakan untuk menjelaskan fitur sistem.
- d) *Sequence diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim antar obyek juga interaksi antar obyek.

- e) *Class Diagram* digunakan untuk menggambar struktur statis *class* dalam sistem.
- f) *Entity Realtionship Diagram* digunakan untuk menunjukkan relasi antar obyek.

3. Implementasi (*Development*)

Tahapan implementasi adalah tahap yang akan mengkonversi desain sistem yang telah dibuat ke dalam bentuk kode program. Sistem yang akan dibangun ditulis dengan Bahasa pemrograman *PHP (Page Hypertext Pre-processor)* dan menggunakan *Laravel* sebagai *framework* pengembangan aplikasi. Sistem ini menggunakan *local server* dan *database* yang terdapat pada aplikasi *XAMPP*. Selain itu, untuk berkomunikasi dengan data dalam media sosial *twitter*, dalam penelitian ini menggunakan *Twitter Search API* dan *Twitter Streaming API* untuk *crawling data tweet*.

4. Pengujian (*Testing*)

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

a) Uji Performansi

Pengujian klasifikasi diukur dengan uji performansi yang meliputi *precision*, *recall* dan *f-measure*. (Rahayu Ponilan et al., 2016) Untuk mempermudah pemahaman mengenai uji perfomansi dapat dilihat pada Tabel 9.

Tabel 9 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

$$precision (P) = \frac{TP}{TP + FP} \dots \dots \dots (8)$$

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

$$recall (P) = \frac{TP}{TP + FN} \dots \dots \dots (9)$$

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

$$f1_{score} = \frac{2 * precision * recall}{precision + recall} \dots \dots \dots (10)$$

b) *Black box testing*

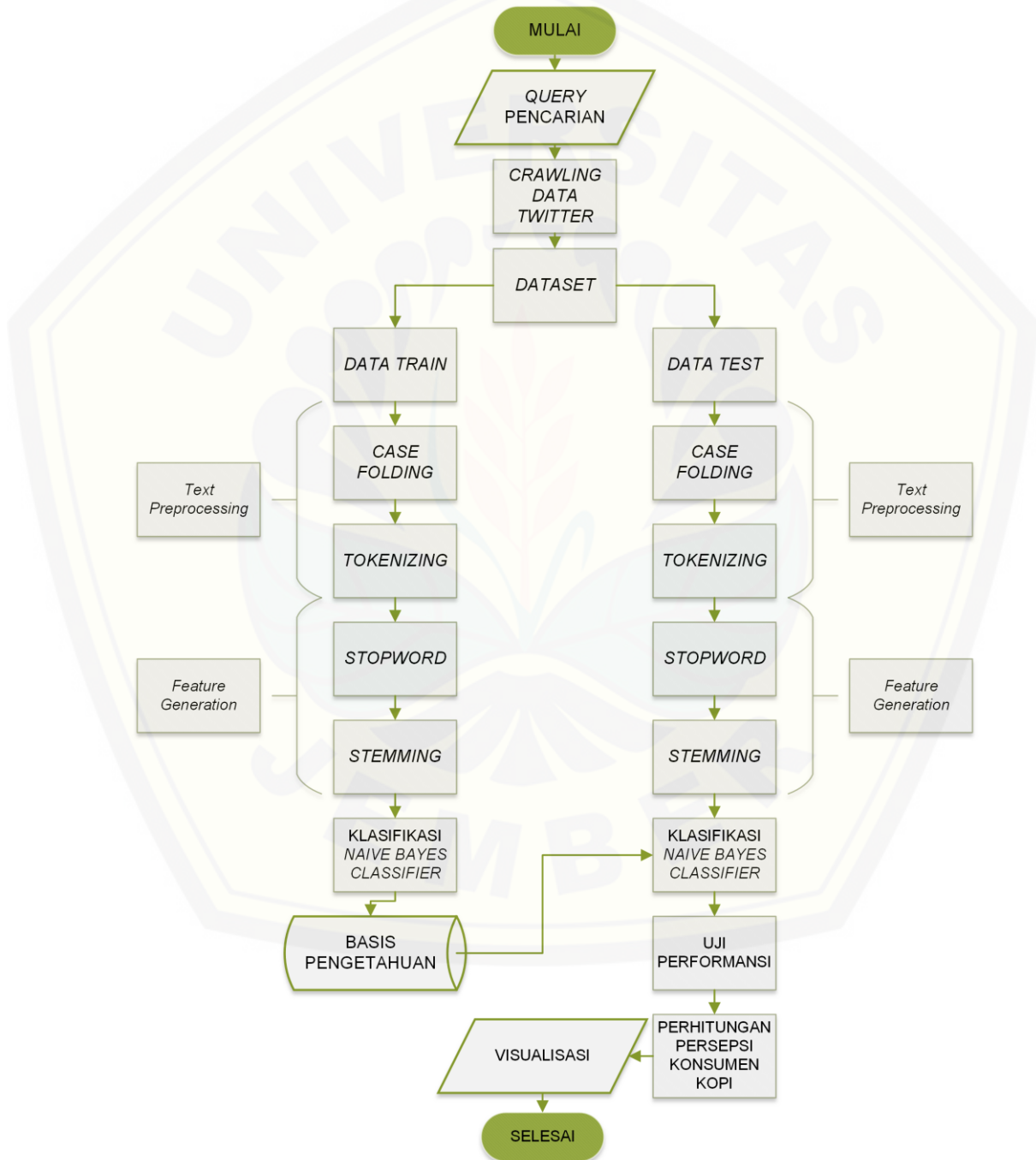
Black box testing melibatkan pengguna/*User*, dimana hanya memperhatikan fungsionalitas yang berkaitan dengan masukan/keluaran (*I/O*) apakah sesuai dengan sistem yang dijalankan. Pengujian dilakukan oleh beberapa pengguna sistem yang diambil secara acak, menggunakan kuesioner.

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 Tahapan Algoritma Sistem

Sistem pengukur persepsi konsumen kopi yang dibangun adalah sebuah sistem berbasis web yang dapat mengukur persepsi konsumen kopi suatu daerah dengan memanfaatkan analisis sentiment pada media sosial. Algoritma yang diterapkan dalam sistem tersebut dapat dilihat sebagai pada Gambar 3.3.



Gambar 3.3 Flowchart Alur Sistem

Tabel 3.1 Penjelasan Alur Sistem

No	Tahap	Input	Proses	Output
1.	<i>Query Pencarian</i>	Kata kunci dan jumlah <i>tweet</i>	Pada tahap ini <i>query</i> pencarian dimasukkan, <i>query</i> kata “kopi luwak” untuk mencari <i>dataset</i> kopi luwak	<i>Query</i>
2.	<i>Crawling Data</i>	<i>Query, Sosial Media API</i>	<i>Crawling data</i> dilakukan dengan menggunakan sosial media <i>API</i> dan dicocokkan dengan <i>query</i> yang telah diinputkan sebelumnya	<i>Dataset</i>
3.	<i>Text Preprocessing</i>	<i>Dataset</i>	<ol style="list-style-type: none"> 1. Pengambilan satu <i>dataset/konten/status</i>. Pengambilan ini nantinya akan dilakukan pengulangan, hingga seluruh <i>dataset</i> berhasil dianalisa 2. <i>Dataset</i> dipisah menjadi <i>data train</i> dan <i>data test</i> 3. <i>Case folding</i>. Pengubahan <i>dataset</i> yang ada menjadi <i>lowercase</i> 4. <i>Tokenizing</i>. Penghilangan tanda baca titik, koma, dan tanda baca lainnya yang tidak diperlukan 	<i>Term</i> yang telah di <i>case folding</i> dan dihilangkan tanda bacanya
4.	<i>Feature Generation</i>	<i>Term</i> yang telah di <i>case folding</i> dan dihilangkan tanda bacanya (<i>tokenizing</i>)	1. <i>Stopping</i> . Penghilangan <i>stopword</i> atau kata-kata yang tidak mengandung sentimen seperti “di”, “pada”, “oleh”. Daftar <i>stopword</i> akan	<i>Term</i> yang telah dihilangkan <i>stopwordnya</i> dan telah di <i>stemming</i>

No	Tahap	Input	Proses	Output
			diinputkan dalam <i>stoplist</i>	
			2. <i>Stemming</i> . Penghilangan imbuhan, baik <i>prefix</i> (kata depan), <i>suffix</i> (kata akhir), <i>konfix</i> . <i>Stemming</i> dilakukan agar tidak terjadi duplikasi dalam DB. Algoritma <i>stemming</i> yang dipakai adalah algoritma <i>enhanced confix-stripping (ECS)</i>	
6.	Klasifikasi Sentimen dan Kategori	<i>Term</i> yang telah dilakukan <i>text preprocessing</i> dan <i>feature generation</i>	Penghitungan klasifikasi sentimen dan kategori dari <i>term</i> yang telah terbentuk. Klasifikasi menggunakan <i>Naïve Bayes Classifier</i> .	Hasil klasifikasi sentimen dan kategori
	Uji Performansi	-	Pengujian klasifikasi diukur dengan uji performansi yang meliputi <i>precision</i> , <i>recall</i> dan <i>f1score</i> berdasarkan data train dan data test	Hasil uji performansi
	Perhitungan Persepsi konsumen kopi	Hasil klasifikasi sentimen dan klasifikasi kategori	Tahap ini dilakukan penghitungan persepsi konsumen menggunakan persentase sentimen positif yang terjadi dibandingkan dengan jumlah keseluruhan sentimen. Kemudian mengkategorikannya sesuai kategori yang telah diklasifikasikan.	Persepsi konsumen kopi
	Visualisasi	Identifikasi persepsi konsumen kopi, hasil	Semua data dan penghitungan yang telah dilakukan	Visualisasi data dan statistik

No	Tahap	Input	Proses	Output
		klasifikasi kategori dan sentimen	divisualisasikan bentuk grafik.	dalam



BAB 4. PERANCANGAN SISTEM

Bab ini menjelaskan tentang perancangan sistem pengukur persepsi konsumen 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 identifikasi persepsi konsumen kopi ini merupakan sebuah sistem informasi berbasis website yang mampu mengukur persepsi konsumen kopi menggunakan sentimen di media sosial dengan metode *Naïve 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*. Persepsi konsumen kopi 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 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 *Naïve Bayes Classifier*, penerapan *text mining*, dan persepsi konsumen kopi, sehingga didapatkan kebutuhan fungsional dan non fungsional yang sesuai. Kebutuhan tersebut dapat

digunakan dalam melakukan penghitungan persepsi konsumen 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

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari aplikasi ini yaitu:

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*.
3. Sistem dapat mengelola data *user* yang meliputi *view*, *insert*, *edit* dan *delete*.
4. Sistem dapat melakukan *crawling* data *twitter*. Kebutuhan ini dimunculkan agar sistem dapat memiliki dataset berupa *data training* dan *data testing* 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 *Naïve Bayes Classifier*, dimana *NBC* merupakan metode yang dipakai untuk melakukan pengklasifikasian yang dibutuhkan untuk mendapatkan persepsi konsumen kopi.
7. Sistem dapat menampilkan persepsi konsumen kopi sesuai dengan hasil klasifikasi. Kebutuhan ini muncul sebagai tujuan dari sistem yaitu menghasilkan hasil klasifikasi persepsi konsumen kopi.

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”.
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 jenis kopi yang dapat dihitung, 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.

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*.
2. Sistem menggunakan *API Search Twitter* untuk *crawling* data.
3. Sistem berbasis website. Website dipakai agar sistem dapat diakses dimanapun dan informasi persepsi konsumen kopi dapat dengan mudah diterima.
4. Tampilan website yang *responsive*. Kebutuhan ini bertujuan untuk memudahkan pengguna dalam mengakses aplikasi di semua komputer dengan berbagai jenis resolusi layar.

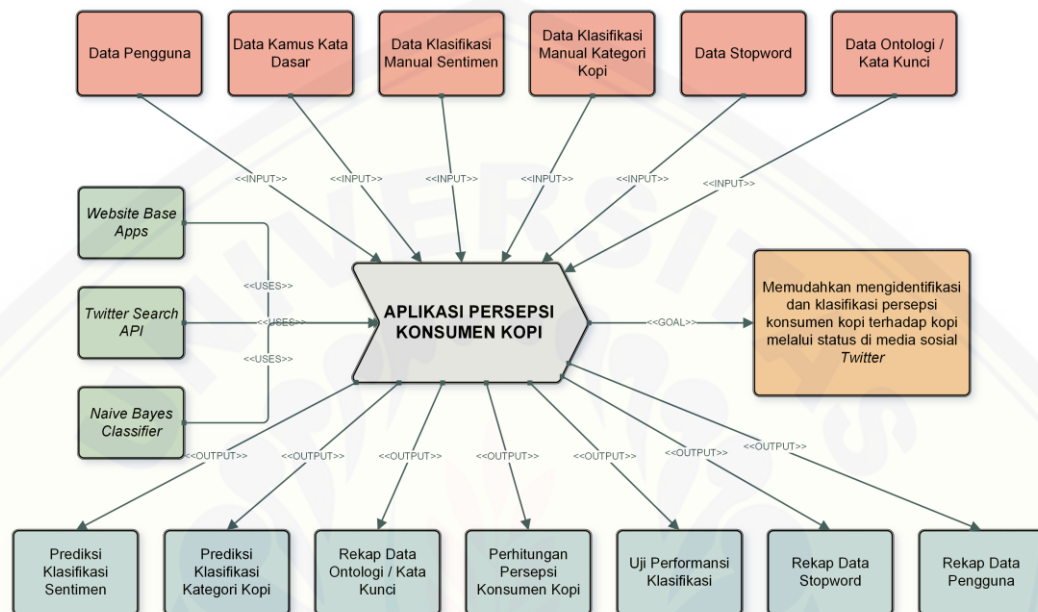
4.3 Desain Sistem

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

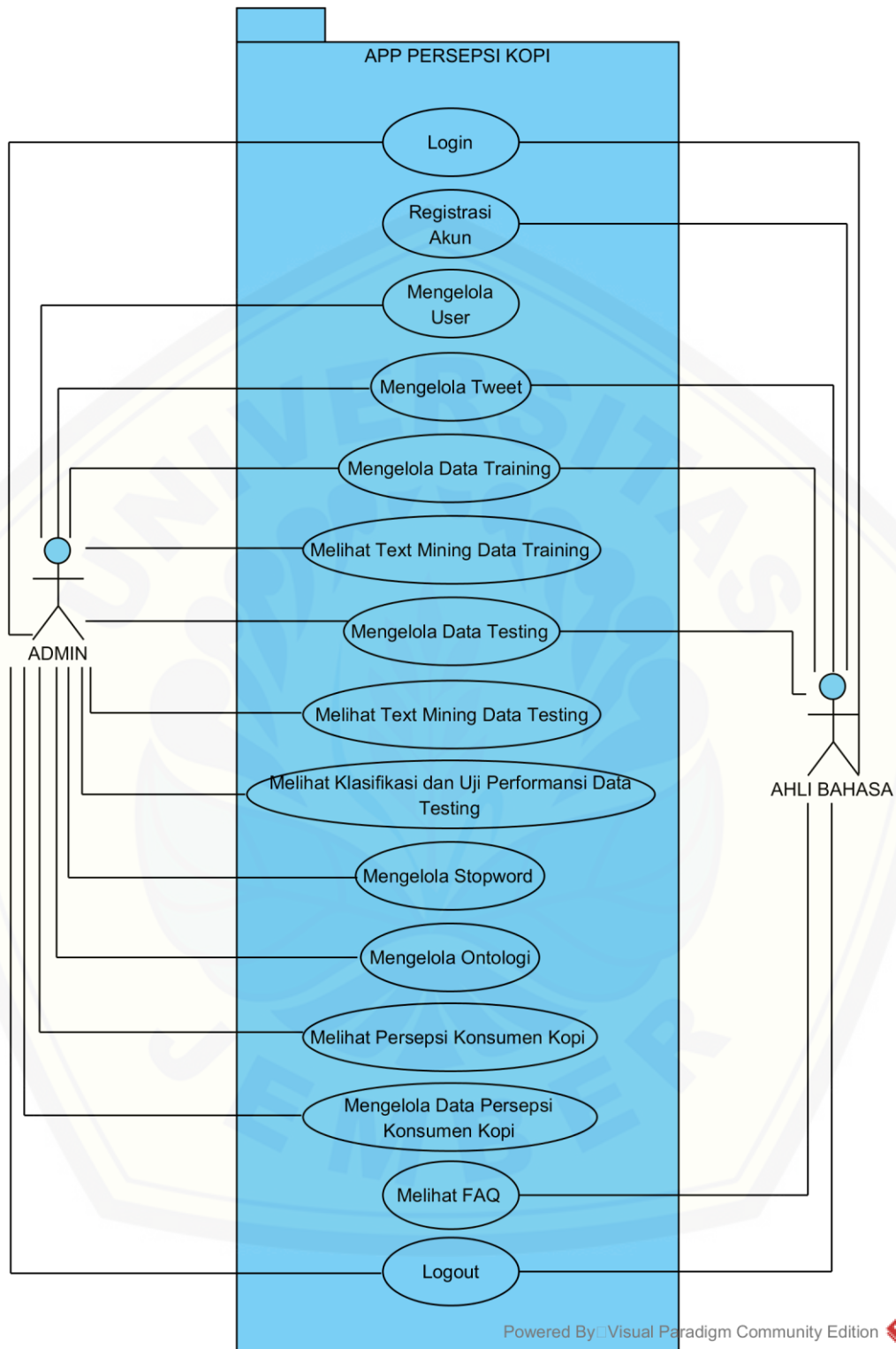
menjadi masukan sistem (*input*), data masukan yang telah diolah sehingga menghasilkan data keluaran (*output*), media yang digunakan (*uses*), serta tujuan yang ingin dicapai (*goal*). *Business process* pada sistem ini dapat dilihat pada gambar 4.1.



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 persepsi konsumen kopi menggunakan analisis sentimen di media sosial. *Use case* sistem pengukur persepsi konsumen kopi 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 persepsi konsumen kopi 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 data training</i> , mengelola <i>data testing</i> , melihat <i>Text Mining data testing</i> , melihat klasifikasi & uji performansi <i>data testing</i> , melihat persepsi konsumen kopi <i>data testing</i> , mengelola <i>stopword</i> , mengelola ontologi, mengelola data persepsi konsumen kopi .
2.	Ahli Bahasa	Aktor ahli bahasa pada sistem ini memiliki peran untuk dapat mengelola data <i>training</i> , mengelola <i>data 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 persepsi konsumen kopi 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.

No.	Usecase	Deskripsi
2.	Registerasi Akun	Menggambarkan proses pendaftaran akun <i>user</i> baru
3.	Mengelola <i>User</i>	Menggambarkan proses pengelolaan data <i>user</i> meliputi melihat melihat, mengubah dan menghapus <i>user</i> .
4.	Mengelola <i>Tweet</i>	Menggambarkan proses pengelolaan data <i>tweet</i> meliputi melihat melihat, mengubah dan menghapus <i>tweet</i> .
5.	Mengelola <i>Data Training</i>	Menggambarkan proses pengelolaan <i>data training</i> meliputi melihat menambahkan, melihat, mengubah dan menghapus <i>data training</i> .
6.	Melihat <i>Mining 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, mengubah dan menghapus <i>data testing</i> .
8.	Melihat <i>Mining 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>Persepsi konsumen kopi Data testing</i>	Menggambarkan proses untuk melihat persepsi konsumen kopi <i>data testing</i> .
11.	Mengelola <i>Stopword</i>	Menggambarkan proses pengelolaan data <i>stopword</i> meliputi melihat menambahkan, melihat, mengubah dan menghapus <i>stopword</i> .
12.	Mengelola <i>Ontologi</i>	Menggambarkan proses pengelolaan data ontologi meliputi melihat menambahkan, melihat, mengubah dan menghapus ontologi.
13.	Mengelola <i>Data Persepsi konsumen kopi</i>	Menggambarkan proses pengelolaan data persepsi konsumen kopi meliputi melihat menambahkan, melihat, mengubah dan menghapus data persepsi konsumen kopi.
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 persepsi konsumen kopi menggunakan sentimen di media sosial adalah sebagai berikut:

a. 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.

b. 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.

c. Skenario *User*

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

d. 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.

e. 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, mengubah, dan menghapus data training dengan memilih menu *training* tabel. Skenario mengelola data training dapat dilihat pada lampiran A.

f. 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.

g. 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, mengubah dan menghapus *data testing* dengan memilih menu *data testing* tabel. Skenario mengelola *data testing* dapat dilihat pada lampiran A.

h. 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.

i. 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 4.4.

Tabel 4.3 Melihat Klasifikasi & Uji Performansi *Data Testing*

No. Usecase	UC09
Nama Usecase	Melihat Klasifikasi & Uji Performansi <i>Data testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Testing</i> Klasifikasi
PostKondisi	Admin berhasil melihat Klasifikasi & Uji Performansi <i>Data testing</i>

Aliran normal**Lihat data klasifikasi**

1. Klik menu <i>Testing</i> Klasifikasi	2. Menampilkan halaman <i>data 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 <i>Persepsi konsumen</i> Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Tweet</i> (varchar: 160) - Probabilitas Positif (double: 10) - Probabilitas Negatif (double: 10) - Probabilitas Netral (double: 10) - Klasifikasi Sentimen (varchar: 30)
---	--

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> - Tabs Klasifikasi Sentimen - Tabs Uji Performansi Sentimen - Tabs Uji Performansi Kategori - Tabs <i>Persepsi Konsumen</i> Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Tweet</i> (varchar: 160) - Probabilitas (varchar: 30)
-----------------------------------	---

Aliran normal**Lihat data Uji Performansi Sentimen**

No. Usecase	UC09
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 <i>Persepsi Konsumen</i> <p>Tabel <i>Average</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai Rata-rata (double: 10) <p>Tabel <i>Recall</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Recall</i> (double: 10) <p>Tabel <i>Precision</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Precision</i> (double: 10) <p>Tabel <i>F1 score</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>F1 score</i> (double: 10) <p>Tabel <i>Support</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Support</i> (double: 10)
Aliran normal	
Lihat data Uji Performansi Kategori	
1. Klik Tabs Uji Performansi Kategori	<p>2. Menampilkan halaman <i>data 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 <i>Persepsi konsumen</i> <p>Tabel <i>Average</i> yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100)

No. Usecase	UC09
	<ul style="list-style-type: none"> - Nilai Rata-rata (double: 10) Tabel <i>Recall</i> yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Recall</i> (double: 10) Tabel <i>Precision</i> yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Precision</i> (double: 10) Tabel <i>F1 score</i> yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>F1 score</i> (double: 10) Tabel <i>Support</i> yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Class</i> (varchar: 100) - Nilai <i>Support</i> (double: 10)

j. Skenario Mengelola *Stopword*

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

k. Skenario Mengelola Ontologi

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

l. Skenario Melihat Persepsi Konsumen *Data Testing*

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

m. Skenario Mengelola Data Persepsi Konsumen

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

n. 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.

o. 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.

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:

a. *Sequence Diagram Login*

Interaksi antar kelas dan *function* pada kelas *login* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

b. *Sequence Diagram Registrasi Akun*

Interaksi antar kelas dan *function* pada kelas registrasi akun yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

c. *Sequence Diagram Mengelola User*

Interaksi antar kelas dan *function* pada kelas mengelola *user* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

d. *Sequence Diagram Mengelola Tweet*

Interaksi antar kelas dan *function* pada kelas mengelola *tweet* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

e. *Sequence Diagram Mengelola Data Training*

Interaksi antar kelas dan *function* pada kelas mengelola *data training* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

f. *Sequence Diagram Melihat Text Mining Data Training*

Interaksi antar kelas dan *function* pada kelas melihat *text mining data training* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

g. *Sequence Diagram Mengelola Data Testing*

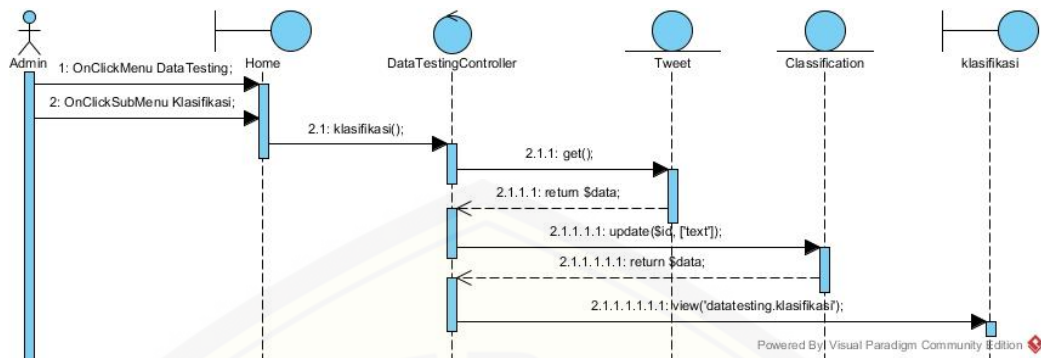
Interaksi antar kelas dan *function* pada kelas mengelola *data testing* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

h. *Sequence Diagram Melihat Text Mining Data Testing*

Interaksi antar kelas dan *function* pada kelas melihat *text mining data testing* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

i. *Sequence Diagram Melihat Klasifikasi dan Uji Performansi Data*

Interaksi antar kelas dan *function* pada kelas melihat klasifikasi dan uji performansi data yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.3.



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

j. *Sequence Diagram* Mengelola *Stopword*

Interaksi antar kelas dan *function* pada kelas *stopword* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

k. *Sequence Diagram* Mengelola Ontologi

Interaksi antar kelas dan *function* pada kelas ontologi yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

l. *Sequence Diagram* Melihat Persepsi Konsumen *Data Testing*

Interaksi antar kelas dan *function* pada kelas persepsi konsumen *data testing* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

m. *Sequence Diagram* Mengelola Data Persepsi Konsumen

Interaksi antar kelas dan *function* pada kelas persepsi konsumen yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

n. *Sequence Diagram* Melihat *FAQ*

Interaksi antar kelas dan *function* pada kelas *FAQ* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

o. *Sequence Diagram* *Logout*

Interaksi antar kelas dan *function* pada kelas *logout* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

4.3.5 Activity Diagram

Activity diagram merupakan diagram yang memodelkan proses-proses yang terjadi pada sistem dan merupakan pemodelan dari skenario yang sudah ditetapkan sebelumnya. *Activity diagram* dalam sistem ini antara lain sebagai berikut:

a. *Activity Diagram Login*

Activity diagram login dapat dilihat pada lampiran C.

b. *Activity Diagram Registrasi Akun*

Activity diagram registrasi akun dapat dilihat pada lampiran C.

c. *Activity Diagram Mengelola User*

Activity diagram mengelola user dapat dilihat pada lampiran C.

d. *Activity Diagram Mengelola Tweet*

Activity diagram mengelola tweet dapat dilihat pada lampiran C.

e. *Activity Diagram Mengelola Data Training*

Activity diagram mengelola data training dapat dilihat pada lampiran C.

f. *Activity Diagram Melihat Text Mining Data Training*

Activity diagram melihat text mining data training dapat dilihat pada lampiran C.

g. *Activity Diagram Mengelola Data Testing*

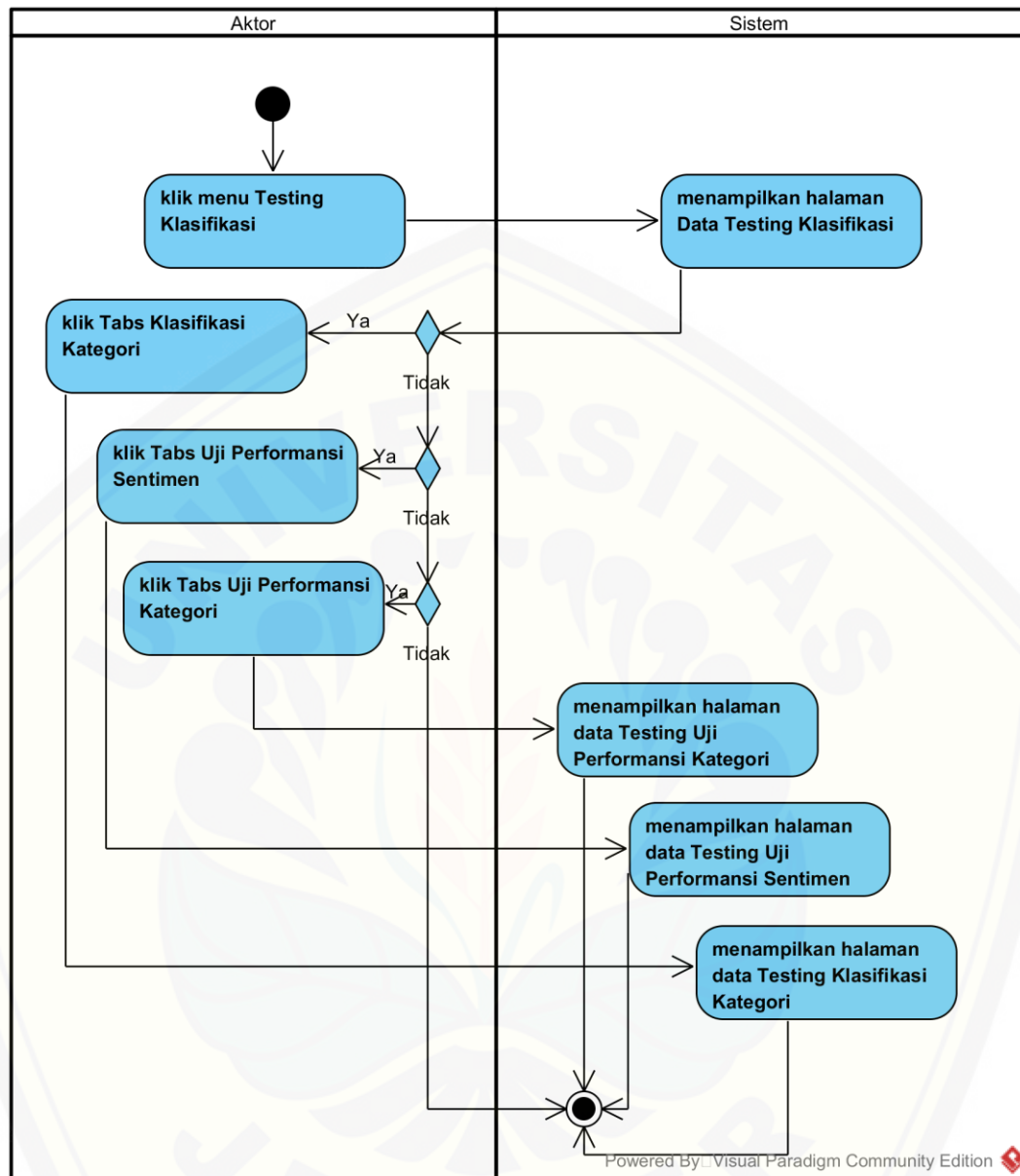
Activity diagram mengelola data testing dapat dilihat pada lampiran C.

h. *Activity Diagram Melihat Text Mining Data Testing*

Activity diagram melihat Text Mining Data Testing dapat dilihat pada lampiran C.

i. *Activity Diagram Melihat Klasifikasi dan Uji Performansi Data*

Activity diagram melihat klasifikasi dan uji performansi data dapat dilihat pada gambar 4.4.



Gambar 4.4 Activity diagram Melihat Klasifikasi dan Uji Performansi Data Testing

j. Activity Diagram Mengelola Stopword

Activity diagram mengelola stopword dapat dilihat pada lampiran C.

k. Activity Diagram Mengelola Ontologi

Activity diagram mengelola ontologi dapat dilihat pada lampiran C.

l. *Activity Diagram* Melihat Persepsi Konsumen *Data Testing*

Activity diagram melihat persepsi konsumen *data testing* dapat dilihat pada lampiran C.

m. *Activity Diagram* Mengelola Data Persepsi Konsumen

Activity diagram mengelola data persepsi konsumen kopi harian dapat dilihat pada lampiran C.

n. *Activity Diagram* Melihat FAQ

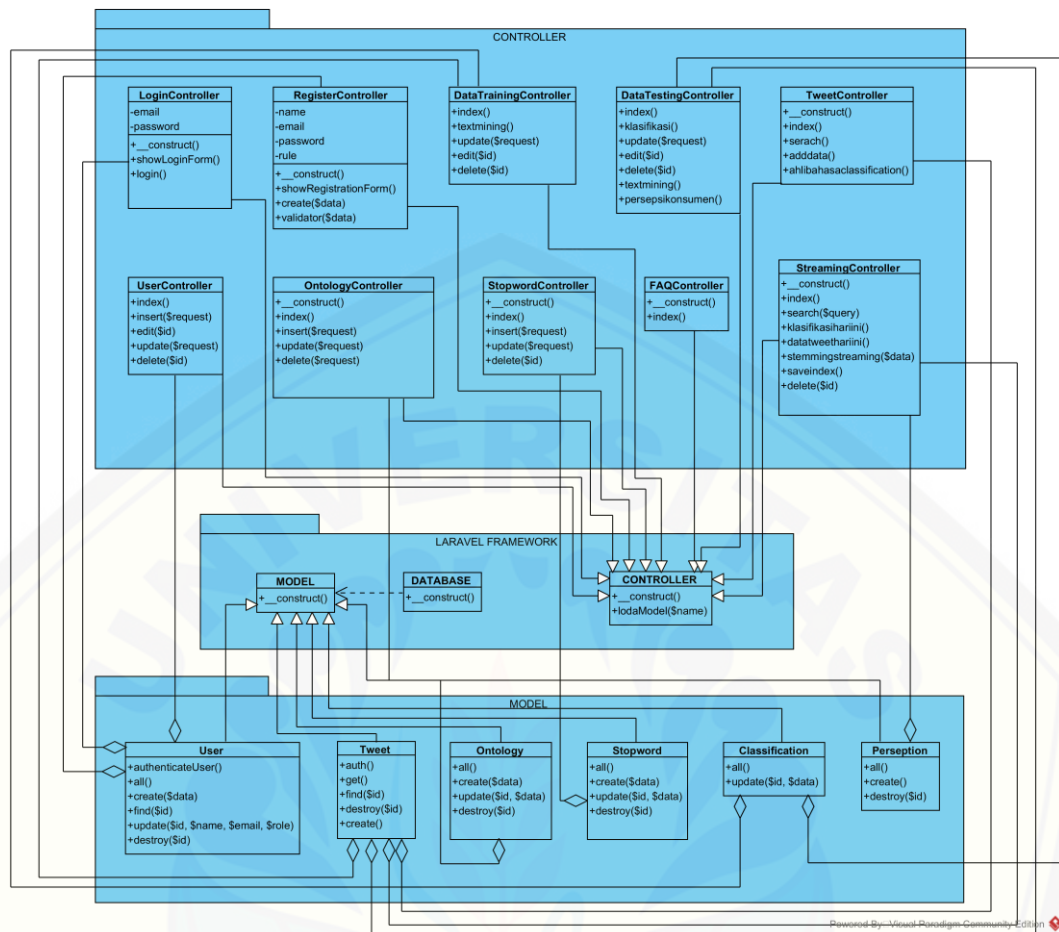
Activity diagram FAQ dapat dilihat pada lampiran C.

o. *Activity Diagram Logout*

Activity diagram logout dapat dilihat pada lampiran C.

4.3.6 *Class Diagram*

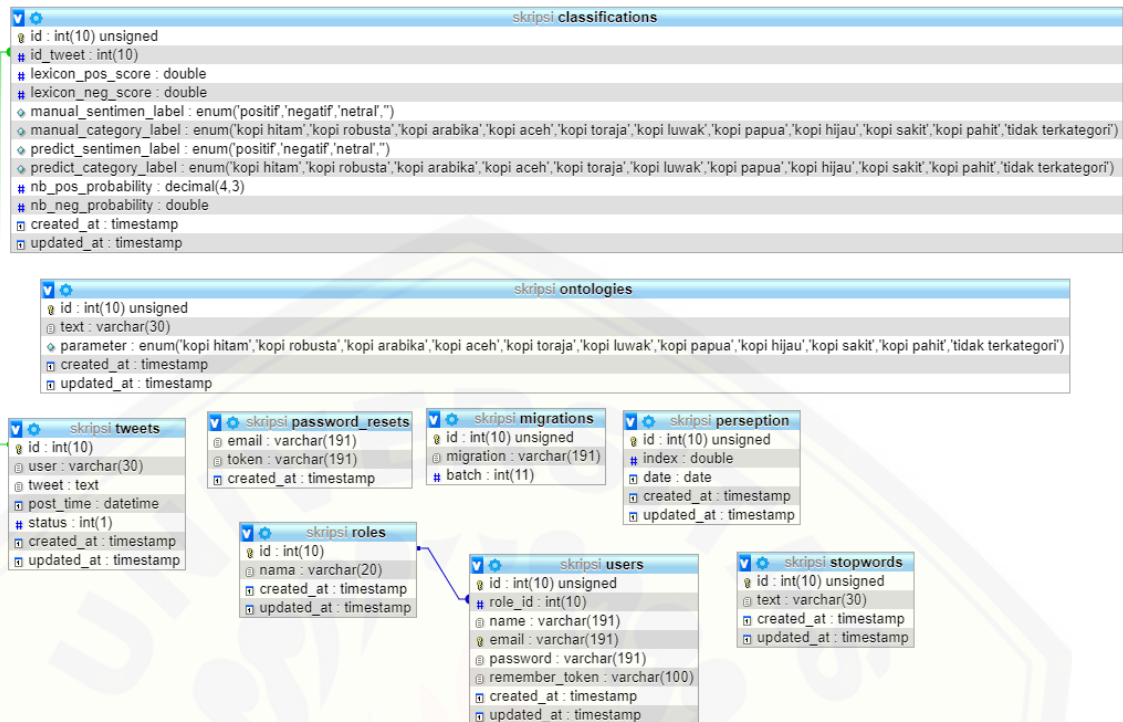
Class diagram merupakan diagram untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya digunakan. *Class diagram* yang digunakan sistem identifikasi persepsi konsumen menggunakan analisis sentimen di media sosial dapat dilihat pada Gambar 4.2.



Gambar 4.5 Class Diagram

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada aplikasi identifikasi persepsi konsumen kopi dengan analisis sentimen ini menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi ditampilkan pada gambar 4.4.



Gambar 4.6 Entity Relationship Diagram (ERD)

4.4 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *blackbox*.

4.5.2 Pengujian 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 dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian *Black Box*

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
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	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>login</i> namun E-mail atau password kosong	Menampilkan <i>span</i> “harap isi bidang ini”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>login</i> namun E-mail kosong	Menampilkan <i>span</i> “harap isi bidang ini”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>login</i> namun password kosong	Menampilkan <i>span</i> “harap isi bidang ini”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>login</i> namun E-mail atau password salah	Menampilkan <i>span</i> “ <i>These credentials do not match our records.</i> ”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Registrasi Akun	Klik tombol daftar sebagai ahli bahasa	Menampilkan halaman yang berisi <i>form</i> registrasi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Masukkan <i>Username, E-mail, Password</i> dan Konfirmasi <i>Password</i> pada halaman	Registrasi berhasil dan sudah terdaftar sebagai ahli bahasa	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
		registrasi lalu klik sign up			
		Klik Sign Up namun <i>form</i> ada yang kosong	Menampilkan span”harap isi bidang ini”	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Mengelola <i>User</i>	Klik menu data <i>user</i>	Menampilkan halaman data <i>user</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Tombol Edit pada halaman data <i>user</i>	Menampilkan halaman <i>form</i> edit data <i>user</i> , tombol <i>save</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		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>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Hapus pada halaman data <i>user</i>	Data di hapus dari <i>database</i> dan menampilkan halaman data <i>user</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
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>	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisikan query dan jumlah <i>tweet</i> lalu klik tombol search	Menampilkan <i>tweet</i> yang sedang dicari pada kolom query	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
		Centang <i>tweet</i> yang diinginkan lalu Klik Tombol Add -> Jadikan Data <i>Training</i>	Menampilkan hasil data <i>training tweet</i> yang diinginkan	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Centang <i>tweet</i> yang diinginkan lalu Klik Tombol Add -> Jadikan Data <i>testing</i>	Menampilkan hasil data <i>testing tweet</i> yang diinginkan	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
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>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Tombol edit pada halaman <i>training</i> tabel	Menampilkan <i>Form</i> Edit <i>Training</i> Tabel dan tombol <i>save</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengubah data <i>Training</i> dan sesuai lalu Klik tombol <i>Save</i>	Mengubah data ke <i>database</i> dan menampilkan halaman <i>Training</i> Tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Hapus pada <i>Training</i> Tabel	Menghapus data <i>Training</i> dan menampilkan halaman <i>Training</i> Tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
6.	Melihat <i>Text Mining</i> Data Training	Klik menu	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		<i>Training Text</i>	halaman	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
		<i>Preprocessing</i>	<i>training</i>		
			<i>preprocessing</i>		
			dan tabs		
		- Tabs <i>Text</i>	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil	
		Tranformation	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal	
		- Tabs			
		<i>Text Unigram</i>			
		- Tabs <i>Text</i>			
		<i>Unigram</i>			
		<i>Frequency</i>			
		Total			
		Klik Tabs <i>Text</i>	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		<i>Transformation</i>	data halaman	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
			<i>text</i>		
			<i>Transformation</i>		
		Klik Tabs <i>Text</i>	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		<i>Unigram</i>	data halaman	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
			<i>Text Unigram</i>		
		Klik Tabs	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		<i>Unigram</i>	data Halaman	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
		<i>Frequency</i>	<i>Unigram</i>		
		Total	<i>Frequency</i>		
			Total		
7.	Mengelola <i>Testing</i> Tabel	Klik menu	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		<i>Testing</i> sub	halaman <i>data</i>	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
		menu Tabel	<i>testing</i> Tabel		
			dan tombol edit		
			serta hapus		
			pada <i>data</i>		
			<i>testing</i>		
		Klik Tombol	Menampilkan	<input checked="" type="checkbox"/> Berhasil	<input checked="" type="checkbox"/> Berhasil
		edit pada	<i>Form</i> Edit	<input type="checkbox"/> Gagal	<input type="checkbox"/> Gagal
			<i>Testing</i> Tabel		

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
		halaman <i>testing</i> tabel	dan tombol <i>save</i>		
		Mengubah <i>data testing</i> dan sesuai lalu Klik tombol <i>Save</i>	Mengubah data ke <i>database</i> dan menampilkan halaman <i>Testing</i> Tabel	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik Hapus pada <i>Testing</i> Tabel	Menghapus <i>data testing</i> dan menampilkan halaman <i>Testing</i> Tabel	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
8.	Melihat <i>Text Mining</i> <i>Data</i> <i>testing</i>	Klik menu <i>Testing Text</i> <i>Preprocessing</i>	Menampilkan halaman <i>Testing</i> <i>Preprocessing</i> dan tabs: -Tabs <i>Text</i> Tranformation -Tabs <i>Text</i> <i>Unigram</i> -Tabs <i>Text</i> <i>Unigram</i> <i>Frequency</i> Total	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
		Klik Tabs <i>Text</i> <i>Transformation</i>	Menampilkan <i>data testing</i> halaman <i>text</i> <i>Transformation</i>	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
		Klik Tabs <i>Text</i> <i>Unigram</i>	Menampilkan <i>data testing</i> halaman <i>Text</i> <i>Unigram</i>	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
		Klik Tabs <i>Unigram</i> <i>Frequency</i> Total	Menampilkan data <i>testing</i> Halaman <i>Unigram</i> <i>Frequency</i> Total	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
9.	Melihat Klasifikasi & Uji Performansi <i>Data</i> <i>testing</i>	Klik menu <i>Testing</i> Sub Menu Klasifikasi	Menampilkan halaman <i>data</i> <i>testing</i> Klasifikasi dan yang berisi beberapa tabs:	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
			-Tabs Klasifikasi Kategori	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
			-Tabs Uji Performansi Sentimen		
			- Tabs Uji Performansi Kategori		
		Klik Tabs Klasifikasi Ketegori	Menampilkan halaman <i>data</i> <i>testing</i> Klasifikasi Kategori	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
		Klik Tabs Uji Performansi Sentimen	Menampilkan halaman <i>data</i> <i>testing</i> Uji Performansi Sentimen	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
		Klik Tabs Uji Performansi Kategori	Menampilkan halaman <i>data</i> <i>testing</i> Uji	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
			Performansi Kategori		
10.	Mengelola <i>Stopword</i>	Klik menu <i>stopword</i>	Menampilkan halaman data <i>stopword</i> dan tombol input, edit dan hapus	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik Tombol Tambah	Menampilkan modal berupa <i>form</i> tambah <i>stopword</i> dan tombol <i>save</i>	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		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 [] Gagal	[√] Berhasil [] Gagal
		Mengisikan <i>form</i> input <i>stopword</i> dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik tombol edit	Menampilkan modal berupa <i>form</i> edit <i>stopword</i> dan tombol <i>save</i>	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		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 [] Gagal	[√] Berhasil [] Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
		Mengubah data <i>stopword</i> dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik Hapus	Menghapus data <i>stopword</i> dari <i>database</i> dan menampilkan halaman <i>stopword</i>	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
11.	Mengelola Ontologi	Klik menu ontologi	Menampilkan halaman ontologi dan tombol input, edit dan hapus	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik Tombol Tambah	Menampilkan modal berupa <i>form</i> tambah ontologi dan tombol <i>save</i>	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		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 [] Gagal	[√] Berhasil [] Gagal
		Mengisikan input ontologi dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik tombol edit	Menampilkan modal berupa <i>form</i> edit	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
			ontologi dan tombol <i>save</i>		
		Mengubah data ontologi dan tidak kosong lalu klik <i>save</i>	Mengubah data dari <i>database</i> dan menampilkan halaman ontologi	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Mengubah data ontologi dan data kosong lalu klik <i>save</i>	Menampilkan span “harap isi bidang ini”	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
		Klik Hapus	Menghapus data ontologi dari <i>database</i> dan menampilkan halaman ontology	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
12.	Melihat Persepsi konsumen <i>Data testing</i>	Klik menu <i>Data testing</i> Persepsi konsumen	Menampilkan halaman <i>Data testing</i> Persepsi konsumen	[√] Berhasil [] Gagal	[√] Berhasil [] Gagal
13.	Mengelola Data Persepsi konsumen	Klik menu Persepsi konsumen	Menampilkan data halaman Persepsi konsumen dan tombol Klasifikas hari ini serta tombol hapus	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal
		Klik Klasifikasi Hari Ini	Menampilkan halaman Persepsi konsumen Hari	[] Berhasil [√] Gagal	[√] Berhasil [] Gagal

No.	Fitur	Aksi	Hasil	Hasil Pengujian	
				30-05-2018	02-07-2018
			Ini dan tombol <i>save</i>		
		Klik <i>save</i> pada Klasifikasi Hari ini	menyimpan ke <i>database</i> dan Menampilkan halaman persepsi konsumen	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Hapus pada Persepsi konsumen	menghapus data dari <i>database</i> dan menampilkan haaman persepsi konsumen	<input type="checkbox"/> Berhasil <input checked="" type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
14.	Melihat <i>FAQ</i>	klik Menu <i>FAQ</i>	Menampilkan halaman <i>FAQ</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
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>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

BAB 6. PENUTUP

Bab ini menjelaskan tentang kesimpulan hasil penelitian yang telah dilakukan serta saran-saran untuk penelitian yang akan dilakukan di waktu mendatang.

6.1 Kesimpulan

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

1. Identifikasi persepsi konsumen kopi di media sosial *Twitter* menggunakan analisis sentimen dilakukan melalui beberapa tahapan. Tahap awal yaitu *crawling data*. *Crawling data* digunakan untuk mendapatkan *dataset* yang berada di media sosial *Twitter* menggunakan *Standard Twitter Search API*. Penggunaan *Twitter Search API* ini memiliki beberapa keterbatasan seperti *tweet* tidak menampilkan keseluruhan *tweet* yang ada pada *twitter (shorten)*, eksekusi pencarian hanya bisa dilakukan 100 kali dalam satu hari, 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* lokal. Tahap selanjutnya penerapan metode *Text Mining* untuk mengidentifikasi persepsi konsumen kopi di media sosial melalui beberapa proses yaitu *Text Preprocessing* yang terdiri dari *Case Folding* dan *Tokenizing*. *Case folding* diterapkan untuk membuat seluruh teks menjadi huruf kecil dan *tokenizing* yang menghilangkan *delimiter* serta karakter-karakter yang tidak diperlukan. Dilanjutkan dengan *Feature Generation* yang menghilangkan kata-kata yang tidak memiliki makna (*stopword*) dan mengubah kata-kata menjadi bentuk dasarnya yaitu *stemming*. Dalam penelitian ini, *stemming* yang digunakan adalah *Enhanced Confix-Stripping (ECS)* yang merupakan penyempurnaan dari algoritma sebelumnya. Tahap ketiga yaitu menerapkan metode *Naive Bayes Classifier* yang diterapkan dalam sistem untuk mengklasifikasikan data *tweet* dengan cara *data training*

yang digunakan berasal dari klasifikasi manual oleh ahli bahasa. *Data training* yang dihasilkan berjumlah 900 *tweet* dengan pelabelan yang bervariasi baik itu pelabelan sentimen maupun pelabelan kategori. Sedangkan *data testing* yang digunakan berjumlah 100 *tweet*. Tahap akhir yaitu pengujian klasifikasi yang diukur dengan uji performansi *precision*, *recall*, dan *f1score*. Hasil dari uji performansi kelas sentimen rata-rata memiliki nilai *precision* sebesar 0.77905982905983, nilai *recall* sebesar 0.72838827838828, dan *f1score* sebesar 0.73307824819453. Sedangkan pada hasil uji performansi kelas kategori rata-rata memiliki nilai *precision* sebesar 0.74154456654457, nilai *recall* sebesar 0.71, dan *f1score* sebesar 0.69409020938231. 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 0,6-0,8 dikarenakan data training yang dipakai berjumlah 900 *tweet* dengan klasifikasi yang berbeda-beda tergantung pada persepsi dari ahli bahasa yang mengklasifikasikannya. Terdapat pula beberapa kesalahan (*error*) yang disebabkan karena *tweet* yang tidak muncul secara utuh (*shorten*) sebagai akibat dari keterbatasan *Standard API Twitter*.

2. Hasil identifikasi persepsi konsumen kopi ditentukan setelah didapatkan kelas sentimen pada data *tweet*. Dalam perhitungan, satu data *tweet* mewakili satu sentimen. Perhitungan identifikasi persepsi pada penelitian ini dilakukan dalam jangka waktu satu minggu menghasilkan rentang nilai 0,7 hingga 0,9. Hasil ini didapat dari hasil *recall* sistem klasifikasi yang berupa nilai rentang antara 0 hingga 1. Semakin mendekati nilai 1, maka hasilnya menunjukkan hasil yang positif atau peningkatan. Sedangkan nilai mendekati 0 berarti hasilnya lebih ke arah negatif atau penurunan. Persepsi konsumen kopi dari hari ke hari mengalami peningkatan dengan puncak pada akhir pekan, yaitu antara hari jumat (0,83769230769231), sabtu (0,84615384615384), minggu (0,86666666666667) hingga senin (0,85714285714285). Hal ini merupakan peningkatan sesuai gaya hidup minum kopi yang ada di suatu wilayah. Tradisi minum kopi tidak sekedar untuk melepas dahaga, tetapi untuk menikmati

aktivitas lain seperti rapat, pertemuan bisnis, reuni, kencan, dan lain-lain. Hal inilah yang membuat tren peminum kopi terus meningkat tajam.

6.2 Saran

1. Implementasi sistem lebih baik menggunakan bahasa pemrograman selain *Hypertext-Preprocessing (PHP)*, seperti *python* atau *Java*, agar aplikasi lebih responsif dalam mengeksekusi data dalam jumlah besar.
2. Pada penelitian ini, dataset hasil proses *crawling* yang didapat masih belum sesuai dengan yang diharapkan. Masih terdapat beberapa kesalahan seperti pembatasan kata atau karakter yang ditampilkan dalam sistem sehingga menghasilkan *short-url*. Maka dari itu, diharapkan pada penelitian selanjutnya memiliki penambahan dan fitur seleksi penyaringan data agar *dataset* yang didapatkan lebih sesuai dengan yang diharapkan.
3. Agar mendapatkan akurasi yang lebih baik, diharapkan menambahkan *data training* lebih banyak. Semakin banyak data latih, maka formulasi untuk data tes akan semakin meningkat dan lebih akurat dalam mengklasifikasikan berbagai jenis sentimen dan kategori kopi yang ada.
4. Diharapkan untuk menambah algoritma lain dalam proses pengolahan data analisis sentimen, contohnya menggunakan *Part-Of-Speech (POS)-Tagging* atau *n-gram*.
5. Penelitian selanjutnya dapat menggunakan media sosial lain *Facebook*, *Youtube*, *Instagram* dan sosial media lainnya untuk menjangkau data lebih banyak dengan hasil yang bervariasi dan akurat dalam mengklasifikasi persepsi konsumen kopi.

DAFTAR PUSTAKA

- Asosiasi Penyelenggara Jasa Internet Indonesia. (2017). *Infografis Penetrasi dan Perilaku Pengguna Internet Indonesia*. 2017: teknopreneur.com.
- Dehaff, M. (2010). *Sentiment Analysis, Hard But Worth It!* Diambil kembali dari http://www.customerthink.com/blog/sentiment_analysis_hard_but_worth_it
- Fink, C. R., Chou, D. S., Kopecky, J. J., & Llorens, A. J. (2011). *Coarse- and Fine-Grained Sentiment Analysis of Social Media Text*. Baltimore: JOHNS HOPKINS APL TECHNICAL DIGEST.
- Fityan, M., & Sinaga, A. (2015). Analisis Faktor-Faktor Yang Mempengaruhi Perilaku Konsumen Dalam Pengambilan Keputusan Pembelian Kopi Instan. *SEPA*, 175 – 180.
- Hidayatullah, A. F., & Ma'arif, M. R. (2016). *Penerapan Text Mining dalam Judul Skripsi*. Yogyakarta.
- Indonesia, S. C. (t.thn.). Diambil kembali dari <http://www.specialtycoffee.co.id/>
- Kaku, R., Mulyanto, A., & Rohandi, M. (2014). *Penerapan Metode Naive Bayes untuk Klasifikasi Jagung Produktif di Gorontalo*. Gorontalo: Universitas Negeri Gorontalo.
- Kementerian Pertanian. (2015, 10). Outlook Kopi Komoditas Pertanian Subsektor Perkebunan. hal. 54-55.
- Liu, B. (2012, Juli 1). *Sentiment Analysis And Opinion Mining*. Diambil kembali dari <http://www.dcc.ufrj.br/~valeriab/DTMSentimentAnalysisAndOpinionMining-BingLiu.pdf>
- Manalu, B. U. (2014). *Analisis Sentimen Pada Twitter Menggunakan Text Mining*. Medan: Universitas Sumatera Utara.
- (2015). *Market Intelligence Kopi*. Japan: ITPC OSAKA.
- Nurdiansyah, Y., Bukhori, S., & Hidayat, R. (2018). Sentiment Analysis System for Movie Review In Bahasa Indonesia using Naive Bayes Classifier

- Method. *IOP Conf. Series: Journal of Physics* (hal. 1008). United Kingdom: IOPscience.
- Nurdiansyah, Y., Muharrom, F. N., & Firdaus. (2018). Implementation of Winnowing Algorithm Based K-Gram to Identify Plagiarism on File Text-Based Document. *The 3rd International Conference on Electrical Systems, Technology and Information (ICESTI 2017)* (hal. 11). France: MATEC Web of Conferences.
- Nuritha, I. (2017). *Pemanfaatan Text Mining dalam Analisis Sentimen Persepsi Masyarakat Terhadap Kopi Organik dengan Naïve Bayes Classifier*. Jember: Universitas Jember.
- Ponilan, I. R., Herdiani, A., & Selviandro, N. (2016). *Pengukuran Happiness Index Masyarakat Kota Bandung pada Media Sosial Twitter Menggunakan Pendekatan Ontologi Top-Down Hierarchy*. Bandung: School of Computing, Telkom University.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. Hartford: Library of Congress Cataloging-in-Publication Data.
- Sudiyarto, Widayati, S., & Kresna, D. M. (2012). *Perilaku Konsumen Kopi Tubruk dan Kopi Instan*. Surabaya: JSEP.
- Suisa, K., & Febrilia, V. (2014). Gaya Hidup Minum Kopi Konsumen Di The Coffee Bean & Tea Leaf Plasa Tunjungan Surabaya. *Jurnal Hospitality dan Manajemen Jasa Vol 2*, 327 - 343.
- Sussolaikah, K., & Alwi, A. (2016). *Sentimen Analysis Terhadap Acara Televisi Mata Najwa Berdasarkan Opini Masyarakat Pada Microblogging Twitter*. Ponorogo: Universitas Muhammadiyah Ponorogo.
- Tahitoe, A. D., & Purwitasari, D. (2010). *Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Tala, F. Z. (1999). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Amsterdam: Universiteit van Amsterdam.

- Tambunan, L. P., Sebayang, T., & Salmiah. (2018). Analisis Sikap Konsumen Terhadap Kopi Bubuk Lokal dan Kopi Instan di Kota Pematangsiantar. *Journal on Social Economic of Agriculture and Agribusiness*.
- Twitter. (2018). *Twitter Developer*. Diambil kembali dari <https://developer.twitter.com/en/docs/tweets/search/overview>
- Yanti, D. (2013). *Analisis Akurasi Algoritma Naive Bayes pada Klasifikasi Dokumen Berkategori*. Medan: Universitas Sumatera Utara.
- Yuliandri, M. T. (2017, Februari). Diambil kembali dari <https://majalah.ottencoffee.co.id/4-alasan-kenapa-orang-katanya-enggan-minum-kopi/>

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 atau password salah</i>	
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 <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 Email 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

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 konfirmasi <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, mengubah dan menghapus 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 (varchar: 10) - Username (varchar: 50) - Email (varchar: 70) - Role (enum: admin, ahli bahasa) - 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>	
	2. Menampilkan halaman data <i>user</i> dengan tabel yang berisi: <ul style="list-style-type: none"> - No (varchar: 10) - Username (varchar: 50)

	<ul style="list-style-type: none"> - Email (varchar: 70) - Role (enum: admin, ahli bahasa) - Action (tombol Edit dan Hapus) Dropdwon Show Tombol Search
3. Klik tombol Edit	
	4. Menampilkan halaman <i>form</i> edit <i>user</i> dengan <i>form</i> yang berisi: <ul style="list-style-type: none"> - Username (varchar: 50) - Email (varchar: 70) - Role (enum: admin, ahli bahasa) 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 (varchar: 10) - Username (varchar: 50) - Email (varchar: 70) - Role (enum: admin, ahli bahasa) - 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 (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 40px;">Tabel berisi - No (varchar: 10) - <i>User</i> (varchar: 30) - <i>Tweet</i> (varchar: 160) - Waktu Post (timestamp) - 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> dengan yang berisi: <ul style="list-style-type: none"> - Field Query (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 40px;">Tabel berisi - No (varchar: 10) - <i>User</i> (varchar: 30) - <i>Tweet</i> (varchar: 160) - Waktu Post (timestamp) - 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 <i>Training</i>	
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 (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 20px;">Tabel berisi - No (varchar: 10) - <i>User</i> (varchar: 30) - <i>Tweet</i> (varchar: 160) - Waktu Post (timestamp) - 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>	

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>	
	2. Menampilkan halaman <i>Tweet</i> yang berisi: <ul style="list-style-type: none"> - Field Query (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 20px;">Tabel berisi

	<ul style="list-style-type: none"> - No (varchar: 10) - User (varchar: 30) - Tweet (varchar: 160) - Waktu Post (timestamp) - Tombol Tambah
Aliran normal Input Data Training dari Tweet	
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 (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 20px;">Tabel berisi - No (varchar: 10) - User (varchar: 30) - <i>Tweet</i> (varchar: 160) - Waktu Post (timestamp) - 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>	
	<p>2. Menampilkan halaman <i>Tweet</i> dengan yang berisi:</p> <ul style="list-style-type: none"> - Field Query (varchar: 160) - Field Jumlah <i>Tweet</i> (int: 4) - Tombol Search <li style="padding-left: 20px;">Tabel berisi

	<ul style="list-style-type: none"> - No (varchar: 10) - User (varchar: 30) - Tweet (varchar: 160) - Waktu Post (timestamp) - Tombol Tambah
3. Masukkan Query dan Jumlah tweet 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	

A.5. Skenario Mengelola *Data Training*

No. Usecase	UC05
Nama Usecase	Mengelola <i>Data Training</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Training</i> Tabel
PostKondisi	Admin berhasil melihat, mengubah 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)

Aliran normal Mengubah 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
3. Klik tombol Edit	
	4. Menampilkan halaman <i>form</i> edit Data <i>Training</i> dengan <i>form</i> yang berisi: <ul style="list-style-type: none"> - Tweet - Sentimen - Kategori <li style="padding-left: 20px;">Tombol <i>Save</i> <li style="padding-left: 20px;">Tombol <i>Close</i>
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	
	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

	<ul style="list-style-type: none"> - Dropdown Show data yang ditampilkan - Tombol Search <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
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, mengubah 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)

Aliran normal Mengubah 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
3. Klik tombol Edit	
	4. Menampilkan halaman <i>form</i> edit <i>user</i> dengan <i>form</i> yang berisi: <ul style="list-style-type: none"> - Tweet - Sentimen - Kategori <li style="padding-left: 20px;">Tombol Save <li style="padding-left: 20px;">Tombol Close
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	
	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

	<ul style="list-style-type: none"> - Tombol Search Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
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>	
	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 (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)

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 Data <i>Training 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> Total <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - Data Awal (varchar: 160) - Hasil <i>Preprocessing</i> (varchar: 160)
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> Total <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - Hasil <i>Preprocessing</i> (varchar: 160) - <i>Stemming</i> (varchar: 160) - <i>Stopword Removal</i> (varchar: 160)
Aliran normal	

Lihat data <i>Unigram</i>	
1. Klik Tabs <i>Unigram</i>	
	2. Menampilkan halaman data <i>Training Unigram</i> yang berisi: <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram Frequency</i> Total Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - Hasil <i>Stopword Removal</i> (varchar: 160) - <i>Unigram</i> (varchar: 10)
Aliran normal	
Lihat data Frekuensi <i>Unigram</i>	
1. Klik Tabs <i>Unigram Frequency</i>	
	2. Menampilkan halaman data <i>Training Unigram Frequency</i> yang berisi: <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram</i> Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Unigram</i> (varchar: 50) - Frekuensi (int: 10)

A.7. Skenario Mengelola *Data testing*

No. Usecase	UC07
Nama Usecase	Mengelola <i>Data testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Data testing</i> Tabel
PostKondisi	Admin berhasil melihat, mengubah dan menghapus <i>data testing</i>
Aliran normal	
Lihat <i>Data testing</i>	
1. Klik menu <i>Testing</i> Tabel	

	<p>2. Menampilkan halaman <i>data 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)
<p>Aliran normal Mengubah <i>Data testing</i></p>	
<p>1. Klik menu <i>Testing</i> Tabel</p>	
	<p>2. Menampilkan halaman <i>data 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)
<p>3. Klik tombol Edit</p>	
	<p>4. Menampilkan halaman <i>form</i> edit <i>Data testing</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - Tweet - Sentimen - Kategori Tombol <i>Save</i> Tombol <i>Close</i>

5. Mengubah <i>data testing</i>	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal Hapus <i>Data testing</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman <i>data 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 <li style="padding-left: 20px;">Tabel yang berisi - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)
3. Klik tombol hapus	
	4. Menghapus <i>data testing</i> dari <i>database</i>
	5. Menampilkan halaman <i>data testing</i>

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

	<p>2. Menampilkan halaman <i>data 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 - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp) - Action (tombol Edit dan Hapus)
<p>Aliran normal Mengubah <i>Data testing</i></p>	
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman <i>data 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 - No (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
3. Klik tombol Edit	
	<p>4. Menampilkan halaman <i>form</i> edit <i>Data testing</i> dengan <i>form</i> yang berisi:</p> <ul style="list-style-type: none"> - Tweet (varchar: 160) - Sentimen (enum: positif, negatif, netral)

	<ul style="list-style-type: none"> - Kategori (enum: 'kopi hitam','kopi robusta','kopi arabika','kopi aceh','kopi toraja','kopi luwak','kopi papua','kopi hijau','kopi sakit','kopi pahit','tidak terkategori') <p>Tombol <i>Save</i> Tombol <i>Close</i></p>
5. Mengubah <i>data testing</i>	
6. Klik tombol <i>save</i>	
	7. Mengubah data
	8. Menampilkan halaman data <i>user</i>
Aliran normal Hapus <i>Data testing</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>Testing</i> Tabel	
	<p>2. Menampilkan halaman <i>data 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 (varchar: 10) - Username (varchar: 30) - Tweet (varchar: 160) - Post Time (timestamp)
3. Klik tombol hapus	
	4. Menghapus <i>data testing</i> dari <i>database</i>
	5. Menampilkan halaman <i>data testing</i>

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

No. Usecase	UC08
--------------------	-------------

Nama Usecase	Melihat <i>Text Mining Data testing</i>
Aktor	Admin
PreKondisi	Admin memilih menu <i>Data testing Text Mining</i>
PostKondisi	Admin berhasil melihat <i>Text Mining Data testing</i>
Aliran normal Lihat data <i>Text Preprocessing</i>	
1. Klik menu <i>Testing Text Preprocessing</i>	
	<p>2. Menampilkan halaman <i>data testing 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 Total</i> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - Data Awal (varchar: 160) - Hasi <i>Preprocessing</i> (varchar: 160)
Aliran normal Lihat data <i>Text Transformation</i>	
1. Klik Tabs <i>Text Transformation</i>	
	<p>2. Menampilkan halaman <i>data 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 Total</i> <p style="padding-left: 40px;">Tabel yang berisi</p> <ul style="list-style-type: none"> - No (varchar: 10) - Hasil <i>Preprocessing</i> (varchar: 160) - <i>Stemming</i> (varchar: 160) - <i>Stopword Removal</i> (varchar: 160)
Aliran normal Lihat data <i>Unigram</i>	
1. Klik Tabs <i>Unigram</i>	

	2. Menampilkan halaman <i>data testing Unigram</i> yang berisi: <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram Frequency</i> Total Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - Hasil <i>Stopword Removal</i> (varchar: 160) - <i>Unigram</i> (varchar: 160)
Aliran normal Lihat data Frekuensi <i>Unigram</i>	
. Klik Tabs <i>Unigram Frequency</i>	
	2. Menampilkan halaman <i>data testing Unigram Frequency</i> yang berisi: <ul style="list-style-type: none"> - Tabs <i>Text Preprocessing</i> - Tabs <i>Text Transformation</i> - Tabs <i>Text Unigram</i> Tabel yang berisi <ul style="list-style-type: none"> - No (varchar: 10) - <i>Unigram</i> (varchar: 50) - Frekuensi (int: 10)

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, mengubah, 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>	
	2. Menampilkan halaman data <i>Stopword</i> , meliputi Tombol

	Tambah, dropdown show, tombol search dan tabel yang berisi: <ul style="list-style-type: none"> - No (varchar: 10) - <i>Stopword</i> (varchar: 50) - Action (edit dan hapus)
3. Klik tombol tambah	
	4. Menampilkan modal berupa <i>form</i> Input <i>stopword</i> yang berisi: <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 <i>Stopword</i> , meliputi Tombol Tambah, dropdown show, tombol search dan tabel yang berisi: <ul style="list-style-type: none"> - No (varchar: 10) - <i>Stopword</i> (varchar: 50) Action (edit dan hapus)
Aliran normal Ubah 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: <ul style="list-style-type: none"> - No (varchar: 10) - <i>Stopword</i> (varchar: 50) Action (edit dan hapus)
3. Klik tombol edit	
	4. Menampilkan Modal <i>form</i> Edit <i>Stopword</i> yang berisi: <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>	
	7. Mengubah data
	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: <ul style="list-style-type: none"> - No (varchar: 10) - <i>Stopword</i> (varchar: 50) 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, mengubah, 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 (varchar: 10) - Kata (varchar: 100) - Parameter (enum: 'kopi hitam','kopi robusta','kopi arabika','kopi aceh','kopi toraja','kopi luwak','kopi papua','kopi hijau','kopi sakit','kopi pahit','tidak terkategori') - 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 (varchar: 10) - Kata (varchar: 100) - Parameter (enum: 'kopi hitam','kopi robusta','kopi arabika','kopi aceh','kopi toraja','kopi luwak','kopi papua','kopi hijau','kopi sakit','kopi pahit','tidak terkategori') <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 (varchar: 10) - Kata (varchar: 100) - Parameter (enum: 'kopi hitam','kopi robusta','kopi arabika','kopi aceh','kopi toraja','kopi luwak','kopi papua','kopi hijau','kopi sakit','kopi pahit','tidak terkategori') <p>Action (edit dan hapus)</p>
3. Klik tombol edit	
	<p>4. Menampilkan Modal <i>form</i> Edit Ontologi yang berisi:</p> <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 (varchar: 10) - Kata (varchar: 100) - Parameter (enum: 'kopi hitam', 'kopi robusta', 'kopi arabika', 'kopi aceh', 'kopi toraja', 'kopi luwak', 'kopi papua', 'kopi hijau', 'kopi sakit', 'kopi pahit', 'tidak terkategori') <p>Action (edit dan hapus)</p>
3. Klik tombol hapus	
	4. Menghapus data
	5. Menampilkan halaman data Ontologi

A.11. Melihat Persepsi Konsumen Kopi *Data Testing*

No. Usecase	UC12
Nama Usecase	Melihat Persepsi konsumen kopi <i>Data testing</i>

Aktor	Admin
PreKondisi	Admin memilih menu <i>Data testing</i> Persepsi konsumen kopi
PostKondisi	Admin berhasil melihat Persepsi konsumen kopi <i>Data testing</i>
Aliran normal	
Lihat data Persepsi konsumen kopi <i>Data testing</i>	
1. Klik menu <i>Data testing</i> Persepsi konsumen kopi	
	<p>2. Menampilkan halaman <i>Data testing</i> Persepsi konsumen kopi yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> (date) - Chart berbentuk Line berisi Persepsi konsumen kopi

A.12. Mengelola Data Persepsi Konsumen Kopi

No. Usecase	UC13
Nama Usecase	Mengelola Data Persepsi konsumen kopi
Aktor	Admin
PreKondisi	Admin memilih menu Persepsi konsumen kopi
PostKondisi	Admin berhasil menambah, melihat dan menghapus persepsi konsumen kopi
Aliran normal	
Lihat Data Persepsi konsumen kopi	
1. Klik menu	
	<p>2. Menampilkan halaman Persepsi konsumen kopi yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Persepsi konsumen kopi - Tabel yang berisi <ul style="list-style-type: none"> - Indeks (varchar: 10) - Tanggal (timestamp) - Action (tombol Hapus) <p>Tombol Klasifikasi Hari ini</p>
Aliran normal	
Menambah Persepsi konsumen kopi	

1. Klik menu	<p>2. Menampilkan halaman <i>Data testing</i> Persepsi konsumen kopi yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Persepsi konsumen kopi - Tabel yang berisi <ul style="list-style-type: none"> - Indeks (varchar: 10) - Tanggal (timestamp) - Action (tombol Hapus) <p>Tombol Klasifikasi Hari ini</p>
3. Klik Tombol Klasifikasi Hari Ini	<p>4. Menampilkan halaman Persepsi konsumen kopi Hari Ini dengan tabel yang berisi:</p> <ul style="list-style-type: none"> - <i>Tweet</i> (varchar: 160) - Sentimen Positif (int: 4) - <i>Tweet</i> Positif (int: 4) - <i>Tweet</i> Negatif (int: 4) - Persepsi konsumen kopi (varchar: 100) - Tanggal (timestamp)
5. Klik Save	<p>6. Kembali ke halaman Persepsi konsumen kopi</p>
<p>Aliran normal Menghapus Persepsi konsumen kopi</p>	
1. Klik menu	<p>2. Menampilkan halaman <i>Data testing</i> Persepsi konsumen kopi yang berisi:</p> <ul style="list-style-type: none"> - <i>Form Date Range</i> - Chart berbentuk Line berisi Persepsi konsumen kopi - Tabel yang berisi <ul style="list-style-type: none"> - Indeks (varchar: 10) - Tanggal (timestamp) - Action (tombol Hapus) <p>Tombol Klasifikasi Hari ini</p>

3. Klik Tombol Hapus	
	4. Menghapus data persepsi konsumen kopi dari <i>database</i>
	5. Menampilkan halaman Persepsi konsumen kopi

A.13. Melihat *FAQ*

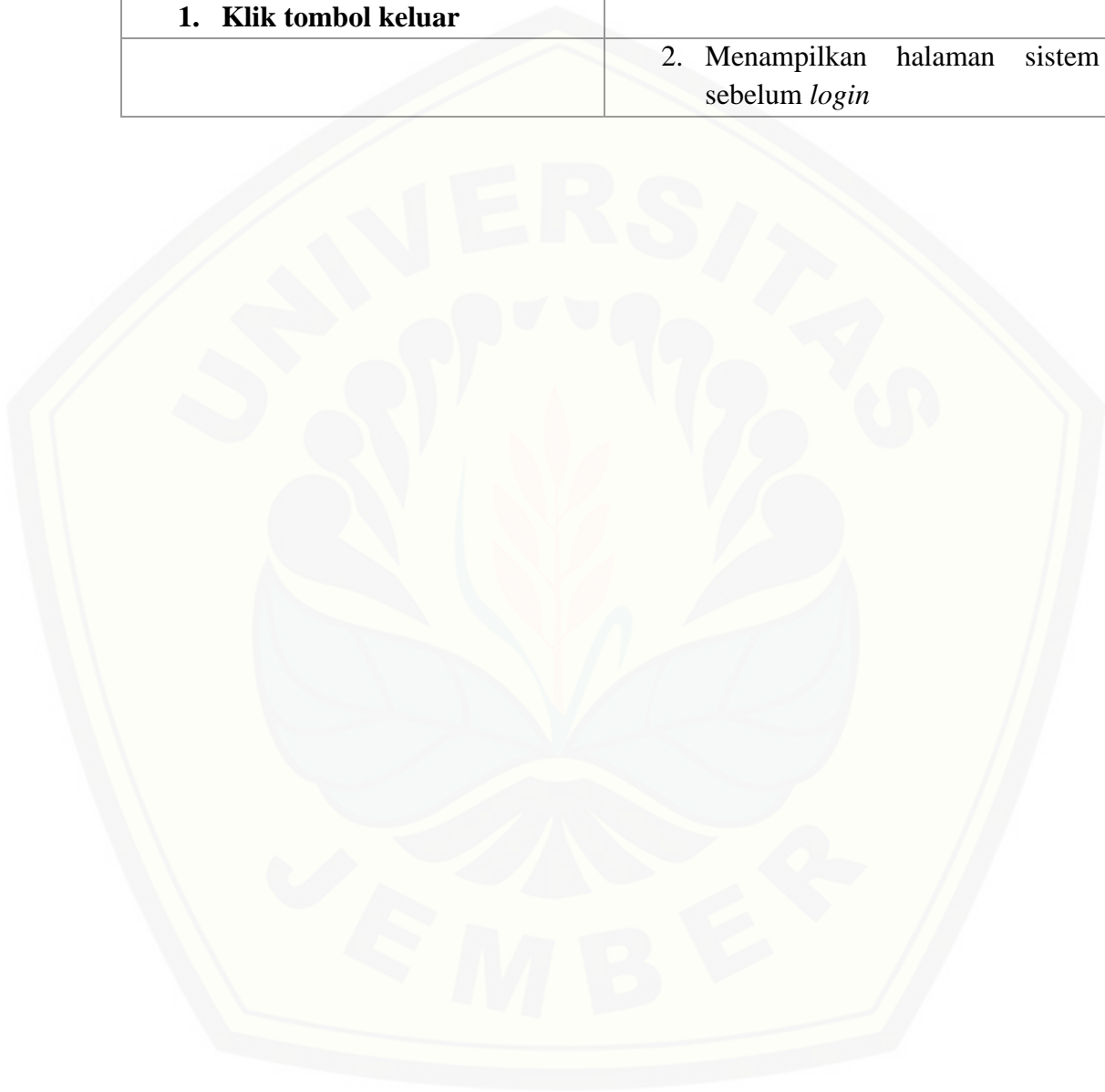
No. Usecase	UC14
Nama Usecase	Melihat <i>FAQ</i>
Aktor	Ahli bahasa
PreKondisi	Ahli bahasa memilih menu <i>FAQ</i>
PostKondisi	Ahli bahasa berhasil melihat data <i>FAQ</i> Tabel
Aliran normal Lihat Halaman Fax	
1. Klik menu Fax	
	2. Menampilkan halaman <i>FAQ</i> Tabel yang berisi: <ul style="list-style-type: 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>

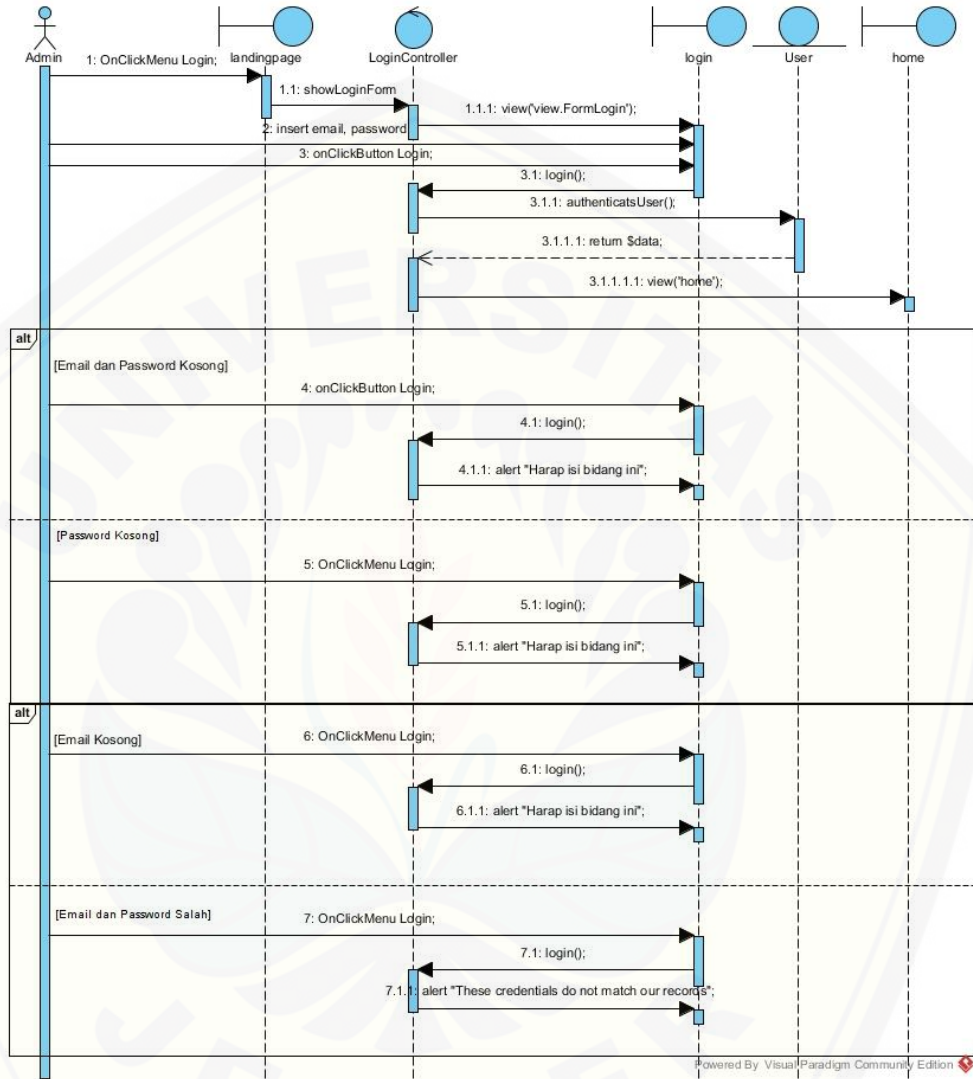
No. Usecase	UC15
Nama Usecase	<i>Logout</i>
Aktor	Ahli Bahasa

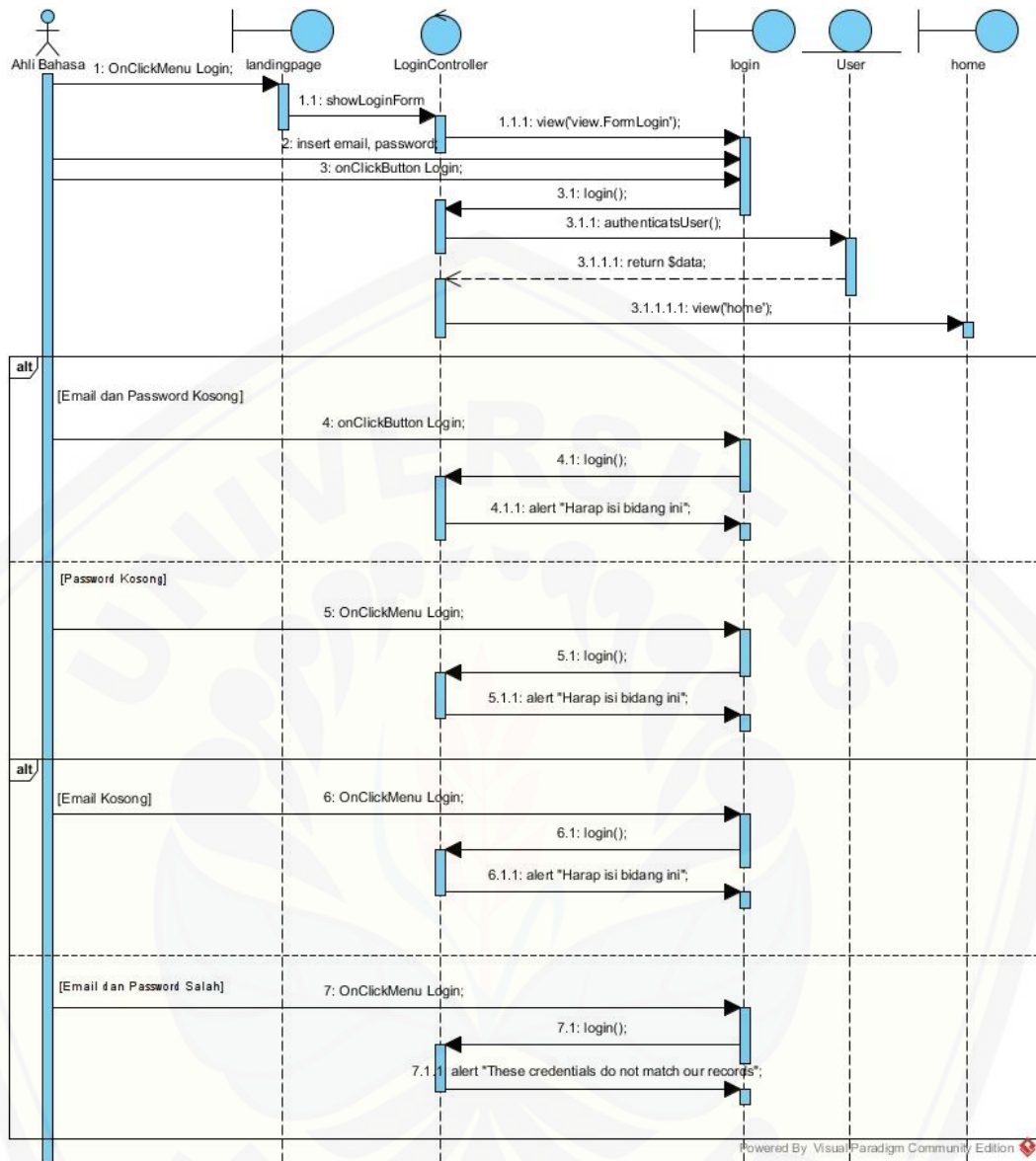
Prekondisi	Ahli Bahasa Memilih Tombol keluar
Postkondisi	Ahli Bahasa Berhasil Keluar
Aliran normal Keluar	
Aksi Aktor	Reaksi Sistem
1. Klik tombol keluar	
	2. Menampilkan halaman sistem sebelum <i>login</i>



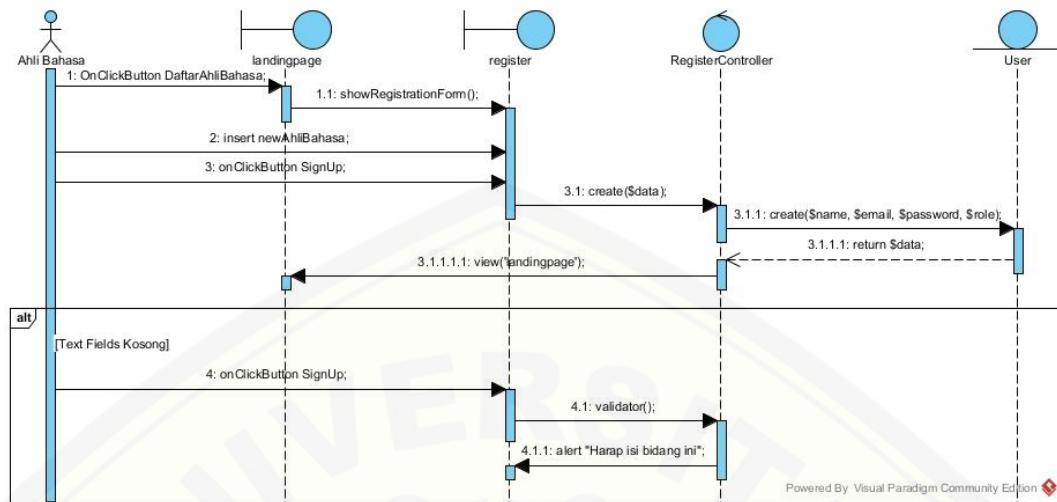
LAMPIRAN B. SEQUENCE DIAGRAM

B.1. Sequence Diagram Login

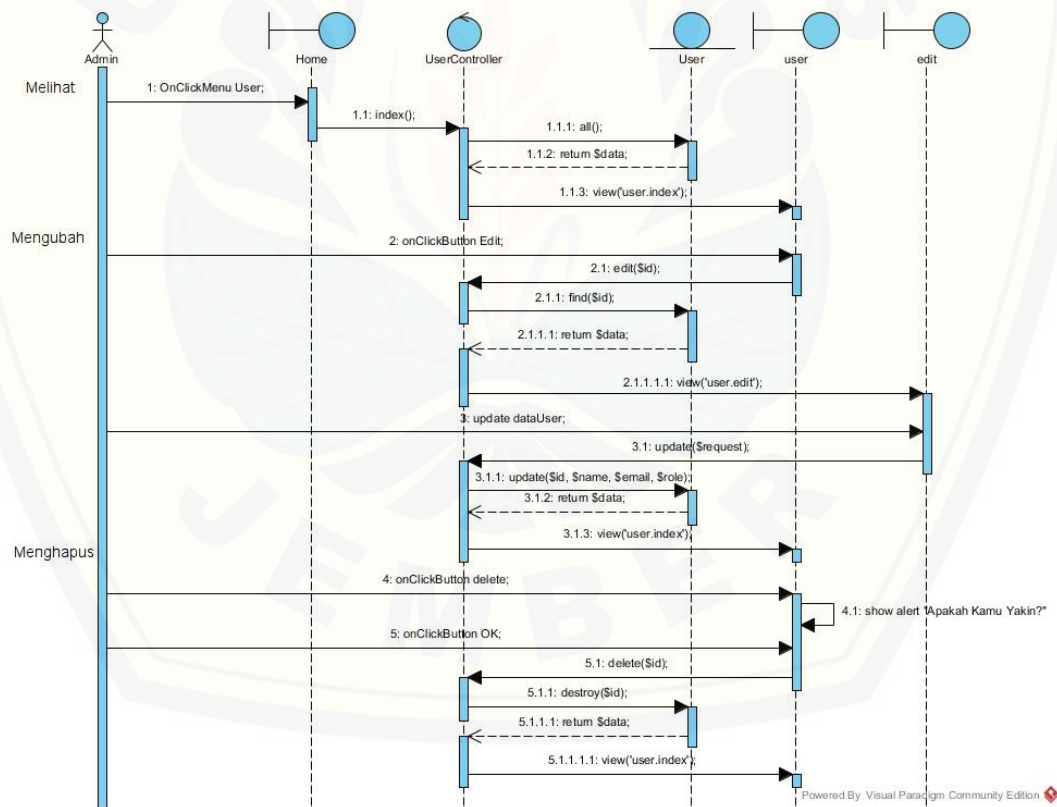




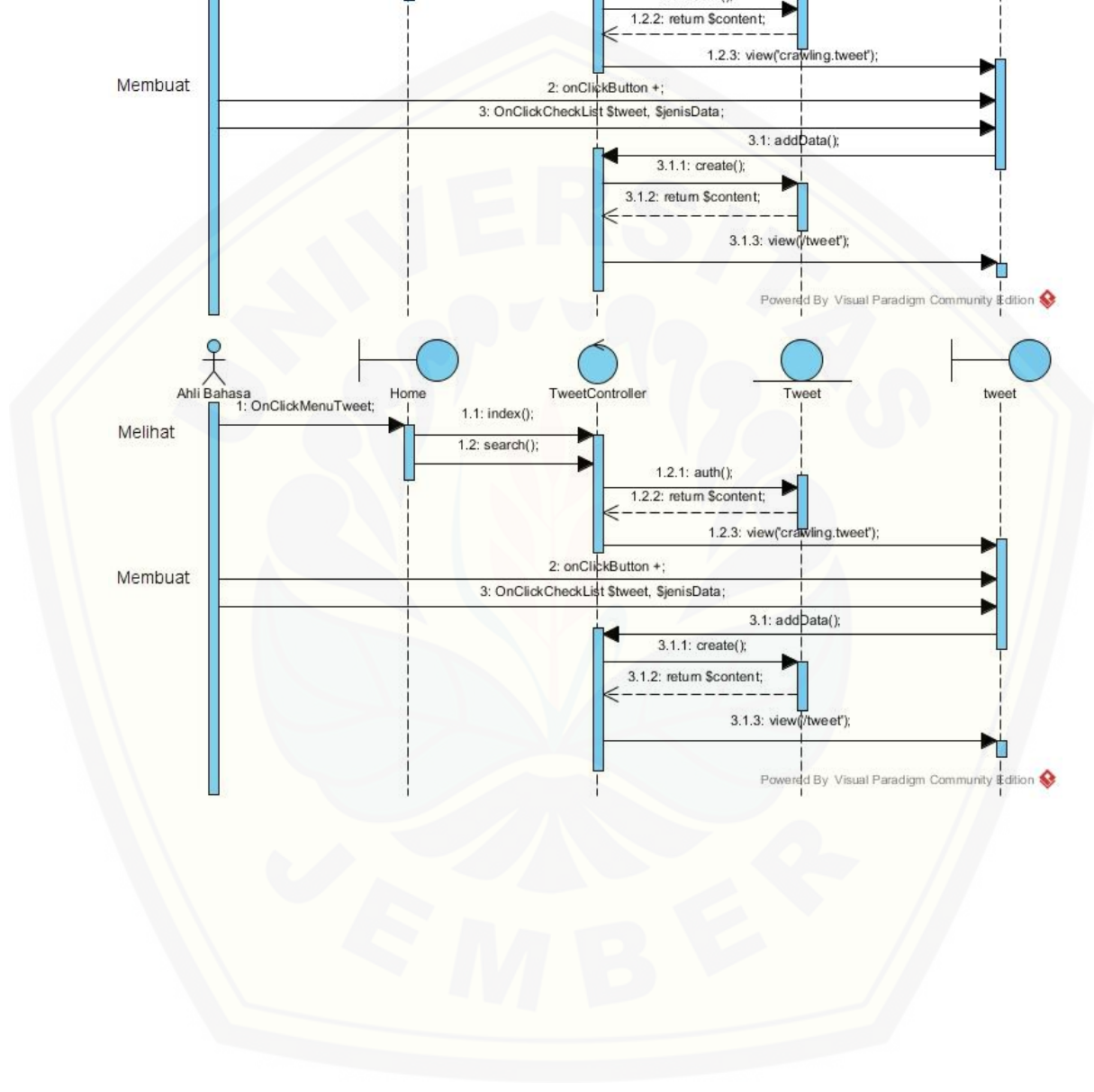
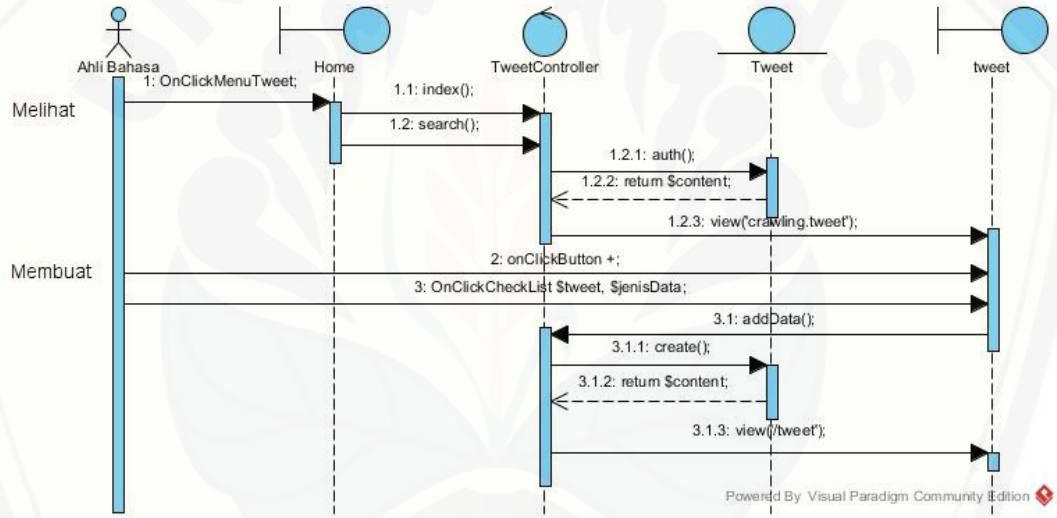
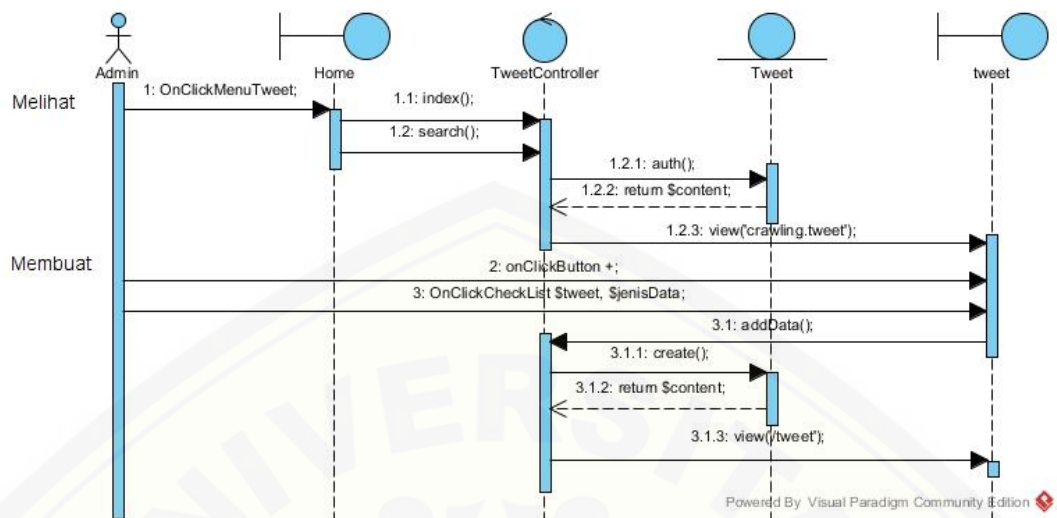
B.2. Sequence Diagram Registrasi Akun



B.3. Sequence Diagram Mengelola User



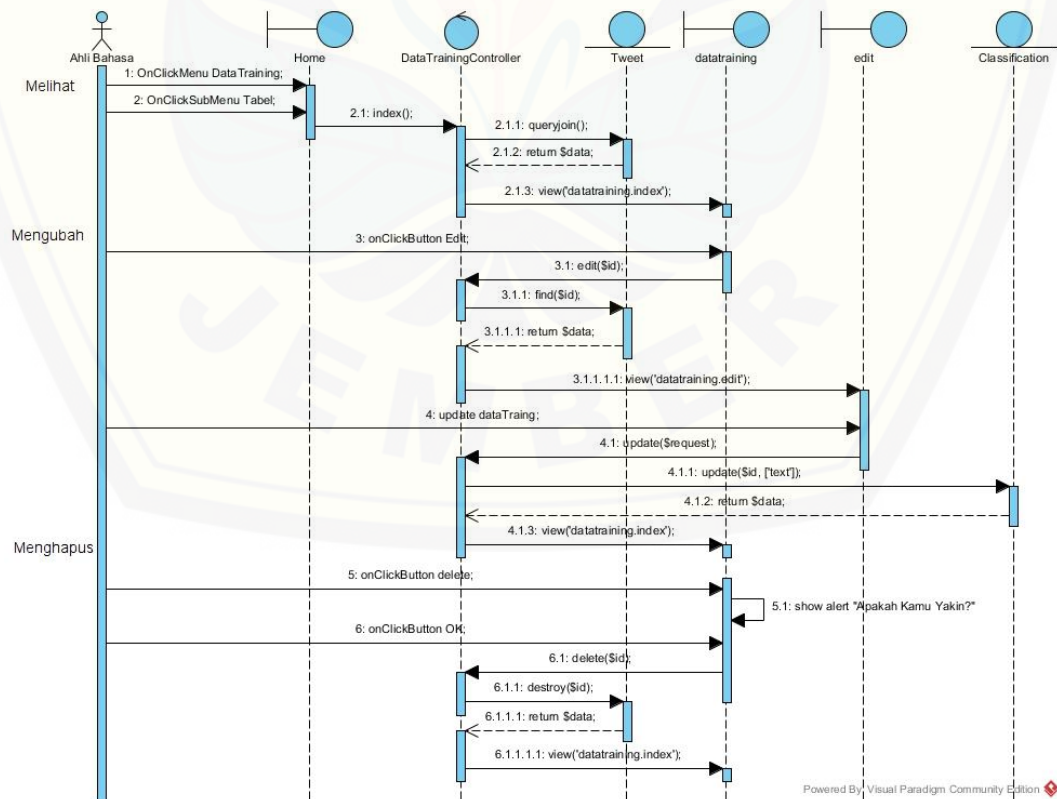
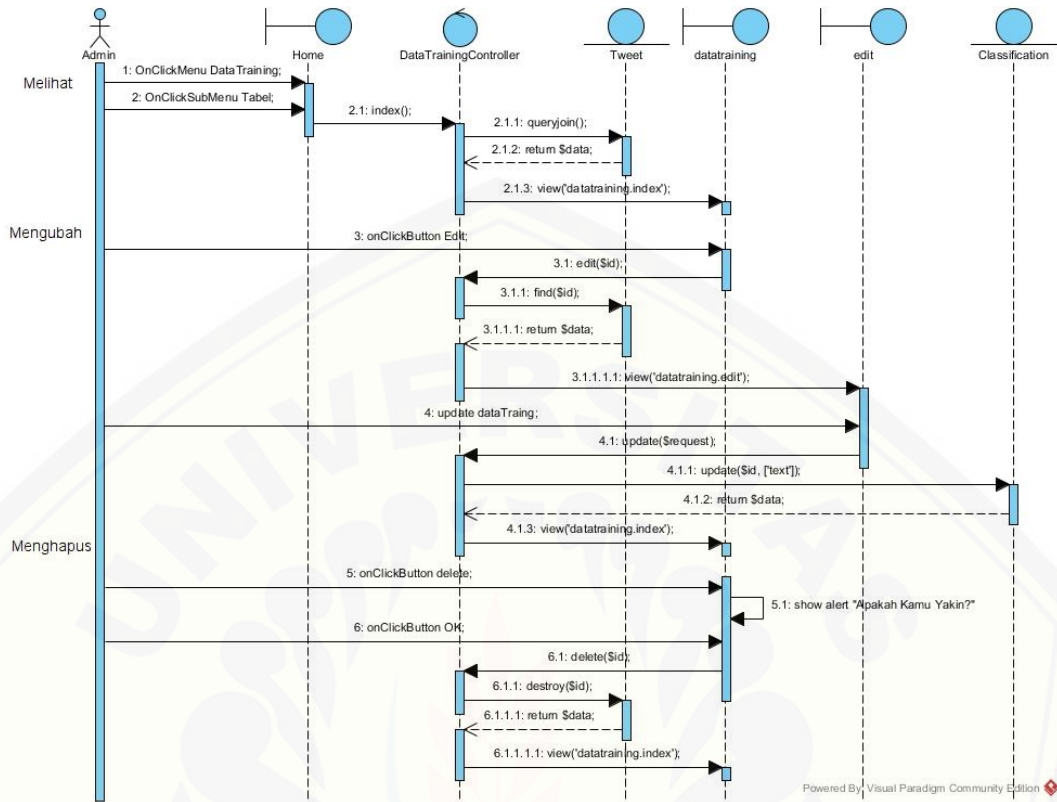
B.4. Sequence Diagram Mengelola Tweet



Powered By Visual Paradigm Community Edition

Powered By Visual Paradigm Community Edition

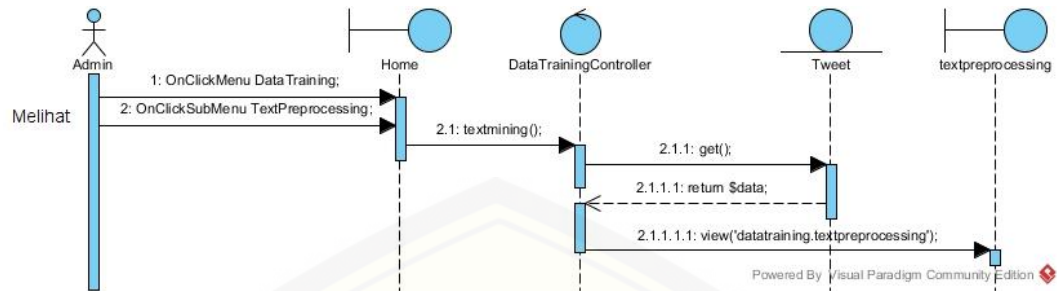
B.5. Sequence Diagram Mengelola Data Training



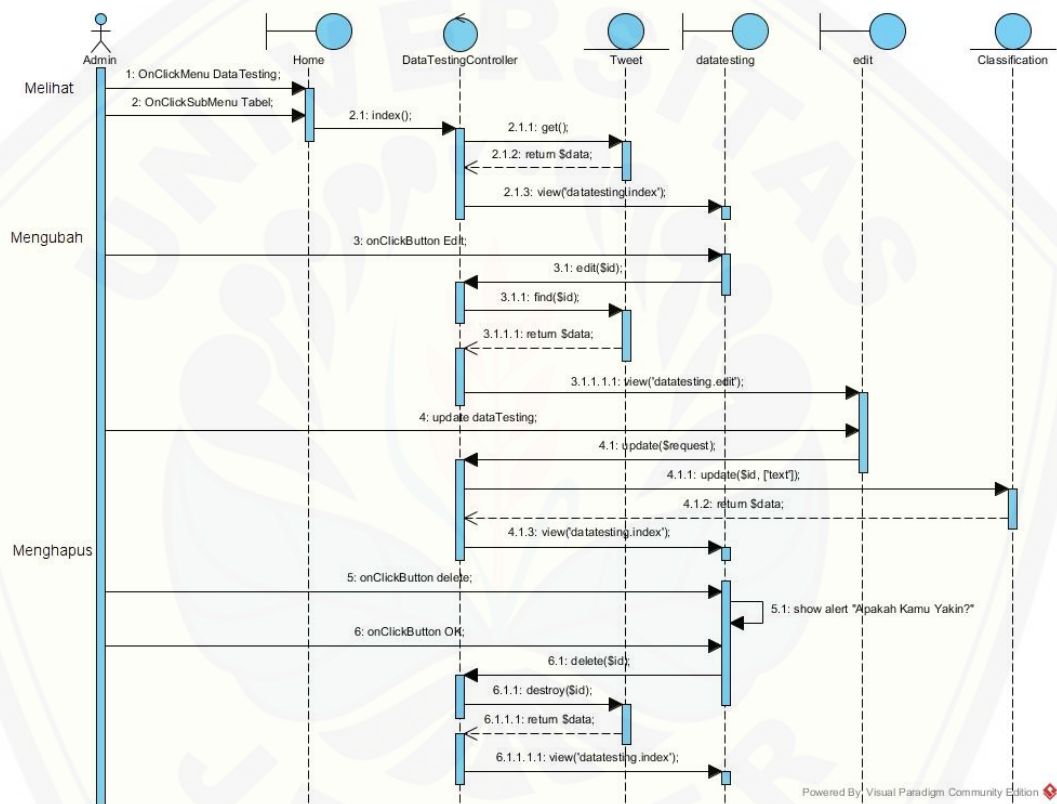
Powered By Visual Paradigm Community Edition

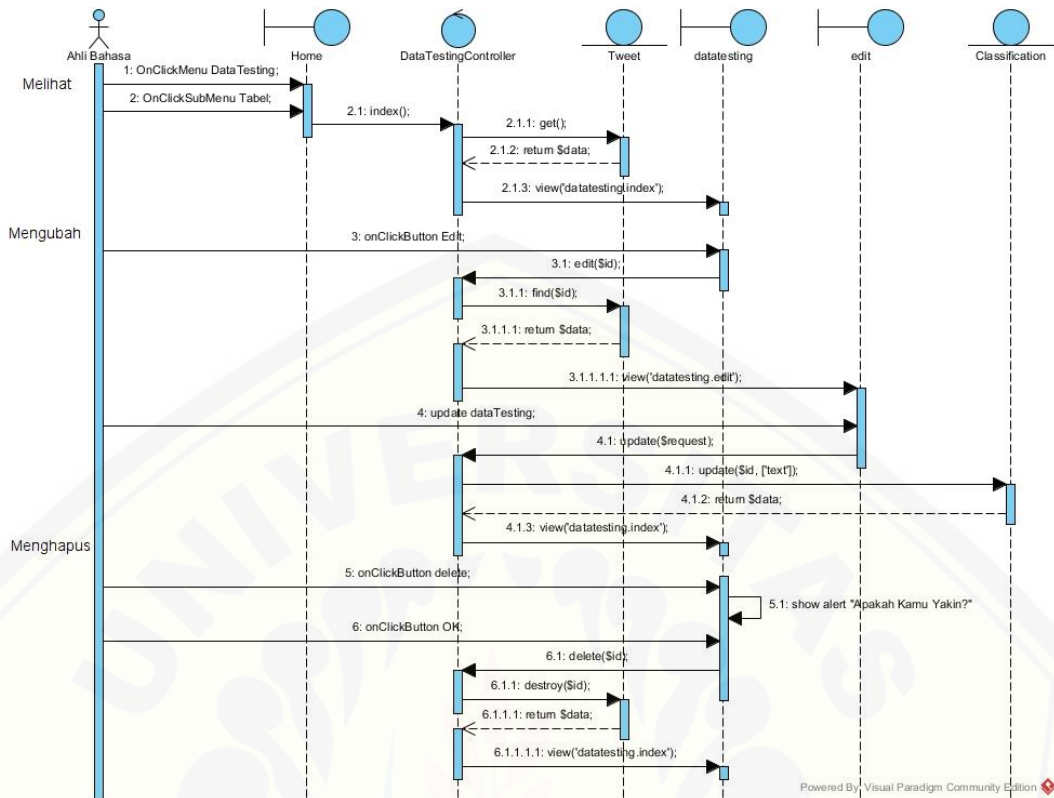
Powered By Visual Paradigm Community Edition

B.6. Sequence Diagram Melihat Text Mining Data Training

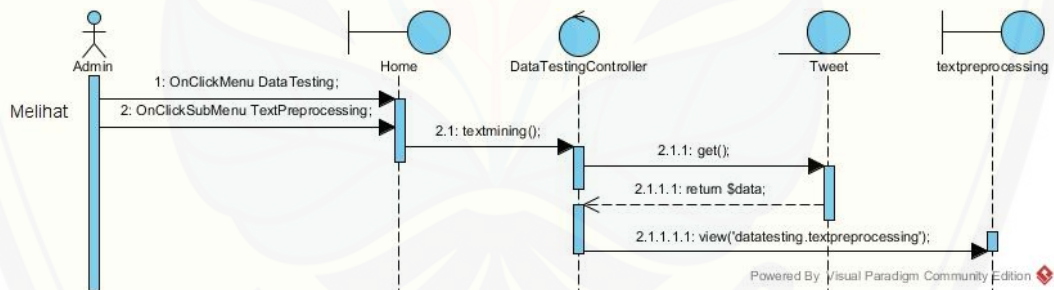


B.7. Sequence Diagram Mengelola Data testing

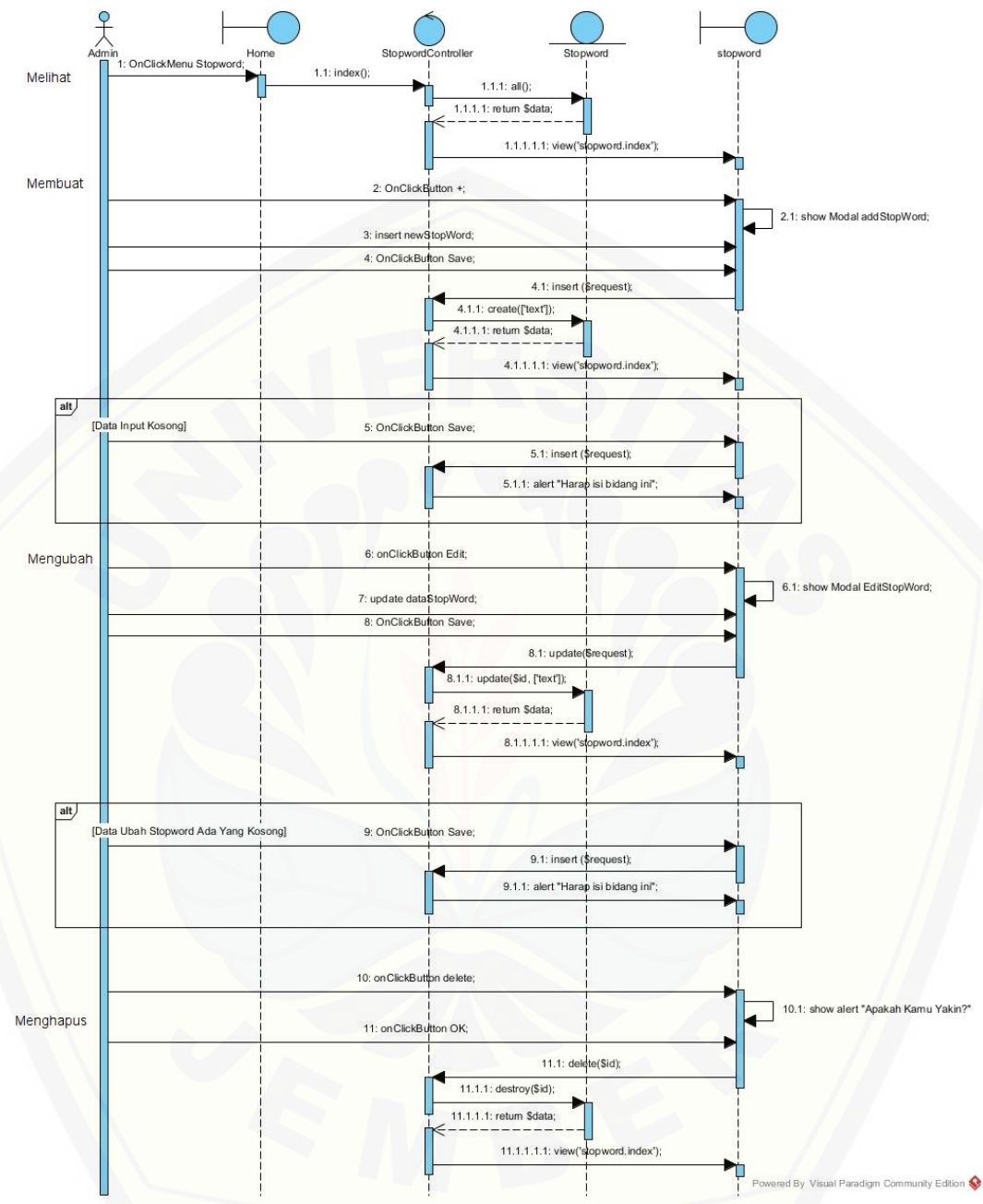




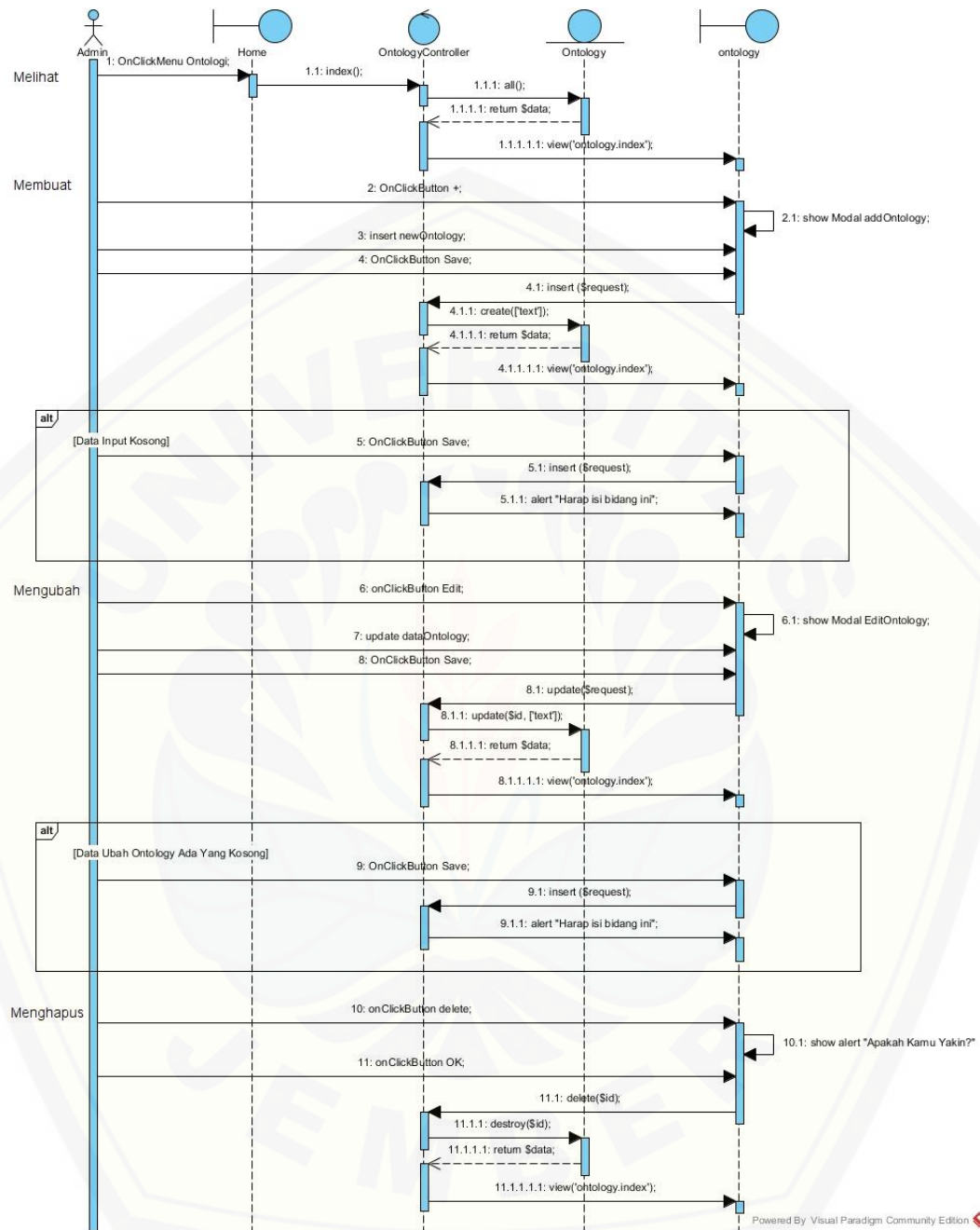
B.8. Sequence Diagram Melihat Text Mining data testing



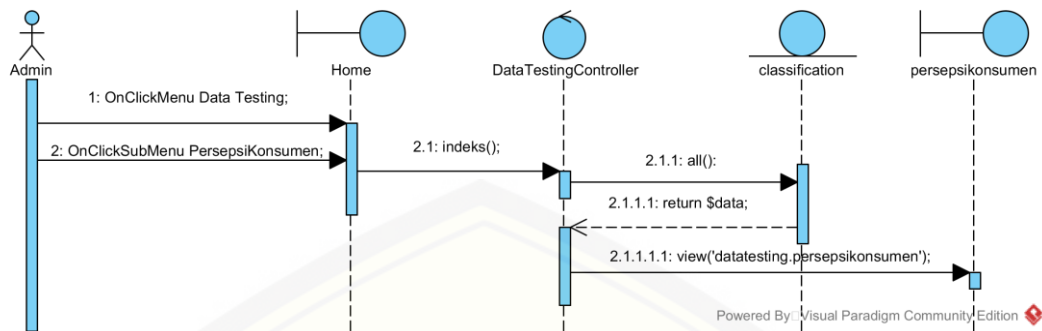
B.9. Sequence Diagram Mengelola Stopword



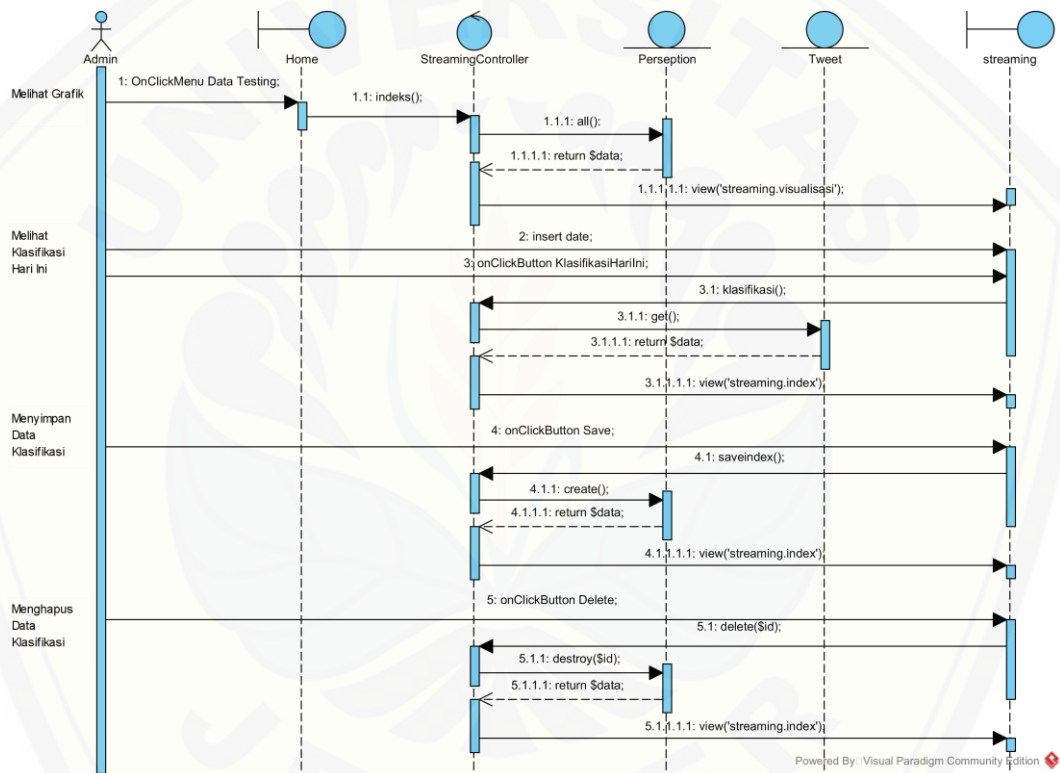
B.10. Sequence Diagram Mengelola Ontologi



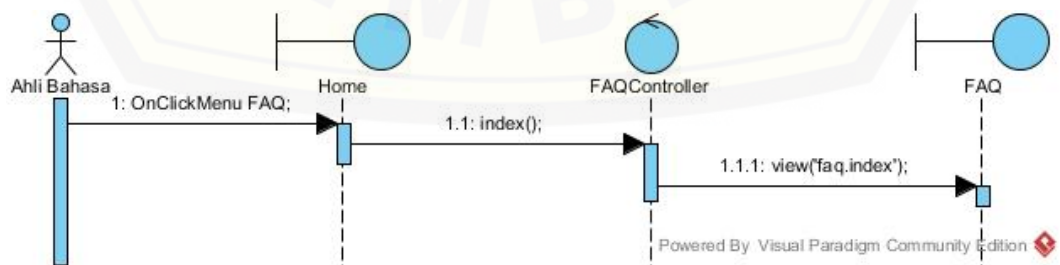
B.11. Sequence Diagram Melihat Persepsi Konsumen Kopi Data testing



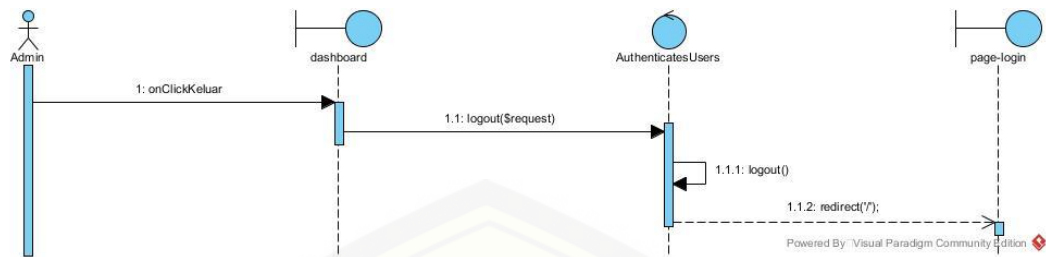
B.12. Sequence Diagram Mengelola Data Persepsi Konsumen Kopi



B.13. Sequence Diagram Melihat FAQ

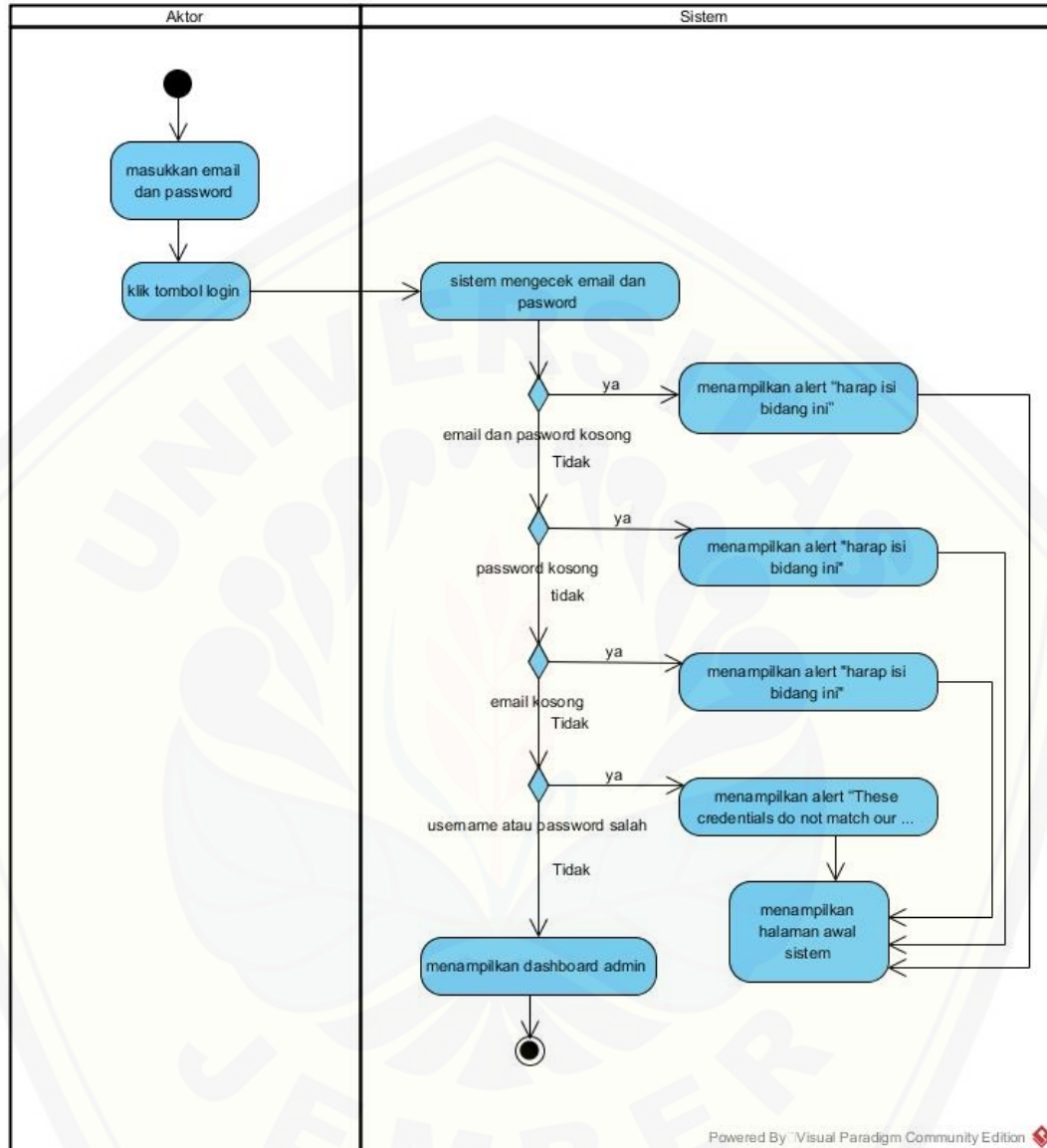


B.14. Sequence Diagram Logout

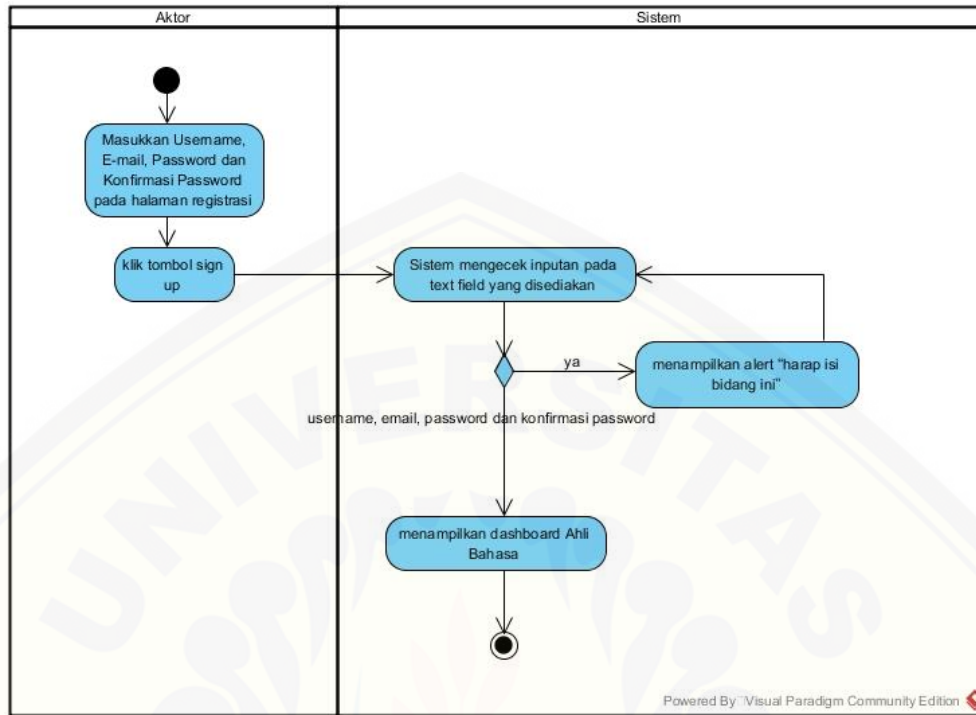


LAMPIRAN C. ACTIVITY DIAGRAM

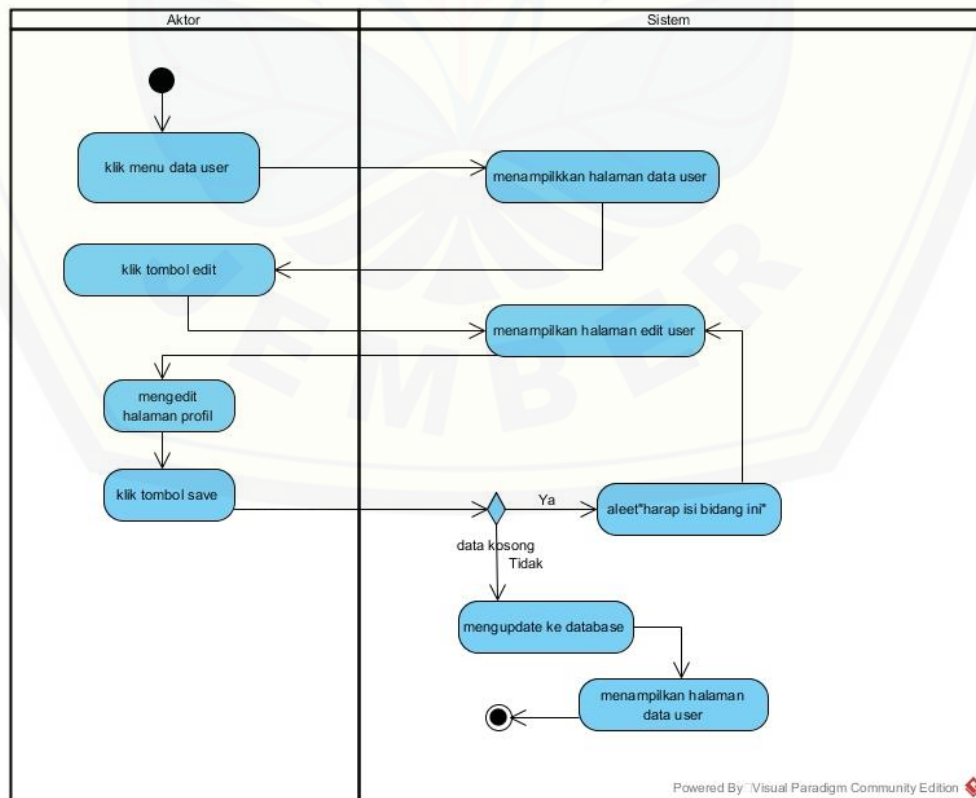
C.1. Activity Diagram Login



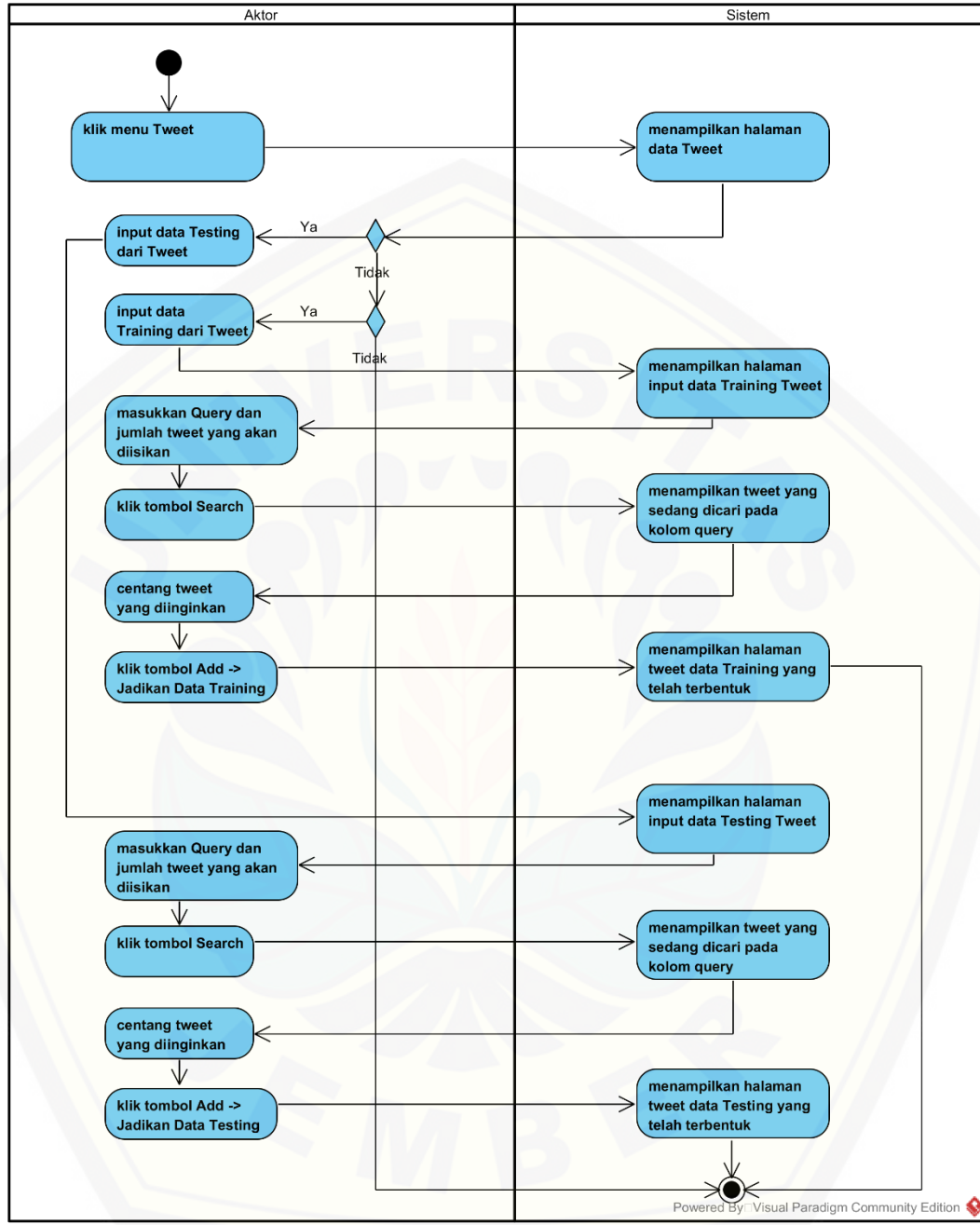
C.3. Activity Diagram Registrasi Ahli Bahasa



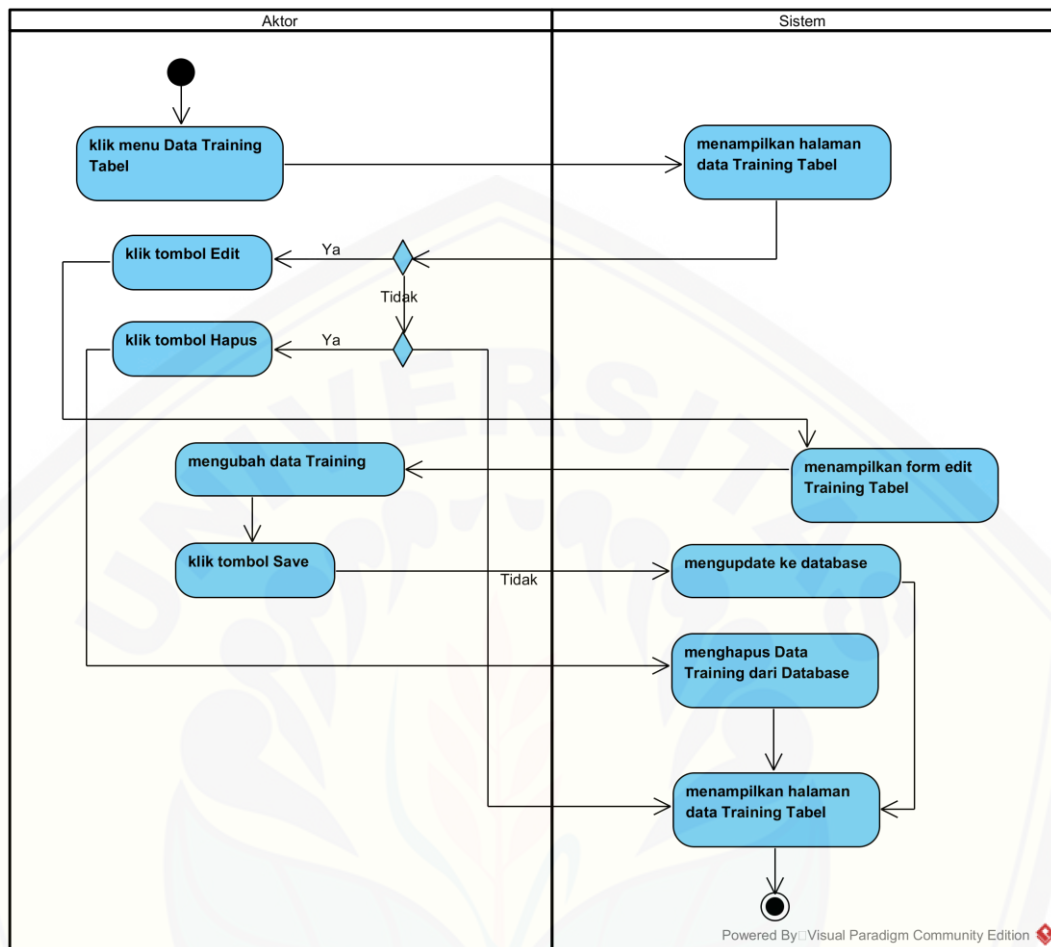
C.3. Activity Diagram Mengelola Data User



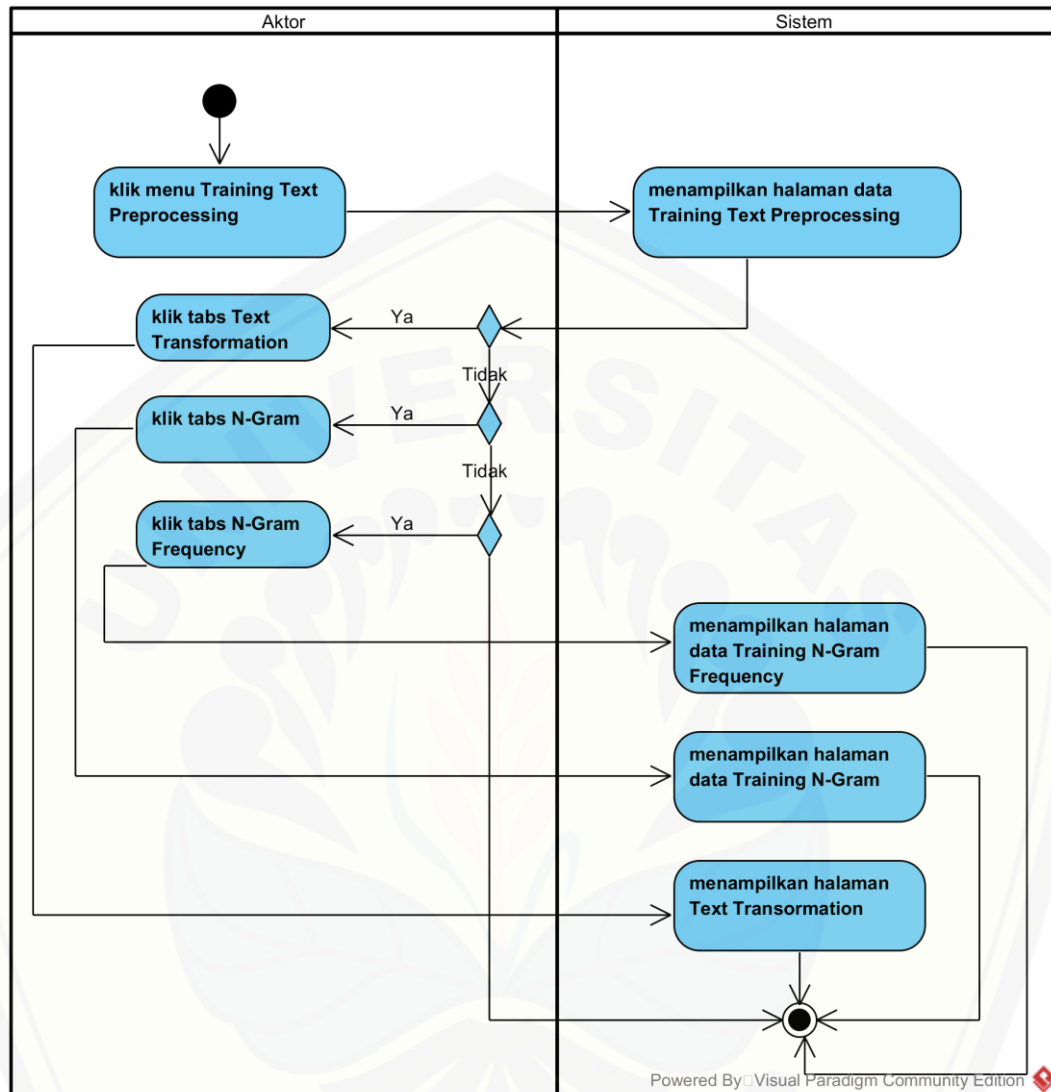
C.4. Activity Diagram Mengelola Tweet



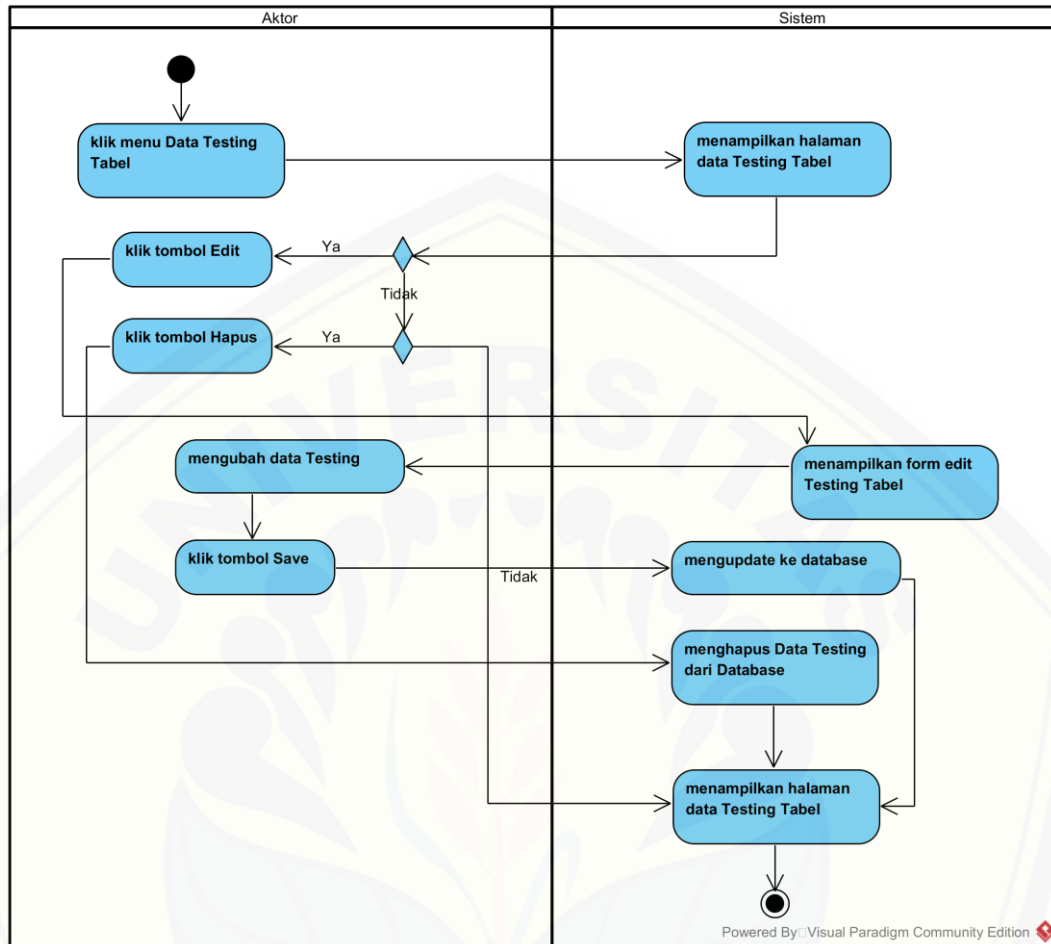
C.5. Activity Diagram Mengelola Data Training



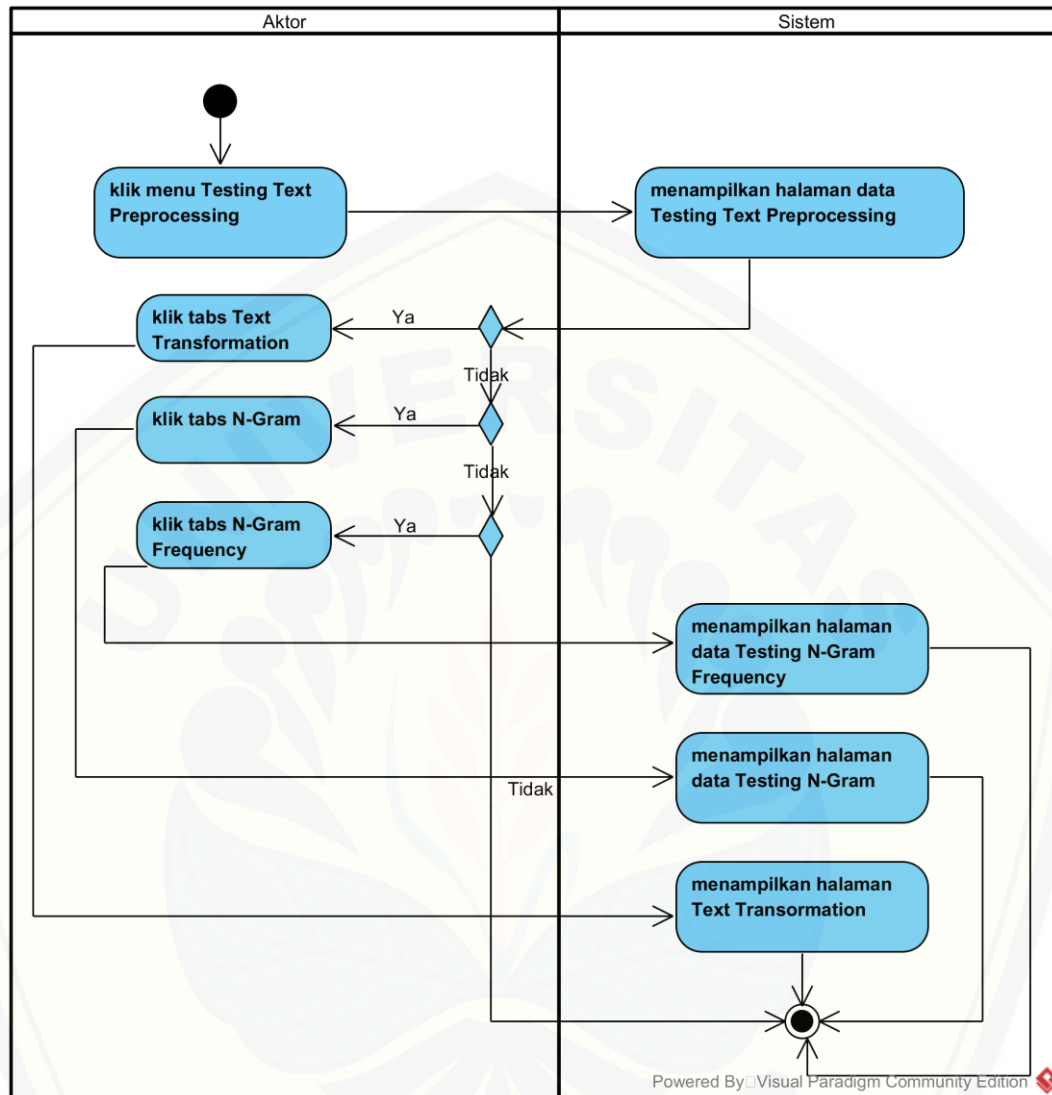
C.6. Activity Diagram Melihat Text Mining Data Training



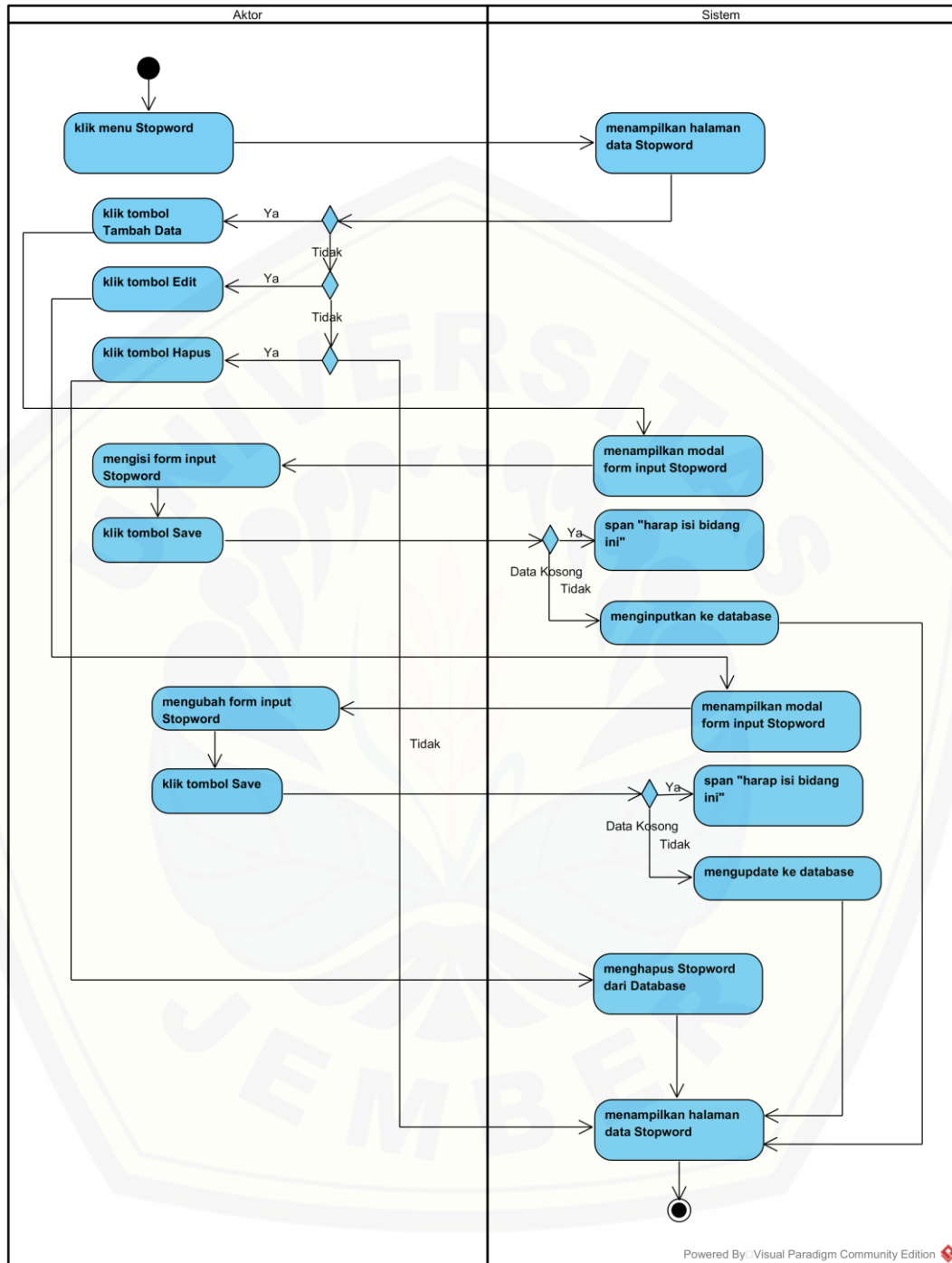
C.7. Activity Diagram Mengelola Data Testing



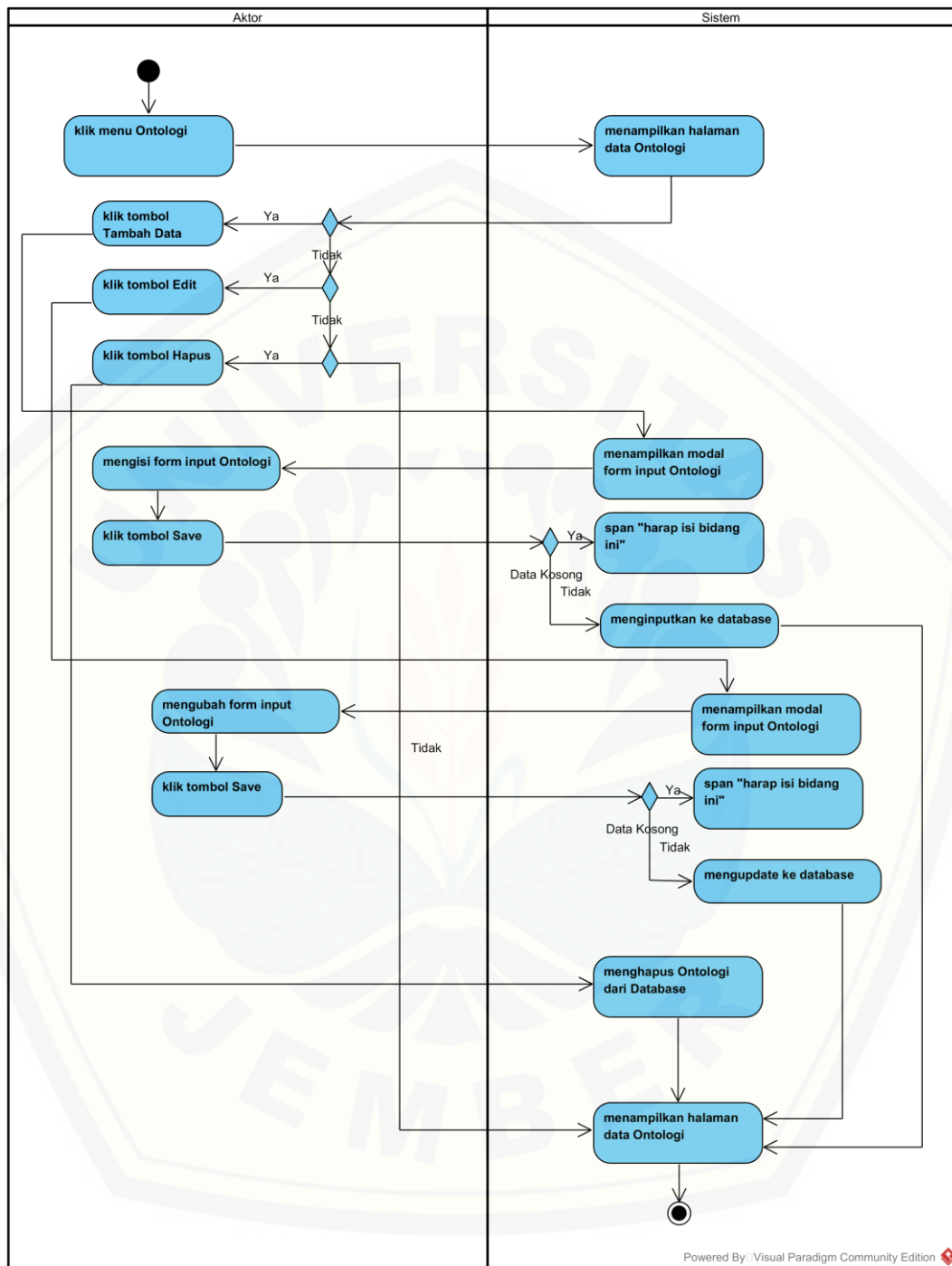
C.8. Activity Diagram Melihat Text Mining Data Testing



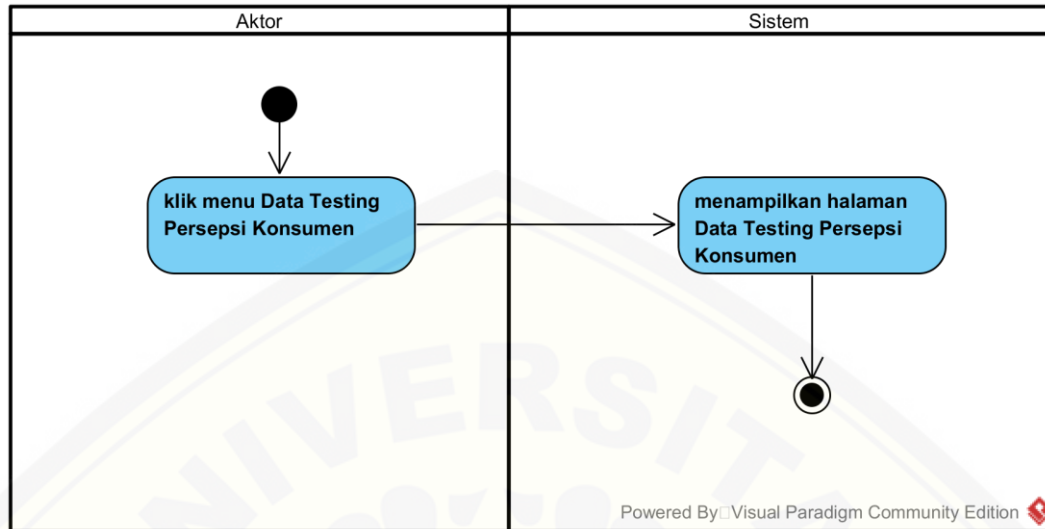
C.9. Activity Diagram Mengelola Stopword



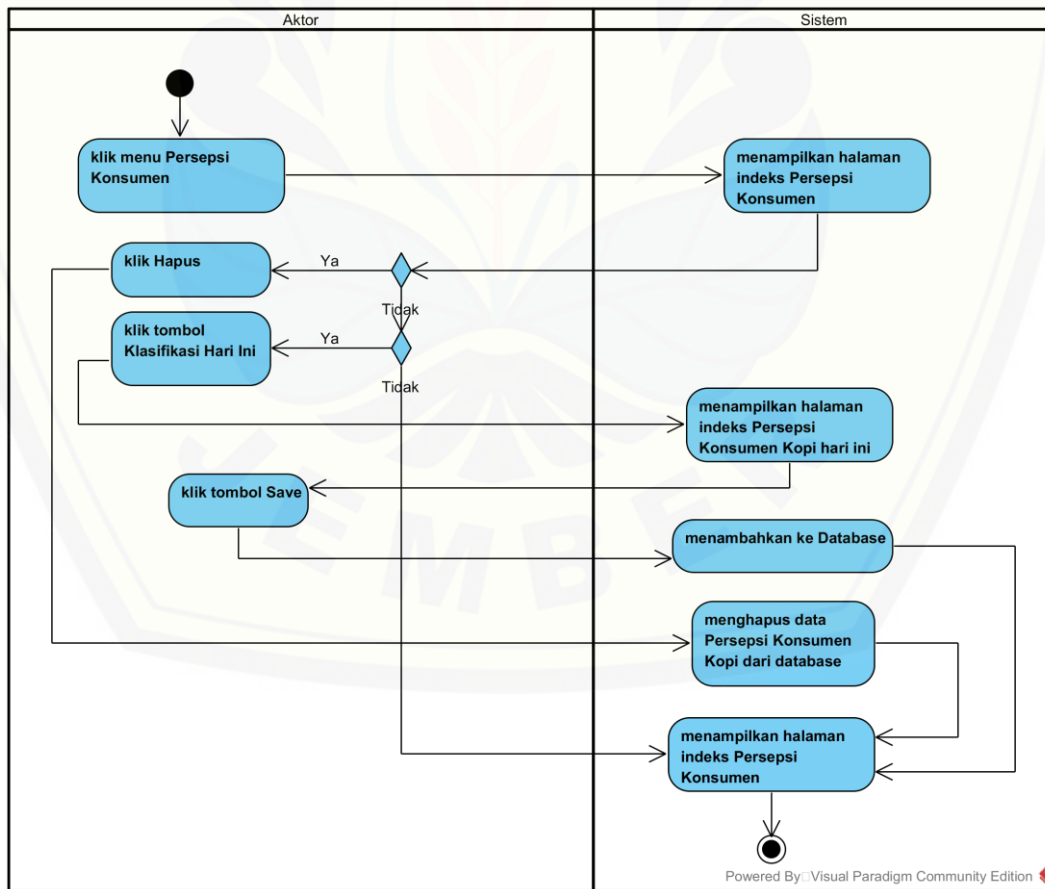
C.10. Activity Diagram Mengelola Ontologi



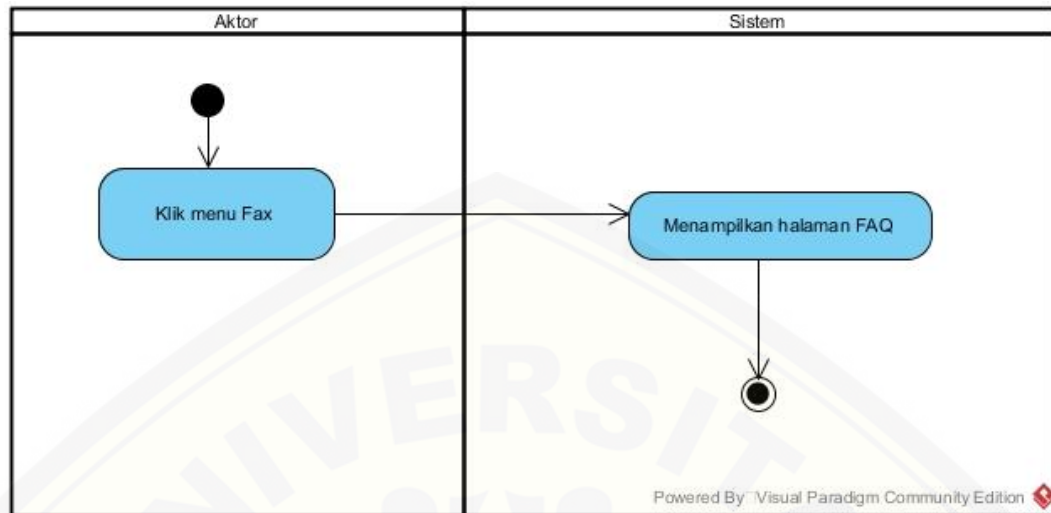
C.11. Activity Diagram Melihat Persepsi Konsumen Kopi Data Testing



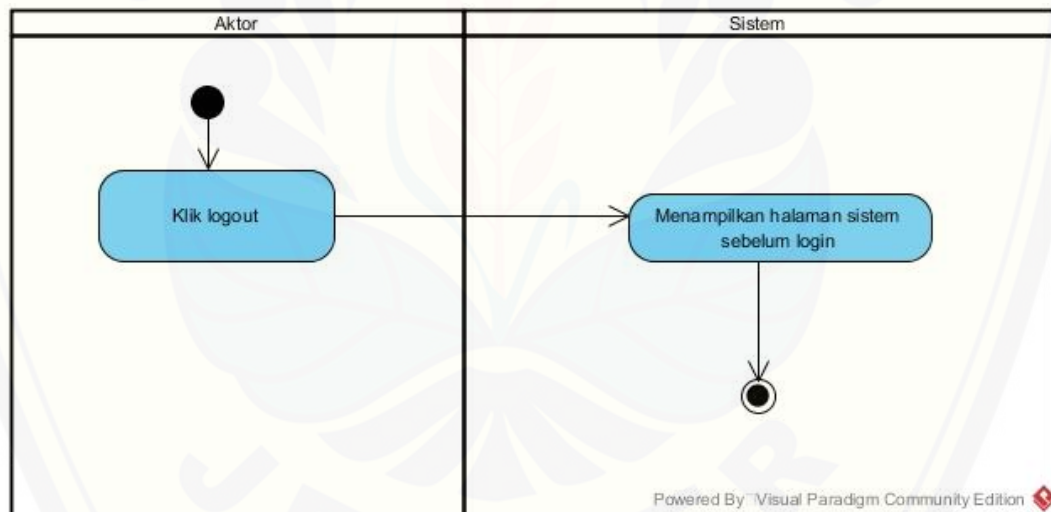
C.12. Activity Diagram Mengelola Data Persepsi Konsumen Kopi



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',
'eyJLZH8xYegJ0h9EIB0m2nsD18shVcJHZtxlNfcHKdzTR5tbyK');
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 Phpm\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,'distribusisentimen'=>$distribusisentimen,
        'distribusicategory'=>$distribusicategory]);
    }

    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;
    }
    $hasilfrequencygram=$this->frequencygram($b);

return
view('datatraining.textpreprocessing',['datatraining'=>$datatraining,'hasilpreprocessing'=>$ha
silpreprocessing,

'hasilstemming'=>$hasilstemming,'hasilstopwordremoval'=>$hasilstopwordremoval,'hasilngra
m'=>$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 frequencygram($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.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 frequencygram($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\NaïveBayes;
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 NaïveBayes();
        $classifierkategori=new NaïveBayes();
        foreach ($hasilngramtraining as $hnt){
            foreach ($hnt as $ss){
                // $train[$i]=$ss;
                $classifiersentimen->train([[ $ss]],[$manualsentimen[$k]]);//menambah
                training
                $classifierkategori->train([[ $ss]],[$manualkategori[$k]]);//menambah
                training
            }
        }
    }
}

```

```

//      $strainlabels=$strainlabels.$manual[$k].", ";
      $i++;
    }
    $k++;
  }

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

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

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

    $klasifikasisentimen[$i]['positif']=0;
    $klasifikasisentimen[$i]['negatif']=0;
    $klasifikasisentimen[$i]['netral']=0;
    $klasifikasisentimen['hasil'][$i]='';
    $klasifikasikategori[$i]['kopi hitam']=0;
    $klasifikasikategori[$i]['kopi robusta']=0;
    $klasifikasikategori[$i]['kopi arabika']=0;
    $klasifikasikategori[$i]['kopi aceh']=0;
    $klasifikasikategori[$i]['kopi toraja']=0;
    $klasifikasikategori[$i]['kopi luwak']=0;
    $klasifikasikategori[$i]['kopi papua']=0;
    $klasifikasikategori[$i]['kopi hijau']=0;
    $klasifikasikategori[$i]['kopi sakit']=0;
    $klasifikasikategori[$i]['kopi pahit']=0;
    $klasifikasikategori[$i]['tidak terkategori']=0;
    foreach ($hnt as $ngram){
      $predictedsentimen = $classifiersentimen->predict([$ngram]);
      $predictedkategori = $classifierkategori->predict([$ngram]);

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

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

      (isset($predictedsentimen['netral']))?$klasifikasisentimen[$i]['netral']+=$predict

```

```

edsentimen['netral']:$klasifikasisentimen[$i]['netral']=0;
    (isset($predictedkategori['kopi hitam']))?$klasifikasikategori[$i]['kopi
hitam']+=$predictedkategori['kopi hitam']:$klasifikasikategori[$i]['kopi
hitam']=0;
    (isset($predictedkategori['kopi
robusta']))?$klasifikasikategori[$i]['kopi robusta']+=$predictedkategori['kopi
robusta']:$klasifikasikategori[$i]['kopi robusta']=0;
    (isset($predictedkategori['kopi
arabika']))?$klasifikasikategori[$i]['kopi arabika']+=$predictedkategori['kopi
arabika']:$klasifikasikategori[$i]['kopi arabika']=0;
    (isset($predictedkategori['kopi aceh']))?$klasifikasikategori[$i]['kopi
aceh']+=$predictedkategori['kopi aceh']:$klasifikasikategori[$i]['kopi aceh']=0;
    (isset($predictedkategori['kopi
toraja']))?$klasifikasikategori[$i]['kopi toraja']+=$predictedkategori['kopi
toraja']:$klasifikasikategori[$i]['kopi toraja']=0;
    (isset($predictedkategori['kopi
luwak']))?$klasifikasikategori[$i]['kopi luwak']+=$predictedkategori['kopi
luwak']:$klasifikasikategori[$i]['kopi luwak']=0;
    (isset($predictedkategori['kopi
papua']))?$klasifikasikategori[$i]['kopi papua']+=$predictedkategori['kopi
papua']:$klasifikasikategori[$i]['kopi papua']=0;
    (isset($predictedkategori['kopi hijau']))?$klasifikasikategori[$i]['kopi
hijau']+=$predictedkategori['kopi hijau']:$klasifikasikategori[$i]['kopi
hijau']=0;
    (isset($predictedkategori['kopi sakit']))?$klasifikasikategori[$i]['kopi
sakit']+=$predictedkategori['kopi sakit']:$klasifikasikategori[$i]['kopi
sakit']=0;
    (isset($predictedkategori['kopi pahit']))?$klasifikasikategori[$i]['kopi
pahit']+=$predictedkategori['kopi pahit']:$klasifikasikategori[$i]['kopi
pahit']=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();
    $reportresultsentimen['f1score']=$reportsentimen->getF1score();
    $reportresultsentimen['support']=$reportsentimen->getSupport();
    $reportresultsentimen['average']=$reportsentimen->getAverage();

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

    $reportresultkategori['precision']=$reportkategori->getPrecision();
    $reportresultkategori['recall']=$reportkategori->getRecall();
    $reportresultkategori['f1score']=$reportkategori->getF1score();
    $reportresultkategori['support']=$reportkategori->getSupport();
    $reportresultkategori['average']=$reportkategori->getAverage();

    return
    view('datatesting.klasifikasi',['datatesting'=>$datatesting,'klasifikasisentimen'
    =>$klasifikasisentimen,'reportresultsentimen'=>$reportresultsentimen,'klasifik
    asikategori'=>$klasifikasikategori,'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 Persepsi Konsumen Kopi *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 Phpm\Metric\ClassificationReport;

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

class DataTestingController extends TextMiningController
{
    public function persepsikonsumenkopi(){
        $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();
        $persepsikonsumenkopi=array();
        foreach ($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;
            $persepsikonsumenkopi[$dp->tanggal]=$positif[$dp->tanggal]/$totalprediksi[$dp-
        >tanggal];
    }
}

```

```

    }
    return
view('datatesting.persepsikonsumenkopi',['persepsikonsumenkopi'=>$persepsikonsumenkopi]
);
}

```

D.13. Kode Program Mengelola Data Persepsi Konsumen Kopi

```

<?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',
'R06seEsYZdaPy2LkpHz54JLhLxBoxsnXg2AdpPuaBipdY'); //isikan dengan
CONSUMER_KEY anda
use Abraham\TwitterOAuth\TwitterOAuth;
use App\Perseption;
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()
    {
        $perseption=Perseption::all();
        return view('streaming.visualisasi',['perseption'=>$perseption]);
    }

    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);
    $hasilbukancorpuستesting=$this->cekCorpus($hasilstemmingtesting);
    $hasilstopwordremovaltesting=$this->stopwordremoval($hasilbukancorpuستesting);
    $hasilngramtesting=$this->ngram($hasilstopwordremovaltesting);

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

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

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

```



```

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

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

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

    arsort($klasifikasimentimen[$i], SORT_NUMERIC);
    $klasifikasimentimen['hasil'][$i]=key($klasifikasimentimen[$i]);

    if ($klasifikasimentimen['hasil'][$i]=='positif'){
        $positif+=1;
    } else if ($klasifikasimentimen['hasil'][$i]=='negatif'){
        $negatif+=1;
    }
    $i++;
}
// dd($tweethariini);
$persepsikonsumenkopi['positif']=$positif;
$persepsikonsumenkopi['negatif']=$negatif;
$persepsikonsumenkopi['hasil']=$positif/($positif+$negatif);
return
view('streaming.index',['tweethariini'=>$tweethariini,'klasifikasimentimen'=>$klasifikasimentimen,'persepsikonsumenkopi'=>$persepsikonsumenkopi]);
}

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['kopi hitam']=Ontology::where('parameter','kopi hitam')->get();
    $ontologi['kopi robusta']=Ontology::where('parameter','kopi robusta')->get();
    $ontologi['kopi arabika']=Ontology::where('parameter','kopi arabika')->get();
    $ontologi['kopi aceh']=Ontology::where('parameter','kopi aceh')->get();
}

```

```

$ontologi['kopi toraja']=Ontology::where('parameter','kopi toraja')->get();
$ontologi['kopi luwak']=Ontology::where('parameter','kopi luwak')->get();
$ontologi['kopi papua']=Ontology::where('parameter','kopi papua')->get();
$ontologi['kopi hijau']=Ontology::where('parameter','kopi hijau')->get();
$ontologi['kopi sakit']=Ontology::where('parameter','kopi sakit')->get();
$ontologi['kopi pahit']=Ontology::where('parameter','kopi pahit')->get();
$ج=0;

foreach ($ontologi as $onto){
    $queryontologi[$ج]=$onto[0]->text;
    for ($i=1;$i<count($onto);$i++){
        $queryontologi[$ج] =$queryontologi[$ج]. ' OR '. $onto[$i]->text;
    }
    $ج++;
}

//
$weethariini=array();
$results=array();
foreach ($queryontologi as $query){
    $results[] = $this->search($query);
}

foreach($results as $result1){
    foreach($result1 as $result2){
        foreach($result2->statuses as $result3) {
            $weethariini[]=$result3;
        }
    }
}
return $weethariini;
}

public function saveindex(){
    $tweet=Perseption::create([
        'index'=>Input::get('persepsikonsumenkopi'),
        'date'=>Input::get('tanggal'),
    ]);
    return Redirect::to('/streaming');
}

function delete($id){
    Perseption::destroy($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. Kode Program *Logout*

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