ANALISIS SENTIMEN PADA TWITTER MENGGUNAKAN *TEXT MINING*

SKRIPSI

Boy Utomo Manalu

071402007

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2014

ANALISIS SENTIMEN PADA TWITTER MENGGUNAKAN *TEXT MINING*

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat mencapai gelar Sarjana Teknologi Informasi

BOY UTOMO MANALU
071402007

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2014

PERSETUJUAN

Judul	:	ANALISIS SENTIMEN PADA TWITTER MENGGUNAKAN *TEXT MINING*
Kategori	:	SKRIPSI
Nama	:	BOY UTOMO MANALU
Nomor Induk Mahasiswa	:	071402007
Program Studi	:	SARJANA (S1) TEKNOLOGI INFORMASI
Departemen	:	TEKNOLOGI INFORMASI
Fakultas	:	ILMU KOMPUTER DAN TEKNOLOGI INFORMASI (FASILKOMTI) UNIVERSITAS SUMATERA UTARA

Diluluskan di
Medan, April 2014

Komisi Pembimbing :

Pembimbing 2	Pembimbing 1
M. Fadly Syahputra, B.Sc, M.Sc.IT	Prof. Dr. Opim Salim Sitompul, M.Sc.
NIP 19830129 200912 1 003	NIP 19610817 198701 1 001

Diketahui/Disetujui oleh
Program Studi S1 Teknologi Informasi
Ketua,

Prof. Dr. Opim Salim Sitompul, M.Sc.
NIP 19610817 198701 1 001

PERNYATAAN

ANALISIS SENTIMEN PADA TWITTER MENGGUNAKAN
TEXT MINING

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, April 2014

Boy Utomo Manalu
071402007

UCAPAN TERIMA KASIH

Alhamdulillah, puji syukur penulis ucapkan kehadirat Allah SWT, serta shalawat dan salam kepada junjungan kita nabi Muhammad SAW, karena atas berkah, rahmat dan hidayahnya penulis dapat menyelesaikan penyusunan skripsi ini. Ucapan syukur yang tidak terhingga kepada Allah SWT yang selalu membimbing dan mengajarkan saya pentingnya kesabaran dan tanggung jawab selama penyusunan skripsi ini.

Dalam penulisan skripsi ini penulis banyak mendapatkan bantuan serta dorongan dari pihak lain. Dalam kesempatan ini dengan segala kerendahan hati, penulis mengucapka terima kasih kepada:

1. Kedua orang tua penulis, yaitu Ayah Drs. A. B. Ch. Manalu, M.Pd beserta Mamak Dra. Rosnah Siregar, M.Si, kepada Kakak-kakak penulis, Syuratti Astuti Rahayu Manalu, S.Pd., M.Hum, Kartika Manalu, M.Pd, dan Salistri Anissa Manalu, S.Pd, M.Hum dan Adik-adik penulis, Bob Rahmat Manalu, S.Pd dan Riza Ramadhan Manalu yang telah memberikan dukungan moril maupun materil kepada penulis selama ini sehingga penulis mampu menyelesaikan skripsi ini. Kepada Lia Silviana, S.TI yang telah memberikan semangat dan bantuannya yang begitu besar sehingga penulis tetap dapat menyelesaikan skripsi ini.
2. Kepada Bapak Prof. Dr. Opim Salim Sitompul, M.Sc, Bapak Muhammad Fadly Syahputra, B.Sc, M.Sc.IT selaku dosen pembimbing penulis yang telah memberikan kritik, saran dan masukan serta bersedia meluangkan waktu, tenaga dan pikiran untuk membantu penulis menyelesaikan skripsi ini.
3. Ketua dan Sekretaris Jurusan Prof. Dr. Opim Salim, M.Sc dan Drs. Sawaluddin, M.IT.
4. Ibu Dra. Elly Rosmaini, M.Si selaku dosen pembimbing akademik saya.
5. Bapak M. Anggia Muchtar, ST, M.MIT dan Bapak Dani Gunawan, ST, MT selaku dosen pembanding dan penguji yang telah banyak memberikan petunjuk, saran dan kritik dalam menyelesaikan skripsi ini.
6. Seluruh Dosen yang mengajar pada program studi Teknologi Informasi Universitas Sumatera Utara.
7. Kepada Staf Tata Usaha Teknologi Informasi dan FASILKOMTI, Roni, Radhy, dan teman-teman Teknologi Informasi stambuk 2007.
8. Seluruh rekan-rekan kuliah sejawat yang tidak dapat disebutkan satu persatu.

Dalam penyusunan skripsi ini penulis menyadari bahwa masih banyak kekurangan, untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun dari semua pihak demi kesempurnaan skripsi ini.

Akhir kata penulis mengharapkan semoga skripsi ini dapat bermanfaat dan membantu semua pihak yang memerlukannya.

ABSTRAK

Twitter salah satu situs *microblogging* memungkinkan penggunanya untuk menulis tentang berbagai topik dan membahas isu-isu yang tejadi pada saat ini. Banyak pengguna yang melakukan *posting* pendapat mereka akan sebuah produk atau layanan yang mereka gunakan. Hal tersebut dapat digunakan sebagai sumber data untuk menilai sentimen pada Twitter. Pengguna sering menggunakan singkatan kata dan ejaan kata yang salah, dimana dapat menyulitkan fitur yang diambil serta mengurangi ketepatan klasifikasi. Dalam penelitian ini penulis menerapkan proses text mining dan proses n-gram karakter untuk seleksi fitur serta menggunakan algoritma Naive Bayes Classifier untuk mengklasifikasi sentimen secara otomatis. Penulis menggunakan 3300 data tweet tentang sentimen kepada provider telekomunikasi. Data tersebut diklasifikasi secara manual dan dibagi kedalam masing-masing 1000 data untuk sentimen positif, negatif dan netral. Kemudian 300 data digunakan untuk testing, dimana tiap sentimen berjumlah 100 tweet. Hasil penelitian ini menghasilkan sebuah sistem yang dapat mengklasifikasi sentimen secara otomatis dengan hasil pengujian 100 tweet mencapai 93 % dengan 2700 data training.

Kata kunci : Twitter, *tweet*, sentimen, *sentiment analysis*, *Naive Bayes Classifier*, *N-gram*

SENTIMENT ANALYSIS ON TWITTER USING TEXT MINING

ABSTRACT

Twitter is a microblogging site allows users to write on various topics and discuss issues that occurred at this time. Many users post their opinion of a product or service that they used. It can be used as a source of data to assess sentiment on Twitter. Users often use abbreviations and wrong spelling words, which can make it difficult for selecting features and reducing the classification accuracy. In this research we apply a text mining and n-grams characters process for selecting feature and using Naive Bayes classifier algorithm for classifying sentiment automatically. We uses the 3300 data of tweets about sentiment to telecommunications providers. The data manually classified and divided into each 1000 data for positive sentiment , negative and neutral. Then 300 of data used for testing, where every sentiment of 100 tweets. The results of this study resulted in a system that can automatically classify sentiment with the test results of 100 tweets reach 93 % in 2700 training data.

Keyword : Twitter, *tweet, sentiment, sentiment analysis, Naive Bayes Classifier, N-gram*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pada saat ini situs *microblogging* telah menjadi alat komunikasi yang sangat populer di kalangan pengguna internet. Dimana jutaan pesan yang muncul setiap hari di situs web populer yang menyediakan layanan *microblogging* seperti Twitter, Tumblr, dan Facebook (Alexa, 2013).

Penulis pesan tersebut menulis tentang kehidupan mereka, berbagi opini tentang berbagai topik dan membahas isu-isu yang tejadi pada saat ini. Format pesan yang bebas dan aksesibilitas dari berbagai platform yang mudah, pengguna internet cenderung untuk beralih dari blog atau milis ke layanan *microblogging* (Agarwal, et al, 2011). Hal tersebut menyebabkan semakin banyak pengguna yang melakukan posting tentang suatu produk dan layanan yang mereka gunakan, atau mengekspresikan pandangan mereka tentang politik dan agama. Twitter sebagai salah satu situs *microblogging* dengan pengguna lebih dari 500 juta dan 400 juta *tweet* perhari (Farber, 2012), memungkinkan pengguna untuk berbagi pesan menggunakan teks pendek disebut *Tweet* (Twitter, 2013). Twitter dapat menjadi sumber data pendapat dan sentimen masyarakat Data tersebut dapat digunakan secara efisien untuk pemasaran atau studi sosial (Pak & Paroubek, 2010).

Sentiment analysis atau *opinion mining* adalah studi komputasional dari opini-opini orang, sentimen dan emosi melalui entitas dan atribut yang dimiliki yang diekspresikan dalam bentuk teks (Liu, 2012). Analisis sentimen akan mengelompokkan polaritas dari teks yang ada dalam kalimat atau dokumen untuk

mengetahui pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif, negatif atau netral (Pang & Lee, 2008).

Text mining adalah salah satu teknik yang dapat digunakan untuk melakukan klasifikasi dokumen dimana t*ext mining* merupakan variasi dari data mining yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar (Feldman & Sanger, 2007).

Analisis sentimen pada Twitter terdapat kelemahan dalam kata-kata yang terdapat pada kalimat yang diposting oleh pengguna situs tersebut. Twitter hanya memungkinkan pengguna menulis sebanyak 140 karakter, hal ini yang menyebabkan para pengguna sering menggunakan singkatan kata dan ejaan kata yang salah.

Cara penulisan yang salah tersebut mengakibatkan terjadi kelemahan pada proses *Text Mining*, dimana dapat menyulitkan fitur yang diambil serta mengurangi ketepatan klasifikasi. Oleh karena itu disini penulis akan menggunakan metode *n-gram* karakter kata untuk mengambil fitur-fitur yang ada pada sebuah *Tweet* yang kemudian akan diklasifikasi dengan Algoritma *Naive Bayes Classifier*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalahnya adalah bagaimana menganalisis sentimen sebuah *tweet* pada Twitter secara otomatis.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengklasifikasikan sentimen pada sebuah *tweet* dengan proses *Text Mining* dan menggunakan metode NBC (*Naive Bayes Classifier)* sehingga bisa mempercepat proses klasifikasi dan mendapatkan kategori sentimen yang sesuai.

1.4 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Mengklasifikasikan sentimen pada Twitter dalam jumlah yang besar secara otomatis.
2. Mencari informasi tentang suatu produk, merek atau tokoh dan menentukan apakah mereka dilihat positif atau negatif di Twitter.

1.5 Batasan Masalah

Agar penyusunan tugas akhir ini tidak keluar dari pokok permasalahan yang dirumuskan, maka ruang lingkup pembahasan dibatasi pada:

1. Algoritma yang digunakan dalam pengklasifikasian ini adalah *Naïve Bayes Classifier* dan tidak membandingkannya dengan algoritma lain.
2. Data yang digunakan terdiri dari *Tweet* provider telekomunikasi berbahasa Indonesia dengan jumlah data yang digunakan 3000 *Tweet*.
3. Proses *Stopword* dan *Stemming* hanya berlaku pada kata-kata berbahasa Indonesia saja.
4. Menggunakan metode *n-gram* kata untuk seleksi fitur karakter kata.
5. Pada tahap proses *Text* Mining pada penelitian ini tidak dilakukan tahap *tagging* atau *Part of Speech Tagging.*

1.6 Metodologi Penelitian

Dalam penelitian ini, penulis melakukan beberapa metode untuk memperoleh data atau informasi dalam menyelesaikan permasalahan. Metode yang dilakukan tersebut antara lain :

1. Studi Literatur

   Dilakukan studi literatur atau studi pustaka yaitu mengumpulkan bahan-bahan referensi baik dari buku, artikel, paper, jurnal, makalah, maupun situs internet.

2 Analisis

Hal-hal yang dilakukan dalam tahap ini adalah :

a. Menganalisis tahap demi tahap dari proses *text mining*.
b. Cara kerja dari algoritma *naïve bayes classifier* dalam mengklasifikasikan *Tweet*.

3 Perancangan

Pada tahap ini dilakukan perancangan arsitektur, perancangan data, dan perancangan antarmuka.

4 Pengkodean

Pada tahap ini akan dilakukan proses implementasi pengkodean program dalam aplikasi komputer menggunakan bahasa pemrograman yang telah ditentukan.

5 Pengujian

Pada tahap ini dilakukan proses pengujian dan percobaan terhadap sistem sesuai dengan spesifikasi yang ditentukan sebelumnya serta memastikan program yang dibuat dapat berjalan seperti yang diharapkan.

6 Penyusunan Laporan

Pada tahap ini dilakukan penulisan dokumentasi hasil analisis dan implementasi.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi menjadi lima bab yaitu :

BAB I Pendahuluan

Bab ini berisikan konsep dasar untuk penyusunan skripsi.

BAB II Landasan Teori

Pada bab ini dibahas beberapa teori yang akan mendukung pembahasan pada bab selanjutnya.

BAB III Analisis dan Perancangan Perangkat Lunak

Pada bab ini dibahas mengenai analisis permasalahan dalam pembuatan aplikasi perangkat lunak serta menjelaskan tentang rancangan struktur program serta merancang interface dari perangkat lunak yang akan dibuat.

BAB IV Implementasi dan Pengujian Perangkat Lunak

Pada bab ini dibahas implementasi dari perangkat lunak yang akan dibuat. Berisikan gambaran antarmuka dari perangkat lunak yang akan dibuat. Selain itu, juga dilakukan pengujian untuk melihat perangkat lunak yang dibuat berhasil dijalankan atau tidak serta untuk menemukan kesalahan (*error*).

BAB V Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 *Text Mining*

Text mining (penambangan teks) adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda (Feldman & Sanger, 2007). *Text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering*, *information extraction* dan *information retrival* (Berry & Kogan, 2010).

Pada dasarnya proses kerja dari *text mining* banyak mengapdopsi dari penelitian *Data Mining* namun yang menjadi perbedaan adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur sedangkan dalam *Data Mining* pola yang diambil dari *database* yang terstruktur (Han & Kamber, 2006). Tahap-tahap t*ext mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman & Sanger 2007, Berry & Kogan 2010) . Dimana penjelasan dari tahap-tahap tersebut adalah sebagai berikut :

2.1.1 Text Preprocessing

Tahap *text preprocessing* adalah tahap awal dari *text mining*. Tahap ini mencakup semua rutinitas, dan proses untuk mempersiapkan data yang akan digunakan pada operasi *knowledge discovery* sistem *text mining* (Feldman & Sanger, 2007). Tindakan yang dilakukan pada tahap ini adalah *toLowerCase*, yaitu mengubah semua karakter huruf menjadi huruf kecil dan *Tokenizing* yaitu proses penguraian deskripsi yang

semula berupa kalimat-kalimat menjadi kata-kata dan menghilangkan delimiter-delimiter seperti tanda titik (.), koma (,), spasi dan karakter angka yang ada pada kata tersebut (Weiss et al, 2005).

2.1.2 Feature Selection

Tahap seleksi fitur (*feature selection*) bertujuan untuk mengurangi dimensi dari suatu kumpulan teks, atau dengan kata lain menghapus kata-kata yang dianggap tidak penting atau tidak menggambarkan isi dokumen sehingga proses pengklasifikasian lebih efektif dan akurat (Do et al, 2006., Feldman & Sanger, 2007., Berry & Kogan 2010). Pada tahap ini tindakan yang dilakukan adalah menghilangkan *stopword* (*stopword removal*) dan *stemming* terhadap kata yang berimbuhan (Berry & Kogan 2010., Feldman & Sanger 2007).

Stopword adalah kosakata yang bukan merupakan ciri (kata unik) dari suatu dokumen (Dragut et al. 2009). Misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagainya. 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*. Daftar kata *stopword* di penelitian ini bersumber dari Tala (2003).

Setelah melalui proses *stopword removal* tindakan selanjutnya adalah yaitu proses *stemming*. *Stemming* adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya *(stem)* (Tala, 2003). Tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun konfiks yang ada pada setiap kata. Jika imbuhan tersebut tidak dihilangkan maka setiap satu kata dasar akan disimpan dengan berbagai macam bentuk yang berbeda sesuai dengan imbuhan yang melekatinya sehingga hal tersebut akan menambah beban *database*. Hal ini sangat berbeda jika menghilangkan imbuhan-imbuhan yang melekat dari setiap kata dasar, maka satu kata dasar akan disimpan sekali walaupun mungkin kata dasar tersebut pada sumber data sudah berubah dari bentuk aslinya dan mendapatkan berbagai macam imbuhan. Karena bahasa Indonesia mempunyai aturan morfologi maka proses *stemming* harus berdasarkan aturan morfologi bahasa Indonesia.

Berdasarkan penelitian sebelumnya, ada beberapa algoritma stemming yang bisa digunakan untuk *stemming* bahasa Indonesia diantaranya algoritma *confix-stripping*, algoritma Porter *stemmer* bahasa Indonesia, algoritma Arifin dan Sutiono, dan Algorima Idris (Tala 2003, Agusta 2009, Asian et al 2005, Adriani et al 2007). Dimana, Algoritma *confix-stripping* adalah algoritma yang akurat dalam *stemming* bahasa Indonesia (Tala 2003, Agusta 2009, Asian et al 2005, Adriani et al 2007).

2.2 *Sentiment Analysis*

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining* yang bertujuan menganlisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik , produk, layanan, organisasi, individu, ataupun kegiatan tertentu (Liu, 2011).

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

Kita dapat mencari pendapat tentang produk-produk, merek atau orang-orang dan menentukan apakah mereka dilihat positif atau negatif di web (Saraswati, 2011). Hal ini memungkinkan kita untuk mencari informasi tentang:

a. Deteksi Flame (rants buruk)
b. Persepsi produk baru.
c. Persepsi Merek.
d. Manajemen reputasi.

Ekspresi atau *sentiment* mengacu pada fokus topik tertentu, pernyataan pada satu topik mungkin akan berbeda makna dengan pernyataan yang sama pada *subject* yang berbeda. Oleh karena itu pada beberapa penelitian, terutama pada review produk, pekerjaan didahului dengan menentukan elemen dari sebuah produk yang sedang dibicarakan sebelum memulai proses *opinion mining* (Barber, 2010).

2.3 Twitter

Twitter adalah sebuah situs web yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan *Tweets* (Twitter, 2013). Mikroblog adalah salah satu jenis alat komunikasi online dimana pengguna dapat memperbarui status tentang mereka yang sedang memikirkan dan melakukan sesuatu, apa pendapat mereka tentang suatu objek atau fenomena tertentu. *Tweets* adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. *Tweets* bisa dilihat secara publik, namun pengirim dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat *Tweets* pengguna lain yang dikenal dengan sebutan pengikut (*follower).*

Tidak seperti Facebook, LinkedIn, dan MySpace, Twitter merupakan sebuah jejaring sosial yang dapat digambarkan sebagai sebuah graph berarah (Wang, 2010), yang berarti bahwa pengguna dapat mengikuti pengguna lain, namun pengguna kedua tidak diperlukan untuk mengikutinya kembali. Kebanyakan akun berstatus publik dan dapat diikuti tanpa memerlukan persetujuan pemilik..

Semua pengguna dapat mengirim dan menerima *Tweets* melalui situs Twitter, aplikasi eksternal yang kompatibel (telepon seluler), atau dengan pesan singkat (SMS) yang tersedia di negara-negara tertentu (Twitter, 2013). Pengguna dapat menulis pesan berdasarkan topik dengan menggunakan tanda # (*hashtag*). Sedangkan untuk menyebutkan atau membalas pesan dari pengguna lain bisa menggunakan tanda @.

Pesan pada awalnya diatur hanya mempunyai batasan sampai 140 karakter disesuaikan dengan kompatibilitas dengan pesan SMS, memperkenalkan singkatan notasi dan slang yang biasa digunakan dalam pesan SMS. Batas karakter 140 juga meningkatkan penggunaan layanan memperpendek URL seperti bit.ly, goo.gl, dan tr.im, dan jasa hosting konten, seperti Twitpic, Tweephoto, memozu.com dan NotePub untuk mengakomodasi multimedia isi dan teks yang lebih panjang daripada 140 karakter (Twitter, 2013). Twitter menggunakan bit.ly untuk memperpendek otomatis semua URL yang dikirim-tampil. Fitur yang terdapat dalam Twitter, antara lain:

1. **Laman Utama (*Home)***

   Pada halaman utama kita bisa melihat *Tweets* yang dikirimkan oleh orang-orang yang menjadi teman kita atau yang kita ikuti (*following*).

2. **Profil (*Profile*)**

   Pada halaman ini yang akan dilihat oleh seluruh orang mengenai profil atau data diri serta *Tweets* yang sudah pernah kita buat.

3. ***Followers***

   Pengikut adalah pengguna lain yang ingin menjadikan kita sebagai teman. Bila pengguna lain menjadi pengikut akun seseorang, maka *Tweets* seseorang yang ia ikuti tersebut akan masuk ke dalam halaman utama.

4. ***Following***

   Kebalikan dari pengikut, *following* adalah akun seseorang yang mengikuti akun pengguna lain agar *Tweets* yang dikirim oleh orang yang diikuti tersebut masuk ke dalam halaman utama.

5. ***Mentions***

   Biasanya konten ini merupakan balasan dari percakapan agar sesama pengguna bisa langsung menandai orang yang akan diajak bicara.

6. ***Favorite***

   Tweets ditandai sebagai favorit agar tidak hilang oleh halaman sebelumnya.

7. **Pesan Langsung (*Direct Message*)**

   Fungsi pesan langsung lebih bisa disebut SMS karena pengiriman pesan langsung di antara pengguna.

8. ***Hashtag***

   Hashtag "#" yang ditulis di depan topik tertentu agar pengguna lain bisa mencari topik yang sejenis yang ditulis oleh orang lain juga

9. *List*

Pengguna Twitter dapat mengelompokkan ikutan mereka ke dalam satu grup sehingga memudahkan untuk dapat melihat secara keseluruhan para nama pengguna (*username*) yang mereka ikuti (*follow*).

10. Topik Terkini (*Trending Topic*)

Topik yang sedang banyak dibicarakan banyak pengguna dalam suatu waktu yang bersamaan.

2.4 Algoritma Confix-stripping

Algoritma *Confix-stripping* mempunyai aturan imbuhan sendiri dengan model sebagai berikut (Adriani et al, 2007) :

[[[AW +]AW +]AW +] Kata-Dasar [[+AK][+KK][+P] (2.1)

AW : Awalan AK : Akhiran

KK : Kata ganti kepunyaan P : Partikel

Tanda kurung besar menandakan bahwa imbuhan adalah opsional.

Dalam algoritma *confix-stripping* ada beberapa kombinasi awalan dan akhiran yang tidak diperbolehkan, yaitu kombinasi awalan dan akhiran yang ada dalam tabel 2.1. Namun ada satu pengecualian pada kombinasi perfiks “ke-“ dan sufiks “-i” yang bboleh diterapkan pada kata “tahu” menjadi kata “ketahui”.

Tabel 2.1 Kombinasi Prefix dan Sufiks yang tidak diperbolehkan

Awalan (Prefiks)	Akhiran (Suffiks)
be-	-i
di-	-an
ke-	-i –kan
me-	-an
se-	-i –kan
te-	-an

2.4.1 Aturan peluruhan kata dasar

Ada beberapa kata dasar yang apabila dilekati oleh awalan “me(N)-“,”pe(N)-“,pe(R)-”te(R)-“,”be(R)-“ akan mengalami peluruhan atau perubahan pada karakter awal dari kata dasar tersebut (Kridalaksana, 2009). Sebagai contoh kata “tanya”, karakter awal dari kata “tanya” akan berubah apabila ditambahkan awalan “me-“ dan menjadi “menanya”. Begitu juga untuk beberapa kata dasar lainnya.

Untuk melakukan proses *stemming* pada kata-kata tersebut harus mengikuti aturan peluruhan yang telah ditetapkan oleh algoritma (Adriani et al, 2007). Aturan-aturan tersebut dijelaskan pada tabel 2.2.

Tabel 2.2 Aturan peluruhan kata dasar (Adriani et al, 2007)

Aturan	Bentuk Awalan	Peluruhan
1	berV…	ber-V… | be-rV…
2	belajar…	bel-ajar
3	beC_1erC_2…	be-C_1erC_2…dimana C_1!={'r'|'l'}
4	terV…	ter-V…|te-rV…
5	terCer…	ter-Cer…dimana C!='r'
6	teC_1erC_2	te-C_1erC_2…dimana C_1!='r'
7	me{l|r|w|y}V…	me-{l|r|w|y}V…
8	mem{b|f|v}…	mem-{b|f|v}…
9	mempe…	mem-pe…
10	mem{rV|V}…	me-m{rV|V}…|me-p{rV|V}…
11	men{c|d|j|z}…	men-{c|d|j|z}…
12	menV…	me-nV…|me-tV…
13	meng{g|h|q|k }…	meng-{g|h|q|k}…
14	mengV…	meng-V…|meng-kV…

Tabel 2.2 Aturan peluruhan kata dasar (Adriani et al, 2007) (Lanjutan)

Aturan	Bentuk Awalan	Peluruhan
15	mengeC	menge-C
16	menyV…	me-ny…| meny-sV…
17	mempV…	mem-pV…
18	pe{w|y}V…	pe-{w|y}V…
19	perV…	per-V…|pe-rV…
20	pem{b|f|v}…	pem-{b|f|v}…
21	pem{rV|V}…	pe-m{rV|V}…|pe-p{rV|V}
22	pen{c|d|j|z}…	pen-{c|d|j|z}…
23	penV…	pe-nV…|pe-tV…
24	peng{g|h|q}	peng-{g|h|q}
25	pengV	peng-V|peng-kV
26	penyV…	pe-nya|peny-sV
27	pelV..	pe-lV…; kecuali untuk kata "pelajar" menjadi "ajar"
28	peCP	pe-CP…dimana C!={r|w|y|l|m|n} dan P!='er'
29	perCerV	Per-CerV… dimana C!={r|w|y|l|m|n}

Pada tabel 2.2 dapat dilihat aturan-aturan peluruhan kata dasar yang apabila dilekati oleh awalan “me-“,”be-“,”te-“,”pe-“. Dimana pada kolom kedua dari tabel tersebut menjelaskan bentuk-bentuk kata dasar yang dilekati awalan “me-“,”be-“,”te-“,”pe-“, sedangkan pada kolom ketiga menjelaskan perubahan-perubahan karakter pada kata dasar yang mungkin terjadi apabila algoritma telah menghilangkan awalan yang telah melekati kata dasar tersebut. Huruf “V” pada tabel tersebut menunjukkan huruf hidup atau huruf vocal, huruf “C” menunjukkan huruf mati atau konsonan , huruf “A” menunjukkan huruf vocal atau huruf konsonan dan huruf “P” menunjukkan pecahan “er”. Sebagai contoh, jika algoritma menerima kata “menyusun”, maka proses *stemming* pada kata tersebut mengikuti aturan ke-16 pada tabel 2.2 yaitu “menyV…” dan perubahan menjadi “me-ny” atau “meny-sV…”. Berdasarkan aturan tersebut maka algoritma akan menghilangkan awalan “me-“ maka akan didapatkan kata “nyusun”, selanjutnya kata “nyusun” akan diperiksa ke dalam *database* kata dasar karena kata “nyusun” bukan kata kata dasar maka tahap selanjutnya algoritma akan menghilangkan kata“meny-“ dan kemudian algoritma akan menambahkan huruf “s” di depan huruf “u”, maka akan didapatkan kata “susun”, selanjutnya kata “susun” akan diperiksa kedalam *database* kata dasar. Karena kata “susun” merupakan kata dasar maka kata tersebut akan diidentifikasikan sebagai kata dasar.

2.5 Morfologi

Morfologi adalah bidang linguistik yang mempelajari morfem dan kombinasi-kombinasinya atau bagian struktur bahasa yang mencakup kata dan bagian-bagian kata, yaitu morfem (Kridalaksana 2009, Muslich 2008). Sedangkan morfem adalah bentuk bahasa yang terkecil yang tidak dapat lagi dibagi menjadi bagian-bagian yang lebih kecil (Alwi et al 2003, Muslich 2008). Misalnya kata “putus”,”me-“,”-kan”, kata tersebut disebut morfem karena tidak dapat dibagi lagi menjadi bagian yang lebih kecil. Morfem terdiri dari 2 bagian yaitu morfem bebas dan morfem terikat (Alwi et al 2003, Muslich 2008), dimana morfem bebas adalah morfem yang dapat berdiri sendiri sedangkan morfem terikat adalah morfem yang tidak dapat berdiri sendiri. Contohnya seperti pada kalimat “Andi memperbesar volume radio”. Pada kalimat tersebut “besar” merupakan morfem bebas karena jika dipecah akan tetap memiliki makna. Sementara itu “mem-“,”per-“ merupakan morfem terikat karena kedua morfem tersebut akan bermakna jika dilekatkan pada bentuk lain.

2.5.1 Proses Morfologi

Proses morfologi adalah proses pembentukkan kata-kata dengan menghubungkan morfem yang satu dengan morfem yang lain (Alwi et al 2003, Muslich 2008, Kridalaksana 2009). Dalam bahasa Indonesia terdapat tiga proses morfologi yaitu proses pembubuhan afiks (afiksasi), proses pengulangan (reduplikasi), dan proses pemajukan. Namun, dalam penelitian ini hanya akan dibahas proses pembubuhan afiks (afiksasi).

2.5.1.1 Afiksasi

Afiksasi adalah proses pembubuhan afiks pada kata dasar (Kridalaksana 2009). Afiks atau imbuhan dalam bahasa Indonesia terdiri atas prefix (awalan), infiks (sisipan), sufiks (akhiran), konfiks (awalan dan akhiran) (Alwi et al 2003, Muslich 2008, Kridalaksana 2009). Penjelasan dari setiap bagian afiks tersebut adalah sebagai berikut:

2.5.1.2 Awalan (Prefiks)

Prefiks atau awalan adalah afiks yang di tempatkan di bagian depan suatu kata dasar. Prefiks dalam bahasa Indonesia terdiri atas :

i. Prefiks be(R)-

Bentuk prefiks “ber-“ ada tiga macam, yaitu “ber-“,”be-“,” dan “bel-“. Bentuk prefiks “ber-“ tidak akan berubah menjadi “be-“ atau “bel-“ apabila satuan dasar kata bentukannya tidak diawali huruf “r”, suku kata awalnya tidak berakhir dengan “er”, dan bukan bergabung dengan kata dasar “ajar”.

Contoh :

ber- + lari => berlari

ber- + agama => beragama

ber- + dua => berdua

ber- + kurang => berkurang

ii. Prefiks me (N)-

Prefiks “me (N)-“ mempunyai beberapa variasi, yaitu “mem-“,”men-“,”meny-“,“meng-“,”menge-“ dan “me-“. Prefiks “me(N)- berubah menjadi mem- jika bergabung dengan kata yang diawali huruf “b”,”f”,”v” dan “p”.

Contoh :

me(N)- + baca => membaca

me(N)- + pukul => memukul

Prefiks “me(N)-“ berubah menjadi “men-“ jika bergabung dengan kata yang diawali oleh huruf “d”,”t”,”j” dan “c”.

Contoh :

me(N)- + data => mendata

me(N)- + tulis => menulis

me(N)- + jadi => menjadi

me(N)- + cuci => mencuci

Prefiks “me(N)-“ berubah menjadi “meny-“ jika bergabung dengan kata yang diawali oleh huruf “s”.

Contoh :

me(N)- + sapu => menyapu

Prefiks “me(N)-“ berubah menjadi “meng-“ jika bergabung dengan kata yang diawali dengan huruf “k”, “g”, dan “h”.

Contoh :

me(N)- + kupas => mengupas

me(N)- + hitung => menghitung

me(N)- + goreng => menggoreng

Prefiks “me(N)-” berubah menjadi “menge-“ jika bergabung dengan kata yang terdiri dari satu suku kata.

Contoh :

me(N)- + bor => mengebor

me(N)- + bom => mengebom

me(N)- + cek => mengecek

Prefiks “me(N)-“ berubah menjadi “me-“ jika bergabung dengan kata yang diawali dengan huruf “r”,”l”,”ny”,”m”,”n”,”ng”,”w” dan “y”.

Contoh :

me(N)- + rusak => merusak

me(N)- + lempar => melempar

me(N)- + nyanyi => menyanyi

me(N)- + merah => memerah

me(N)- + naik => menaik

me(N)- + ngangah => mengangah

me(N)- + wujudkan => mewujudkan

me(N)- + yakini => meyakini

iii. Prefiks pe(R)-

Prefiks “pe(R)-“ identik dengan prefik “ber-“. Perhatikan contoh berikut :

berawat => perawat

bekerja => pekerja

Prefiks “pe(R)-“ mempunyai variasi “pe-“,”per-“, dan”pel-“. Prefiks “pe(R)-“ berubah menjadi “pe-“ jika bergabung dengan kata yang diawali huruf “r” dan kata yang suku kata pertamanya berakhiran “er”.

Contoh :

pe(R)- + rawat => perawat

pe(R)- + kerja => pekerja

Prefiks “pe(R)-“ berubah menjadi “pel-“ jika bergabung dengan kata “ajar”.

Contoh :

pe(R)- + ajar => pelajar

Prefiks “pe(R)-“ berubah menjadi “per-“ bila bergabung dengan kata dasar yang tidak berawalan “r”,”l”, dan kata yang suku pertamanya tidak berakhiran “er”.

iv. Prefiks pe (N)-

Prefiks “pe(N)” mempunyai beberapa variasi. Prefiks “pe(N)-“ sejajar dengan prefiks “me(N)-“. Variasi “pe(N)-“ memiliki variasi “pem-“,”pen-“,”peny-“,”peng-“,”pe-“, dan “penge-“.

Prefiks “pe(N)-“ berubah menjadi “pen-“ jika bergabung dengan kata yang diawali oleh huruf “t”,”d”,”c” dan “j”.

Contoh :

penuduh

pendorong

pencuci

penjudi.

Prefiks “pe(N)-“ berubah menjadi “pem-“ jika bergabung dengan kata yang diaawali oleh huruf “b” dan “p”.

Contoh :

pembaca

pemukul

Prefiks “pe(N)-“ berubah menjadi “peny-“ jika bergabung dengan kata yang diawali oleh huruf “s”.

Contoh :

penyapu

Prefiks “pe(N)-“ berubah menjadi “peng-“ jika bergabung dengan kata yang diawali oleh huruf “g” dan “k”.

Contoh :

penggaris

pengupas

Prefiks “pe(N)-“ berubah menjadi “penge-“ jika bergabung dengan kata yang terdiri atas satu suku kata.

Contoh :

pengebom

pengecat

Prefiks “pe(N)-“ berubah menjadi “pe-“ jika bergabung dengan kata yang diawali oleh huruf “r”,”l”,”ny”,”m”,”n”,”ng”,”w” dan “y”.

Contoh :

pemarah

pelupa

perasa

v. Prefiks te(R)-

Bentuk prefiks “te(R)-“ berubah menjadi “ter-“ apabila bergabung denga kata dasar yang mempunyai huruf awal bukan “r”.

Contoh :

te(R)- + ambil => terambil

te(R)- + kuasai => terkuasai

te(R)- + isi => terisi

Bentuk prefiks “te(R)-“ akan berubah menajdi “te-“ apabila bergabung dengan kata dasar yang huruf awalnya dala “r”.

Contoh :

te(R)- + rabah => terabah

te(R)- + rendah => terendah

vi. Prefiks di-

Prefiks “di-“ hanya memiliki satu bentuk yaitu “di-“ dan tidak akan mengalami perubahan jika digabung dengan kata dasar apapun.

Contoh:

di- + tarik => ditarik

di- + kurung => dikurung

di- + ambil => diambil

vii. Prefiks ke-

Prefiks “ke-“ hanya memiliki satu bentuk yaitu “ke-“ dan tidak akan mengalami perubahan jika digabung dengan kata dasar apapun.

Contoh:

ke- + tua => ketua

ke- + hendak => kehendak

viii. Prefiks se-

Prefiks “se-“ memiliki dua macam bentuk yaitu “se-“ dan “sen-“. Prefiks “se-“ akan berubah menjadi “sen-“ apabila bergabung dengan kata dasar “diri” yaitu menajdi “sendiri”.

Contoh:

se- + buah => sebuah

se- + lembar => selembar

se- + piring => sepiring

2.5.1.3 Sisipan (Infiks)

Sisipan atau infiks adalah afiks yang disisipkan ditengah kata dasar. Ada 4 infiks dalam Bahasa Indonesia, yaitu “-el-“,”-em-“,”-in-“ dan “-er-”. Contoh :

-el- + getar => geletar

-em- + getar => gemetar

-er- + gigi => gerigi

-in- + kerja => kinerja

2.5.1.4 Akhiran (Sufiks)

Akhiran atau sufiks adalah afiks yang ditempatkan di bagian belakang kata dasar. Sufiks dalam Bahasa Indonesia adalah “-i”,”-an”, dan “-kan”,”-kah”,”-lah”,”-pun”,”-ku”,”-mu”,”-nya”. Dimana akhiran “-kah”,”-lah”,”-pun” termasuk dalam partikel penegasan dan akhiran ”-ku”,”-mu”,”-nya” termasuk dalam kata ganti kepunyaan Contoh :

-i + basah => basahi

-an + minum => minuman

-kan + ambil => ambilkan

-lah + biar => biarlah

-pun + apa => apapun

-kah + mana => manakah

-tah + apa => apatah

-nya + nama => namanya

-ku + milik => milikku

-mu + diri => dirimu

2.5.1.5 Konfiks

Konfiks adalah afiks yang berupa morfem terbagi, yang bagian pertama berposisi pada awal kata dasar, dan bagian yang kedua berposisi pada akhir bentuk dasar dimana proses pengimbuhannya dilakukan secara bersamaan Konfiks dalam bahasa Indonesia adalah “per-/-an”, “ke-/-an”, “ ber-/-an”. Contoh :

per-/-an => pertempuran

ke-/-an => keadaan

ber-/-an => bermunculan

2.6 N-gram

N-gram adalah potongan n karakter dalam suatu string tertentu atau potongan n kata dalam suatu kalimat tertentu (Cavnar & Trenkle, 1994). Misalnya dalam kata "Teknik" akan didapatkan n-gram sebagai berikut.

Tabel 2.3 Contoh pemotongan N-gram berbasis karakter

Nama	n-gram karakter
Uni-gram	T, E, K, N, I, K
Bi-gram	_T, TE, EK, KN, NI, IK, K_
Tri-gram	_TE, TEK, EKN, KNI, NIK, IK_, K_ _
Quad-gram	_TEK, TEKN, EKNI, KNIK, NIK_, IK_ _, K_ _ _

Karakter blank "_" digunakan untuk merepresentasikan spasi di depan dan diakhir kata. Dan untuk word-based n-gram contohnya adalah sebagai berikut.
Kalimat : "N-gram adalah potongan n karakter dalam suatu string tertentu"

Tabel 2.4 Contoh pemotongan N-gram berbasis kata

Nama	n-gram kata
Uni-gram	n-gram, adalah, potongan, n, karakter, dalam, suatu, sring, tertentu
Bi-gram	n-gram adalah, adalah potongan, potongan n, n karakter, karakter dalam, dalam suatu, suatu string, string tertentu
Tri-gram	n-gram adalah potongan, adalah potongan n, potongan n karakter, n karakter dalam, karakter dalam suatu, dalam suatu string, suatu string tertentu
Dst…	

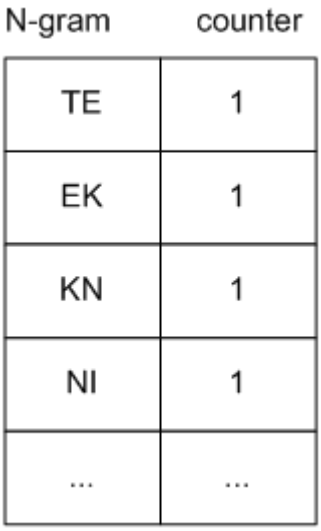
2.7.1 N-Gram Based Text Categorization

Bahasa manusia memiliki beberapa kata yang muncul (digunakan) lebih sering dibandingkan dengan kata yang lain.

2.7.1.1 Learning

Setelah dilakukan *preprocessing* terhadap dokumen-dokumen dalam training set, maka selanjutnya dilakukan *learning* terhadap dokumen-dokumen tersebut. Langkah-langkah *learning* yang dilakukan adalah sebagai berikut :

- Fitur-fitur (token) yang telah didapatkan ditransformasikan ke dalam bentuk n-gram dengan n = 2, 3, dan 4.
- Masukkan tiap-tiap n-gram yang telah didapatkan dalam suatu tabel *hash* sebagai counter untuk menghitung frekuensi n-gram dalam dokumen. Tabel *hash* tersebut menggunakan mekanisme penanganan duplikasi konvensional untuk menjamin bahwa setiap n-gram memiliki counter-nya masing-masing. Contoh implementasi dari mekanisme ini dijelaskan dalam gambar di bawah ini.



Gambar 2.1 Contoh penggunaan tabel hash

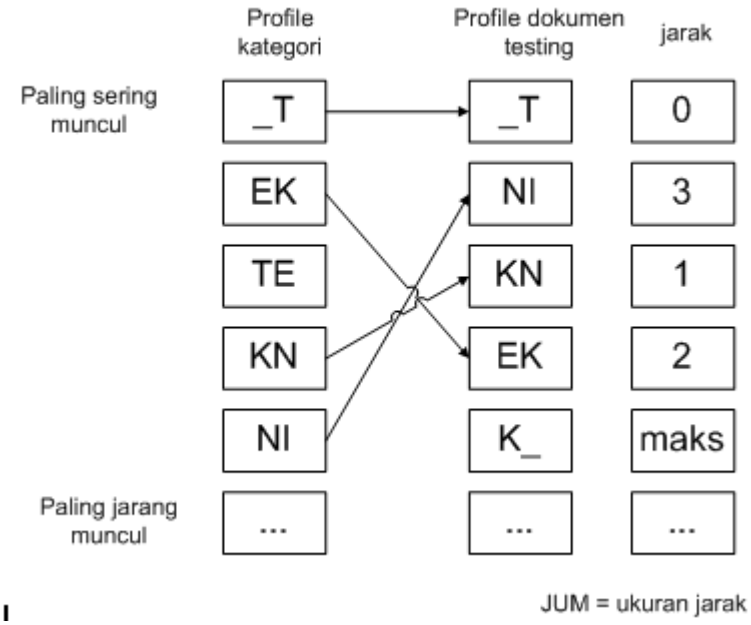
Ketika muncul n-gram “TE” lagi, maka frekuensi (counter) “TE” ditambah 1, tidak lagi ditambahkan baris baru dalam tabel hash tersebut. Sehingga duplikasi dapat dicegah. Setelah semuanya dihitung, keluarkan semua *N-gram* beserta jumlah kemunculannya. Urutkan *N-gram* dalam urutan terbalik berdasarkan jumlah kemunculannya.

Hasil akhir dari proses diatas adalah *N-gram frequency profile* dari dokumen. Setelah didapatkan *N-gram* frequency profile dari dokumen (per kategori dalam *training* set), untuk *testing*-nya maka dilakukan pengukuran jarak profil kategori dengan profil dokumen yang akan diketahui kategorinya.

2.7.1.2 Testing

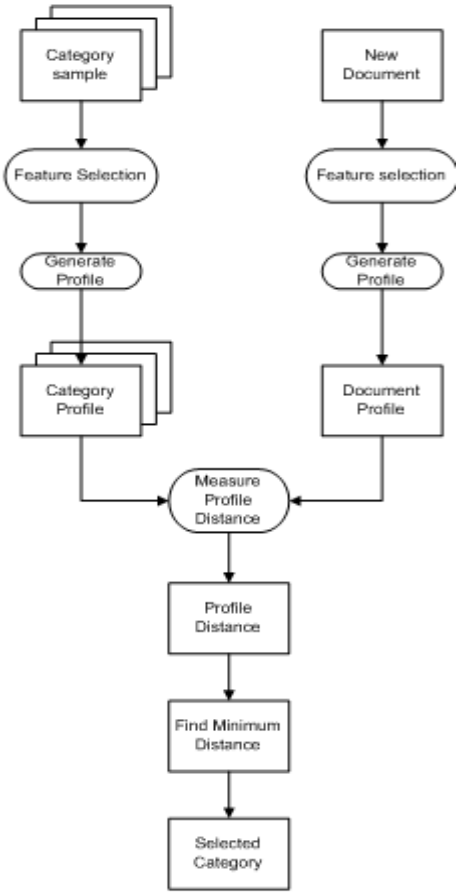
Seperti yang telah dijelaskan dalam sebelumnya, untuk melakukan testing terhadap sebuah dokumen, maka dilakukan langkah-langkah seperti pada proses *training* terhadap dokumen dalam test-set. Dengan demikian, didapatkan *N-gram frequency*

profile untuk dokumen *testing*. Kemudian langkah yang selanjutnya untuk mengetahui kategori dari dokumen *testing* adalah dengan menghitung jarak antara profil dokumen *testing* dengan profil dari masing-masing kategori dalam dokumen *training*. Pengukuran jarak (*distance measure*) dilakukan dengan mekanisme *out-of-place measure*. Cara kerja mekanisme ini adalah sebagai berikut. Untuk setiap *N-gram* dalam profil dalam dokumen *testing*, temukan profil yang sama pada profil kategori dalam dokumen *training*. Kemudian hitung seberapa jauh profil tersebut dari tempat yang seharusnya jika dokumen tersebut termasuk dalam suatu kategori. Untuk lebih jelasnya, dapat dilihat dalam gambar dibawah ini.



Gambar 2.2 Contoh penghitungan jarak dengan mekanisme *out-of-place measure*

N-gram yang muncul dalam dokumen *testing* namun tidak muncul dalam profil kategori diberi jarak maksimal yaitu jumlah keseluruhan *N-gram* yang terbentuk. Kategori dari dokumen *testing* tersebut merupakan kategori dengan ukuran jarak (*distance measure*) terkecil. Sebagai catatan, profil diatas hanya untuk menjelaskan saja, dan bukan refleksi dari *N-gram frequency statistic* yang sebenarnya. Proses kategorisasi teks secara umum dapat dilihat pada gambar di bawah ini.



Gambar 2.3 Gambaran umum kategorisasi teks dengan menggunakan *N-gram* (Cavnar & Trenkle, 1994).

2.7 Naïve Bayes Classifier

Algoritma *naive bayes classifier* merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger 2007). Dalam penelitian ini yang menjadi data uji adalah dokumen *weets*. Ada dua tahap pada klasifikaasi dokumen. Tahap pertama adalah pelatihan terhadap dokumen yang sudah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi dokumen yang belum diketahui kategorinya.

Dalam algoritma *naïve bayes classifier* setiap dokumen direpresentasikan dengan pasangan atribut "$x_1, x_2, x_3,...x_n$" dimana x_1 adalah kata pertama, x_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan kategori *Tweet*. Pada saat

klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}), dimana persamaannya adalah sebagai berikut :

$$V_{MAP} = \underset{V_j \, e \, V}{\arg\max} \frac{P(x_1,x_2,x_3,...x_n|V_j)P(V_j)}{P(x_1,x_2,x_3,...x_n)} \quad (2.2)$$

Untuk $P(x_1, x_2, x_3,...x_n)$ nilainya konstan untuk semua kategori (V_j) sehingga persamaan dapat ditulis sebagai berikut :

$$V_{MAP} = \underset{V_j \, e \, V}{\arg max} P(x_1, x_2, x_3, \dots x_n|V_j)P(V_j) \quad (2.3)$$

Persamaan diatas dapat disederhanakan menjadi sebagai berikut :

$$V_{MAP} = \underset{V_j \, e \, V}{\arg max} \prod_{i=1}^{n} P(x_i|V_j)P(V_j) \quad (2.4)$$

Keterangan :

V_j = Kategori *tweet* j =1, 2, 3,…n. Dimana dalam penelitian ini j_1 = kategori *tweet* sentimen negatif, j_2 = kategori *tweet* sentimen positif, dan j_3 = kategori *tweet* sentiment netral

$P(x_i|V_j)$ = Probabilitas x_i pada kategori V_j

$P(V_j)$ = Probabilitas dari V_j

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

$$P(V_j) = \frac{|docs\ j|}{|contoh|} \quad (2.5)$$

$$P(x_i|V_j) = \frac{n_k + 1}{n + |kosakata|} \quad (2.6)$$

Keterangan :

|docs j| = jumlah dokumen setiap kategori j

|contoh| = 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

2.8 Unified Modelling Language (UML)

Unified Modelling Language adalah sebuah "bahasa" yang telah menjadi standard industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Dharwiyanti dan Wahono, 2003). Dengan menggunakan UML kita dapa membuat model untuk semua jenis aplikasi piranti lunak dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan kelas dan operasi dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek. Tujuan perancangan UML adalah sebagai berikut (Hariyanto, 2004) :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap untuk mengembangkan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen kolaborasi, *framework* dan *patern*.

Unified Modeling Language (UML) menyediakan sejumlah diagram untuk menggambarkan pemodelan berorientasi objek yang dilakukan. UML membagi diagram menjadi dua tipe yaitu :

1. Diagram Struktur

Diagram ini untuk menvisualisasi, menspesifikasi, membangun dan mendokumentasi aspek atatik dari sistem. Diagram struktur di UML terdiri dari :

a. Diagram kelas (*Class diagram*)
b. Diagram objek (*Object diagram*)
c. Diagram komponen (*Component diagram*)
d. Diagram *deployment* (*Deployment Diagram*)

2. Diagram Perilaku

Diagram ini untuk menvisualisasi, menspesifikasi, membangun dan mendokumentasi aspek dinamis dari sistem. Diagram perilaku di UML tediri dari :

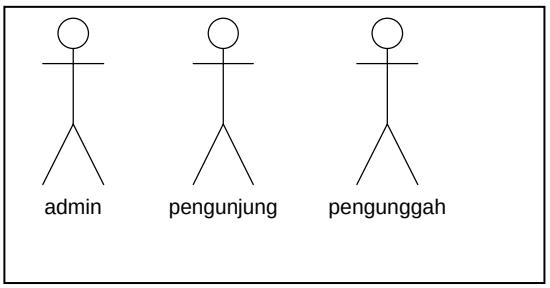
a. Diagram use-case (*Use case diagram*)
b. Diagram sekuen (*Sequence diagram*)
c. Diagram kolaborasi (*Colaboration diagram*)
d. Diagram statechart (S*tatechart diagram*)
e. Diagram aktifitas (A*ctivity diagram*)

2.8.1 Diagram Use case

Diagram use case merupakan salah satu diagram untuk memodelkan aspek perilaku sistem atau digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem (Hariayanto, 2004). Diagram use case terdiri dari beberapa elemen yaitu :

1. Aktor

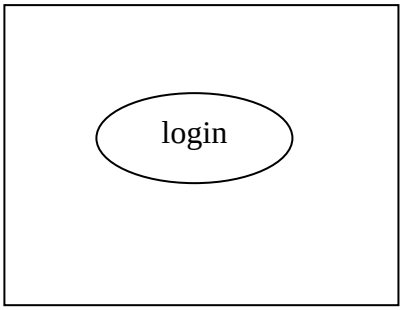
Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan, yaitu siapa dan apa yang menggunakan sistem. Aktor mempresentasikan peran bukan pemakai individu dari sistem. Aktor memiliki nama, nama yang dipilih seharusnya menyatakan peran aktor.



Gambar 2.4 Aktor

2. Use-case

Use case adalah cara spesifik penggunaan sistem oleh aktor. *Use case* melihatkan interaksi antara aktor-aktor dan sistem. *Use case* mengemukakan suatu kerja yang tampak.

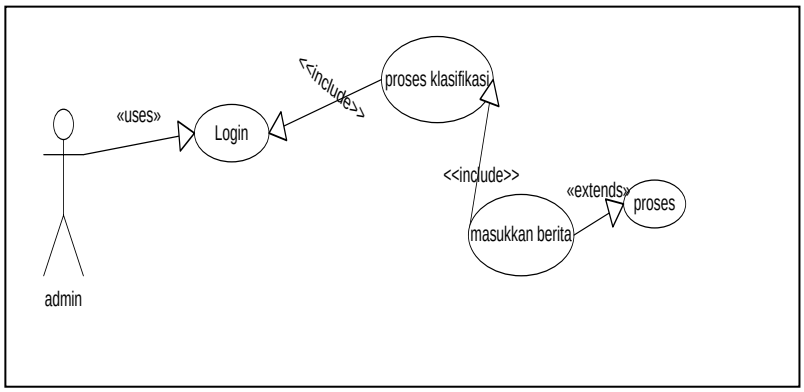


Gambar 2.5 *Use case*

3. Keterhubungan

Keterhubungan *use case* dengan *use case* yang lain berupa generalisasi *use case* yaitu:

a. *Include*, perilaku *use case* merupakan bagian dari *use case* lain.
b. *Extend*, perilaku *use case* memperluas *use case* yang lain.



Gambar 2.6 Keterhubungan

2.8.2 Spesifikasi Use Case

Spesifikasi *use case* memberikan gambaran lengkap spesifikasi pada *use case*. Spesifikasi *use case* sistem rekomendasi dilakukan berdasarkan *case* yang ada pada *use case* diagram. Spesifikasi *use case* biasanya terdiri dari :

a. Tujuan *use case* yaitu menjelaskan apa tujuan dari *case* yang terjadi.

b. Deskripsi yaitu yang menjelaskan apa yang terjadi pada *case.*

c. Skenario yaitu menjelaskan cara kerja *case* mulai dari awal hingga akhir.

d. Kondisi awal yaitu keadaan apa yang terjadi sebelum *case* berlangsung

e. Kondisi akhir yaitu keadaan apa atau apa *output* apa yang dihasilkan setelah *case* berlangsung.

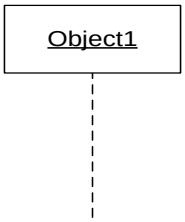
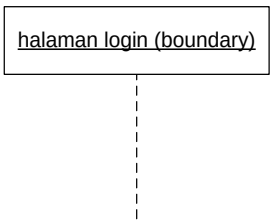
2.8.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display dan sebagainya) berupa *message* (pesan) yang digambarkan terhadap waktu.

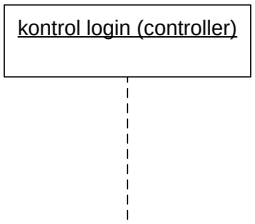
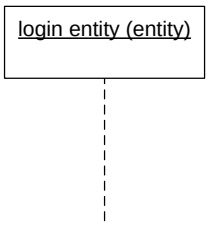
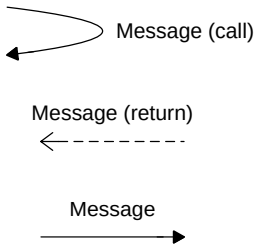
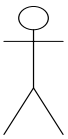
Sequence diagram digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. *Sequence diagram* menunjukkan objek sebagai garis vertical dan tiap kejadian sebagai panah horizontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan.

S*equence diagram* memiliki beberapa elemen yaitu sebagai berikut :

Tabel 2.5 Elemen-elemen *sequence diagram*

No	Nama	Penjelasan	Gambar
1.	Objek lifeline	Menggambarkan batasan objek	
2.	Boundary	Berhubungan dengan proses input output ataupun interface	

Tabel 2.5 Elemen-elemen *sequence diagram* (Lanjutan)

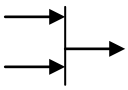
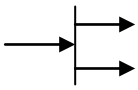
No	Nama	Penjelasan	Gambar
3.	Controller	Berhubungan dengan proses	
4.	Entity	Berhubungan dengan input-output data	
5.	Message arrow	Menggambarkan alir proses, perintah atau pengiriman data	
6.	Aktivasi	Menggambarkan aktivitas objek	
7.	Actor	Mengambarkan actor suatu objek	

2.8.4 Diagram Aktivasi (Activity Diagram)

Diagram aktifitas adalah diagram *flowchart* yang diperluas untuk menunjukkan aliran kendali satu aktivitas ke aktivitas yang lain. Diagram aktifitas digunakan untuk

memodelkan aspek dinamis sistem. Diagram aktivitas berupa operasi-operasi dan aktivitas-aktivitas di *use case* (hariyanto, 2004).

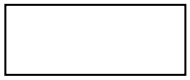
Tabel 2.6 Simbol-simbol diagram aktifitas

Simbol	Keterangan
	Start point
	End Point
	Activities
	Join (Penggabungan)
	Fork (Percabangan)
Swimline	Sebuah cara mengelompokkan aktivitas berdasarkan aktor (mengelompokkan aktivitas dalam sebuah urutan yang sama)

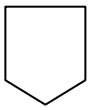
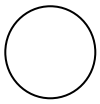
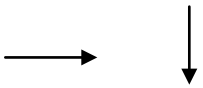
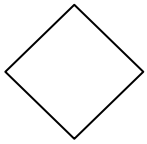
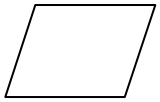
2.9 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur suatu program (Setiawan, 2006). Simbol-simbol dari *flowchart* memiliki fungsi yang berbeda antara satu simbol dengan simbol lainnya (Davis, 1999). Fungsi dari simbol-simbol *flowchart* adalah sebagai berikut :

Tabel 2.7 Fungsi simbol-simbol *flowchart.*

Simbol	Fungsi
	simbol *process*, yaitu menyatakan suatu tindakan (proses) yang dilakukan didalam program.

Tabel 2.7 Fungsi simbol-simbol *flowchart* (Lanjutan).

Simbol	Fungsi
	Simbol *offline connector* yaitu menyatakan penghubung bila flowchart terputus disebabkan oleh pergantian halaman (misalnya tidak cukup dalam satu halaman).
	Simbol *online connector*, berfungsi untuk menyatakan sambungan dari prose ke proses yang lainnya dalam halaman yang sama.
	Simbol arus/*flowline*, yaitu menyatakan jalannya arus suatu proses.
	Simbol *decision* yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu : ya/ tidak.
	Simbol *input/output*, menyatakan proses input atau output tanpa tergantung jenis peralatannya.
	Simbol *terminal* yaitu menyatakan permulaan atau akhir suatu program.
	Simbol *document*, mencetak keluaran dalam bentuk dokumen.

2.10 Bahasa Pemograman PHP dan Database MySQL

PHP (*Hypertext Preprocessor*) adalah bahasa computer yang dibuat untuk pengembangan web dinamis. Pada umumnya PHP digunakan di server namun juga dapat berdiri sendiri sebagai aplikasi *graphical* (www.php.net, 2008).

Penggunaan PHP dan MySQL dipilih karena PHP dan MySQL memiliki beberapa kelebihan seperti dinyatakan oleh Nugroho, B (2008) kelebihannya sebagai berikut:

1. Bahasa pemograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web *Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan Apache dengan konfigurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *opensource* yang dapat digunakan di berbagai mesin (Linux, Unix, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Sedangkan database MySQL memiliki beberapa kelebihan, yaitu:

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi.
2. *Open Source*

   MySQL dapat didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma- cuma.
3. *Multiuser*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat meproses lebih banyak SQL per satuan waktu.
5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed* atau *unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.
6. *Command* dan *functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.
7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang perizinan yang mendetail serta *password* terenkripsi.
8. *Scalability* dan *limits*

MySQL mampu mengani database dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 juta ribu serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *client* menggunakan protocol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
10. *Localization*

    MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. *Interface*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. *Clients* dan *tools*

    MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk adminsitrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.
13. Struktur Tabel
14. MySQL memiliki struktur table yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle

2.11 Penelitian Terdahulu

Penelitian mengenai klasifikasi sentimen telah dilakukan oleh Bo Pang (2002). Pada papernya, Bo Pang melakukan klasifikasi sentimen terhadap review film dengan menggunakan berbagai teknik pembelajaran mesin. Teknik pembelajaran mesin yang digunakan yaitu Naïve Bayes, Maximum Entropy, dan Support Vector Machines (SVM). Pada penelitian itu juga digunakan beberapa pendekatan untuk melakukan ekstraksi fitur, yaitu unigram, unigram+bigram, unigram+Part of Speech (POS), adjective, dan unigram+posisi. Hasil dari eksperimen yang dilakukan dipenelitian ini

menemukan bahwa SVM menjadi metode terbaik ketika dikombinasikan dengan unigram dengan akurasi 82.9% (Pang, et. al, 2002).

Suhaad Prasad (2011) mencoba untuk menggunakan Naïve Bayes dengan berbagai macam pendekatan yakni, Bernoulli, Bernoulli Chi Square, Multinomial Unigram, Linear Bigram, Back off Bigram, Empirical Bigram, dan Weighted-Normalized Complement Naïve Bayes (WCNB). Dari hasil uji coba diketahui bahwa Multinomial Unigram, Bernouli ChiSquare, dan Linear Bigram menunjukan hasil yang cenderung lebih baik dari pendekatan lain (Prasad, 2011).

Penelitian Analisis Sentimen Sentimen pada Opini Terhadap Tokoh Publik dilakukan oleh Ismail Sunni dan Dwi Hendratmo Widyantoro (2012). Mereka menggunakan F3 (F3 is Factor Finder) yang memiliki beberapa metode praproses yang diperkirakan mampu menangani permasalahan model bahasa yang ditemukan. F3 menggunakan Naïve Bayes untuk melakukan analisis sentimen karena telah teruji di bebagai penelitian. Sedangkan untuk mengetahui perubahan sentimen, F3 akan menampilkan perubahan sentimen dalam bentuk kurva menggunakan metode Tf-Idf dengan discounted-cumulative untuk menangani karakter topik yang muncul di Twitter yang berkelanjutan.Hasil analisis dan pengujian menunjukkan tahapan praproses tidak memiliki pengaruh yang signifikan terhadap akurasi (69.4%-72.8%) klasifikasi sentimen. Sedangkan untuk pengekstrakan topik menunjukkan bahwa penggunaan Tf-Idf dengan discounted cumulative mampu meningkatkan jumlah topik terekstrak yang sesuai.

Penelitian yang serupa juga dilakukan oleh Paulina Aliandu (2013). Penelitian ini melakukan eksperimen untuk melakukan klasifikasi sentimen terhadap data yang diperoleh dari Twitter dengan mengambil *Tweet* akun Presiden RI @SBYudhoyono baik sentimen positif, negatif ataupun netral. Aliandu menerapkan Naive Bayes Method untuk klasifikasi sentimen tersebut dan dapat mengklasifikasi dengan baik dengan akurasi 79,42% (Aliandu, 2013).

Tabel 2.8 Penelitian Terdahulu

No	Peneliti / Tahun	Judul	Keterangan
1	Pang, 2002	Thumbs Up ? Sentiment Classification Using Machine Learning Techniques	Ektraksi fitur dilakukan dengan unigram, unigram+bigram, unigram+Part of Speech (POS), adjective, dan unigram+posisi. Hasil dari eksperimen SVM menjadi metode terbaik ketika dikombinasikan dengan unigram dengan akurasi 82.9%
2	Prasad, 2011	Microblogging Sentiment Analysis Using Bayesian Classification Methods	Menggunakan Naïve Bayes dengan berbagai macam pendekatan yakni, Bernoulli, Bernoulli Chi Square, Multinomial Unigram, Linear Bigram, Back off Bigram, Empirical Bigram, dan Weighted-Normalized Complement Naïve Bayes (WCNB). Dari hasil uji coba diketahui bahwa Multinomial Unigram, Bernouli ChiSquare, dan Linear Bigram menunjukan hasil yang cenderung lebih baik dari pendekatan lain
3	Sunni & Widyantoro (2012).	Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik	Penelitian ini menerapkan F3 (F3 is Factor Finder) yang memiliki beberapa metode praproses menggunakan Naïve Bayes. Sedangkan untuk mengetahui perubahan sentimen, F3 akan menampilkan perubahan sentimen dalam bentuk kurva menggunakan metode Tf-Idf dengan discounted-cumulative untuk menangani karakter topik yang muncul di Twitter yang berkelanjutan. Hasil analisis dan pengujian menunjukkan tahapan praproses tidak memiliki pengaruh yang signifikan terhadap akurasi (69.4%-72.8%)
4	Aliandu, 2013	Twitter Used by Indonesian President: An Sentiment Analysis of Timeline	Penelitian ini melakukan eksperimen untuk melakukan klasifikasi sentimen terhadap data yang diperoleh dari Twitter dengan mengambil *Tweet* akun Presiden RI @SBYudhoyono baik sentimen positif, negatif ataupun netral. Aliandu menerapkan Naive Bayes Method untuk klasifikasi sentimen tersebut dan dapat mengklasifikasi dengan baik dengan akurasi 79,42%

BAB 3

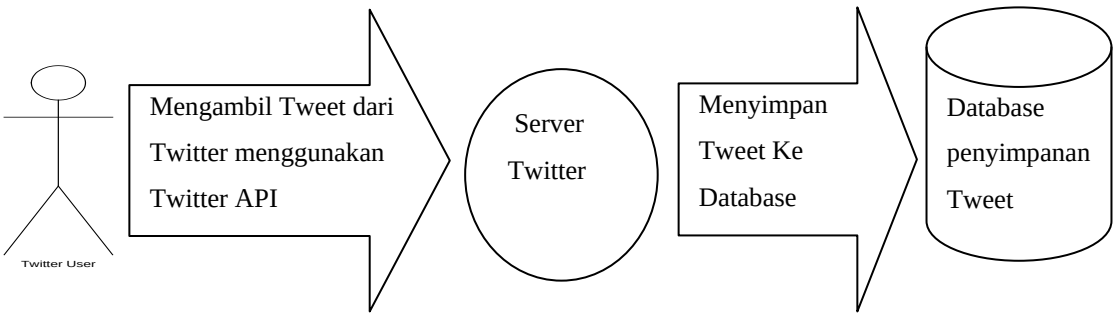
ANALISIS DAN PERANCANGAN

3.1 Analisis Data

Dalam penelitian ini data terdiri dari 4 bagian yaitu data *Tweet*, data *stopword*, data kata dasar, dan data *knowledge*.

3.1.1 Data Tweet

Data *Tweet* dalam penelitian ini dperoleh dengan memanfaatkan API yg disediakan oleh Twitter. Dengan memanfaatkan API tersebut dibangunlah sebuah aplikasi untuk mengambil data *Tweet* tersebut dari Twitter kemudian disimpan ke dalam Database. Skema dari proses pengambilan *Tweet* dapat dilihat pada gambar 3.1.



Gambar 3.1 Skema dari proses pengambilan *tweet*

Pada saat pengumpulan data, penulis menggunakan Twitter API *Search*, kemudian memasukkan keyword-keyword yang berhubungan dengan Provider Telekomunikasi yang dikombinasikan dengan kata-kata sentimen. Penulis mengikuti

teknik pengumpulan data yang digunakan oleh peneliti sebelumnya (Pak & Paroubek, 2010., Nur & Santika 2011., Agarwal et al. 2011) dimana mereka menggunakan *emoticon* sebagai penanda sebuah *tweet* mengandung sentimen positif, negatif, atau netral. Selain itu penulis juga mengikuti teknik yang dilakukan pada penelitian seblumny dimana mereka menggunakan kata-kata bermakna sentimen sebagai penanda sentimen pada tweet tersebut (Kouloumpis, 2011., Wicaksono et al.,2013). Berdasarkan teknik pengumpulan data yang dilakukan peneliti-peneliti sebelumny diatas, penulis menggabung kedua teknik tersebut yang kemudian menggunakan *emoticon* dan kata-kata sentimen yang digabungkan dengan nama-nama provider telekomunikasi yang ada di Indonesia. Penulis menggabungkan sebuah provider dengan sebuah kata sentimen atau *emoticon* yang kemudian digunakan menjadi kata kunci pencarian (*keyword*). Berikut ini adalah daftar kata-kata yang digunakan dalam penelitian ini yang digunakan sebagai kata kunci (*keyword*).

Tabel 3.1 Tabel *Keyword*

Negative Word	bodoh, tolol, gagal, bermasalah, lelet, kurang, susah, lambat, parah, bohong, pending, payah
Positive Word	bisa, ok, best, pintar, lancar, cepat, cepet, untung, baik, bagus, gampang, membantu, senang, kencang, kenceng, menolong, tanggap
Negative Emotion Icon	:-(, :(, =(, ;(
Positive Emotion Icon	:-), :), =), :D
Nama Provider	telkomsel, indosat, im3, kartu xl, smartfren, simpati, Axis, tri 3, provider, xlcare

Data *Tweet* yang diambil untuk data *training* adalah sebesar 3300 data, dimana data ini terbagi menjadi beberapa bagian seperti yang ditunjukkan pada Tabel 3.1. Data yang diambil adalah data *Tweet* yang mengandung sentimen terhadap Provider Telekomunikasi.

Untuk kebutuhan *training*, data yang berhasil dikumpulkan tersebut akan dikategorikan secara manual yang dilakukan oleh Penulis dan menilai sentimen yang terkandung di dalam *Tweet* tersebut dan menandai *Tweet* tersebut ke dalam 3 kategori sentimen yaitu *Tweet* yang mengandung sentimen negatif, positif dan netral.

Tabel 3.2 Tabel *Tweet*

Jenis *Tweet* Sentimen	**Negatif**	**Positif**	**Netral**
Tweet Provider Telekomunikasi	1100	1100	1100

Dalam penyimpanan data *Tweet* tersebut di dalam database dibagi menjadi 3 tabel yaitu tabel *tweet training*, tabel *tweet testing*, dan tabel hasil. Penjelasan untuk masing-masing tabel adalah sebagai berikut :

3.1.1.1 Tabel Tweet Training

Tabel *tweet training* adalah tabel *database* yang menyimpan *Tweet-Tweet* yang akan digunakan untuk proses *training*. Tabel *training* memiliki 5 field yaitu id_str, user, ext, jenis *Tweet*, dan kategori sentimen. Rancangan tabel *tweet training* dapat dilihat pada tabel 3.2

Tabel 3.3 Tabel *Tweet training*

Id_Str	**User**	**Text**	**Jenis *Tweet***	**Sentimen**
440737-968970-821632	@8lue_fire	@telkomsel iyo.di syarat dan ketentuan hadiah cm tertulis gadget keren.nah aq tanya gadget kerennya ap?	provider	netral
438512-559277-092866	@eLfiraRosana	Keren memang telkomsel, via twitter fast respon :D gak nyesel pake telkomsel :D thank youuu telkomsel atas pelayanan yang memuaskan :)	provider	positif

Tabel 3.3 Tabel *Tweet training* (Lanjutan)

Id_Str	User	Text	Jenis *Tweet*	Sentimen
444075-303816-536066	@NainggolanVaber	Jokowi jgn nyapres lebih baik urus Jakarta katanya apa bedanya anda dgn koruptor mementingkan diri sendiri bkn Indonesia?	politik	negatif

3.1.1.2 Tabel Tweet Testing

Tabel *tweet testing* adalah tabel *database* yang menyimpan *Tweet* yang akan digunakan untuk proses *testing*. Tabel *testing* memiliki 5 *field* yaitu id_str, user, ext, jenis *Tweet*, dan kategori sentimen. Pada tabel *training*, *Tweet* juga telah ditandai dengan kategori sentimennya. Rancangan tabel *tweet training* dapat dilihat pada tabel 3.3.

Tabel 3.4 Tabel *Tweet testing*

Id_Str	User	Text	Jenis Tweet	Sentimen
443349-567581-347841	@Jiyeon7_Tara	Wacana Jokowi-JK Tak Buat Golkar Risau: Golkar sudah mantap mengusung Ketua Umum Aburizal Bakrie	provider	netral
440327-905319-854081	@RoniJohansyah	Update status gagal terus,giliran sms sampah,cepet masuk @Telkomsel http://t.co/Ul8fxv3vNx	provider	negatif
441320-417718-837248	@Min_Yuna	@Telkomsel dibenerin donk signalnya. saya tuh pake internet pengen cepet, bukan meringankan kerjaan, malah semakin bikin sibuk krn lelet	provider	negatif

3.1.1.3 Tabel Pengetahuan

Tabel hasil adalah tabel database yang menyimpan hasil training. Rancangan tabel testing dapat dilihat pada tabel 3.3.

Tabel 3.5 Tabel pengetahuan

ngram	sentimen	Frekuensi	Probabilitas
_i	negatif	2	0,26
in	negatif	5	0,31
nt	negatif	2	0,26
te	negatif	5	0,31
er	negatif	2	0,26
rn	negatif	2	0,26
ne	negatif	2	0,26
et	positif	2	0,26
t_	positif	2	0,26

3.1.2 Data Stopword

Data *stopword* didapat dari jurnal Tala (2003) dimana datanya berjumlah 753 data dan dari *Tweet-Tweet* yang digunakan dalam penelitian. Data *stopword* di dalam *database*. Rancangan tabel *stopword* dapat dilihat pada table 3.4.

Tabel 3.6 Tabel *stopword*

id_stopword	*Stopword*
1	Ada
2	Dari
3	Karena

3.1.3 Data Kata Dasar

Data kata dasar didapat dari kamus bahasa Indonesia online dimana datanya berjumlah 28533 data. Data kata dasar disimpan di dalam *database*. Rancangan tabel kata dasar dapat dilihat pada tabel 3.5.

Tabel 3.7 Tabel kata dasar

id_katadasar	Katadasar
1	Ajar
2	Makan
3	Lari

3.1.4 Data Knowledge

Data *knowledge* adalah data hasil dari proses *training* yang telah dilakukan. Data digunakan sebagai *N-gram* karakter kata pada saat proses dilakukan. Data *N-gram Tweet* ini disimpan dalam *database knowledge*. Rancangan tabel *keyword* dapat dilihat pada table 3.6.

Tabel 3.8 Tabel *keyword Tweet*

N-gram	sentimen	frekuensi	Probabilitas
_i	negatif	2	0,26
in	negatif	5	0,31
nt	negatif	2	0,26
te	negatif	5	0,31
er	negatif	2	0,26
ne	negatif	2	0,26
et	positif	2	0,26
t_	positif	2	0,26

3.2 Analisis Sistem

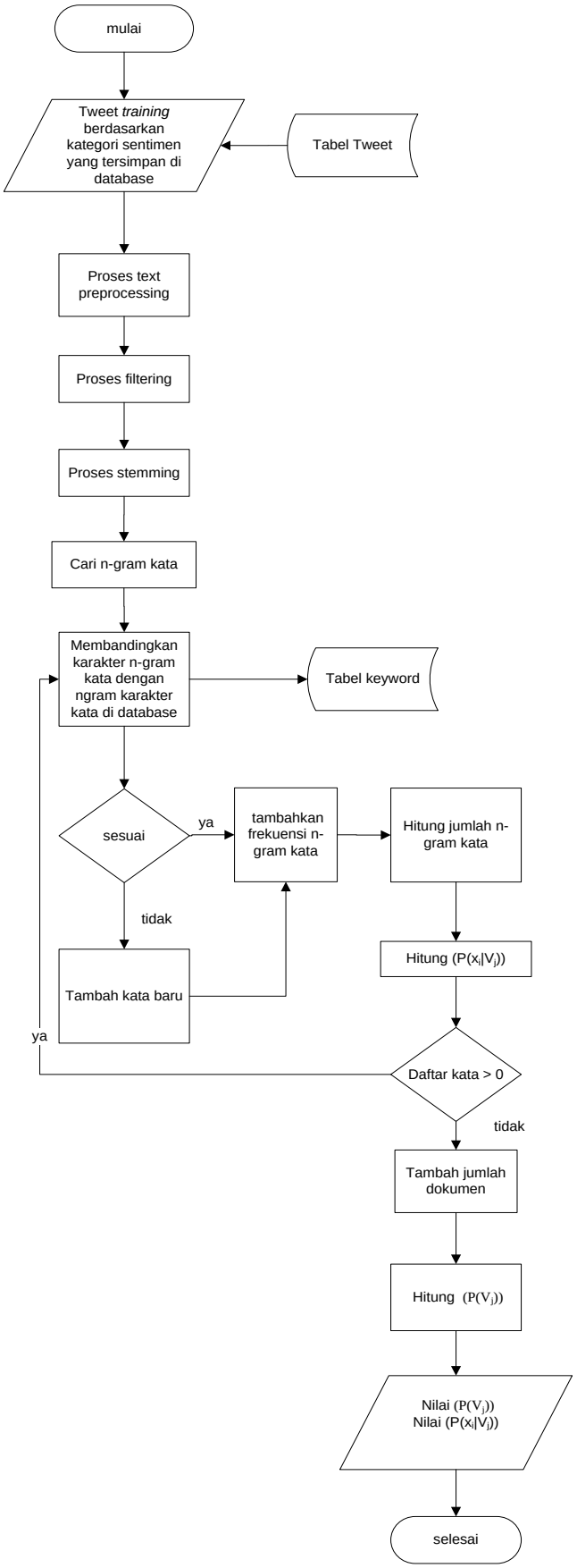
Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem yang meliputi perangkat lunak (*software*), pengguna (*user*) serta hasil analisis terhadap sistem dan elemen-elemen yang terkait. Analisis ini diperlukan sebagai dasar bagi tahapan perancangan sistem. Analisis sistem ini meliputi desain data, deskripsi sistem, dan implementasi desain dan semua yang diperlukan dalam aplikasi pengklasifikasian sentimen.

Dalam penelitian ini sistem mempunyai 2 tahapan proses yaitu tahapan pertama adalah tahap pembelajaran atau *training* yaitu tahap pengklasifikasian

terhadap *Tweet* yang sudah diketahui kategorinya. Tujuan dari tahap *training* adalah untuk mencari *keyword* beserta probabilitasnya yang nantinya akan digunakan pada proses *testing*. Sedangkan tahap kedua adalah *tahap testing* yaitu tahap pengklasifikasian terhadap *Tweet* yang belum diketahui kategorinya.

Pada tahap pembelajaran atau *training* proses-proses yang dilakukan adalah sebagai berikut :

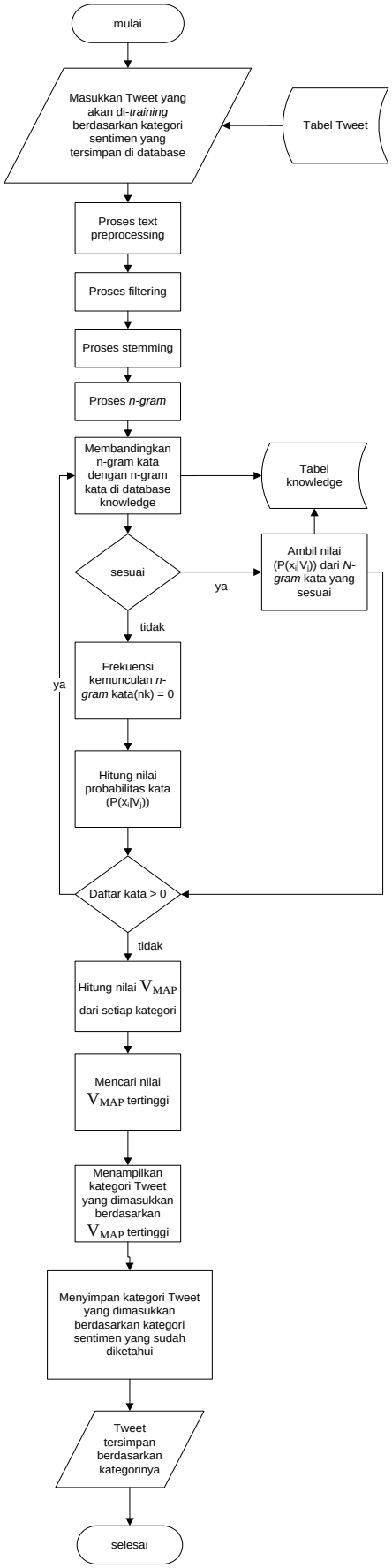
1. Memasukkan *Tweet training* berdasarkan kategori sentimen yang tersimpan di *database*.
2. Kemudian sistem akan melakukan proses *text preprocessing*.
3. Setelah melakukan proses *text preprocessing* sistem melakukan proses *filtering.*
4. Selanjutnya sistem melakukan proses *stemming*.
5. Kata hasil *stemming* kemudia dicari *N-gram* katanya. *N-gram* kata yang muncul dibandingkan dengan *N-gram* kata yang ada di dalam *database*.
6. Jika sesuai, maka tambahkan jumlah frekuensi kemunculan *N-gram* kata (n_k). Jika tidak sesuai, maka kata tersebut dijadikan sebagai *N-gram* kata baru dan tambahkan jumlah kemunculan kata (n_k) tersebut.
7. Hitung probabilitas setiap *N-gram* kata ($P(x_i|V_j)$).
8. Jika daftar kata hasil *stemming* lebih dari nol maka proses akan kembali ke langkah nomor 5. Jika tidak, proses akan berlanjut ke langkah berikutnya.
9. Tambahkan jumlah dokumen.
10. Hitung probabilitas dokumen *Tweet* setiap kategori sentimen ($P(V_j)$).
11. Hasilnya adalah nilai probabilitas setiap *N-gram* kata dan nilai probabilitas *Tweet* setiap kategori sentimen.
12. Proses *training* selesai. *Flowchart* dari proses *training* adalah sebagai berikut :



Gambar 3.2 *Flowchart* proses *training*

Sedangkan pada tahap *testing* proses-proses yang dilakukan adalah sebagai berikut :

1. Memasukkan *Tweet testing* yang tersimpan di *database*.
2. Kemudian sistem akan melakukan proses *text preprocessing*.
3. Setelah melakukan proses *text preprocessing* sistem melakukan proses *filtering* yaitu penghilangan *stopword.*
4. Selanjutnya sistem melakukan proses *stemming* yaitu mengubah kata berimbuhan menjadi kata dasar.
5. Kata hasil *stemming* akan dicari *N-gram* katanya. *N-gram* kata yang muncul dibandingkan dengan *N-gram* kata yang ada di dalam *database*. Jika sesuai maka nilai probabilitas *N-gram* kata dari *N-gram* kata yang di database *keyword* dijadikan nilai probabilitas kata ($(P(x_i|V_j))$ yang dimasukkan. Jika tidak maka frekuensi kemunculan *N-gram* kata (nk) bernilai nol dan nilai probabilitas *N-gram* kata ($(P(x_i|V_j))$ dihitung.
6. Jika daftar kata hasil *stemming* lebih dari nol maka proses akan kembali ke langkah nomor 5. Jika tidak, proses akan berlanjut ke langkah berikutnya.
7. Hitung nilai V_{MAP} dari setiap kategori sentimen.
8. Mencari nilai V_{MAP} tertinggi diantara sentimen negatif, positif atau netral.
9. Menampilkan kategori sentimen *Tweet* yang dimasukkan berdasarkan nilai V_{MAP} tertinggi
10. Proses *testing* selesai. *Flowchart* dari proses *testing* adalah sebagai berikut :



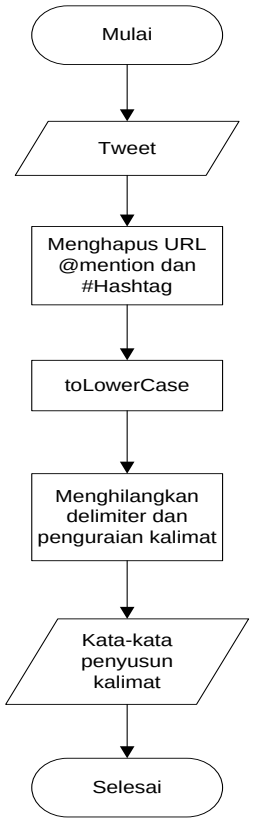
Gambar 3.3 *Flowchart* proses *testing*

Dalam penelitian ini pembuatan sistem juga menggunakan metode *text mining*. Dimana, langkah-langkah dari setiap tahap *text mining* adalah sebagai berikut :

3.2.1 Text Preprocessing

Langkah-langkah proses *text preprocessing* adalah sebagai berikut :

1. Setelah teks dokumen dimasukkan maka sistem akan merubah semua karakter huruf menjadi huruf kecil melalui proses *toLowerCase.*
2. Kemudian dilakukan penghapusan URL , *@mention* dan *@hashtag* yang ada pada *Tweet* tersebut.
3. Kemudian dilakukan penghapusan delimiter yaitu karakter angka dan karakter simbol kecuali karakter huruf serta penguraian terhadap kalimat-kalimat yang ada di teks *Tweet* tersebut.
4. Hasilnya adalah kata-kata penyusun kalimat yang ada di *Tweet*.
5. Proses *text preprocessing* selesai.
6. *Flowchart* dari proses *text preprocessing* adalah sebagai berikut:



Gambar 3.4 *Flowchart Text Preprocessing*

Contoh :

Misal terdapat *input* kalimat seperti :

Update status gagal terus, giliran sms sampah,cepet masuk @Telkomsel http://t.co/Ul8fxv3vNx

Gambar 3.5 Contoh kalimat yang akan di *input*

Maka setelah melalui proses *RemoveURLMentionHashtag* maka *Tweet* tersebut berubah menjadi seperti ini :

Update status gagal terus, giliran sms sampah,cepet masuk

Gambar 3.6 Contoh kalimat yang akan di *input*

Maka setelah melalui proses *ToLowerCase* maka huruf besar dalam kalimat tersebut berubah menjadi huruf kecil :

update status gagal terus, giliran sms sampah,cepet masuk dasar.

Gambar 3.7 Contoh kalimat setelah *ToLowerCase*

Kemudian setelah proses penghilangan delimiter dan penguraian kalimat maka hasilnya adalah sebagai berikut :

Tabel 3.9 Hasil dari proses *text preprocessing*

update	status	gagal	terus
giliran	sms	sampah	cepat
masuk	dasar		

3.2.2 Feature Selection

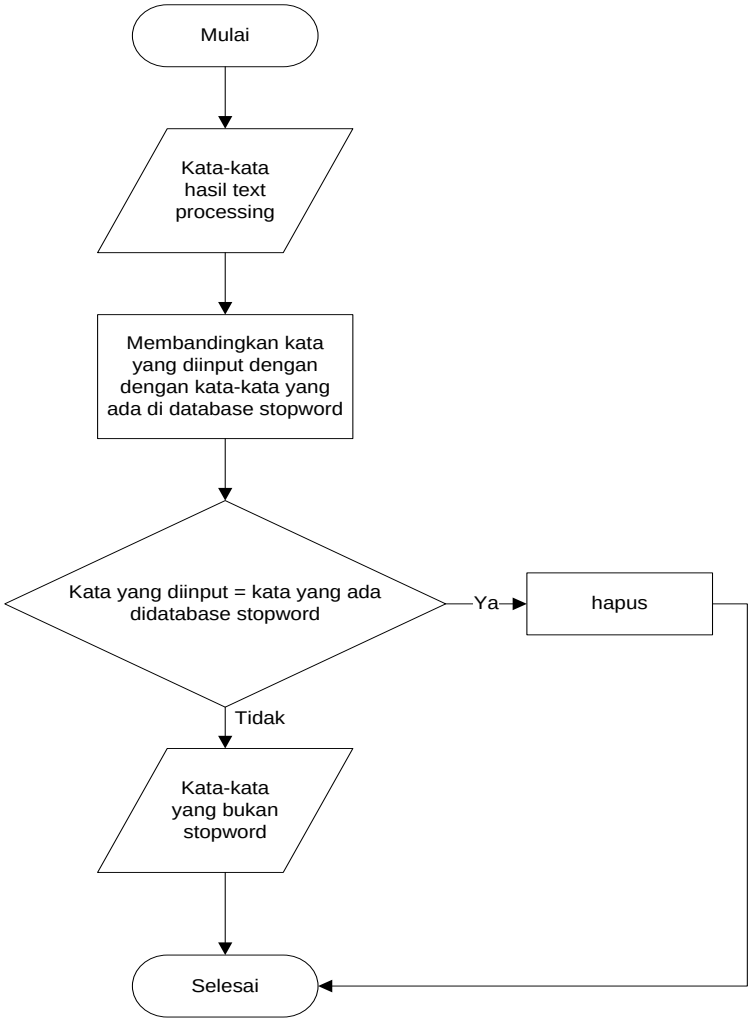
Pada tahap ini terdapat dua proses yang dilakukan, adalah sebagai berikut :

3.2.2.1 Stopword Removal (Filtering)

Langkah-langkah untuk proses *filtering* adalah sebagai berikut :

1. Kata-kata penyusun kalimat hasil dari tahap *text preprocessing* dijadikan sebagai masukkan.
2. Kemudian dibandingkan dengan kata-kata yang ada di *database stopword.*

3. Jika kata yang dimasukkan sama dengan kata di *database stopword* maka kata yang dimasukkan dihapus. Namun jika kata yang dimasukkan tidak sama dengan kata yang ada di *database stopword* maka tersebut tidak dihapus
4. Proses *filtering* selesai. *Flowchart* dari proses *filtering* adalah sebagai berikut :



Gambar 3.8 *Flowchart* proses *filtering*

Contoh :

Misalkan terdapat masukkan yang merupakan hasil dari proses *text processing* sebagai berikut :

Tabel 3.10 Hasil dari proses *text preprocessing* yang dijadikan *input*.

telkomsel	di	medan	ini
memang	parah	dan	gak
bisa	internetan	lagi	

Dan misalnya terdapat *stopword* yang dalam *database stopword* sebagai berikut :

Tabel 3.11 Kumpulan *stopword*

dan	di	ingin	ini
kepada	dalam	selalu	lalu
yaitu	bahwa	terdiri	sekali
dulu	sekalian	enggak	bagian

Kemudian sistem akan membandingkan antara kata-kata yang dimasukkan dengan kata-kata yang ada di dalam *database stopword*. Selanjutnya sistem akan menghapus kata-kata yang dimasukkan apabila kata-kata yang dimasukkan sama dengan kata-kata yang ada di *database stopword*. Maka *ouput*-nya menjadi sebagai berikut:

Tabel 3.12 Hasil dari proses *filtering*

telkomsel	medan	parah	gak
bisa	internetan		

3.2.2.2 Stemming

Berdasarkan algoritma *confix stripping* langkah-langkah proses *stemming* adalah sebagai berikut :

1. Kata yang belum di-*stemming* dibandingkan ke dalam *database* kamus kata dasar. Jika ditemukan, maka kata tersebut diasumsikan sebagai kata dasar dan algoritma berhenti. Jika kata tidak sesuai dengan kata dalam kamus, lanjut ke langkah 2.
2. Jika kata di-*input* memiliki pasangan awalan-akhiran “be-lah”,”be-an”,”me-i”,”di-i”,”pe-i”, atau “te-i” maka langkah *stemming* selanjutnya adalah 5, 3, 4, 5, 6, tetapi jika kata yang di-*input* tidak memiliki pasangan awalan-akhiran tersebut, langkah *stemming* berjalan normal yaitu 3, 4, 5, 6, 7.
3. Hilangkan partikel dan kata ganti kepunyaan. Pertama hilangkan partikel (“-lah”, “-kah”, “-tah”, “-pun”). Setelah itu hilangkan juga kata ganti kepunyaan (“-ku”, “-mu”, atau “-nya”). Contoh : kata “bajumulah”, proses *stemming* pertama menjadi “bajumu” dan proses stemming kedua

menjadi “baju”. Jika kata “baju” ada di dalam kamus maka algoritma berhenti. Sesuai dengan model imbuhan, menjadi :

[[[AW+]AW+]AW+] Kata Dasar [+AK]

4. Hilangkan juga akhiran (“-i”,”-an”, dan “-kan”), sesuai dengan model imbuhan, maka menjadi:

[[[AW+]AW+]AW+] Kata Dasar

Contoh: kata “membelikan” di-*stemming* menjadi ”membeli”, jika tidak ada dalam *database* kata dasar maka dilakukan proses penghilangan awalan.

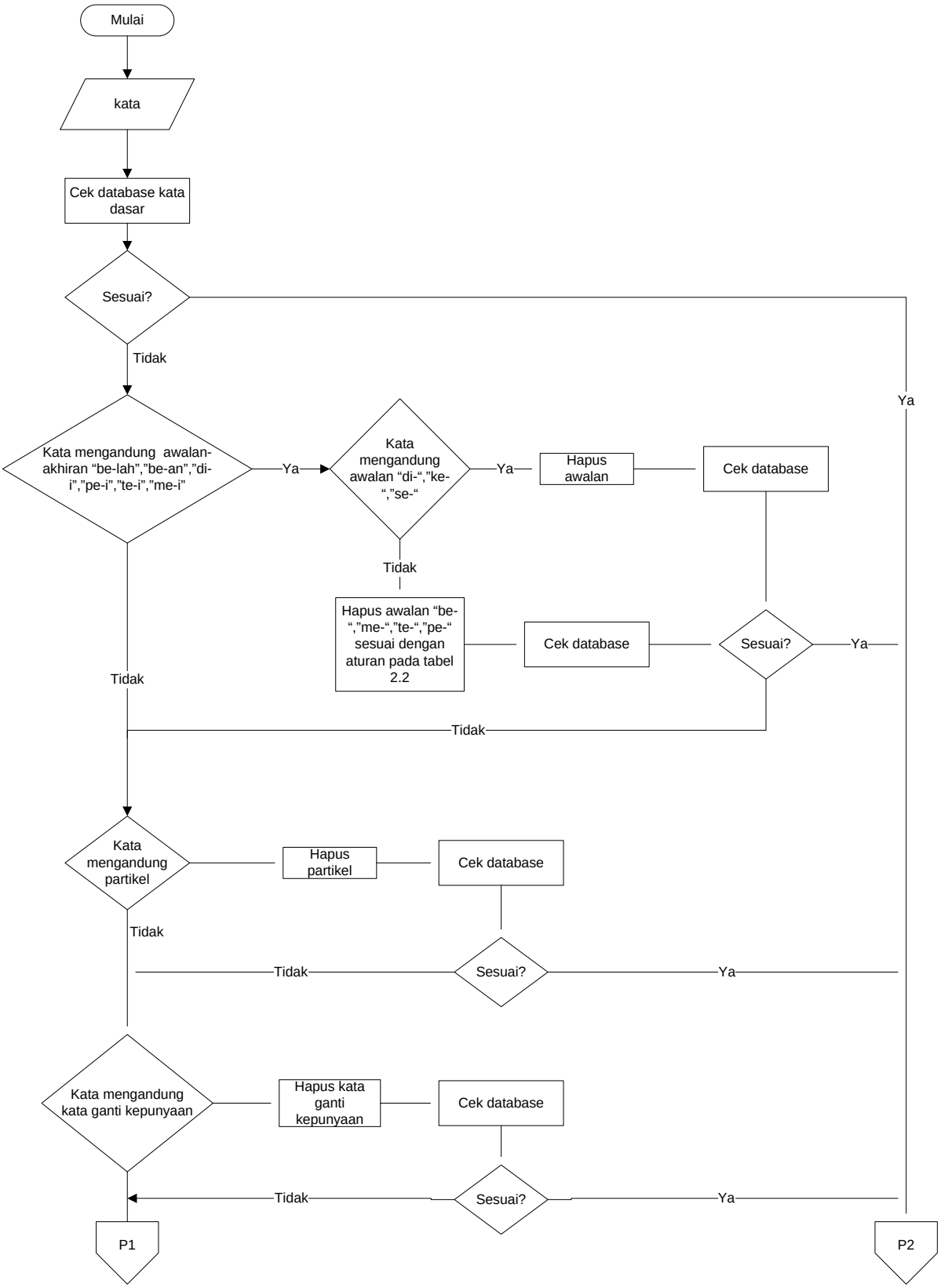
5. Penghilangan awalan (“be-“, ”di-“, ”ke-“, ”me-”, ”pe-”, ”se-”, dan “te-“) mengikuti langkah-langkah berikut:
   a. Algoritma akan berhenti jika:
      i. Awalan diidentifikasi bentuk sepasang imbuhan yang tidak diperbolehkan dengan akhiran (berdasarkan tabel 2.1) yang dihapus pada langkah 3.
      ii. Diidentifikasi awalan yang sekarang identik dengan awalan yang telah dihapus sebelumnya atau,
      iii. Kata tersebut sudah tidak memiliki awalan.

   b. Identifikasi jenis awalan dan peluruhannya bila diperlukan.jenis awalan ditentukan dengan aturan dibawah ini.

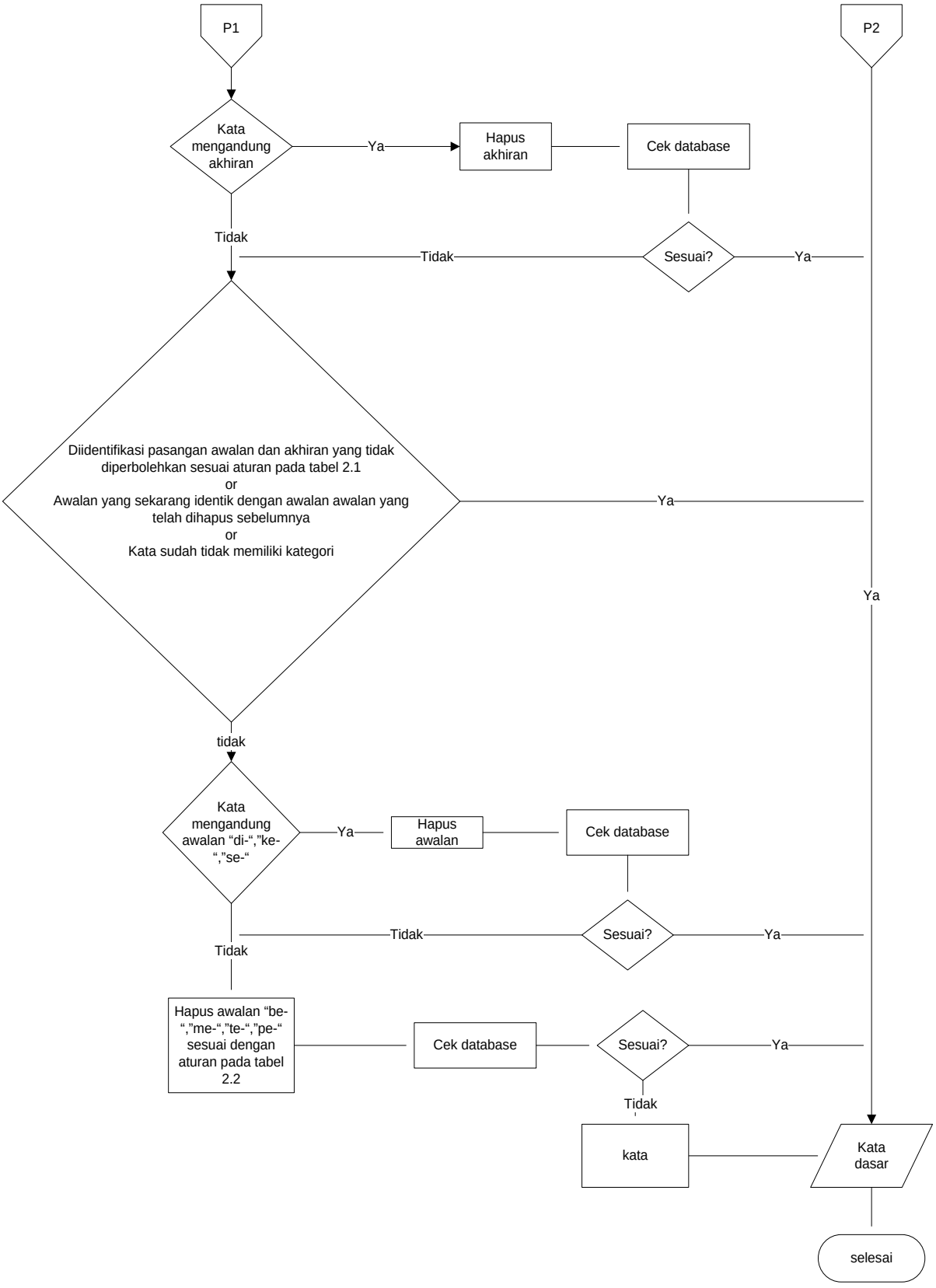
      i. Jika awalan dari kata adalah “di-“, “ke-“, atau “se-“ maka awalan dapat langsung dihilangkan.
      ii. Hapus awalan “te-“, “be-“, “me-“, atau “pe-“ yang menggunakan aturan peluruhan yang dijelaskan pada tabel 2.2.

Sebagai contoh kata “menangkap”, setelah menghilangkan awalan “me-“ maka kata yang didapat adalah “nangkap”. Karena kata “nangkap” tidak ditemukan dalam database kata dasar maka karakter “n” diganti dengan karakter “t” sehingga dihasilkan kata “tangkap” dan kata “tangkap” merupakan kata yang sesuai dengan kata yang ada di database kata dasar, maka algoritma berhenti.

6. Jika semua langkah gagal, maka kata yang diuji pada algoritma ini dianggap sebagai kata dasar. *Flowchart* dari proses *stemming* adalah sebagai berikut :



Gambar 3.9 *Flowchart* proses *stemming*



Gambar 3.9 *Flowchart* proses *stemming* (Lanjutan)

3.2.3 Contoh penggunaan algoritma naïve bayes classifier

Berikut ini akan diberikan contoh pembelajaran dan klasifikasi *naïve bayes classifier*. Contoh ini hanya akan menampilkan 3 *Tweet* yang berupa *N-gram* kata hasil dari proses-proses *text mining*. Dimana dokumen *Tweet* yang pertama dan yang kedua digunakan pada tahap pembelajaran (data latih), sedangkan dokumen *Tweet* ketiga akan digunakan sebagai dokumen test yang akan diklasifikasi. Dan dalam contoh ini terdapat dua *Tweet* dengan kategori sentimen positif dan sentimen negatif.

Dokumen 1: Dokumen pembelajaran 1 (Internet Cepat)

Tabel 3.13 Daftar kata sentimen positif

no.	***keyword***	**frekuensi (n_k)**
1	_i	1
2	in	1
3	nt	1
4	te	1
5	er	1
6	rn	1
7	ne	1
8	et	1
9	t_	2
10	_c	1
11	ce	1
12	ep	1
13	pa	1
14	at	1

Dari tabel 3.11 diketahui :

Jumlah frekuensi keseluruhan sentimen positif (n) = 15

Jumlah kata (|kosakata|) = 14

Sehingga dari nilai-nilai tersebut kita bisa mencari nilai probabilitas setiap kata dengan menggunakkan rumus $P(x_i|V_j)$ yaitu sebagai berikut :

P(_i|positif) = (1+1)/(15+14) = 0,068

P(in|positif) = (1+1)/(15+14) = 0,068

P(nt|positif) = (1+1)/(15+14) = 0,068

Untuk hasil keseluruhan dapat dilihat pada tabel 3.12 berikut ini :

Tabel 3.14 Probabilitas kata *tweet* positif

no.	*keyword*	frekuensi (n_k)	probabilitas
1	_i	1	0,068
2	in	1	0,068
3	nt	1	0,068
4	te	1	0,068
5	er	1	0,068
6	rn	1	0,068
7	ne	1	0,068
8	et	1	0,068
9	t_	2	0,10
10	_c	1	0,068
11	ce	1	0,068
12	ep	1	0,068
13	pa	1	0,068
14	at	1	0,068

Diketahui : jumlah dokumen sentimen positif = 1

jumlah dokumen sentimen negatif = 0

maka nilai $P(V_j)$:

P(positif) = 1/1 = 1

P(negatif) = 0/1 = 0

Dokumen 2 : Dokumen pembelajaran 2

Tabel 3.15 Daftar kata sentimen negatif

No.	*Keyword*	Frekuensi (n_k)
1	_i	1
2	in	1
3	nt	1
4	te	1
5	er	1
6	rn	1
7	ne	1
8	et	1
9	t_	2
10	_l	1
11	le	1

Tabel 3.15 Daftar kata sentimen negatif (Lanjutan)

No.	*Keyword*	Frekuensi (n_k)
12	el	1
13	le	1
14	et	1

Dari tabel 3.13 diketahui :

Jumlah frekuensi keseluruhan sentimen negatif (n) = 15

Jumlah kata (|kosakata|) = 19

Sehingga dari dari nilai-nilai tersebut kita bisa mencari nilai probabilitas setiap *N-gram* kata dengan menggunakan rumus $P(x_i|V_j)$ yaitu sebagai berikut :

1. Pada sentimen negatif

   P(in|negatif) = (1+1)/(15 + 19) = 0,05

   P(_t|negatif) = (2+1)/(15 + 19)= 0,08

   Untuk hasil keseluruhan dapat dilihat pada tabel 3.4.

2. Pada sentimen positif

   P(ce|positif) = (1+1)/(15+19) = 0,05

   P(ec|positif) = (1+1)/(15+19) = 0,05

   Untuk hasil keseluruhan dapat dilihat pada tabel 3.14 berikut ini:

Tabel 3.16 Probabilitas *N-gram* kata sentimen negatif

no.	*keyword*	frekuensi (n_k)	probabilitas
1	_i	1	0,05
2	in	1	0,05
3	nt	1	0,05
4	te	1	0,05
5	er	1	0,05
6	rn	1	0,05
7	ne	1	0,05
8	et	1	0,05
9	t_	2	0,08
10	_l	1	0,05
11	le	1	0,05
12	el	1	0,05
13	le	1	0,05
14	et	1	0,05

Tabel 3.17 Perubahan nilai probabilitas pada daftar *N-gram* kata sentimen positif

no.	*keyword*	frekuensi (n_k)	probabilitas
1	_i	1	0,05
2	in	1	0,05
3	nt	1	0,05
4	te	1	0,05
5	er	1	0,05
6	rn	1	0,05
7	ne	1	0,05
8	et	1	0,05
9	t_	2	0,08
10	_c	1	0,05
11	ce	1	0,05
12	ep	1	0,05
13	pa	1	0,05
14	at	1	0,05

Diketahui : jumlah dokumen sentimen positif = 1

jumlah dokumen setiment negatif= 1

maka nilai $P(V_j)$:

P(positif) = 1/2 = 0,5

P(negatif) = 1/2 = 0,5

Dokumen 3 : Dokumen yang akan diklasifikasi

Tabel 3.18 Daftar kata yang akan diklasifikasi

no.	*keyword*
1	_i
2	in
3	nt
4	te
5	er
6	rn
7	ne
8	et
9	t_

Tabel 3.18 Daftar kata yang akan diklasifikasi (Lanjutan)

no.	***keyword***
10	_c
11	ce
12	ep
13	pe
14	et
15	t_

Pada tahap klasifikasi dimulai dengan pencarian nilai probabilitas terhadap kata-kata yang ada pada tabel 3.16 yaitu dengan membandingkan kata-kata pada tabel diatas dengan tabel pada dokumen pembelajaran satu dan dokumen pembelajaran dua. Bila kata pada tabel 3.16 sama dengan kata pada dokumen pembelajaran satu dan dokumen pembelajaran dua maka nilai probabilitas yang ada pada dokumen pembelajaran dijadikan nilai probabilitas pada kata di tabel 3.16. Namun jika kata tidak sama maka nilai frekuensi pada tabel 3.16 sama dengan nol. Perhitungannya adalah sebagai berikut :

Tabel 3.19 Pencarian nilai probabilitas pada kata yang akan diklasifikasi pada kategori sentimen positif

No.	Kata yang akan diklasifikasi (tabel 3.6)	*N-gram tweet* positif(tabel 3.15)	Frekuensi n-gram karakter kata yang akan diklasifikasi(n_k)	Nilai probabilitas n-gram karakter kata yang akan diklasifikasi (P(xi|Vj))
1	_i	_i	1	(1+1)/(15+19) = 0,05
2	in	in	1	(1+1)/(15+19) = 0,05
3	nt	nt	1	(1+1)/(15+19) = 0,05
4	te	te	1	(1+1)/(15+19) = 0,05
5	er	er	1	(1+1)/(15+19) = 0,05
6	rn	rn	1	(1+1)/(15+19) = 0,05
7	ne	ne	1	(1+1)/(15+19) = 0,05
8	et	et	1	(1+1)/(15+19) = 0,05
9	t_	t_	2	(2+1)/(15+19) = 0,08
10	_c	_c	1	(1+1)/(15+19) = 0,05
11	ce	ce	1	(1+1)/(15+19) = 0,05
12	ep	ep	1	(1+1)/(15+19) = 0,05
13	pe	pa	0	(0+1)/(15+19) = 0,02
14	et	et	1	(1+1)/(15+19) = 0,05

Berdasarkan nilai probabilitas diatas kita bisa menghitung nilai dari V_{MAP} untuk yaitu

$= \underset{V_j \, e \, V}{\arg max} \prod_{i=1}^{n} P(x_i|V_j)P(V_j)$

$= \underset{(positif)}{\arg max} \prod_{i=1}^{10} P(x_i|V_j)P(V_j)$

= (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,08) (0,05) (0,05) (0,05) (0,02) (0,05)

= 3,90625E-19

Perhitungan probabilitas pada sentimen negatif:

Tabel 3.20 Pencarian nilai probabilitas pada kata yang akan diklasifikasi pada kategori setimen negatif

No.	*N-gram* karakter yang akan diklasifikasi (tabel 3.6)	*N-gram tweet* negatif(tabel 3.16)	Frekuensi *N-gram* karakter kata yang akan diklasifikasi(n_k)	Nilai probabilitas *N-gram* karakter kata yang akan diklasifikasi (P(xi|Vj))
1	_i	_i	1	(1+1)/(15+19) = 0,05
2	in	in	1	(1+1)/(15+19) = 0,05
3	nt	nt	1	(1+1)/(15+19) = 0,05
4	te	te	1	(1+1)/(15+19) = 0,05
5	er	er	1	(1+1)/(15+19) = 0,05
6	rn	rn	1	(1+1)/(15+19) = 0,05
7	ne	ne	1	(1+1)/(15+19) = 0,05
8	et	et	1	(1+1)/(15+19) = 0,05
9	t_	t_	2	(2+1)/(15+19) = 0,08
10	_c	_l	0	(0+1)/(15+19) = 0,02
11	ce	le	0	(0+1)/(15+19) = 0,02
12	ep	el	0	(0+1)/(15+19) = 0,02
13	pe	le	0	(0+1)/(15+19) = 0,02
14	et	et	1	(1+1)/(15+19) = 0,05

Berdasarkan nilai probabilitas diatas kita bisa menghitung nilai dari V_{MAP} untuk yaitu

$= \underset{V_j \, e \, V}{\arg max} \prod_{i=1}^{n} P(x_i|V_j)P(V_j)$

$= \underset{(negatif)}{\arg max} \prod_{i=1}^{10} P(x_i|V_j)P(V_j)$

= (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,05) (0,08) (0,02) (0,02) (0,02) (0,02) (0,05)

= 2,5E-20

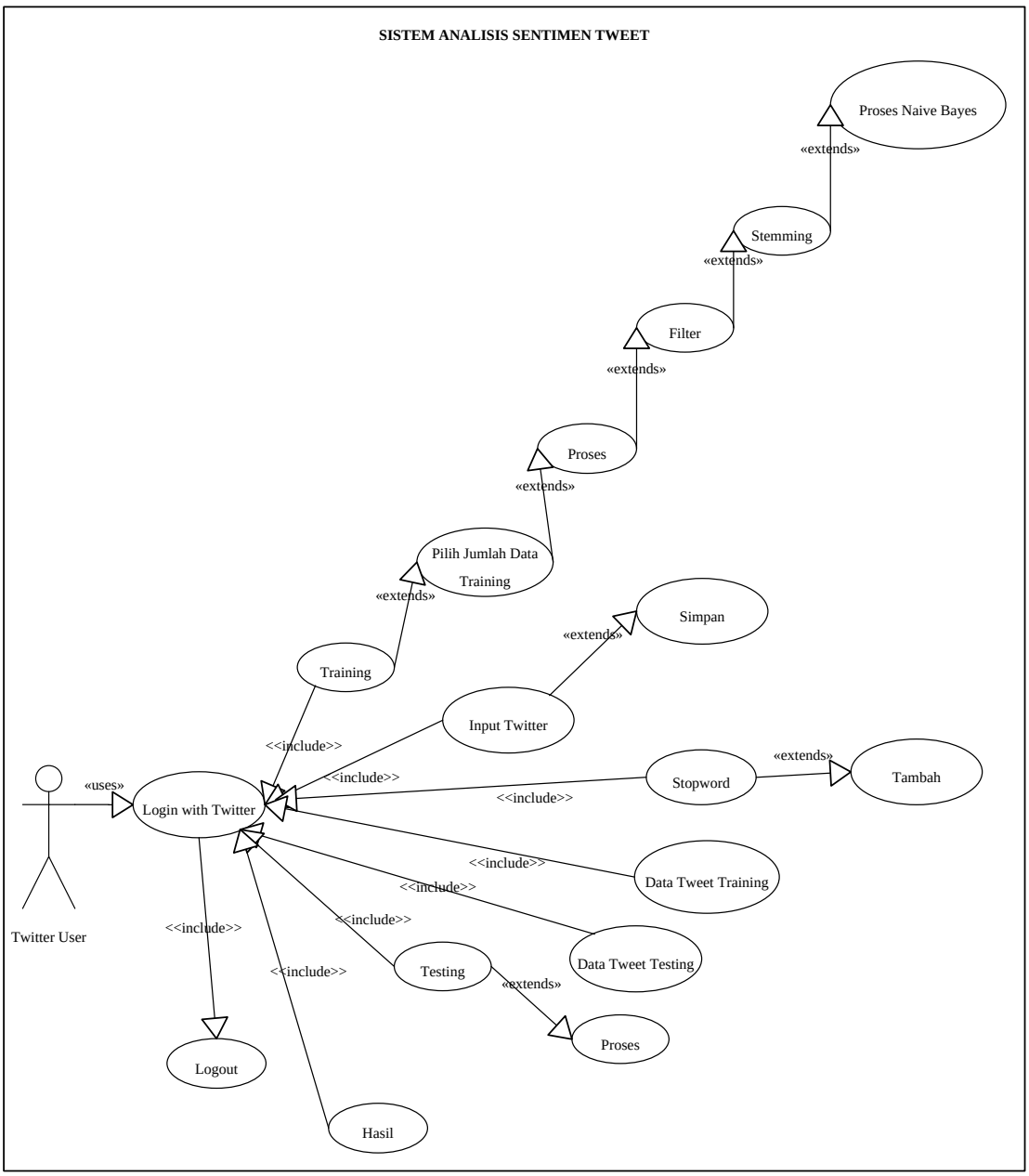
Pada hasil perhitungan tersebut, didapat bahwa nilai Vmap untuk kategori sentimen positif memiliki nilai tertinggi dibandingkan kategori sentimen negatif.

Sehingga, dapat disimpulkan bahwa dokumen ketiga merupakan *Tweet* yang memiliki nilai sentimen positif.

3.3 Perancangan Sistem

3.3.1 Diagram Use Case

Pada sistem analisis sentimen pada Twitter ini mempunyai 1 Aktif yaitu Twitter *User*. Gambaran diagram *use case* sistem dapat dilihat pada gambar 3.1.



Gambar 3.10 Diagram *Use Case*

3.3.2 Definisi Use Case

Berikut adalah deskripsi pendefinisian *use case* berdasarkan diagram *use case* yang digambarkan pada gambar 3.19 berikut ini.

Tabel 3.21 Definisi *use case*

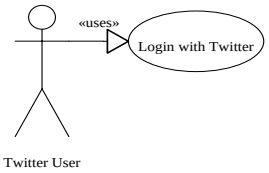
No.	*Use Case*	Deskripsi
1.	*Login*	Merupakan proses pengecekkan hak akses siapa yang berhak mengakses pengolahan data sistem klasifikasi *Tweet* ini.
2.	*Training*	Merupakan proses untuk mengklasifikasi *Tweet* yang sudah diketahui kategorinya dimana proses ini melewati proses *text preprocessing*, proses *filtering*, proses *stemming*, memilih kategori *Tweet* yang sesuai, dan proses *naïve bayes.*
3.	*Testing*	Merupakan proses untuk mengklasifikasi *Tweet* yang belum diketahui kategorinya dimana proses ini melewati proses *text preprocessing*, proses *filtering*, proses *stemming*, proses *naïve bayes* dan melihat hasil *testing*.
4.	*Stopword*	Merupakan proses untuk melihat data *stopword.*
5.	*Tweet training*	Merupakan proses untuk melihat *Tweet* yang digunakan untuk proses *training*
6.	*Tweet testing*	Merupakan proses untuk melihat *Tweet* yang digunakan untuk proses *testing*
7.	Hasil	Merupakan proses untuk melihat *Tweet* hasil dari proses *testing*
8.	Input *Tweet*	Merupakan proses untuk memasukkan *Tweet*
9.	*Logout*	Merupakan proses untuk keluar dari sistem.

3.3.3 Model Spesifikasi Use Case

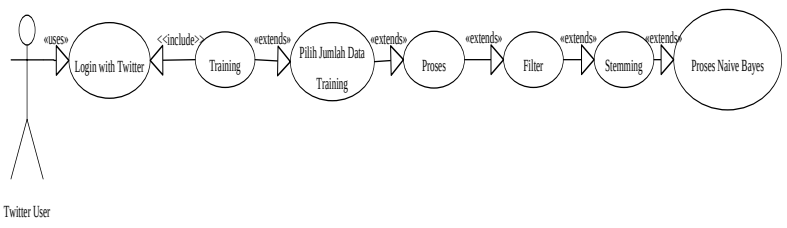
Spesifikasi *use case* sistem pengklasifikasi *Tweet* dilakukan berdasarkan diagram *use case* yang digambarkan pada gambar 3.7. Berikut adalah tabel spesifikasi setiap *use case*.

3.3.2.1 Model Spesifikasi Use Case User

Tabel 3.22 Spesifikasi *use case login*

Tipe Use Case	**Penjelasan**
1.	Use case : *Login* 
Tujuan	Untuk masukkan kedalam sistem analisis sentimen
Deskripsi	Proses untuk melakukan *login* sistem
Aktor	Twitter *User*
Kondisi awal	*User* membuka halaman *login*
Skenario	Skenario dasar : *1. User* membuka halaman *login* *2. User* mengisi username dan *password* *3. User* menekan tombol *login* 4. Sistem akan menampilkan halaman *User*
Kondisi Akhir	Pengunggah masuk ke dalam sistem analisis sentimen

Tabel 3.23 Spesifikasi *use case training*

Tipe Use Case	**Penjelasan**
2.	Use case : *Training* 

Tabel 3.23 Spesifikasi *use case training* (Lanjutan)

Tipe Use Case	Penjelasan
Deskripsi	Proses untuk melakukan *training* terhadap *Tweet* yang sudah diketahui kategori sentimennya.
Aktor	Twitter *User*
Kondisi awal	Twitter *User* berada di halaman *User*
Skenario	Skenario dasar : 1. *User* berada di halaman *User* 2. *User* membuka menu *training* *3.* *User* memilih jumlah *Tweet* yang akan di-*training* 4. Kemudian *User* menekan tombol “proses” 5. Sistem akan menampilkan hasil dari proses *text preprocessing* 6. Kemudian *User* menekan tombol “proses *filtering”* 7. Sistem akan menampilkan hasil dari proses *filtering* 8. Kemudian *User* menekan tombol “proses *stemming”* 9. Sistem akan menampilkan hasil dari proses *stemming* 10. Kemudian *User* menekan tombol “proses training” 11. Sistem akan menampilkan pesan bahwa proses *training* berhasil dan sistem akan menyimpan *Tweet* yang sudah di-*training.*
Kondisi Akhir	Hasil *Training* disimpan sesuai dengan jumlah data yang dipilih *User*

Table 3.24 Spesifikasi *use case* proses *testing*

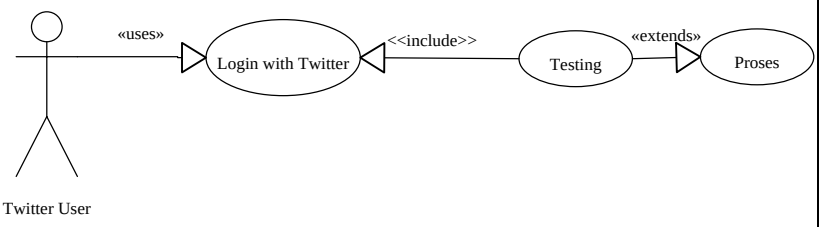
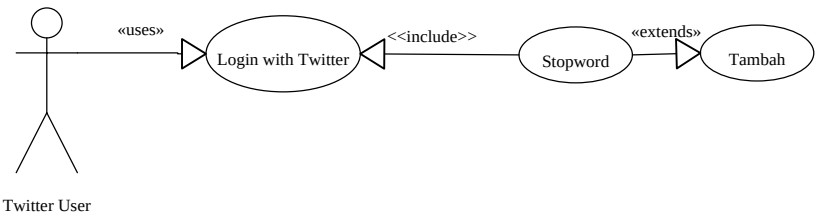
Type Use Case	Penjelasan
3.	Use Case : Testing 

Table 3.24 Spesifikasi *use case* proses *testing* (Lanjutan)

Tipe Use Case	Penjelasan
Tujuan	Menampilkan halaman proses *testing*
Deskripsi	Proses untuk melakukan *testing* terhadap *Tweet* yang sudah diketahui kategorinya.
Aktor	*User*
Kondisi awal	*User* berada di halaman *User*
Skenario	Skenario dasar : 1. *User* berada di halaman *User* 2. *User* membuka menu *testing* 3. *User* memilih jumlah *Tweet* yang akan di testing 4. Kemudian *User* menekan tombol “proses” 5. Kemudian sistem akan menampilkan *Tweet* sesuai dengan banyaknya *Tweet* yang kita pilih 6. Kemudian *User* menekan tombol “proses testing” 7. Sistem akan menampilkan hasil dari proses *testing* dan sistem akan menyimpan *Tweet* yang sudah di-*testing.*
Kondisi Akhir	Sistem menampilkan hasil dari proses *testing*.

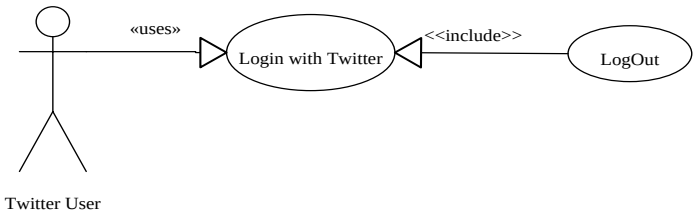
Tabel 3.25 Spesifikasi *use case* melihat data *stopword*

Type Use Case	Penjelasan
4.	Use case : *stopword* 
Tujuan	Menampilkan halaman data *stopword*
Deskripsi	Proses untuk melihat data *stopword*
Aktor	*User*
Kondisi awal	*User* harus *login* terlebih dahulu

Tabel 3.25 Spesifikasi *use case* melihat data *stopword* (Lanjutan)

Type Use Case	Penjelasan
Skenario	Skenario dasar : 1. *User* berada di halaman *User* 2. *User* memilih menu data *stopword* 3. Sistem akan menampilkan halaman data *stopword* Skenario alternatif : 1. *User* dapat menambah *stopword*
Kondisi Akhir	*User* berada di halaman data *stopword*

Tabel 3.26 spesifikasi *use case logout*

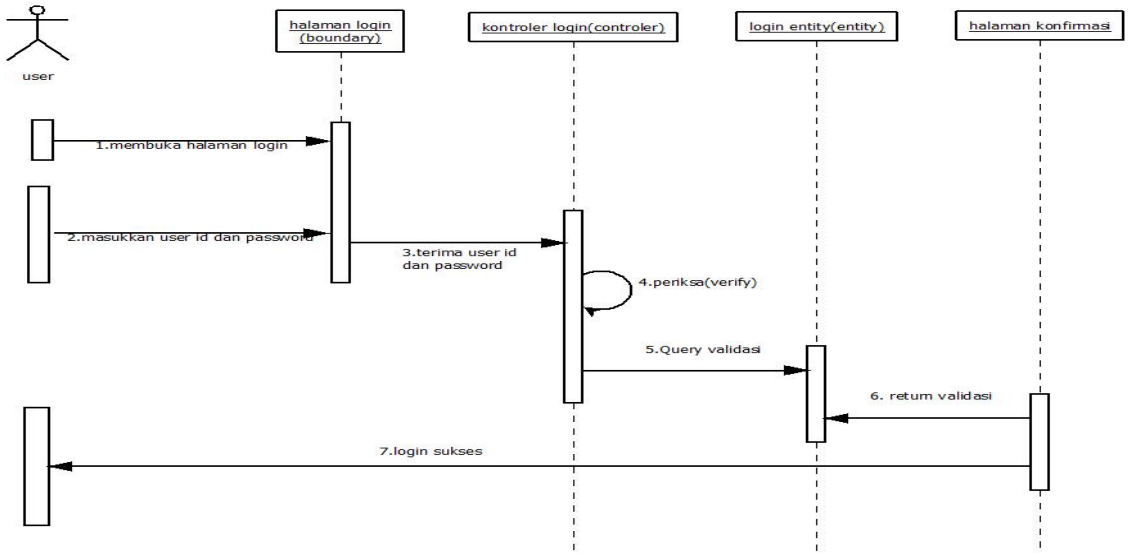
Type Use Case	Penjelasan
	Use case : *Logout* 
Tujuan	Keluar dari halaman *User*
Deskripsi	Proses untuk keluar dari halaman *User*
Aktor	*User*
Kondisi awal	*User* harus *login* terlebih dahulu
Skenario	Skenario dasar : *1.* *User* berada di halaman *User* 2. *User* memilih *logout* 3. Sistem akan menampilkan halaman *login*
Kondisi Akhir	*User* berada di halaman data *Login*

3.3.4 Model Interaksi Diagram Sequence

Berikut ini merupakan *sequence diagram* yang menggambarkan interaksi antar objek di dalam dan sekitar sistem :

1. Diagram Sequence *Login*

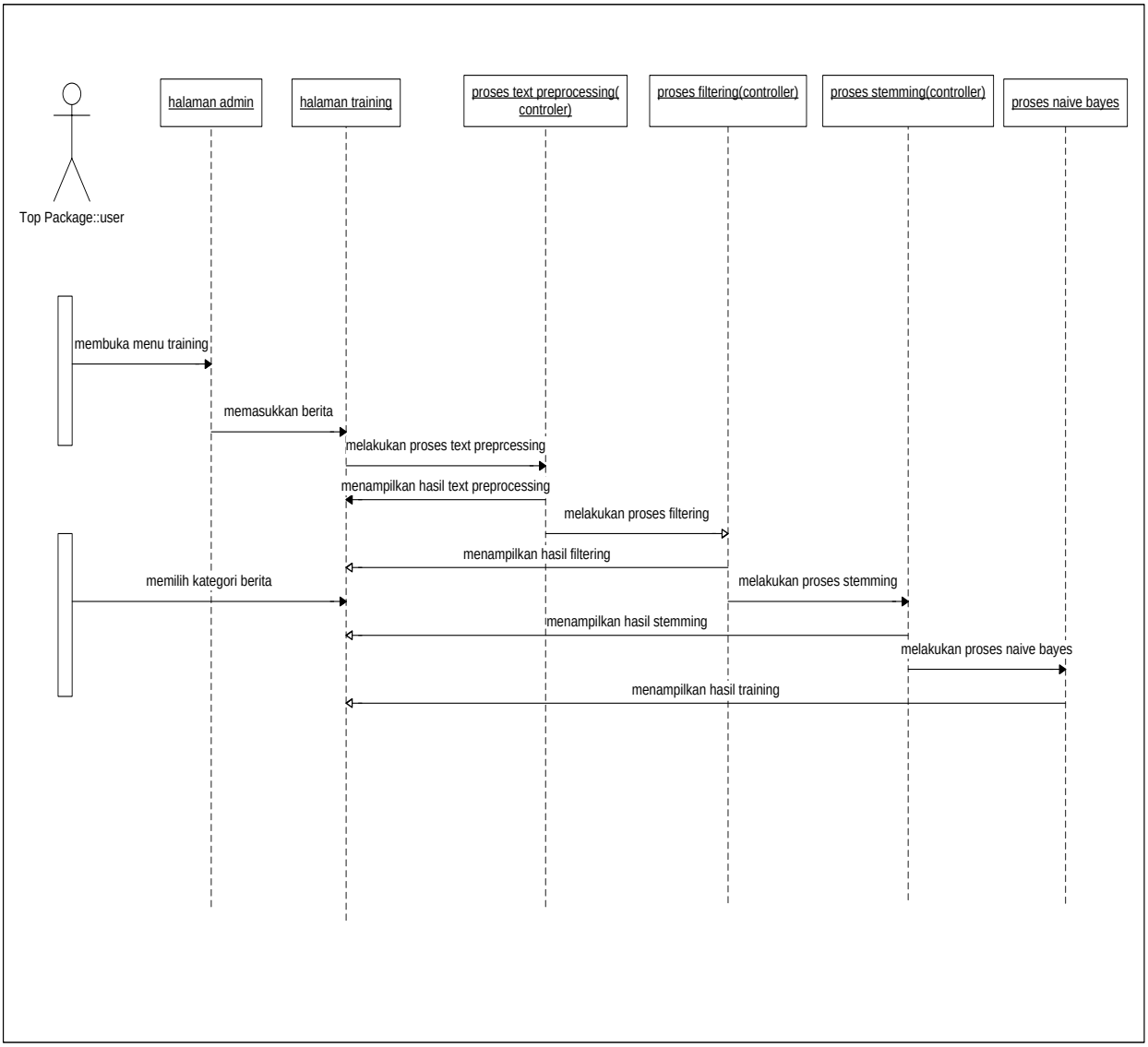
Sequence diagram login menggambarkan alur sistem untuk melakukan *login* ke dalam sistem. Sequence ini dimulai ketika *user* memilih menu *login*, kemudian *username* dan *password*. Kemudian sistem akan memeriksa apakah *username* dan *password* yang di masukkan benar atau salah jika benar atau salah sistem akan memberikan konfirmasi. Sequence diagram untuk proses ini adalah sebagai berikut :



Gambar 3.11 *Sequence diagram login*

2. Sequence Diagram Proses *Training*

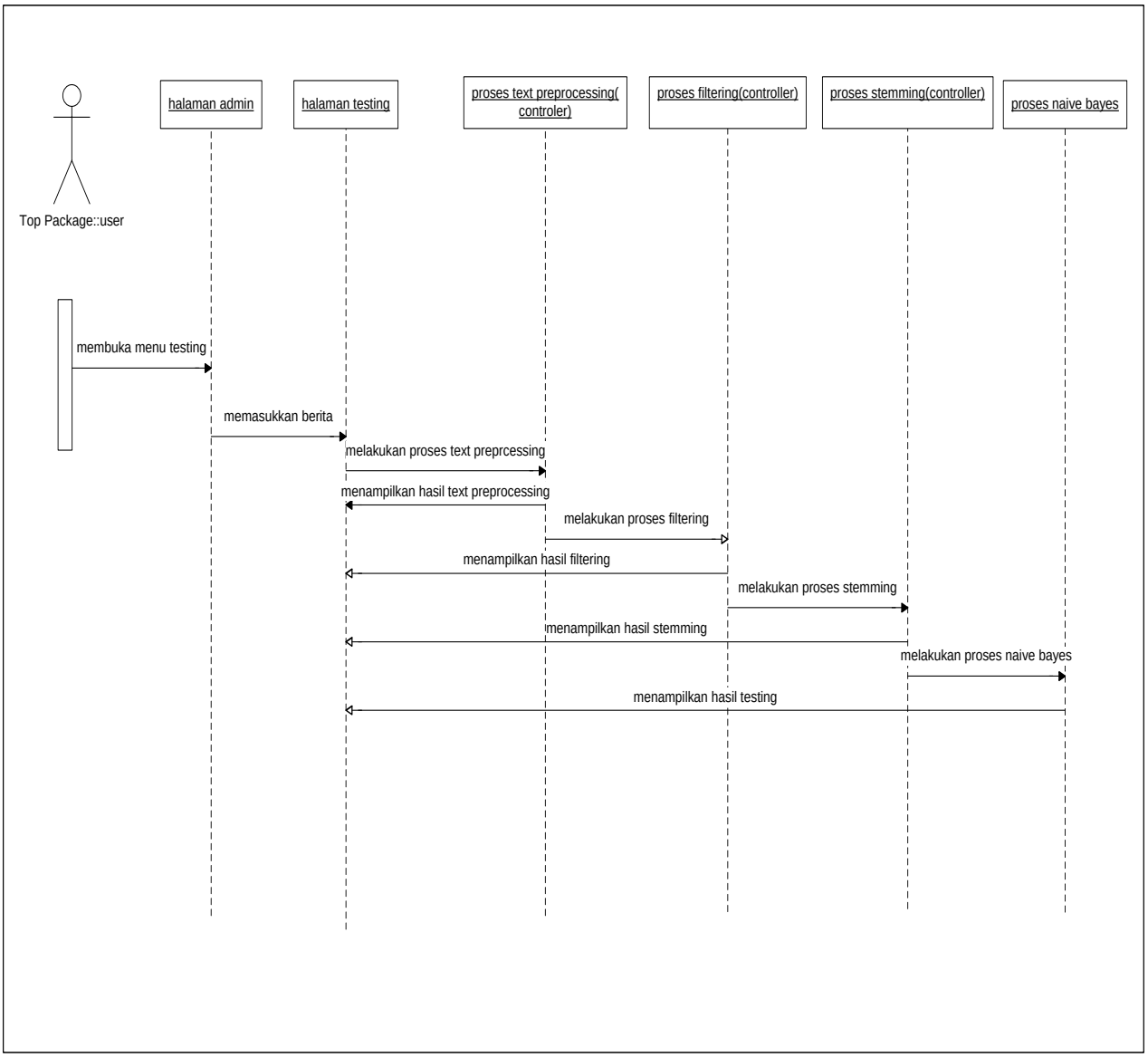
Sequence diagram proses *training* menggambarkan alur sistem melakukan proses *training* terhadap *Tweet* yang sudah diketahui kategorinya. proses ini dimulai ketika user memasukkan *Tweet* yang sudah diketahui kategorinya. Kemudian sistem akan melakukan tahap-tahap *text mining* yaitu *text preprocessing*, *filtering*, *stemming* dan sistem juga akan melakukan proses *naïve bayes classifier*. Pada akhirnya sistem akan memberitahukan kepada *user* bahwa proses *training* sudah berhasil. Sequence diagram untuk proses ini adalah sebagai berikut :



Gambar 3.12 *Sequence diagram* proses *training*

3. Sequence Diagram Proses *testing*

Sequence diagram proses *testing* menggambarkan alur sistem untuk melakukan proses klasifikasi terhadap *Tweet* yang belum diketahui kategorinya. Proses ini dimulai ketika *user* memilih menu proses *testing*, kemudian *user* akan memasukkan *tweet* yang belum diketahui kategorinya. Kemudian sistem akan melakukan tahap-tahap *text mining* yaitu *text preprocessing*, *filtering*, *stemming* dan sistem juga akan melakukan proses *naïve bayes classifier*. Pada akhirnya sistem akan menampilkan hasil dari proses klasifikasi, sesuai kategori *Tweet* tersebut. *Sequence diagram* untuk proses ini adalah sebagai berikut :

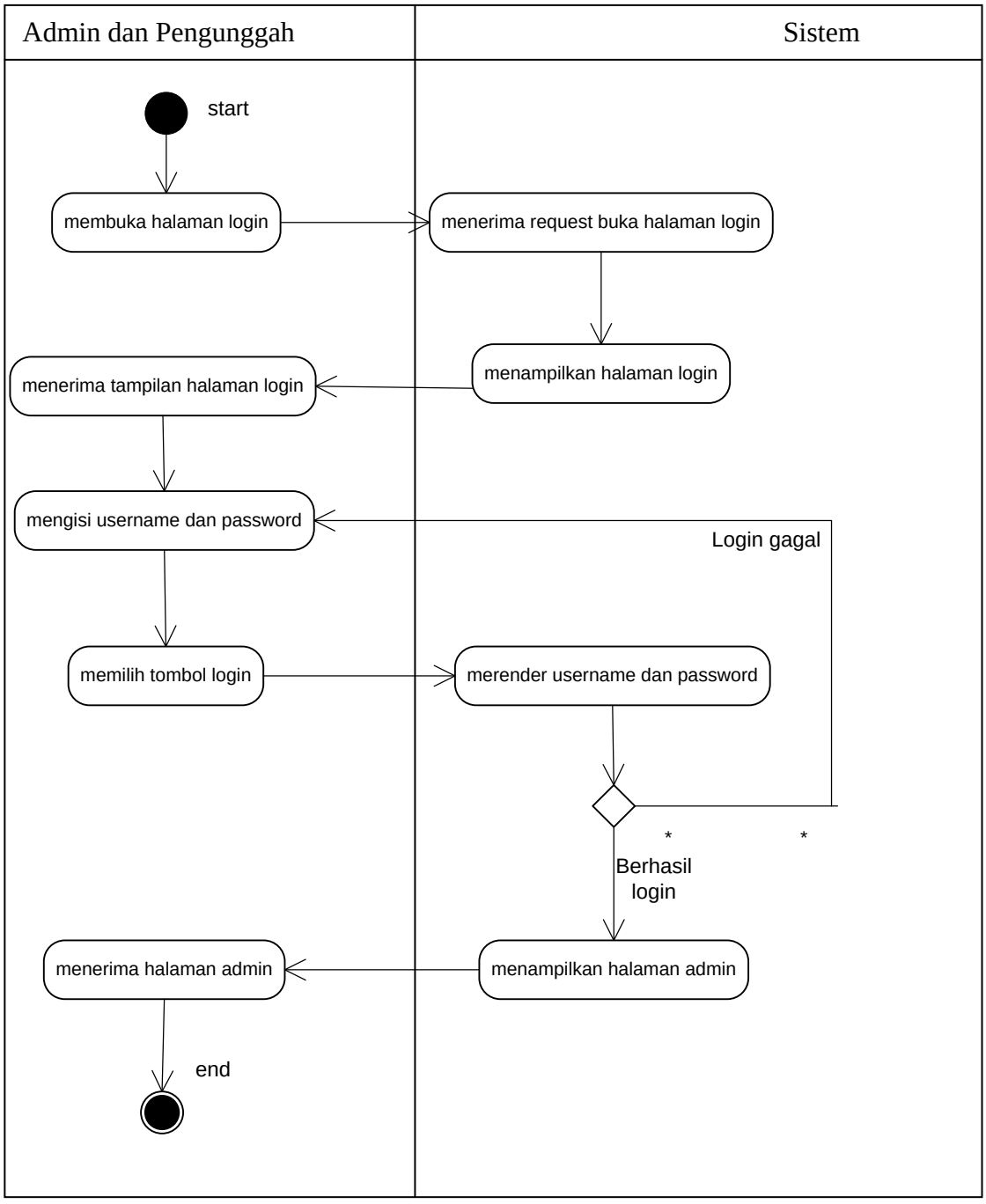


Gambar 3.13 *Sequence diagram* proses *testing*

3.3.5 Diagram Aktifitas

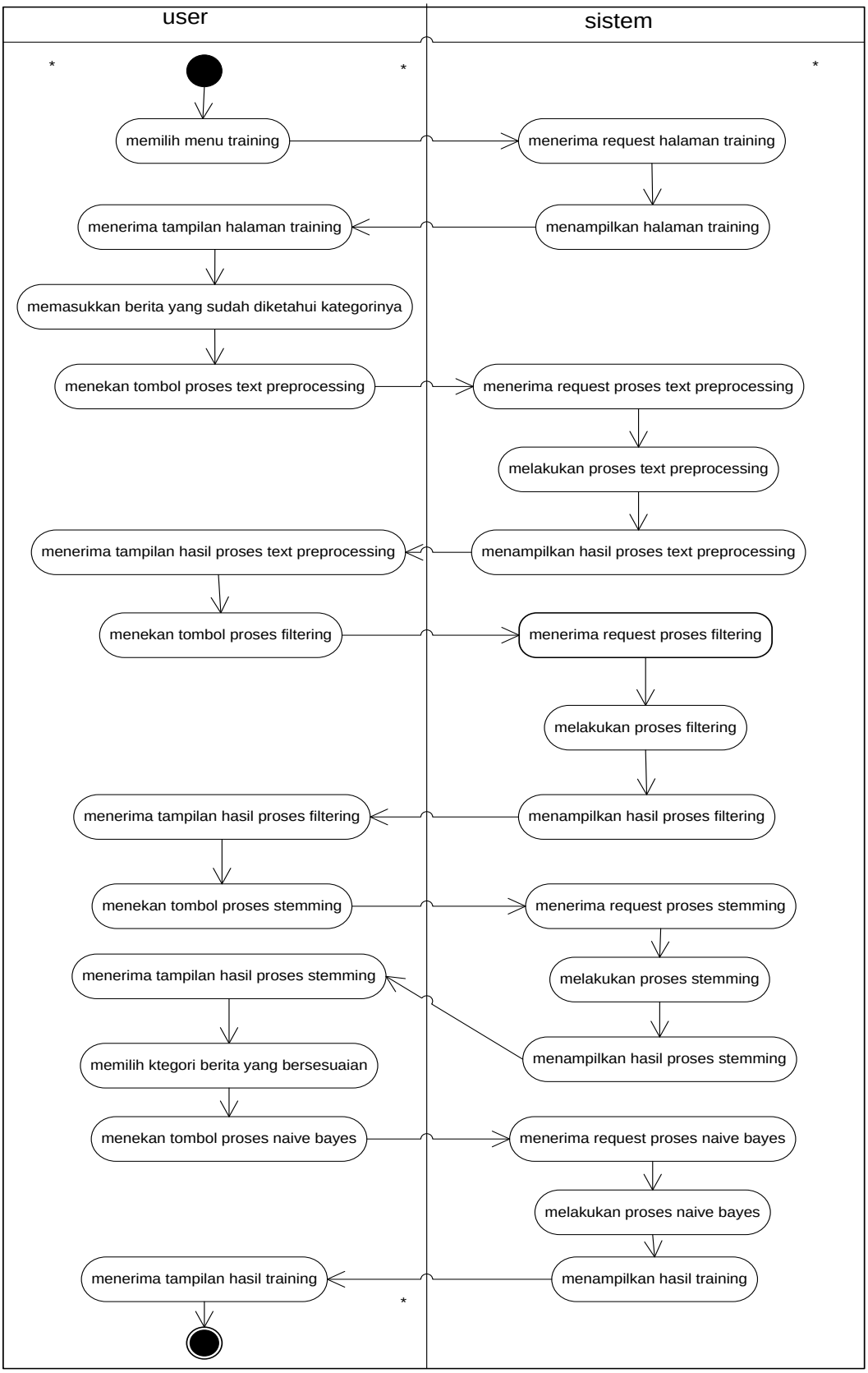
Diagram aktifitas ini digunakan untuk menggambarkan berbagai alur aktifitas yang sedang berjalan. Berikut merupakan diagram aktifitas dari sistem klasifikasi *Tweet* :

3.3.5.1 Diagram Aktifitas Login



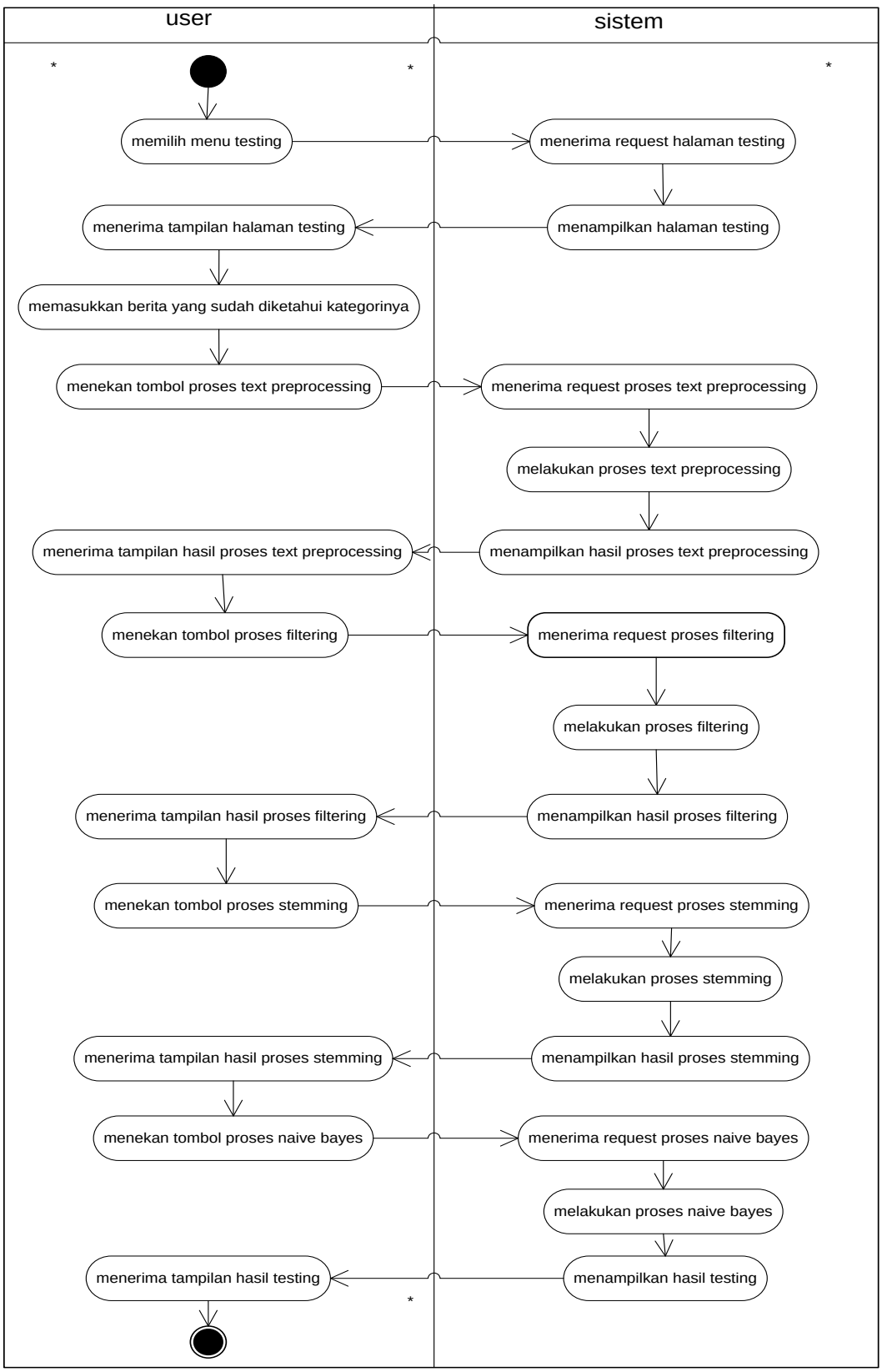
Gambar 3.14 Diagram aktifitas *login*

3.3.5.2 Diagram Aktifitas Proses Training



Gambar 3.15 Diagram aktifitas proses *training*

3.3.5.3 Diagram Aktifitas Proses testing



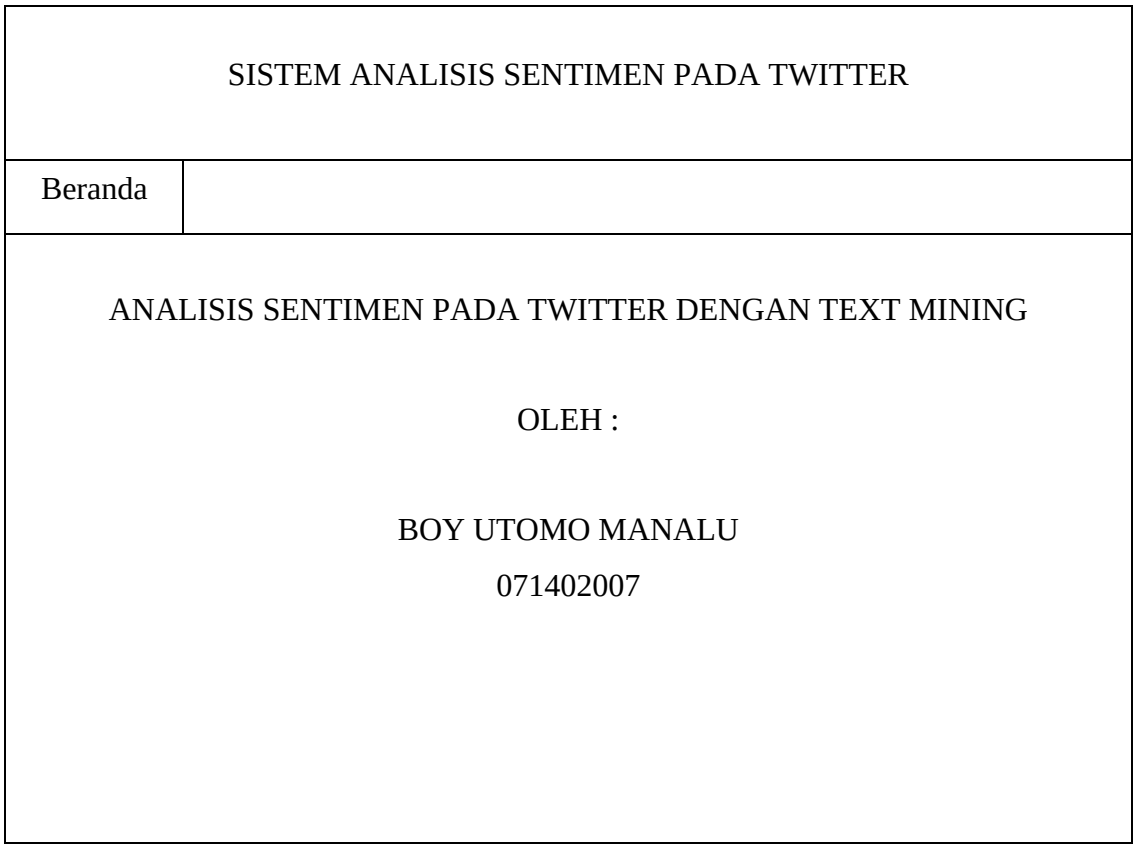
Gambar 3.16 Diagram aktifitas proses *testing*

3.4 Perancangan Tampilan Antarmuka

Perancangan tampilan antarmuka digunakan untuk menggambarkan tampilan antarmuka sebelumnya. Beberapa rancangan tampilan antarmuka yang digunakan dalam skripsi ini sebagai berikut :

3.4.1 Rancangan Halaman Utama

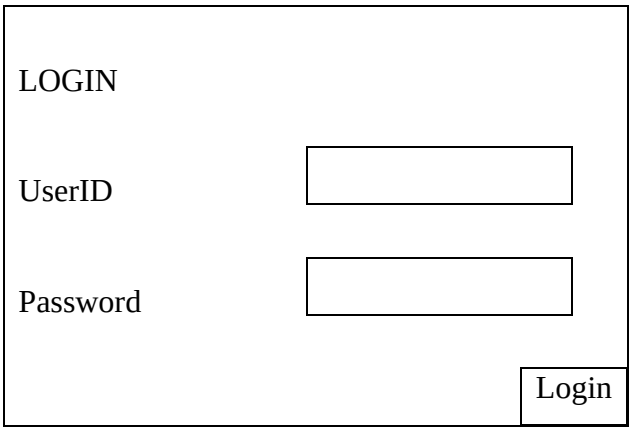
Rancangan halaman utama dapat dilihat pada gambar 3.14.



Gambar 3.17 Rancangan halaman utama

3.4.2 Rancangan Halaman Login

Rancangan halaman login digunakan untuk merancang halaman login yang berfungsi sebagai otentifikasi pengunggah untuk masuk kedalam sistem. Rancangan halaman *login* dapat dilihat pada gambar 3.15



Gambar 3.18 Rancangan halaman *login*

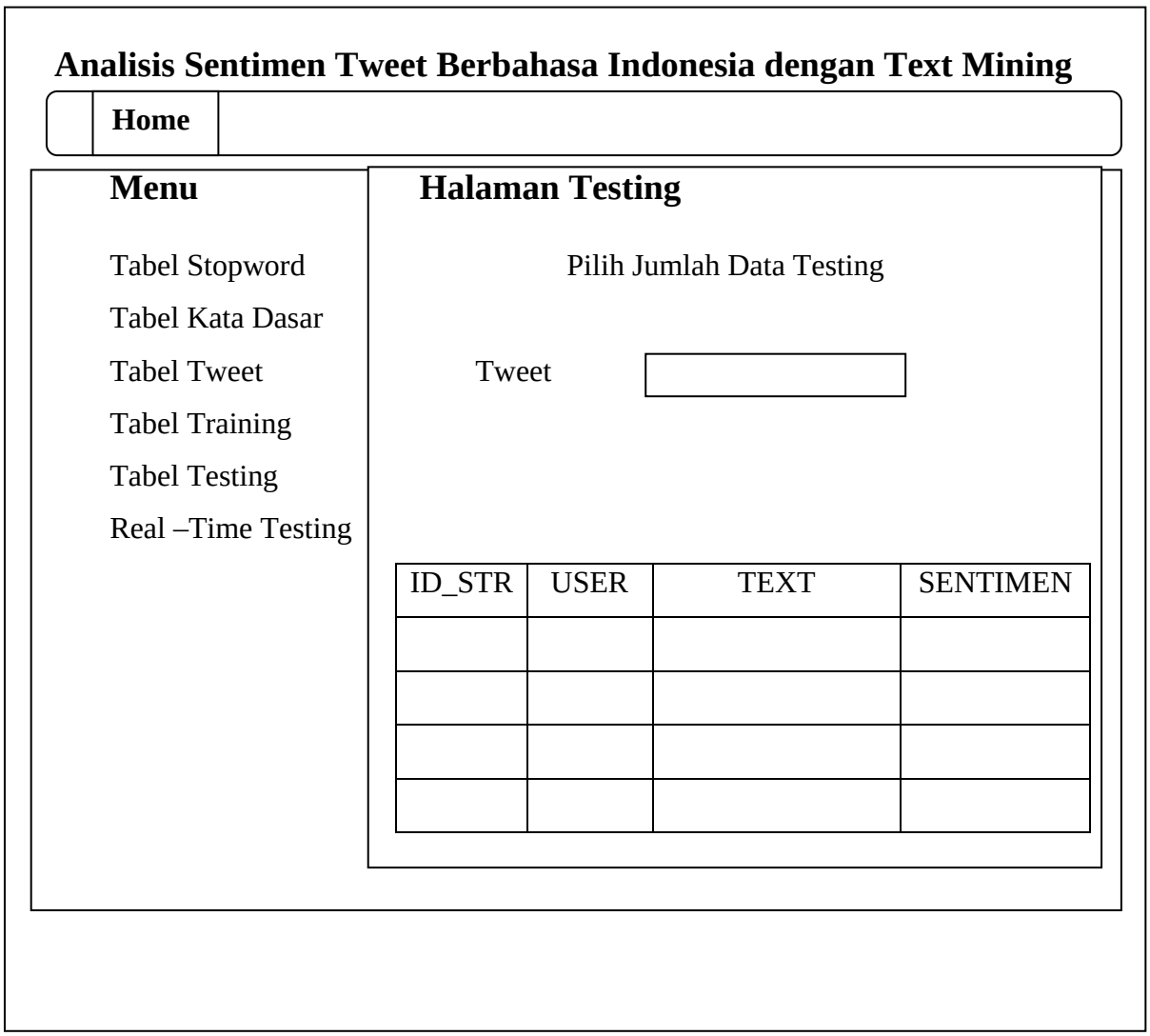
3.4.3 Rancangan Halaman Tweet training

Halaman *tweet training* adalah halaman yang menampilkan tweet yang digunakan untuk *training* dimana sentimen *Tweet* tersebut sudah diketahui kategorinya. Rancangan halaman *tweet training* dapat lihat pada gambar 3.16.

Gambar 3.19 Rancangan halaman *tweet training*

3.4.4 Rancangan Halaman Tweet Testing

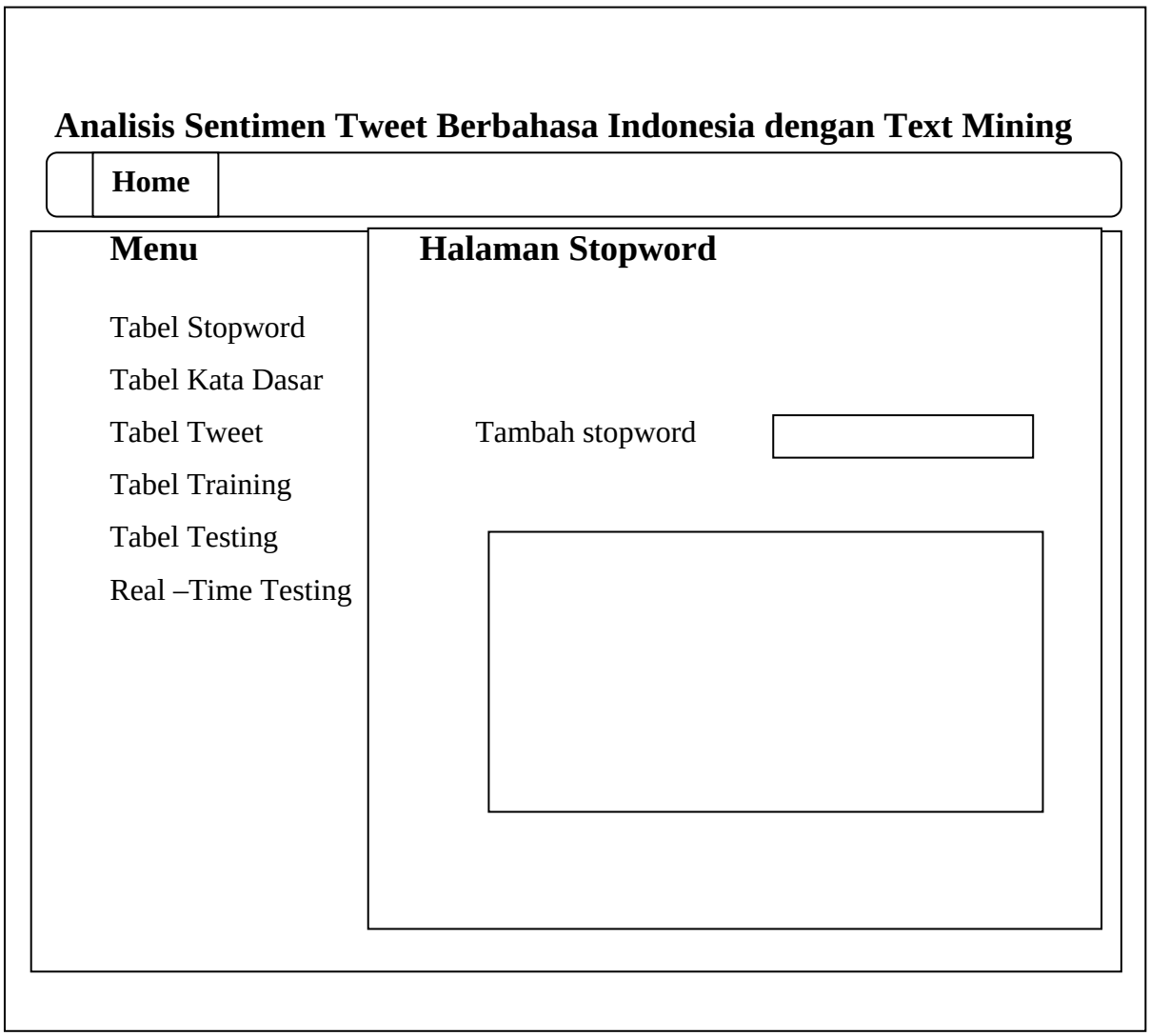
Halaman *tweet testing* adalah halaman yang menampilkan *tweet* yang digunakan untuk *testing* dimana *tweet* tersebut belum diketahui kategorinya. Rancangan halaman *tweet testing* dapat lihat pada gambar 3.17.



Gambar 3.20 Rancangan halaman *tweet testing*

3.4.5 Rancangan Halaman Stopword

Rancangan halaman tampilan *stopword* digunakan untuk merancang halaman *stopword.* Tampilan ini akan menampilkan *stopwordID* dan *stopword.* Dalam tampilan ini juga terdapat *form* untuk menambah data *stopword*. Rancangan tampilan halaman *stopword* dapat dilihat pada gambar



Gambar 3.21 Rancangan halaman *stopword*

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Setelah melalui tahap analisis dan perancangan, tahap selanjutnya untuk mengembangkan suatu perangkat lunak adalah tahap implementasi dan pengujian sistem. Untuk mengetahui apakah implementasi perangkat lunak tersebut berhasil atau tidak, diperlukan pengujian. Berikut ini hasil implementasi dan pengujian dari aplikasi yang telah dibangun.

4.1 Implementasi Sistem

Berdasarkan hasil analisis dan perancangan sistem yang telah dilakukan, maka dilakukan implementasi sistem klasifikasi *Tweet* dengan metode *text mining* yang menggunakan algoritma *naïve bayes classifier* ke dalam bentuk program dengan menggunakan bahasa pemrograman PHP.

4.1.1 Spesifikasi Perangkat Keras dan Perangkat Lunak yang Digunakan

Lingkungan implementasi merupakan lingkungan perangkat lunak yang digunakan untuk membangun dan mengoperasikan perangkat lunak. Pada bagian ini semua analisis dan perancangan akan direpresentasikan ke dalam bentuk perangkat lunak yang dapat menunjang aktifitas pengguna dalam kehidupan sehari-hari.

Spesifikasi perangkat keras yang digunakan :

1. Processor AMD E-450 APU
2. Memory RAM yang digunakan 2 GB
3. Kapasitas Hardisk 250GB

Spesifikasi perangkat lunak yang digunakan :

1. Windows 7 Ultimate
2. Apache Server 2.4
3. PHP 5.6
4. MySQL

4.1.2 Tampilan Utama Sistem

Bentuk tampilan utama sistem dapat dilihat pada gambar 4.1.

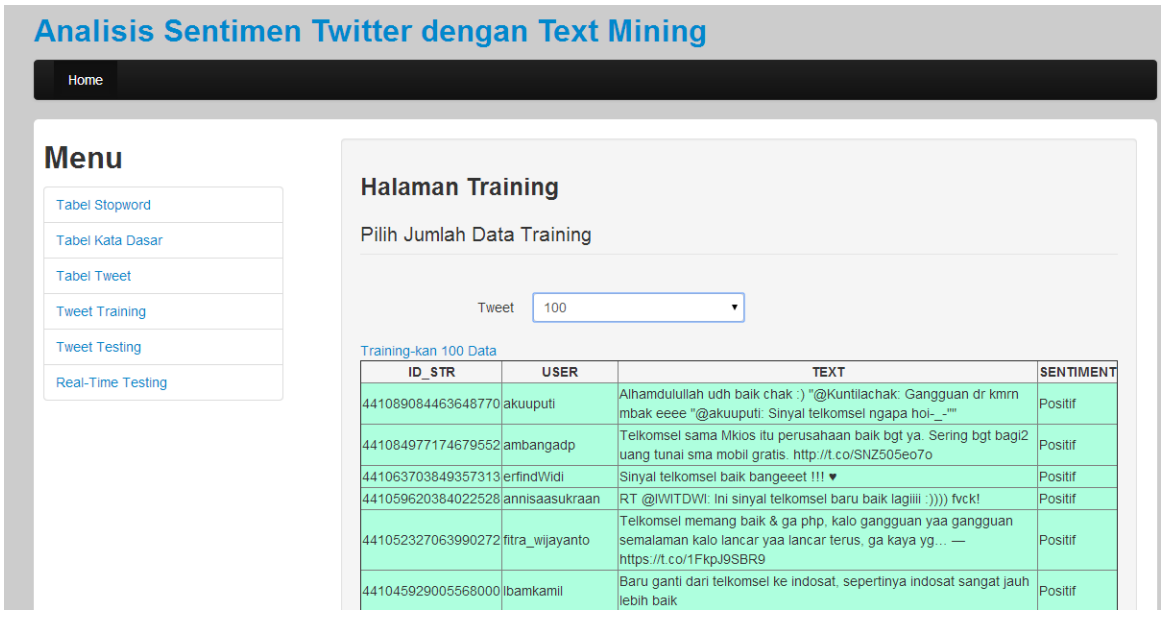
Gambar 4.1 Tampilan halaman utama sistem

Pada tampilan sistem terdapat menu-menu *Stopword*, Tabel Kata Dasar, Tabel *Tweet*, *Tweet Training*, *Tweet Testing*, dan *Real Time Testing*.

Pada Gambar 4.2 menampilkan hasil dari menu Tabel *Tweet* dimana pada tabel ini akan ditampilkan semua *Tweet* yang ada di dalam *Database* sistem ini.

Gambar 4.2 Tampilan menu *Tabel Tweet*

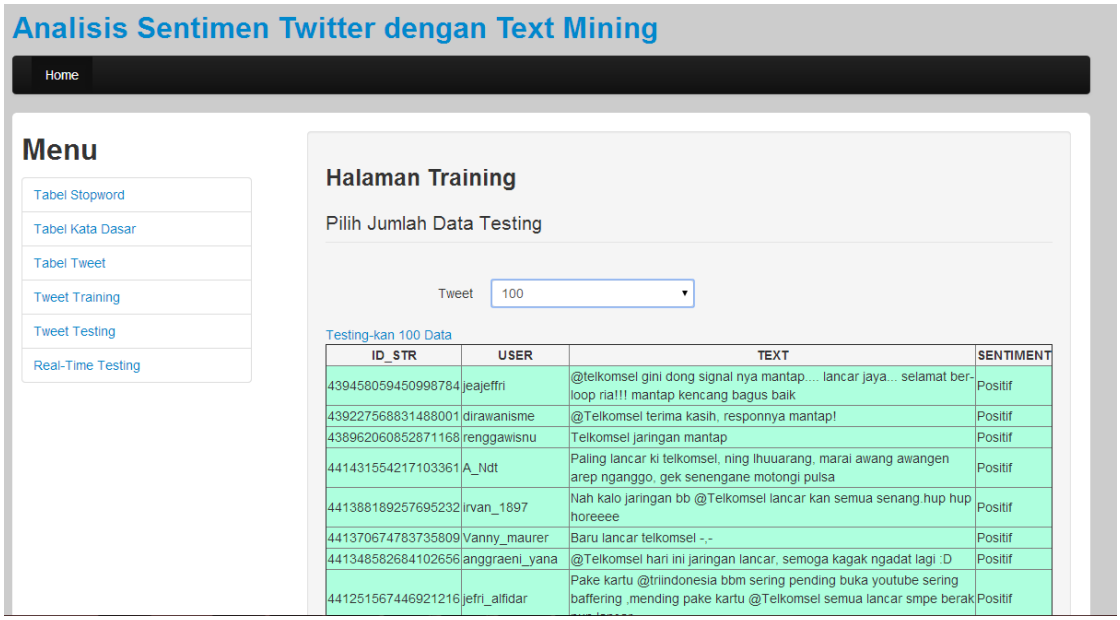
Pada tampilan menu *Tweet Training*, akan menampilkan sebuah form dimana Pengguna dapat memilih berapa jumlah *Tweet* yang akan dilatih, jumlah *Tweet* yang dipilih akan memanggil sejumlah pilihan tersebut untuk data latih yang mengandung sentimen Negatif, Positif dan Netral. Tampilan *Tweet Training* dapat kita lihat pada gambar 4.3.



Gambar 4.3 Tampilan isi *Tweet*

4.1.3 Tampilan Tweet Testing

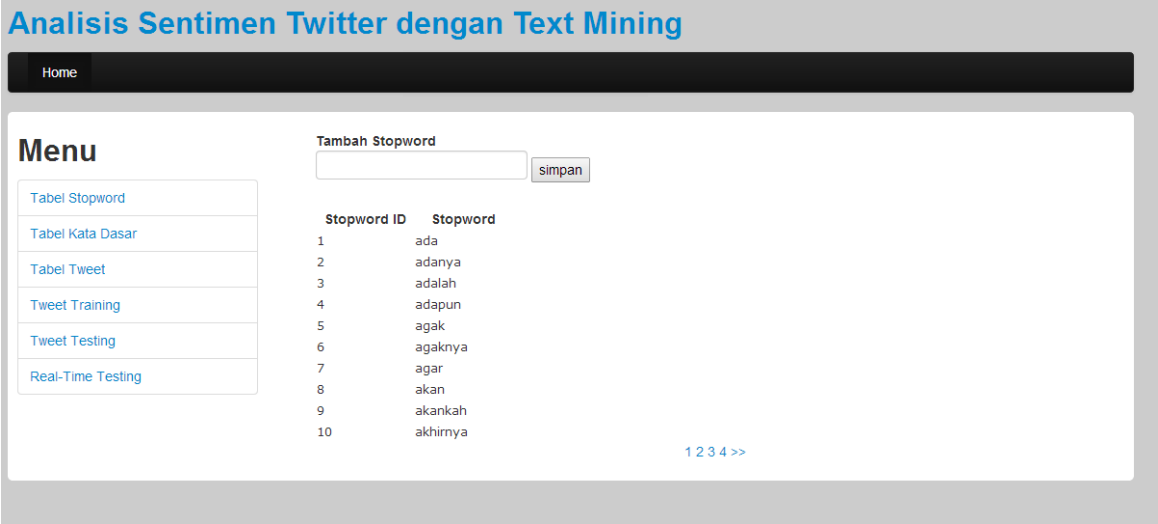
Pada tampilan menu *Tweet Testing*, akan menampilkan sebuah form dimana pengguna dapat memilih berapa jumlah *Tweet* yang akan diuji, jumlah *Tweet* yang dipilih akan memanggil sejumlah pilihan tersebut untuk data latih yang mengandung sentimen Negatif, Positif dan Netral. Tampilan Tweet testing dapa kita lihat pada gambar 4.3.



Gambar 4.4 Tampilan isi *tweet*

4.1.4 Tampilan Stopword

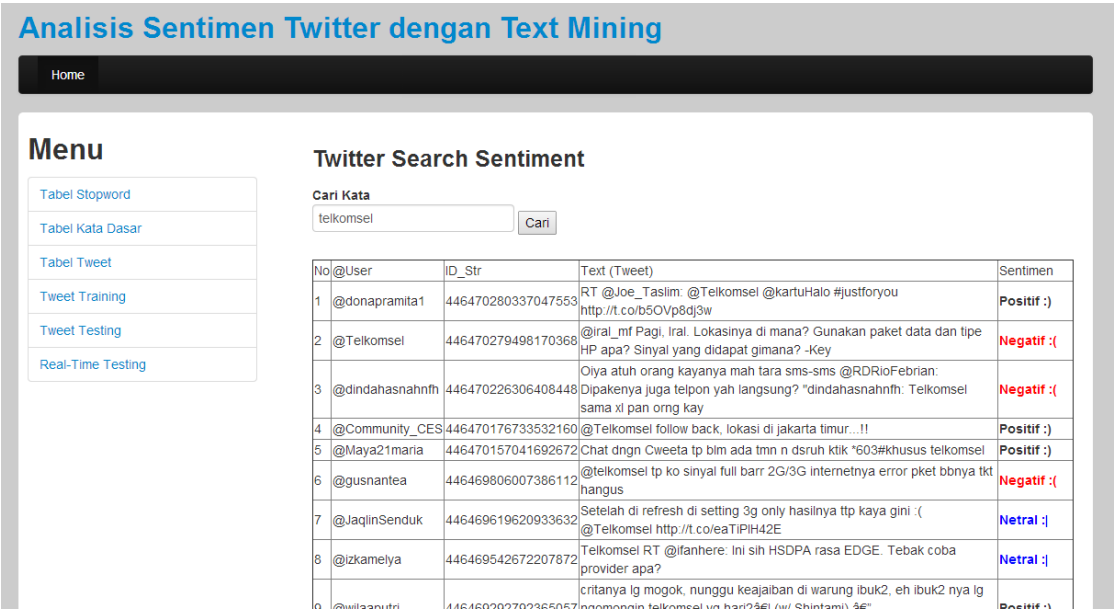
Tampilan *stopword* adalah tampilan yang menampilkan data *stopword*. dalam tampilan ini terdapat juga *textbox* untuk menambah *stopword* baru. Tampilan *stopword* dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan *stopword*

4.1.5 Tampilan Realtime Testing

Menu *Realtime Testing* akan menampilkan sebuah form input berupa teks pencarian, dimana sistem akan menggunakan Twitter API Search untuk melakukan pencari terhadak *keyword* tertentu dan kemudian sistem akan mengolah data yang masuk dari Twitter tersebut dan kemudian diuji apa sentimen yang terdapat pada kata atau *tweet* tersebut. Hasil Tampilan dan Pengujian *Realtime Testing* terdapat pada Gambar 4.6.



Gambar 4.7 Tampilan *Realtime testing*

4.2 Pengujian Sistem

Pada sub bab ini akan dijelaskan mengenai analisis hasil pengujian sistem yang berfungsi untuk mengetahui kinerja dari program dalam melakukan proses klasifikasi. Pengujian yang dilakukan pada sistem ini adalah pengujian *testing*.

Penulis melakukan pengujian *Training* dan *Testing* terhadap sistem analisis sentimen ini dengan skenario mengunakan data latih sebanyak 100 sampai 500 *Tweet* untuk data latihnya dan menggunakan 100 *Tweet* untuk data tes-nya.

100

1	@telkomsel gini dong signal nya mantap.... lancar jaya... selamat ber-loop ria!!! mantap kencang bagus baik	1	Negatif :(
2	@Telkomsel terima kasih, responnya mantap!	1	Positif :)
3	Telkomsel jaringan mantap	1	Negatif :(
4	Paling lancar ki telkomsel, ning lhuuarang, marai awang awangen arep nganggo, gek senengane motongi pulsa	1	Positif :)
5	Nah kalo jaringan bb @Telkomsel lancar kan semua senang.hup hup horeeee	1	Negatif :(
6	Baru lancar telkomsel -,-	1	Positif :)
7	@Telkomsel hari ini jaringan lancar, semoga kagak ngadat lagi :D	1	Negatif :(
8	Pake kartu @triindonesia bbm sering pending buka youtube sering baffering ,mending pake kartu @Telkomsel semua lancar smpe berak pun lancar	1	Positif :)
9	RT @missnovaq: Lagi diporotin sama @telkomsel nih, BIS masih aktif tapi twitteran kok pake pulsa. Layanan macet tapi motong pulsanya lancarâ€¦	0	Netral :|
10	@Telkomsel adakah penanganan khusus untuk para phreaker ? Tadi sempat coid, sekrang lancar jaya :D	1	Negatif :(
11	Direct dan injek masih lancar jaya :D @Telkomsel	1	Netral :|
12	Thank u ya @Telkomsel sudah dibantu, semoga lancar terus ya jaringannya â˜º	1	Positif

Gambar 4.8 Proses *Testing*

90	sinyal modem smartfren ciamik cepet buat online, deadline desain bisa langsung dikirim ke costumer @smartfrenworld #SFMegaBazaar14 #SFDay1	1	Negatif :(
91	kecepatan download tri kok lebih cepet dari pada smartfren ya -____-	1	Netral :|
92	angin cepet., hahaha :D RT @ardiekate: ada angin apa smartfren jd cpt gini	1	Netral :|
93	BARU NGERASAIN SMARTFREN CEPET, JADI GA MAU PINDAH HAHA. sampe 2,5mbps. donwload 9detik beres yg 13mb haha	1	Positif :)
94	Yeay pagi ini,smartfren lagi baik;3 cepet banget,gak lelet._.	1	Positif :)
95	Koneksi stabil, buka animeindo/youtube wafenya cepet... download or browsingan jadi nyaman :3 semoga aja smartfren ulimitednya gini terus :v	1	Netral :|
96	smartfren udah pulih. dan mendadak cepet. yay	1	Negatif :(
97	@fitrisumiyati56 iyaaa da apaa pelan pelan :D aku ngga bisa pelan lagi pake smartfren jadii cepet :D	1	Positif :)
98	pengen cepet cepet liat gadget smartfrennya yang keren dan canggih itu , donk ! smartfren kece badai (cont) http://t.co/u8MJiq3VnF	1	Netral :|
99	cepet gilaaa ni smartfren wahaha	1	Negatif :(
100	Disini smartfren cepet banget tapi coba aja kalo dibawa daerah lemotttt	1	Positif :)

Total Tweet : 100
Total Negatif: 0
Total Positif : 33
Total Netral:

Total Akurasi 33%

Gambar 4.9 Proses *Testing* (Lanjutan)

4.3 Hasil Pengujian

Pada pengujian yang dilakukan pada sistem analisis sentimen ini, dilakukan dengan menguji data *testing* pada beberapa kali proses *training*. Hasil Pengujian yang diperoleh pada pengujian ini akan dijelaskan dalam Tabel 4.1 dibawah ini:

Tabel 4.1 Hasil Pengujian

Training			Data Testing	Akurasi
Positif	Negatif	Netral		
100	100	100	100	51%
200	200	200	100	72%
300	300	300	100	87%
400	400	400	100	89%
500	500	500	100	92%
600	600	600	100	90%
700	700	700	100	90%
800	800	800	100	91%
900	900	900	100	93%
1000	1000	1000	1000	88%

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

1. Aplikasi ini mampu melakukan mengklasifikasi sentimen yang ada pada sebuah *Tweet* secara otomatis.
2. Proses klasifikasi semakin akurat jika data latih yang digunakan dalam pembelajaran berjumlah banyak, akan tetapi dapat juga mengurangi keakuratan jika kata-kata yang terdapat pada *Tweet* tersebut mengalami bias atau bermakna ganda.
3. Seleksi fitur menggunakan *N-gram* kata dapat meningkatkan kemampuan analisis sentimen pada *Tweet*.

5.2 Saran

Penulis menyarankan pengembangan penelitian lebih lanjut sistem pengklasifikasian *Tweet* sebagai berikut:

1. Pada penelitian selanjutnya dapat menggunakan metode klasifikasi dan seleksi fitur yang lebih baik, seperti penggunaan *Part of Speech tagging* untuk mengetahui posisi sebuah kata dalam kalimat.

2. Untuk penelitian berikutnya diharapkan sistem ini tidak hanya untuk mengklasifikasi untuk sentimen terhadap Provider Telekomunikasi tetapi juga terhadap tokoh politik atau produk yang lain.
3. Bahasa yang digunakan juga tidak hanya bahasa Indonesia tetapi dapat menggunakan bahasa daerah atau bahasa asing seperti bahasa Inggris dan bahasa asing lainnya.

DAFTAR PUSTAKA

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S.M.M., Williams, H.E. 2007. Stemming Indonesian : A Confix-Stripping Approach. Transaction on Asian Langeage Information Processing. Vol. 6, No. 4, Articel 13. Association for Computing Machinery : New York .

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. 2011. Sentiment Analysis of Twitter Data. http://www.cs.columbia.edu/ ~julia/papers/Agarwaletal11.pdf. Diakses tanggal 20 Desember 2013

Agusta, L. 2009. Perbandingan Algoritma stemming Porter dengan algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia.Proceeding Konferensi Nasional Sistem dan Informatika. Yogyakarta. Hal 196-201.

Alexa. 2013. The top 500 sites on the web. http://www.alexa.com/topsites. Diakses tanggal 5 Februari 2013.

Aliandu, P. 2013. Twitter Used by Indonesian President: An Sentiment Analysis of Timeline. Dalam Information Systems International Conference (ISICO), 2 – 4 December 2013.al. 713-717. Bali: Indonesia

Alpaydin, E. 2010. Introduction to Machine Learning: Second Edition. Massachusetts London, England: The MIT Press.

Alwi, H., Dardjowidjojo, S., Lapoliwa, A.M., 2003. Tata Bahasa Baku Bahasa Indonesia: Edisi Ketiga. Pusat Bahasa Departemen Pendidikkan Nasional. Balai Pustaka : Jakarta.

Asian, J., Williams, H.E., Tahaghoghi, S.M.M. 2005. Stemming Indonesia. Proceedings of the Twenty-eighth Australasian conference on Computer Science. Vol. 38, hal. Australia : Association for Computing Machinery.

Berry, M.W. & Kogan, J. 2010. Text Mining Aplication and theory. WILEY : United Kingdom.

Dharwiyanti, S dan Wahono, S.R., 2003. Pengantar Unified Modeling Language. IlmuKomputer.com.

Dragut, E., Fang, F., Sistla, P., Yu, S. & Meng, W. 2009. Stop Word and Related Problems in Web Interface Integration. http://www.vldb.org/pvldb/2/vldb09-384.pdf. Diakses tanggal 8 Desember 2011.

Farber, Dan. 2012. *Twitter hits 400 million tweets per day, mostly mobile.* http://www.cnet.com/news/twitter-hits-400-million-tweets-per-day-mostly-mobile/. Diakses tanggal 27 Maret 2014.

Feldman, R & Sanger, J. 2007. The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press : New York.

Han, J & Kamber, M. 2006 Data Mining: Concepts and Techniques Second Edition. Morgan Kaufmann publisher : San Francisco.

Hariyanto, B., 2004. Rekayasa Sistem Berorientasi Objek. Bandung: Informatika Bandung.

Herlian,Milkha. Text Mining. http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm /6Text%20Mining.pdf. Diakses tanggal 3 Juli 2011.

http://www.php.net/.(2010). Introduction to PHP. Diakses tanggal 1-03-2014.

Ikonomakis, M., Kotsiantis, S., Tampakas, V. 2005. Text Classification Using Machine Learning Techniques. WSEAS TRANSACTIONS on COMPUTERS. Volume 4. Issue 8, 966-974.

Kim, S., Han, K., Rim, H., and Myaeng, S. 2006. Some Effective Techniques for Naive Bayes Text Classification.TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. IEEE.1(11).

Kononenko, I. 1993. Inductive and Bayesian learning in medical diagnosis. Applied Artificial Intelligence : 317-337

Kotsiantis, S.B., Zaharakis,D.I., Pintelas, P.E. 2007. Machine learning: a review of classification and combining techniques.Artificial Intelegence review. Volume 26. Number 3, 159-190. Springer : New York.

Kouloumpis, E., Wilson, T., Moore, J. 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG!. Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Menlo Park, California. pp. 538-541.

Kridalaksana, H. 2009. Pembentukkan Kata dalam Bahasa Indonesia. Gramedia Pustaka Utama : Jakarta.

Kwak, H., Lee, C., Park, H., & Moon, S. 2010. What is Twitter, a Social Network or a News Media?. Dalam Proceedings of International World Wide Web Conference Committee (IW3C2) WWW 2010: hal.591-600. Raleigh, North Carolina, USA: ACM.

Lin, S. 2008. A document classification and retrival system for R&D in semiconductor industry-A hybrid approach.Expert System 18, 2:4753-4764.

Liu, Bing. 2012. *Sentiment Analysis And Opinion Mining*. Chicago: Morgan & Claypool Publisher. http://www.dcc.ufrj.br/~valeriab/DTMSentiment-AnalysisAndOpinionMining-BingLiu.pdf. Diakses tanggal 10 Januari 2014.

Makice, K. 2009. Twitter API: Up and Running. California: O'Reilly.

Nur,Y., Santika, D. D. 2011. Analisis Sentimen Pada Dokumen Berbahasa Indonesia Dengan Pendekatan Support Vector Machine. *Konferensi Nasional Sistem dan Informatika 2011; Bali, November 12, 2011 KNS&I11-002*, pp. 9-14.

Pak, A. & Paroubek, P. 2010.Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Dalam Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10. Valletta: Malta

Pang, B., Lee, L., & Vithyanathan, S. (2002). Thumbs Up ? SentimentClassification Using Machine Learning Techniques. Dalam Proceedings of The ACL-02 conference on Empirical methods in natural language processing, pp. 79-86. Stroudsburg: Association for computationalLinguistic.

Pop, I. 2006. An approach of the Naive Bayes classifier for the document classification 1. General Matthematics 14, 4:135-138.

Prasad, S. 2011. Micro-blogging Sentiment Analysis Using Bayesian Classification Methods. http://www-nlp.stanford.edu/courses/cs224n/2010/reports/suhaasp.pdf. Diakses tanggal 20 Desember 2013.

Russell, M. A. 2011. 21 Recipes for Mining Twitter. California: O'Reilly.

Sunni, I. & Widyantoro, D. H. 2012. Analisis Sentimen dan Ekstraksi Topik PenentuSentimen pada Opini Terhadap Tokoh Publik. *Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika Volume 1, Number 2, Juli 2012*. https://www.academia.edu/2101269/-Analisis_Sentimen_dan_Ekstraksi_Topik_Penentu_Sentimen_pada_Opini_Terhadap_Tokoh_Publik. Diakses 20 Januari 2014

Tala, Fadillah Z. 2003. A Study of Stemming Efects on Information Retrieval in Bahasa Indonesia. Institute for Logic, Language and ComputationUniversiteit van Amsterdam The Netherlands. http://www.illc.uva.nl/Research/Reports/MoL-2003-02.text.pdf. Diakses tanggal 29 September 2011.

Thomas, K., Grier, C., Ma, J., Paxson, V., & Song, D. 2011. Design and Evaluation of a Real-Time URL Spam Filtering Service. Dalam Proceedings of the IEEE Symposium on Security and Privacy. California: IEEE.

Trappey, A.J.C., Hsu, F, Trappey, C.V., Lin, C. 2006. Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications.Volume 31*. Issue 4, pp. 755-765.

Twitter. 2013. https://support.twitter.com/. Diakses tanggal 10 Desember 2013.

Wang, A. H. 20100. Don't Follow Me: Twitter Spam Detection. *Proceedings of 5th International Conference on Security and Cryptography (SECRYPT) Athens 2010*: pp. 1-10. California:IEEE.

Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F.J. 2005. *Text Mining : Predictive Methods fo Analyzing Unstructered Information*. Springer : New York

Wicaksono, A. I., Nio, E., & Myaeng, S. H. Unsupervised Approach for Sentiment Analysis on Indonesian Movie Review. the 6th Conference of Indonesian Students Association in Korea (CISAK -2013). http://cisak.perpika.kr/wp-content/uploads/2013/07/2013-05.pdf . Diakses tanggal 10 Februari 2014

Witten, E. H., Frank, E., & Hall, M. A. 2011. *Data Mining Practical Machine Learning Tools and Techniques Third Edition*. Burlington, MA, USA: Elsevier Inc.

LAMPIRAN A: LISTING PROGRAM

Nama File: training.php

```
<?php

set_time_limit(0);
ini_set('memory_limit','64M');

require("config.php");
include('class_koneksi.php');
include('Enhanced_CS.php');
include('class_tokenizer.php');


require("trainer.php");

$trainer = new trainer;

$limit=$_GET[banyak];
echo "Batas $limit <br>";


$db = mysql_connect(MYSQL_HOST,MYSQL_USER,MYSQL_PASS);
mysql_select_db(MYSQL_DB,$db);
mysql_query('TRUNCATE TABLE knowledge_base;');
/*-------------------------------------------------------------------
----------------------*/
echo "<h1>Hapus Pembelajaran Sebelumnya </h1>";
/* loading previus learn */
echo "<h1>Loading Pembelajaran Sebelumnya</h1>";flush();
$query = mysql_query("select belongs,ngram,repite from
knowledge_base",$db);
$previouslearn = array();
while ( $row = mysql_fetch_array($query) )
    $previouslearn[$row['belongs']][$row['ngram']] = $row['repite'];
mysql_free_result($query);
$trainer->setPreviousLearn($previouslearn);

/* traine */
echo "<h1>Training Negatif </h1>";flush();
$query = mysql_query("select * from tweet where sentiment=0 limit
$limit",$db);
$sql=mysql_query("select comment_content as text,comment_approved as
state from wp_comments",$db);
echo "<h2>Loading Tweet</h2>";flush();
while ( $row = mysql_fetch_array($query) ){
      $a=hapusTMU($row['text']);
      ob_start();
      cariStem($a);
      $myStr2 = ob_get_contents();
      ob_end_clean();
      //echo $myStr2;
    $text =$myStr2;
    $trainer->add_example($text,$row['sentiment']);
}
mysql_free_result($query);

/* learn */
echo "<h2>Learning</h2>";flush();
$trainer->extractPatterns();
```

```
/* save what is learned */
echo "<h1>Simpan Hasil Learning</h1>";flush();
foreach ($trainer->knowledge as $tipo => $v) {
    foreach($v as $k => $y) {
        $k = addslashes($k);
        $sql = "replace knowledge_base
values('$k','$tipo','".$y['cant']."','".$y['bayesian']."')";
        mysql_query($sql,$db) or die(mysql_error($db).":".$sql);
    }
}
echo "<h1></h1>";flush();

mysql_query("create temporary table opttable as
select ngram, count(*) total, min(percent) as nmin, max(percent) as
nmax
from knowledge_base group by ngram having count(ngram) > 1",$db);

mysql_query("delete from knowledge_base where ngram in (select ngram
from opttable where (nmax-nmin) < 0.30)",$db);

/*-----------------------------------------------------------------
-----------------------------------------------------*/

echo "<h1>Hapus Pembelajaran Sebelumnya </h1>";
/* loading previus learn */
echo "<h1>Loading Pembelajaran Sebelumnya</h1>";flush();
$query = mysql_query("select belongs,ngram,repite from
knowledge_base",$db);
$previouslearn = array();
while ( $row = mysql_fetch_array($query) )
    $previouslearn[$row['belongs']][$row['ngram']] = $row['repite'];
mysql_free_result($query);
$trainer->setPreviousLearn($previouslearn);

/* traine */
echo "<h1>Training Positif </h1>";flush();
$query = mysql_query("select * from tweet where sentiment=1 limit
$limit",$db);
$sql=mysql_query("select comment_content as text,comment_approved as
state from wp_comments",$db);
echo "<h2>Loading Tweet</h2>";flush();
while ( $row = mysql_fetch_array($query) ){
      $a=hapusTMU($row['text']);
      ob_start();
      cariStem($a);
      $myStr2 = ob_get_contents();
      ob_end_clean();
      //echo $myStr2;
    $text =$myStr2;
    $trainer->add_example($text,$row['sentiment']);
}
mysql_free_result($query);

/* learn */
echo "<h2>Learning</h2>";flush();
$trainer->extractPatterns();

/* save what is learned */
echo "<h1>Simpan Hasil Learning</h1>";flush();
foreach ($trainer->knowledge as $tipo => $v) {
    foreach($v as $k => $y) {
        $k = addslashes($k);
        $sql = "replace knowledge_base
values('$k','$tipo','".$y['cant']."','".$y['bayesian']."')";
        mysql_query($sql,$db) or die(mysql_error($db).":".$sql);
    }
}
echo "<h1></h1>";flush();
```

```
mysql_query("create temporary table opttable as
select ngram, count(*) total, min(percent) as nmin, max(percent) as
nmax
from knowledge_base group by ngram having count(ngram) > 1",$db);

mysql_query("delete from knowledge_base where ngram in (select ngram
from opttable where (nmax-nmin) < 0.30)",$db);


/*-----------------------------------------------------------------
-----------------------------------------------------*/

echo "<h1>Hapus Pembelajaran Sebelumnya </h1>";
/* loading previus learn */
echo "<h1>Loading Pembelajaran Sebelumnya</h1>";flush();
$query = mysql_query("select belongs,ngram,repite from
knowledge_base",$db);
$previouslearn = array();
while ( $row = mysql_fetch_array($query) )
    $previouslearn[$row['belongs']][$row['ngram']] = $row['repite'];
mysql_free_result($query);
$trainer->setPreviousLearn($previouslearn);

/* traine */
echo "<h1>Training Netral</h1>";flush();
$query = mysql_query("select * from tweet where sentiment=2 limit
$limit",$db);
$sql=mysql_query("select comment_content as text,comment_approved as
state from wp_comments",$db);
echo "<h2>Loading Tweet</h2>";flush();
while ( $row = mysql_fetch_array($query) ){
      $a=hapusTMU($row['text']);
      ob_start();
      cariStem($a);
      $myStr2 = ob_get_contents();
      ob_end_clean();
      //echo $myStr2;
    $text =$myStr2;
    $trainer->add_example($text,$row['sentiment']);
}
mysql_free_result($query);

/* learn */
echo "<h2>Learning</h2>";flush();
$trainer->extractPatterns();

/* save what is learned */
echo "<h1>Simpan Hasil Learning</h1>";flush();
foreach ($trainer->knowledge as $tipo => $v) {
    foreach($v as $k => $y) {
        $k = addslashes($k);
        $sql = "replace knowledge_base
values('$k','$tipo','".$y['cant']."','".$y['bayesian']."')";
        mysql_query($sql,$db) or die(mysql_error($db).":".$sql);
    }
}
echo "<h1></h1>";flush();

mysql_query("create temporary table opttable as
select ngram, count(*) total, min(percent) as nmin, max(percent) as
nmax
from knowledge_base group by ngram having count(ngram) > 1",$db);

mysql_query("delete from knowledge_base where ngram in (select ngram
from opttable where (nmax-nmin) < 0.30)",$db);
```

```
?>
<?
function hapusTMU($str){
//$str = "Hello this is a test @someone #tag1 #tag2
http://bit.ly/123";
$str = preg_replace('/#([\w-]+)/i', '', $str); // #tag
$str = preg_replace('/@([\w-]+)/i', '', $str); // @mention
$str = preg_replace('/(http|https|ftp|ftps)\:\/\/[a-zA-Z0-9\-
\.]+\.[a-zA-Z]{2,3}(\/\S*)?/', '', $str);
return $str;
}
function cariStem($teksinput){

      $kon = new database;
      $kon->database();
      $pre = new Preprocessing;
      if (!empty($teksinput)){
            $teks = strtolower($teksinput);
            $tokenKarakter=array('’','–
',' ','/',',','?','.',':',';',',','!','[',']','{','}','(',')','-
',' ','+','=','<','>','\'','"','\\','@','#','$','%','^','&','*','`','
~','0','1','2','3','4','5','6','7','8','9','â€','”','“');
            $teks= str_replace($tokenKarakter,' ',$teks);
            $teks = $pre->tokenText($teks);
            $teks = $pre->removeStopword();
            $teks = $pre->text;
                  /* Use tab and newline as tokenizing characters as
well  */
            $tok = strtok($teks, " \n\t");
            $testes="";
            while ($tok !== false) {
                  $teks = Enhanced_CS(trim($tok)).' ';
                  $tok = strtok(" \n\t");
                  $testes=$testes." ".$teks;
                  $tampung[]=$teks;
            }
            echo $testes;
      }
}
?>
```

Nama File: trainer.php

```
if (defined("TRAINER_CLASS") ) return true;
define("TRAINER_CLASS",true);
require(dirname(__FILE__)."/ngram.php");


class trainer {
    var $examples;
    var $ngram;
    var $knowledge;

    function trainer() {
        $this->ngram = new ngram;
    }

    function add_example($text, $clasification) {
        $this->examples[$clasification][] = $text;
    }

    function setPreviousLearn($f) {
        $this->previous = $f;
    }

    function extractPatterns() {
        $previous = & $this->previous;
```

```
        $examples = & $this->examples;
        $ngram = & $this->ngram;
        $knowledge = & $this->knowledge;

        foreach($examples as $tipo => $texts) {
            $params[$tipo] = 0;
            $ngram->setInitialNgram( isSet($previous[$tipo]) ?
$previous[$tipo] : array() );
            foreach ($texts as $text) {
                $ngram->setText($text);
                for($i=1; $i <= 3;$i++) {
                    $ngram->setLength($i);
                    $ngram->extract();
                }
            }

            $actual = & $knowledge[$tipo];
            foreach( $ngram->getnGrams() as $k => $v) {
                $actual[$k]['cant'] = $v;
                $params[$tipo] += $v;
            }
        }
        $this->computeBayesianFiltering($params);
    }

    function computeBayesianFiltering($param) {
        $knowledge = & $this->knowledge;
        //print_r($param);
        //
        foreach($knowledge as $tipo => $caracterist) {
            foreach($caracterist as $k => $v) {
                 $t = ($v['cant']/$param[$tipo]);
                 $f = 0;
                 foreach($param as $k1 => $v1)
                     if ( $k1 != $tipo) {

                        $f += isset($knowledge[$k1][$k]['cant']) ?
$knowledge[$k1][$k]['cant'] / $v1 : 0;
                     }
                 $knowledge[$tipo][$k]['bayesian'] = $t / ($t + $f);
            }
        }
    }
}
?>
```

Nama File: cek.php

```
if (defined("SPAM_CLASS") ) return true;
define("SPAM_CLASS",true);

require(dirname(__FILE__)."/ngram.php");


class spam {
    var $_source;

    function spam($callback='') {
        if ( !is_callable($callback) ) {
            trigger_error("$callback is not a valid
funciton",E_USER_ERROR);
        }
        $this->_source = $callback;
    }
```

```
    function isItSpam($text,$type) {
        $ngram = new ngram;
        $ngram->setText($text);

        for($i=3; $i <=5;$i++) {
            $ngram->setLength($i);
            $ngram->extract();
        }

        $fnc = $this->_source;
        $ngrams =  $ngram->getnGrams();
        $knowledge =  $fnc( $ngrams,$type );
        $total=0;
        $acc=0;
        foreach($ngrams as $k => $v) {
            if ( isset($knowledge[$k]) ) {
                $acc += $knowledge[$k] * $v;
                $total++;
            }
        }
        $percent = ($acc/$total);
        $percent = $percent > 1.0 ? 1.0 : $percent;
        return $percent * 100;
    }



    function isItSpam_v2($text,$type) {
        $ngram = new ngram;
        $ngram->setText($text);

        for($i=3; $i <= 5;$i++) {
            $ngram->setLength($i);
            $ngram->extract();
        }

        $fnc = $this->_source;
        $ngrams =  $ngram->getnGrams();
        $knowledge =  $fnc( $ngrams,$type );
        $total=0;
        $acc=0;

        /**
         *  N = total ngram yg digunakan.
         *  K = product semua n-grams
         *  H = chi2Q( -2N K, 2N);
         *  S = chi2Q( -2N ( (1.0 - ngram(1)) ( 1.0 - ngram(2)) .. (
1.0 - ngram(N)) ), 2N)
         *  I = ( 1 + H - S ) / 2
         *
         */
        $N = 0;
        $H = $S = 1;

        foreach($ngrams as $k => $v) {
            if ( !isset($knowledge[$k]) ) continue;
            $N++;
            $value = $knowledge[$k] * $v;
            $H *= $value;
            $S *= (float)( 1 - ( ($value>=1) ? 0.99 : $value) );
        }

        $H = $this->chi2Q( -2 * log( $N *  $H), 2 * $N);
        $S = (float)$this->chi2Q( -2 * log( $N *  $S), 2 * $N);
        $percent = (( 1 + $H - $S ) / 2) * 100;
        return is_finite($percent) ? $percent : 100;
    }
```

```
    function chi2Q( $x,  $v) {
        $m = (double)$x / 2.0;
        $s = exp(-$m);
        $t = $s;

        for($i=1; $i < ($v/2);$i++) {
            $t *= $m/$i;
            $s += $t;
        }
        return ( $s < 1.0) ? $s : 1.0;
    }
}
?>
```

Nama File: testing.php

```
<?

echo $_GET[banyak];

include('class_koneksi.php');
include('Enhanced_CS.php');
include('class_tokenizer.php');
?>
<?

set_time_limit(0);
ini_set('memory_limit','64M');
require("spam.php");
require("config.php");

$db = mysql_connect(MYSQL_HOST,MYSQL_USER,MYSQL_PASS);
mysql_select_db(MYSQL_DB,$db);

$spam = new spam("handler");
/**/
?><table border=1> <?
$no=1;
$aaaa=0;
$bbbb=0;
$limit=$_GET[banyak];
$query = "SELECT * FROM tweet_test LIMIT $limit";
$hasil = mysql_query($query);
while ($data = mysql_fetch_array($hasil))
{
   echo "<tr>";
   echo "<td>".$no."</td>";
   echo "<td>".$data['text']."</td>";
   //$text=$data['text'];
   $a=hapusTMU($data['text']);
      ob_start();
      cariStem($a);
      $myStr2 = ob_get_contents();
      ob_end_clean();
      //echo $myStr2;
    $text =$myStr2;
   $sentx=$data['sentiment'];
   if($sentx==0)
      $senty="Negatif";
      elseif($sentx==1)
      $senty="Positif";
      else
      $senty="Netral";

   echo "<td>".$senty."</td>";

   $hit=$data['sentiment'];
```

```
   ?><td><?
   if( $spam->isItSpam_v2($text,'0')< $spam->isItSpam_v2($text,'1')
&& $spam->isItSpam_v2($text,'1')< $spam->isItSpam_v2($text,'2'))
            {
                  echo "<font color=blue><b>Netral :| </font></b>";
                  $anet++;
                  if ($hit==2){
                                    $cccc+=1;
                              }
                  }
            elseif( $spam->isItSpam_v2($text,'0')< $spam-
>isItSpam_v2($text,'1') && $spam->isItSpam_v2($text,'1')> $spam-
>isItSpam_v2($text,'2'))
            {
                  echo "<b>Positif :)</b>";
                  $apos++;
                  if ($hit==1){
                                    $bbbb+=1;
                              }
                  }
            elseif( $spam->isItSpam_v2($text,'0')> $spam-
>isItSpam_v2($text,'1') && $spam->isItSpam_v2($text,'1')< $spam-
>isItSpam_v2($text,'2'))
            {
                  if($spam->isItSpam_v2($text,'0')> $spam-
>isItSpam_v2($text,'2'))
                        {
                              echo "<font color=red><b>Negatif
:(</b></font>";
                              $aneg++;
                              if ($hit==0){
                                    $aaaa+=1;
                              }
                        }
                        else{
                              echo "<font color=blue><b>Netral :|
</font></b>";
                              $anet++;
                              if ($hit==2){
                                    $cccc+=1;
                              }
                        }
                  }
            elseif( $spam->isItSpam_v2($text,'0')> $spam-
>isItSpam_v2($text,'1') && $spam->isItSpam_v2($text,'1')> $spam-
>isItSpam_v2($text,'2'))
            {
                  echo "<font color=red><b>Negatif :(</font></b>";
                  $aneg++;;
                  if ($hit==0){
                                    $aaaa+=1;
                              }
                  }
            else {
                  echo "<b>Positif :)</b>";
                  $apos++;
                  if ($hit==1){
                                    $bbbb+=1;
                              }
                  }
            ?><td><?
  $no++;
}
?>
<tr><td colspan=4>
Total Tweet : <?=$limit;?><br>
Total Negatif: <?=$aneg;?><br>
```

```
Total Positif : <?=$apos;?><br>
Total Netral: <?=$anet;?><br>

<br>Total Akurasi <?=(($aaaa+$bbbb+$cccc)/$limit)*100;?>%
</table>
<?

function handler($ngrams,$type) {
    global $db;

    $info = array_keys($ngrams);

    $sql = "select ngram,percent from knowledge_base where belongs =
'$type' && ngram in ('".implode("','",$info)."')";
    $r = mysql_query($sql,$db);

    while ( $row = mysql_fetch_array($r) ) {
        $t[ $row['ngram'] ]  = $row['percent'];
    }

    return $t;
}

?>

<?
function hapusTMU($str){

$str = preg_replace('/#([\w-]+)/i', '', $str); // #tag
$str = preg_replace('/@([\w-]+)/i', '', $str); // @mention
$str = preg_replace('/(http|https|ftp|ftps)\:\/\/[a-zA-Z0-9\-
\.]+\.[a-zA-Z]{2,3}(\/\S*)?/', '', $str);
return $str;
}
function cariStem($teksinput){

      $kon = new database;
      $kon->database();
      $pre = new Preprocessing;
      if (!empty($teksinput)){
            $teks = strtolower($teksinput);
            $tokenKarakter=array('’','—
',' ','/',',',',','?','.',':',';',' ',',','!','[',']','{','}','(',')','-
',' ','+','=','<','>','\'','"','\\','@','#','$','%','^','&','*','`','
~','0','1','2','3','4','5','6','7','8','9','â€','”','“');
            $teks= str_replace($tokenKarakter,' ',$teks);
            $teks = $pre->tokenText($teks);
            $teks = $pre->removeStopword();
            $teks = $pre->text;

            $tok = strtok($teks, " \n\t");
            $testes="";
            while ($tok !== false) {
                  $teks = Enhanced_CS(trim($tok)).' ';
                  $tok = strtok(" \n\t");
                  $testes=$testes." ".$teks;
                  $tampung[]=$teks;
            }
            echo $testes;
      }
}
?>
```