BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Pada bab ini akan membahas tentang analisis dan perancangan sistem analisis sentimen dengan menggunakan algoritma *naïve bayes classifier*. Langkah-langkah yang dilakukan dalam analisis sistem ini yaitu:

1. Analisis masalah.
2. Analisis sumber data.
3. Analisis *preprocessing.*
4. Analisis penerapan algoritma *naïve bayes classifier.*
5. Analisis pembobotan kata (TF-IDF).

III.1.1 Analisis Masalah

Twitter adalah salah satu *microblogging* yang sangat popular di tengah masyarakat. Hal ini terlihat dari banyaknya jumlah pengguna Twitter yang mencapai 231,7 juta [5]. Biasanya Twitter digunakan sebagai sarana untuk menyampaikan pendapat. Informasi yang terkandung dalam *tweets* ini sangat berharga sebagai alat penentu kebijakan. Salah satunya adalah untuk menilai suatu produk yang dikeluarkan oleh perusahaan atau yang sering disebut dengan analisis sentimen.

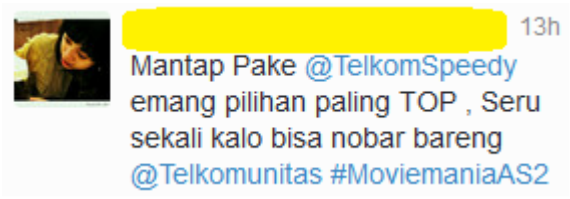
Perusahaan dapat memanfaatkan salah satu *microblogging* ini sebagai layanan *feedback* untuk pengguna produknya. Salah satu perusahaan yang menyediakan layanan *feedback* adalah Telkom Speedy. *Feedback* dari pengguna biasanya berupa opini-opini selama menggunakan produk dari perusahaan tersebut. Hal ini sangat berguna bagi perusahaan dalam meninjau kembali produknya. Opini dari pengguna itu nantinya akan dilakukan analisis sentimen apakah termasuk opini positif atau opini negatif. Namun, analisis sentimen ini mendapat tantangan berupa model bahasa tidak formal yang digunakan di Twitter. Maka dari itu, sebelum melakukan analisis sentimen, terlebih dahulu

harus dilakukan *preprocessing* pada data *tweets* yang akan digunakan. Hal ini berguna untuk mengatasi model bahasa yang tidak formal yang sering digunakan pada Twitter.

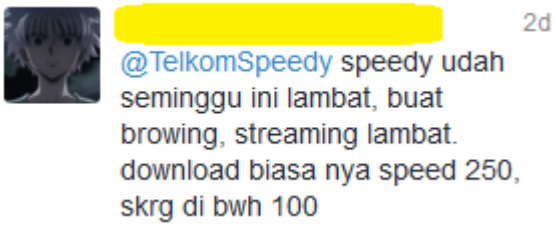
Selain itu, pengklasifikasian sentimen saat ini dilakukan secara manual oleh manusia. Permasalahan ini berdampak pada kualitas dan kecepatan dalam menganalisis sentimen yang sangat banyak. Maka dari itu, penggunaan aplikasi yang dapat melakukan analisis sentimen secara otomatis merupakan salah satu solusi untuk mengatasi masalah ini.

III.1.2 Analisis Sumber Data

Data yang digunakan pada penelitian ini diambil dari kumpulan *tweets* bahasa Indonesia yang diambil dari *official* akun Twitter Telkom Speedy. Data *tweets* ini diperoleh dengan membuat program *crawling* yang menggunakan Tweetinvi API. Dalam proses *crawling*, secara otomatis akan mengambil data *tweets* yang mengandung kata "TelkomSpeedy", "speedy reguler", "speedy instant", dan "speedy gold". Data *tweets* yang terkumpul nantinya akan melewati tahap *preprocessing* dan selanjutnya akan diklasifikasikan. Dalam sistem analisis sentimen ini, *tweets* akan diklasifikasikan ke dalam dua kelas (kategori), yaitu kelas sentimen positif dan kelas sentimen negatif. Contoh dari *tweets* yang termasuk sentimen positif dapat dilihat pada Gambar III.1, sedangkan *tweets* yang termasuk sentimen negatif dapat dilihat pada Gambar III.2.



Gambar III.1 Contoh Sentimen Positif



Gambar III.2 Contoh Sentimen Negatif

Data yang dibutuhkan dalam penelitian ini terdiri dari dua jenis, yaitu data latih dan data uji. Data latih yang digunakan ini diambil dari kumpulan *tweets* yang telah dilabeli dengan kelas sentimennya secara manual. Data inilah yang digunakan sebagai data latih untuk membentuk model analisis sentimen. Model ini nantinya akan digunakan untuk mengklasifikasikan *tweets* pada kelas sentimennya. Pada penelitian ini, metode klasifikasi yang digunakan adalah *naïve bayes classifier*. Sebagian dari hasil *crawling*, nantinya akan digunakan sebagai data uji. Data uji ini menggunakan kumpulan *tweets* yang belum memiliki label.

Setiap orang memiliki ciri khas dalam penulisan *tweets*. Dari hasil observasi yang dilakukan pada *official* akun Twitter Telkom Speedy, terdapat beberapa karakteristik dalam penulisan *tweets*, seperti:

1. Penulisan kata yang disingkat.

   Keterbatasan karakter yang dapat ditulis dalam suatu *tweets* (maksimal 140 karakter) membuat banyak kata dalam *tweets* disingkat. Contohnya dapat dilihat pada Gambar III.3.



Gambar III.3 Contoh *Tweets*

2. Penggunaan titik (.) atau koma (,) pada akhiran *tweets*.

   Ada beberapa orang yang biasa menggunakan dua sampai empat titik (.) atau koma (,) pada akhiran *tweets*. Contohnya seperti pada Gambar III.4.



Gambar III.4 Contoh *Tweets*

3. Penggunaan *emoticon.*

   Ada beberapa orang yang menyatakan sentimennya dalam *tweets* dengan menggunakan *emoticon*. Contohnya seperti berikut:



Gambar III.5 Contoh *Tweets*

III.1.3 Analisis Preprocessing

Pemrosesan teks merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk lebih efektif dalam proses pemrosesan dilakukan langkah transformasi data ke dalam suatu format yang memudahkan untuk kebutuhan pemakai. *Preprocessing* merupakan salah satu langkah yang penting dalam analisis sentimen. Sama halnya *preprocessing* pada *Information Retrieval (IR),* tahapannya terdiri dari *tokenizing,* normalisasi fitur, *case folding*, *stopword removal* dan *stemming*. Namun pada *preprocessing* analisis sentimen, ada tambahan tahapan seperti *convert emoticon* dan *convert negation*. Tahapan dari *preprocessing* adalah sebagai berikut:

1. *Case Folding*

   Pada tahap ini, semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Berikut merupakan langkah-langkah *case folding* dalam contoh salah satu contoh *tweets*: “Puas dgn layanan primanya @TelkomSpeedy, akhirnya bisa

konek @wifi_id lg di rumah. Makasih atas bantuan maintenancenya yang cepat :)".

1. Memeriksa ukuran setiap karakter dari awal sampai akhir karakter.
2. Jika ditemukan karakter yang menggunakan huruf kapital (*uppercase*), maka huruf tersebut akan diubah menjadi huruf kecil (*lowercase*).

Gambaran tahap *case folding* dapat dilihat pada Gambar III.6.



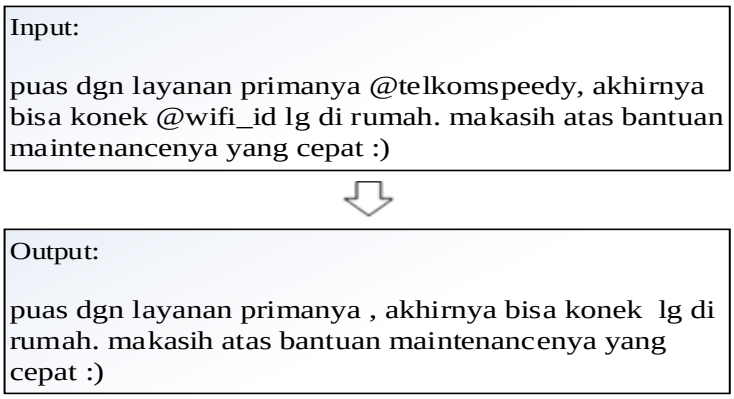
Gambar III.6 Contoh *Case Folding*

2. Normalisasi Fitur

Ada beberapa komponen khas yang biasa ada di *tweet* yakni, *username*, URL (*Uniform Resource Locator*), dan "RT" (tanda *retweet*). Karena username, URL, dan "RT" tidak memiliki pengaruh apapun terhadap nilai sentimen, maka ketiga komponen di atas akan dibuang. Komponen *username* diidentifikasi dengan kemunculan karakter '@'. Selain *username*, karakter '@' biasa juga digunakan untuk pemanggilan suatu tempat seperti @FloatingMarket. Namun nama tempat tersebut tidak memiliki pengaruh pada analisis sentimen sehingga nama tempat pun harus dihapus. Pada komponen URL dikenali melalui ekspresi regular (seperti http, www). Berikut langkah-langkah pada tahap normalisasi fitur:

1. Kata yang digunakan hasil dari *case folding*.
2. Hasil dari *case folding* akan diperiksa apakah terdapat *username*, URL, dan RT.
3. Jika terdapat *username*, URL, dan RT maka akan dihilangkan.

Gambaran tahap normalisasi fitur dapat dilihat pada Gambar III.7.



Gambar III.7 Contoh Normalisasi Fitur

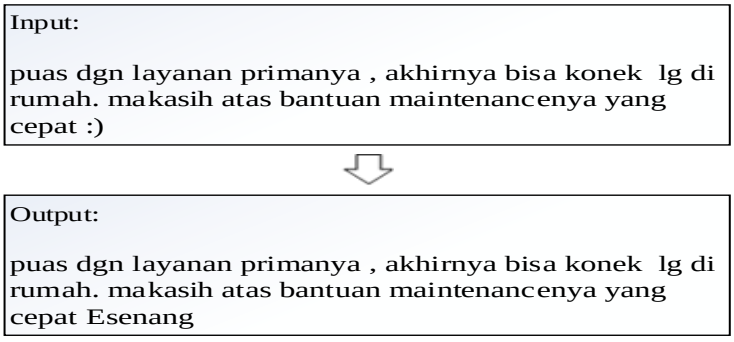
3. *Convert E**m**oticon*

Pada tahap ini, kumpulan *tweets* yang terdapat *emoticon* (*emotion icon*) akan dikonversikan ke dalam *string* yang bersesuaian. Namun, tidak semua akan diimplementasikan, karena tidak semua *emoticon* sering digunakan oleh pengguna Twitter. *Emoticon* yang digunakan berdasarkan *Western Style*, dapat dilihat pada Tabel II.1.

Berikut merupakan langkah-langkah dalam tahap *convert emoticon*:

1. Kata yang digunakan berasal dari hasil normalisasi fitur.
2. Membandingkan setiap karakter dengan tabel *list emoticon* pada
3. Jika terdapat *emoticon*, maka *emoticon* tersebut akan diubah ke dalam bentuk *string*.

Gambaran tahap *convert emoticon* dapat dilihat pada Gambar III.8.



Gambar III.8 Contoh *Convert Emoticon*

4. *Convert Negation*

Pada tahap ini, setiap *tweets* yang mengandung kata-kata yang bersifat negasi akan diubah nilai sentimennya. Kata-kata yang bersifat negasi seperti

“bukan”, “tidak”, “enggak”, “ga”, “jangan”, “nggak”, “tak”, dan “gak”. *Convert negation* dilakukan jika terdapat kata negasi sebelum kata yang bernilai positif, maka kata tersebut akan diubah nilainya menjadi negatif dan begitupun sebaliknya. Langkah-langkah pada tahap *convert negation* adalah sebagai berikut:

1. Kata yang digunakan adalah hasil dari *convert emoticon*.
2. Setiap kata akan diperiksa dan dibandingkan dengan kumpulan kata-kata yang bersifat negasi pada *database*.
3. Jika setelah kata yang bersifat negasi terdapat kata yang termasuk sentimen positif, maka sentimen tersebut akan diubah menjadi negatif.
4. Jika setelah kata yang bersifat negasi terdapat kata yang termasuk sentimen negatif, maka sentimen tersebut akan diubah menjadi positif.



Gambar III.9 Contoh *Tweets* yang mengandung kata negasi

Dalam contoh Gambar III.9, terdapat kata “menyesal” yang merupakan sentimen negatif. Namun, didepan kata “menyesal” terdapat kata yang bersifat negasi yaitu “tidak”, sehingga sentimennya menjadi positif.

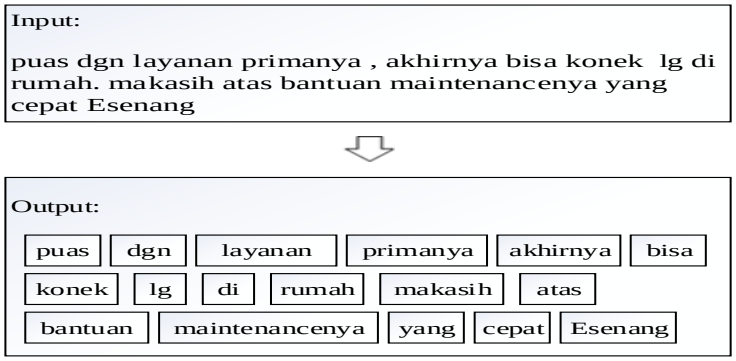
5. *Tokenizing*

Pada tahap ini akan dilakukan pengecekan *tweets* dari karakter pertama sampai karakter terakhir. Apabila karakter ke-i bukan tanda pemisah kata seperti titik(.), koma(,), spasi dan tanda pemisah lainnya, maka akan digabungkan dengan karakter selanjutnya. Langkah-langkah pada tahap *tokenizing* adalah sebagai berikut:

1. Kata yang digunakan adalah hasil dari *convert negation*.
2. Memotong setiap kata dalam teks berdasarkan pemisah kata seperti titik(.), koma(,), dan spasi.
3. Bagian yang termasuk dalam daftar *emoticon* tidak dibuang.

4. Bagian yang hanya memiliki satu karakter *non* alfabet dan angka akan dibuang.

Gambaran tahap *tokenizing* dapat dilihat pada Gambar III.10.



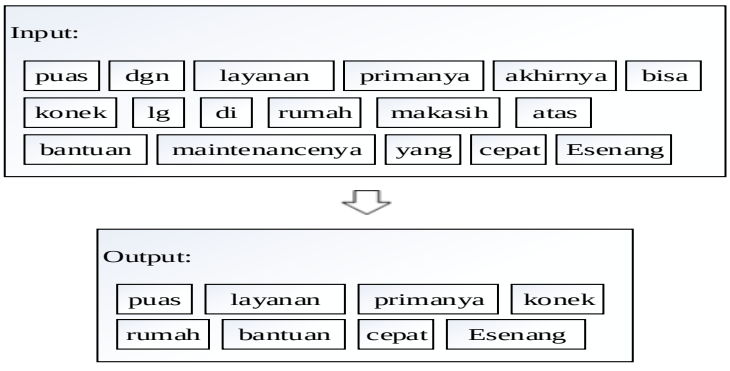
Gambar III.10 Contoh *Tokenizing*

6. *Stopword Removal*

Pada tahap ini, kumpulan *tweets* yang telah melewati tahap *tokenzing* akan melalui tahap *stopword removal*. Setiap kata pada *tweets* akan diperiksa. Jika terdapat kata sambung, kata depan, kata ganti atau kata yang tidak ada hubungannya dalam analisis sentimen, maka kata tersebut akan dihilangkan. Langkah-langkah pada *stopword removal* adalah sebagai berikut:

1. Kata hasil *tokenizing* akan dibandingkan dengan daftar *stopword*.
2. Dilakukan pengecekan apakah kata sama dengan daftar *stopword* atau tidak.
3. Jika kata sama dengan yang ada pada daftar *stopword*, maka akan dihilangkan.

Contoh dari tahap *stopword removal* dapat dilihat pada Gambar III.11.

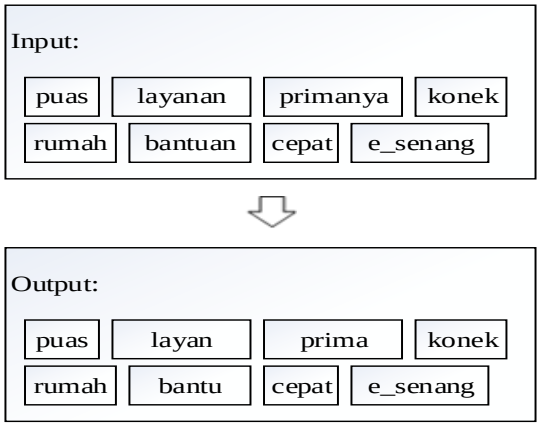


Gambar III.11 Contoh *Stopword Removal*

7. *Stemming*

Kata-kata yang muncul di dalam *tweets* sering mempunyai banyak varian morfologik. Oleh karena itu, setiap kata-kata direduksi ke bentuk *stemmed word* (*term*) yang cocok. Kata-kata tersebut akan diambil bentuk kata dasarnya dengan cara menghilangkan awalan atau akhiran. Langkah-langkah pada tahap *stemming* adalah sebagai berikut:

1. Kata yang digunakan adalah dari hasil *stopword removal*.
2. Setiap kata dalam *tweets* akan diperiksa dari awal hingga akhir kata.
3. Jika terdapat kata yang mengandung imbuhan, maka imbuhan pada kata tersebut akan dihilangkan.



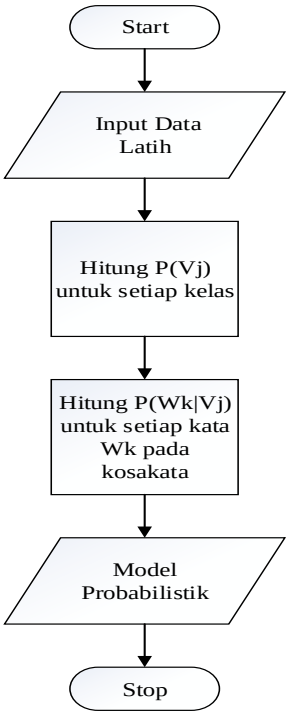
Gambar III.12 Contoh *Stemming*

III.1.4 Analisis Penerapan Algoritma Naive Bayes Classifier

Pada tahap ini, metode yang digunakan dalam pengklasifikasian sentimen adalah *naïve bayes classifier* (NBC). Metode ini terdiri dari dua proses, yaitu sebagai berikut:

1. Proses Pelatihan *Naive Bayes Classifier*

Secara umum proses ini dibagi menjadi beberapa tahap. Tahap-tahap tersebut dapat dilihat pada Gambar III.13 berikut:



Gambar III.13 *Flowchart* Proses Pelatihan *Naive Bayes Classifier*

Untuk proses pembelajaran *naïve bayes classifier*, maka sebelumnya harus diperhatikan hal-hal berikut:

a. Kosakata

   |kosakata| adalah jumlah kata unik pada semua data latih. Data latih disini berarti kumpulan *tweets* yang sudah diklasifikasikan. Pada analisis sentimen, kata dibagi menjadi dua kelas (kategori) yaitu:

   1. Data 1 (D1) = kelas sentimen positif.
   2. Data 2 (D2) = kelas sentimen negatif.

   Contoh himpunan data latih terdapat pada Tabel III.1.

Tabel III.1 Contoh Himpunan Data Latih

Data	*Keyword* (kemunculan)	Kelas Sentimen
D1	Bagus (2), cepat (3), ramah (1)	Positif
D2	Kecewa (5), Lambat (2), rugi (3)	Negatif

   |kosakata| yang dihasilkan dari data latih berjumlah 6 kata.

b. Pada setiap kelas, didapatkan nilai probabilitas setiap kosakata terhadap sekumpulan *tweets* melalui persamaan II.4.

Tabel III.2 Nilai P(Vj) untuk setiap kelas

Data	*Keyword* (kemunculan)	Kelas Sentimen (V)	$P(V_j)$
D1	Bagus (2), cepat (3), ramah (1)	Positif	$\frac{1}{2} = 0.5$
D2	Kecewa (5), Lambat (2), rugi (3)	Negatif	$\frac{1}{2} = 0.5$

Untuk setiap kata w_k pada kelas V_j diterapkan perhitungan berdasarkan persamaan II.5. Sebagai contoh untuk menampilkan perhitungannya, akan diambil satu kata pada masing-masing kelas, yaitu perhitungan terhadap kata "lambat".

Tabel III.3 Nilai P(lambat) untuk setiap kelas

V_j	Sentimen Positif		Sentimen Negatif	
	n_k	n	n_k	n
	0	6	2	10
$P(w_k \mid V_j)$	$\frac{1}{12}$		$\frac{3}{16}$	

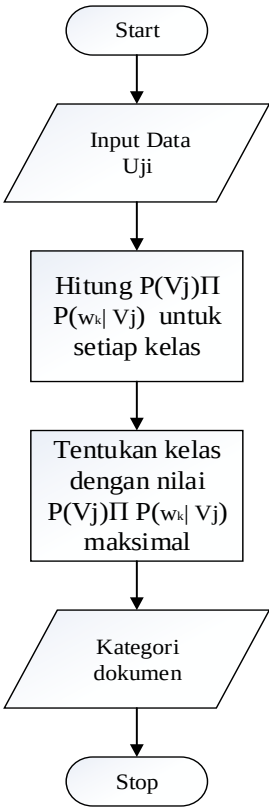
Hal yang sama diterapkan pada setiap kata w_k sehingga diperoleh nilai $P(w_k)$ untuk setiap kelas V_j dan didapatkan model probabilistik seperti pada Tabel III.4.

Tabel III.4 Model Probabilistik

V	$P(V_j)$	$P(w_k \mid V_j)$					
		bagus	cepat	ramah	kecewa	lambat	rugi
Positif	$\frac{1}{2}$	$\frac{3}{12}$	$\frac{4}{12}$	$\frac{2}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$
Negatif	$\frac{1}{2}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{6}{16}$	$\frac{3}{16}$	$\frac{4}{16}$

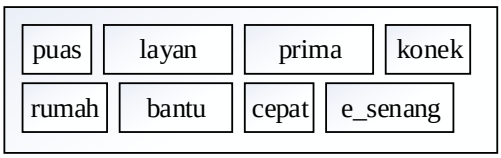
2. Proses Klasifikasi *Naïve Bayes Classifier*

Pada proses ini, data uji akan melewati proses klasifikasi berdasarkan data latih. Flowchart untuk tahap klasifikasi dapat dilihat pada Gambar III.14.



Gambar III.14 *Flowchart* Proses Klasifikasi *Naive Bayes Classifier*

Data uji adalah data *tweets* yang belum diklasifikasikan. Data uji ini adalah hasil dari tahap *preprocessing*.



Gambar III.10 Hasil *Preprocessing*

Contoh:

Data uji (D4): puas (1), layan (1), prima (1), konek (1), rumah (1), bantu (1), cepat (1), Esenang (1).

Pada tahap klasifikasi, dimulai dengan menghitung nilai Vmap untuk tiap kelas dengan persamaan II.3.

Berdasarkan acuan dari hasil pelatihan, berikut adalah hasil perhitungannya.

$$Vmap = \underset{Vj\{Positif, "Negatif"\}}{argmax} P(Vj)\ \Pi\ P(w_k \mid Vj)$$

$$Vmap = \underset{Vj\{Positif, "Negatif"\}}{argmax} P(Vj)\ P("puas"|Vj)P("layan"|Vj)P("prima"|Vj)$$

$$P("konek"|Vj)P("rumah"|Vj)P("bantu"|Vj)P("cepat"|Vj)P("Esenang"|Vj)$$

- Nilai Vmap untuk Sentimen Positif

  Vmap(“Positif) = P(“Positif”) P(“puas”|“Positif”) P(“layan”|“Positif”)
  P(“prima”|“Positif”) P(“konek”|“Positif”)
  P(“rumah”|“Positif”) P(“bantu”|“Positif”)
  P(“cepat”|“Positif”) P(“Esenang”|“Positif”)

  $= \frac{1}{2} \text{ x } \frac{1}{12} \text{ x } \frac{1}{12} \text{ x } \frac{1}{12} \text{ x } \frac{1}{12} \text{ x } \frac{1}{12} \text{ x } \frac{1}{12} \text{ x } \frac{4}{12} \text{ x } \frac{1}{12}$

  = 0.000000002

- Nilai Vmap untuk Sentimen Negatif

  Vmap(“Negatif) = P(“Negatif”) P(“puas”|“ Negatif”) P(“layan”|“ Negatif”)
  P(“prima”|“ Negatif”) P(“konek”|“ Negatif”)
  P(“rumah”|“ Negatif”) P(“bantu”|“ Negatif”)
  P(“cepat”|“ Negatif”) P(“Esenang”|“ Negatif”)

  $= \frac{1}{2} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16} \text{ x } \frac{1}{16}$

  = 0.0000000001

Kelas suatu *tweets* ditentukan berdasarkan nilai Vmap terbesar. Pada perhitungan diatas, didapat bahwa nilai Vmap untuk kelas sentimen positif memiliki nilai tertinggi dibandingkan dengan kelas sentimen negatif sehingga *tweets* tersebut termasuk kelas sentimen positif. Jika nilai Vmap untuk sentimen positif dan sentimen negatif sama, maka akan dianggap sebagai sentimen negatif karena dengan menganggap sentimen negatif, setidaknya perusahaan akan meninjau kembali kekurangan produknya.

III.1.5 Analisis Pembobotan Kata (TF-IDF)

Dalam analisis sentimen, pembobotan kata digunakan untuk mendapatkan suatu topik atau *keyword* dari kumpulan sentimen. Salah satu metode pembobotan adalah TF-IDF (*Term Frequency – Inverse Document Frequency*).

Nilai bobot suatu kata (*term*) menyatakan kepentingan bobot tersebut dalam merepresentasikan *tweets*. Pada pembobotan TF-IDF, bobot akan semakin besar jika frekuensi kemunculan kata semakin tinggi, tetapi bobot akan berkurang jika kata tersebut semakin sering muncul pada *tweets* lainnya

Contoh: Terdapat 3 *tweets* (sudah melewati *preprocessing*) seperti berikut:
D1 : puas layan prima Esenang internet bantu cepat Esenang
D2 : internet lambat kecewa Esedih
D3 : internet lancar video stabil salut
Dari persamaan II.1, diketahui:

$$\text{Idf} = \log\left(\frac{N}{df}\right) \qquad \text{(III.5)}$$

N = 3 (jumlah *tweets*)
df = Banyaknya *tweets* dimana suatu kata (*term*) muncul

Tabel III.5 Tabel Pembobotan Kata

Kata	tf D1	tf D2	tf D3	df	$\frac{N}{df}$	Idf	W D1	W D2	W D3
puas	1	0	0	1	$\frac{3}{1} = 3$	0.48	0.48	0	0
layan	1	0	0	1	$\frac{3}{1} = 3$	0.48	0.48	0	0
prima	1	0	0	1	$\frac{3}{1} = 3$	0.48	0.48	0	0
internet	1	1	1	3	$\frac{3}{3} = 1$	0	0	0	0
bantu	1	0	0	1	$\frac{3}{1} = 3$	0.48	0.48	0	0
cepat	1	0	0	1	$\frac{3}{1} = 3$	0.48	0.48	0	0
Esenang	2	0	0	2	$\frac{3}{2} = 1.5$	0.18	0.36	0	0
lambat	0	1	0	1	$\frac{3}{1} = 3$	0.48	0	0.48	0
kecewa	0	1	0	1	$\frac{3}{1} = 3$	0.48	0	0.48	0
Esedih	0	1	0	1	$\frac{3}{1} = 3$	0.48	0	0.48	0
lancar	0	0	1	1	$\frac{3}{1} = 3$	0.48	0	0	0.48
video	0	0	1	1	$\frac{3}{1} = 3$	0.48	0	0	0.48
stabil	0	0	1	1	$\frac{3}{1} = 3$	0.48	0	0	0.48
salut	0	0	1	1	$\frac{3}{1} = 3$	0.48	0	0	0.48

Keterangan:

tf D1 = Banyaknya muncul kata yang muncul di *tweets* 1 (D1)

tf D2 = Banyaknya muncul kata yang muncul di *tweets* 2 (D2)

tf D3 = Banyaknya muncul kata yang muncul di *tweets* 3 (D3)

W D1 = Bobot kata di *tweets* 1

W D2 = Bobot kata di *tweets* 2

W D3 = Bobot kata di *tweets* 3

Berdasarkan pada Tabel III.5, dapat disimpulkan bahwa kata yang merepresentasikan ketiga *tweets* diatas adalah puas, layan, prima, bantu, cepat, lambat, kecewa, e_sedih, lancar, video, stabil, dan salut.

III.2 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dilakukan berdasarkan kebutuhan perangkat lunak analisis sentimen berdasarkan hasil observasi. Spesifikasi kebutuhan perangkat lunak akan dibagi kedalam dua bagian yaitu SKPL-F (Spesifikasi Kebutuhan Perangkat Lunak Fungsional) dan SKPL-NF (Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional). Berikut ini adalah tabel spesifikasi kebutuhan perangkat lunak fungsional dan non fungsional pada tabel Tabel III.6 dan Tabel III.7.

Tabel III.6 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode	Kebutuhan
SKPL-F001	Perangkat lunak dapat melakukan *crawling tweets* pada akun Twitter @TelkomSpeedy
SKPL-F002	Perangkat lunak dapat melakukan *request tweets* pada Tweetinvi API
SKPL-F003	Perangkat lunak dapat melakukan *case folding* pada *tweets.*
SKPL-F004	Perangkat lunak dapat melakukan normalisasi fitur pada *tweets.*
SKPL-F005	Perangkat lunak dapat melakukan *convert emoticon* pada *tweets.*
SKPL-F006	Perangkat lunak dapat melakukan *convert negation* pada *tweets.*
SKPL-F007	Perangkat lunak dapat melakukan *tokenizing* pada *tweets.*
SKPL-F008	Perangkat lunak dapat melakukan *stopword removal* pada *tweets.*
SKPL-F009	Perangkat lunak dapat melakukan *stemming* pada *tweets.*
SKPL-F010	Perangkat lunak dapat melakukan klasifikasi *tweets* berdasarkan sentimennya.
SKPL-F011	Perangkat lunak dapat menggambarkan persentasi dari hasil klasifikasi *tweets* dalam bentuk diagram *pie*.
SKPL-F012	Perangkat lunak dapat menentukan kata-kata yang menjadi pemicu sentimen (ekstraksi *keyword*).

Tabel III.7 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional

Kode	Kebutuhan
SKPL-NF001	Pengguna yang menggunakan perangkat lunak ini adalah orang yang memiliki jabatan *manager operation*.
SKPL-NF002	Terhubung dengan akses internet.
SKPL-NF003	Perangkat keras yang digunakan adalah komputer dengan spesifikasi minimal *processor* 2.0 GHz, RAM 1 GB, *hard disk* 40 GB, monitor, *keyboard* dan *mouse*.
SKPL-NF004	Bahasa pemrograman yang digunakan adalah C#.

III.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah langkah dalam menganalisis sumber daya yang akan digunakan, yang direkomendasikan oleh pembangun perangkat lunak (*software developer*) kepada pengguna agar perangkat lunak yang dibangun menjadi *user friendly* dan perangkat keras yang mendukung secara maksimal terhadap kinerja perangkat lunak. Perangkat keras dan perangkat lunak yang digunakan harus sesuai dengan kebutuhan, sehingga sistem yang dibangun akan berjalan dengan baik.

Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu:

1. Analisis kebutuhan perangkat keras.
2. Analisis kebutuhan perangkat lunak.
3. Analisis kebutuhan perangkat pikir.

III.3.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras adalah seluruh komponen atau unsur peralatan yang digunakan untuk menunjang pembangunan aplikasi. Spesifikasi perangkat keras yang digunakan di PT. Telekomunikasi Indonesia, Tbk adalah sebagai berikut:

1. Processor 3.0 GHz
2. VGA 512 MB
3. Hard disk 500 GB
4. RAM 2 GB
5. Monitor

6. Keyboard
7. Mouse

Spesifikasi perangkat keras minimum untuk aplikasi yang akan dibangun pada unit personal komputer agar dapat menjalankan aplikasi secara optimal adalah sebagai berikut:

1. Processor 1.5 GHz
2. VGA On-Board
3. Hard disk 40 GB
4. RAM 1 GB
5. Monitor
6. Keyboard
7. Mouse

Berdasarkan analisis spesifikasi perangkat keras yang ada di PT. Telekomunikasi Indonesia, Tbk, spesifikasi perangkat keras yang digunakan sudah memenuhi syarat untuk menerapkan sistem yang akan dibangun.

III.3.2 Analisis Kebutuhan Perangkat Lunak

Sistem operasi yang digunakan saat ini di PT. Telekomunikasi Indonesia, Tbk menggunakan Windows 8, sedangkan perangkat lunak yang dibutuhkan untuk membangun dan menerapkan aplikasi ini adalah sebagai berikut:

1. Sistem operasi Windows 7.
2. XAMPP
3. Visual Studio 2012

Berdasarkan analisis spesifikasi perangkat lunak yang ada di PT. Telekomunikasi Indonesia, Tbk, spesifikasi perangkat lunak yang digunakan sudah memenuhi syarat untuk menerapkan sistem yang akan dibangun. Dibutuhkan pengadaan perangkat lunak XAMPP dan Visual Studio 2012.

III.3.3 Analisis Kebutuhan Perangkat Pikir

Sistem yang akan dibangun ini nantinya akan digunakan oleh *manager operation* PT. Telekomunikasi Indonesia, Tbk. Adapun karakteristik dari pengguna adalah sebagai berikut:

1. Mampu menggunakan komputer.
2. Mampu mengoperasikan internet.
3. Mampu memahami maksud dari suatu diagram.

Berdasarkan hasil dari analisis pengguna, dapat diambil kesimpulan bahwa pengguna yang ada sudah memenuhi syarat sebagai pengguna sistem ini.

III.4 Analisis Kebutuhan Fungsional

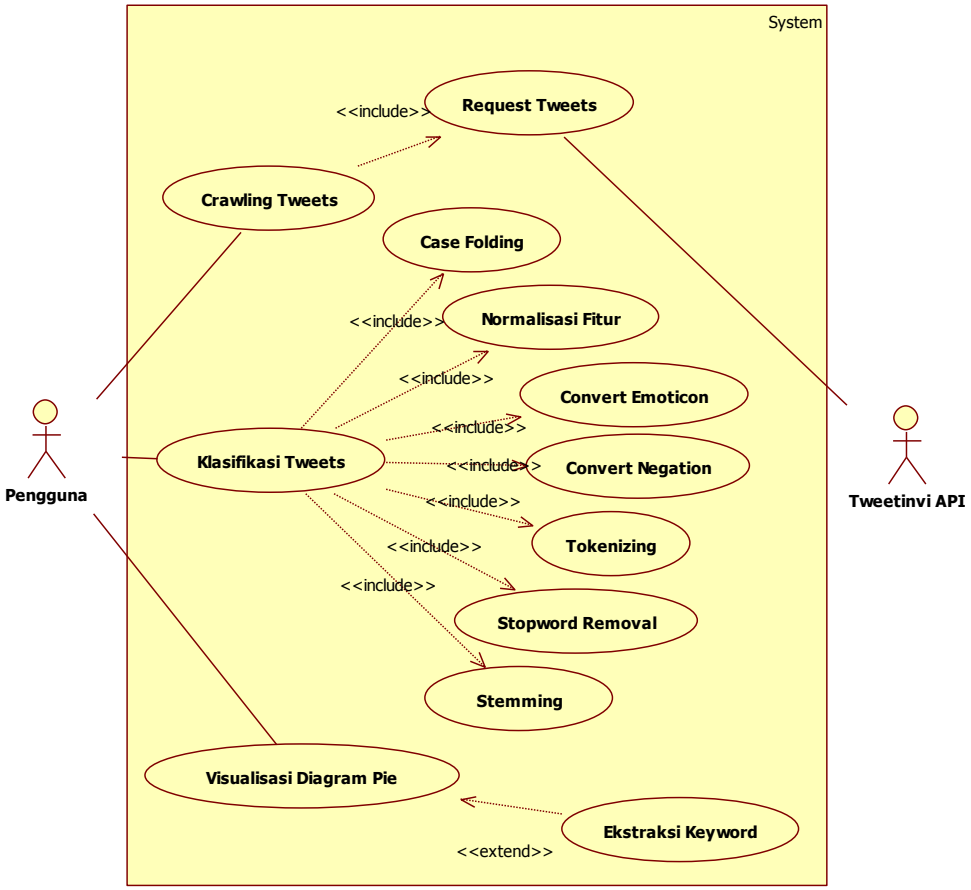
Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar berjalan dengan baik. Kebutuhan fungsional dalam pembangunan aplikasi ini dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis ini antara lain melakukan identifikasi aktor, pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

III.4.1 Identifikasi Aktor

Identifikasi aktor dimaksudkan untuk mengetahui siapa saja aktor yang terlibat di dalam sistem ini. Aktor pada sistem ini adalah pengguna (*user*) yang yang memiliki jabatan sebagai *Manager Operation*.

III.4.2 Use Case Diagram

Use case diagram memperlihatkan hubungan-hubungan yang terjadi antara aktor dengan *use case* dalam sistem. *Use case* dibuat untuk membantu calon pengguna sistem untuk mendapat pemahaman yang utuh tentang sistem yang akan dibangun. *Use case diagram* analisis sentimen dapat dilihat pada Gambar III.15.



Gambar III.15 *Use Case Diagram* Analisis Sentimen

III.4.2.1 Identifikasi Use Case

Identifikasi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Hasil dari identifikasi *use case* dijelaskan pada Tabel III.8.

Tabel III.8 Identifikasi *Use Case* Analisis Sentimen

No.	*Use Case*	Deskripsi
1.	*Crawling Tweets*	Proses untuk melakukan pengambilan *tweets* dari Twitter.
2.	*Request Tweets*	Proses untuk meminta *tweets* pada Tweetinvi API.
3.	*Case Folding*	Proses untuk mengubah kata ke dalam huruf kecil (*lowercase*).
4.	Normalisasi Fitur	Proses yang digunakan untuk menghilang *username*, URL dan RT pada *tweets*.
5.	*Convert Emoticon*	Proses untuk mengganti setiap *emoticon* menjadi kata.
6.	*Convert Negation*	Proses untuk mengubah nilai sentimen jika terdapat kata negasi
7.	*Tokenizing*	Proses untuk memecah *tweets* menjadi potongan kata-kata.
8.	*Stopword Removal*	Proses yang digunakan untuk menghapus setiap kata yang tidak ada kaitannya dengan analisis sentimen.

9.	*Stemming*	Proses untuk mereduksi setiap kata pada *tweets* sehingga mendapatkan bentuk kata dasar.
10.	Klasifikasi *Tweets*	Proses yang digunakan untuk mengklasifikasikan *tweets* menjadi sentimen positif atau sentimen negatif.
11.	Visualisasi Diagram *Pie*	Proses untuk menggambarkan persentasi dari sentimen positif dan sentimen negatif dalam bentuk diagram *pie*.
12.	Ekstraksi *Keyword*	Proses untuk mendapatkan kata kunci (*keyword*) yang menjadi topik dari kumpulan sentimen.

III.4.2.2 Skenario Use Case Diagram

Setiap urutan langkah-langkah dalam sistem dideskripsikan dengan skenario *use case diagram*. Berikut adalah skenario dari masing-masing *use case* analisis sentimen:

a. *Use Case Crawling Tweets*

Proses ini dilakukan untuk mengambil data *tweets* dari *official* akun Twitter Telkom Speedy (@TelkomSpeedy). Proses *crawling* secara otomatis akan mengambil data *tweets* yang mengandung kata "TelkomSpeedy", "speedy reguler", "speedy instant", dan "speedy gold" dan akan disimpan dalam *database*. Skenario untuk *use case crawling* dapat dilihat pada Tabel III.9.

Tabel III.9 Skenario *Use Case Crawling*

Use Case Name	*Crawling Tweets*	
Related Requirements	SKPL-F001	
Goals	Mengambil data *tweets* dari akun @TelkomSpeedy dan menyimpannya ke dalam *database*	
Preconditions	Sistem tidak melakukan pengambilan data *tweets*	
Successful End Condition	Dapat mengambil data *tweets* dan dapat disimpan dalam *database*	
Failed End Condition	Tidak dapat mengambil data *tweets* dan tidak dapat disimpan dalam *database*	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem untuk melakukan proses pengambilan *tweets*
Main Flow	*Steps*	*Action*
	1	Pilih menu *Crawling*
	2	Klik tombol Ambil *Tweets*
	3	Sistem memeriksa ketersediaan akses internet
	4	Sistem melakukan *request* data *tweets* ke Tweetinvi API
	5	Sistem menyimpan data *tweets* ke *database*
Extension	*Steps*	*Branching Action*
	3.1	Jika tidak ada akses internet, munculkan pesan

b. *Use Case Request Tweets*

Proses ini digunakan untuk meminta *tweets* pada Tweetinvi API. Skenario untuk *use case request tweets* dapat dilihat pada Tabel III.10.

Tabel III.10 Skenario *Use Case Request Tweets*

Use Case Name	*Requst Tweets*	
Related Requirements	SKPL-F002	
Goals	Melakukan *request* data *tweets* ke Tweetinvi API	
Preconditions	Sistem terhubung dengan jaringan internet	
Successful End Condition	Tweetinvi API mengirimkan data *tweets* pada sistem	
Failed End Condition	Tweetinvi API tidak mengirimkan data *tweets* pada sistem	
Primary Actors	Tweetinvi API	
	Trigger	*Request tweets* pada Tweetinvi API
Main Flow	*Steps*	*Action*
	1	Sistem melakukan *request* data *tweets* ke Tweetinvi API
	2	Tweetinvi API mengirimkan data *tweets* ke sistem

c. *Use Case Case Folding*

Proses ini digunakan untuk mengubah kata ke dalam huruf kecil (*lowercase*). Skenario *use case* ekstraksi *keyword* dapat dilihat pada Tabel III.11.

Tabel III.11 Skenario *Use Case Case Folding*

Use Case Name	*Case Folding*	
Related Requirements	SKPL-F003	
Goals	Mengubah kata ke dalam huruf kecil (*lowercase*)	
Preconditions	Sistem sudah melakukan proses *crawling*	
Successful End Condition	Berhasil mengubah kata menjadi *lowercase*	
Failed End Condition	Masih terdapat huruf capital pada *tweets*	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem mengubah setiap kata menjadi *lowercase*
Main Flow	*Steps*	*Action*
	1	Pilih menu Klasifikasi
	2	Klik tombol Proses *Tweets*
	3	Sistem memeriksa huruf kapital pada *tweets*
	4	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	3.1	Jika terdapat huruf kapital, maka akan diubah menjadi huruf kecil (*lowercase*)

d. *Use Case* Normalisasi Fitur

Proses ini digunakan untuk menghapus *username*, URL dan RT pada *tweets*. Skenario untuk *use case* normalisasi fitur dapat dilihat pada Tabel III.12.

Tabel III.12 Skenario *Use Case* Normalisasi Fitur

Use Case Name	Normalisasi Fitur	
Related Requirements	SKPL-F004	
Goals	Menghapus *username*, URL dan RT pada *tweets*	
Preconditions	*Tweets* sudah melewati tahap *case folding*	
Successful End Condition	*Username*, URL dan RT pada *tweets* berhasil dihapus	
Failed End Condition	Terdapat u*sername*, URL dan RT pada *tweets*	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem menghapus *username*, URL, dan RT pada *tweets*
Main Flow	*Steps*	*Action*
	1	Sistem memeriksa RT, *username* dan URL dari hasil *case folding*
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat RT, *username* dan URL maka akan dihapus

e. *Use Case Convert Emoticon*

Proses untuk mengganti setiap *emoticon* yang terdapat pada *tweets* menjadi kata yang sesuai dengan *emoticon* tersebut. Skenario untuk *use case convert emoticon* dapat dilihat pada Tabel III.13.

Tabel III.13 Skenario *Use Case Convert Emoticon*

Use Case Name	*Convert Emoticon*	
Related Requirements	SKPL-F005	
Goals	Mengubah setiap *emoticon* menjadi kata yang sesuai dengan *emoticon* tersebut	
Preconditions	*Tweets* sudah melewati tahap normalisasi fitur	
Successful End Condition	Setiap *emoticon* berhasil diubah menjadi kata yang sesuai dengan *emoticon*nya	
Failed End Condition	*Emoticon* tidak dikonversi	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem mengubah setiap *emoticon* menjadi kata
Main Flow	*Steps*	*Action*
	1	Sistem memeriksa *emoticon* pada *tweets* dari hasil normalisasi fitur
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat *emoticon*, ubah sesuai dengan *string* dari *emoticon* tersebut

f. *Use Case Convert Negation*

Proses ini digunakan untuk mengubah nilai sentimen. Jika ditemukan kata negasi sebelum kata yang bernilai positif, maka kata tersebut akan diubah nilainya menjadi negatif dan begitupun sebaliknya. Skenario untuk *use case convert negation* dapat dilihat pada Tabel III.14.

Tabel III.14 Skenario *Use Case Convert Negation*

Use Case Name	*Convert Negation*	
Related Requirements	SKPL-F006	
Goals	Mengubah nilai sentimen jika ditemukan kata negasi	
Preconditions	*Tweets* sudah melewati tahap *convert emoticon*	
Successful End Condition	Berhasil mengubah setiap nilai sentimen yang terdapat kata negasi	
Failed End Condition	Nilai sentimen tidak dikonversi meskipun pada *tweets* terdapat kata negasi.	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem mengubah nilai sentimen jika terdapat kata negasi
Main Flow	*Steps*	*Action*
	1	Sistem memeriksa kata negasi pada *tweets* dari hasil *convert emoticon*
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat kata positif setelah kata negasi, maka ubah nilai sentimen menjadi negatif
	2.2	Jika terdapat kata negatif setelah kata negasi, maka ubah nilai sentimen menjadi positif

g. *Use Case Tokenizing*

Proses ini digunakan untuk memecah setiap *tweets* menjadi potongan kata-kata. Skenario untuk *use case tokenizing* dapat dilihat pada Tabel III.15.

Tabel III.15 Skenario *Use Case Tokenizing*

Use Case Name	*Tokenizing*	
Related Requirements	SKPL-F007	
Goals	Memecah setiap *tweets* menjadi potongan kata-kata.	
Preconditions	*Tweets* sudah melewati tahap *convert negation*	
Successful End Condition	Berhasil memecah setiap *tweets* menjadi potongan-potongan kata.	
Failed End Condition	Sistem tidak melakukan *tokenizing* sehingga *tweets* masih dalam bentuk kalimat	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem memecah setiap *tweets* menjadi potongan kata-kata
	Steps	*Action*

Main Flow	1	Sistem memeriksa karakter *delimiter* pada *tweets* dari hasil *convert negation*
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat karakter *delimiter*, pisahkan *tweets* menjadi potongan kata

h. *Use Case Stopword Removal*

Proses ini digunakan untuk menghapus setiap kata yang tidak ada kaitannya dengan analisis sentimen. Skenario untuk *use case stopword removal* dapat dilihat pada Tabel III.16.

Tabel III.16 Skenario *Use Case Stopword Removal*

Use Case Name	*Stopword Removal*	
Related Requirements	SKPL-F008	
Goals	Menghapus kata-kata yang termasuk *stopword*	
Preconditions	*Tweets* sudah melewati tahap *tokenizing*	
Successful End Condition	Setiap kata yang termasuk *stopword* berhasil dihapus	
Failed End Condition	Terdapat kata yang termasuk *stopword* pada *tweets*	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem menghapus kata-kata yang termasuk *stopword*
Main Flow	*Steps*	*Action*
	1	Sistem memeriksa kata *stopword* pada *tweets* dari hasil *tokenizing*
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat kata *stopword*, maka hapus kata *stopword* tersebut

i. *Use Case Stemming*

Proses ini digunakan untuk mereduksi setiap kata pada *tweets* sehingga mendapatkan bentuk kata dasar. Skenario untuk *use case stemming* dapat dilihat pada Tabel III.17.

Tabel III.17 Skenario *Use Case Stemming*

Use Case Name	*Stemming*
Related Requirements	SKPL-F009
Goals	Menghilangkan imbuhan kata sehingga mendapatkan bentuk kata dasar
Preconditions	*Tweets* sudah melewati tahap *stopword removal*
Successful End Condition	Setiap kata dalam *tweets* menjadi bentuk kata dasar
Failed End Condition	Masih terdapat kata yang mengandung imbuhan
	Pengguna

Primary Actors	*Trigger*	Pengguna meminta sistem menghilangkan imbuhan pada setiap kata
Main Flow	*Steps*	*Action*
	1	Sistem memeriksa kata berimbuhan dari hasil *stopword removal*
	2	Sistem menyimpan data *tweets* dalam *array*
Extension	*Steps*	*Branching Action*
	2.1	Jika terdapat kata yang mengandung imbuhan, maka hapus imbuhan tersebut

j. *Use Case* Klasifikasi *Tweets*

Proses ini digunakan untuk mengklasifikasikan *tweets* menjadi sentimen positif atau sentimen negatif. Metode klasifikasi yang digunakan adalah *Naïve Bayes Classifier*. Skenario *use case* klasifikasi *tweets* terdapat pada Tabel III.18.

Tabel III.18 Skenario *Use Case* Klasifikasi *Tweets*

Use Case Name	Klasifikasi *Tweets*	
Related Requirements	SKPL-F010	
Goals	Mengklasifikasikan *tweets* ke dalam sentimen positif atau sentimen negative	
Preconditions	*Tweets* sudah melalu tahap *preprocessing*	
Successful End Condition	Setiap *tweets* berhasil diklasifikasikan sesuai sentimennya	
Failed End Condition	*Tweets* tidak diklasifikasi	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem mengklasifikasikan *tweets* sesuai dengan sentimennya
Main Flow	*Steps*	*Action*
	1	Pilih menu Klasifikasi
	2	Klik tombol Proses *Tweets*
	3	Sistem memeriksa ketersediaan data *tweets*
	4	Sistem melakukan tahapan *preprocessing* (*tokenizing*, normalisasi fitur, *case folding*, *convert emoticon*, *convert negation*, *stemming* dan *stopword removal*)
	5	Sistem melakukan perhitungan klasifikasi *Naïve Bayes*
	6	Sistem menyimpan data *tweets* ke *database*
Extension	*Steps*	*Branching Action*
	2.1	Jika data *tweets* tidak ada, munculkan pesan

k. *Use Case* Visualisasi Diagram *Pie*

Proses ini digunakan untuk menggambarkan persentasi dari sentimen positif dan sentimen negatif dalam bentuk diagram *pie*. Skenario untuk *use case* visualisasi diagram *pie* dapat dilihat pada Tabel III.19.

Tabel III.19 Skenario *Use Case* Visualisasi Diagram *Pie*

Use Case Name	Visualisasi Diagram *Pie*	
Related Requirements	SKPL-F011	
Goals	Sistem menampilkan diagram *pie* berdasarkan dari persentasi sentimen positif dan negatif	
Preconditions	Diketahui persentasi dari sentimen positif dan sentimen negative	
Successful End Condition	Menggambarkan diagram *pie* sesuai dengan persentasinya	
Failed End Condition	Diagram *pie* tidak tampil	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem menampilkan persentasi sentimen positif dan negatif dalam bentuk diagram *pie*
Main Flow	*Steps*	*Action*
	1	Pilih menu Visualisasi
	2	Pilih tanggal
	3	Sistem menghitung persentasi dari jumlah sentimen positif dan sentimen negatif (dihitung perhari)
	4	Sistem menampilkan persentasi sentimen positif dan negatif dalam bentuk diagram *pie*

l. *Use Case* Ekstraksi *Keyword*

Proses ini digunakan untuk mendapatkan kata kunci (*keyword*) yang menjadi topik dari kumpulan sentimen berdasarkan nilai TF-IDF tertinggi. Skenario untuk *use case* ekstraksi *keyword* dapat dilihat pada Tabel III.20.

Tabel III.20 Skenario *Use Case* Ekstraksi *Keyword*

Use Case Name	Ekstraksi *Keyword*	
Related Requirements	SKPL-F012	
Goals	Mendapatkan kata kunci (*keyword*) dengan nilai TF-IDF tertinggi	
Preconditions	Setiap kata belum dilakukan pembobotan	
Successful End Condition	Mendapatkan kata kunci (*keyword*) dengan nilai TF-IDF tertinggi	
Failed End Condition	Setiap kata nilai bobotnya nol	
Primary Actors	Pengguna	
	Trigger	Pengguna meminta sistem menampilkan kata kunci dari kumpulan *tweets* (diambil perhari)
Main Flow	*Steps*	*Action*
	1	Pilih menu Visualisasi
	2	Pilih tanggal
	3	Sistem menghitung nilai TF-IDF (berdasarkan *tweets* perhari)
	4	Sistem menampilkan kata yang memiliki nilai TF-IDF tertinggi
	Steps	*Branching Action*

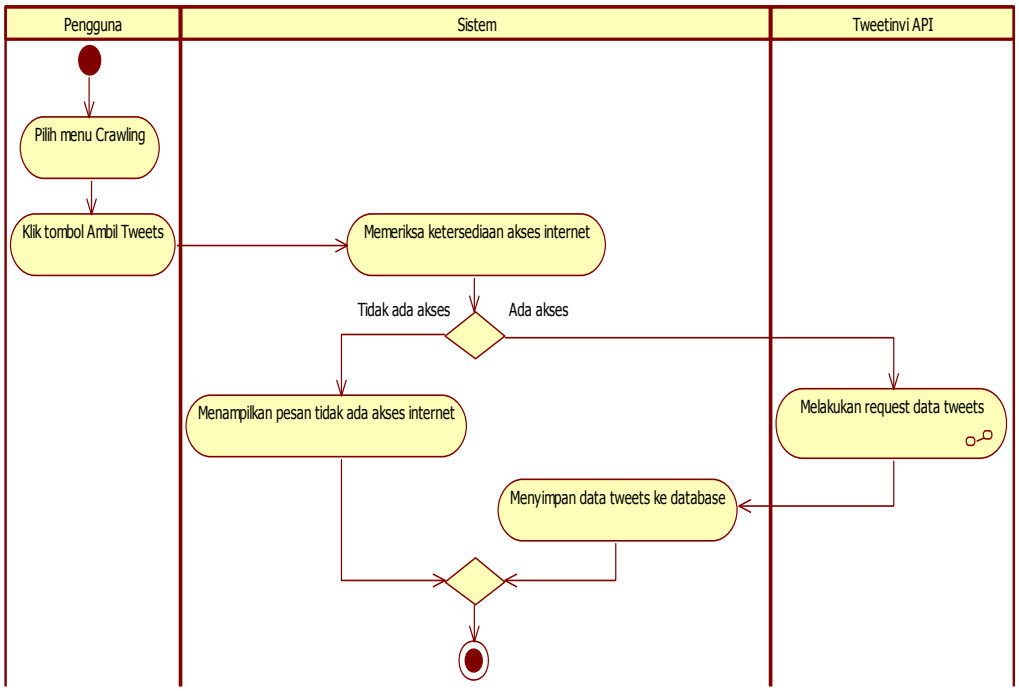
Extension	3.1	Jika tanggal tidak dipilih, maka munculkan pesan
	3.2	Jika tanggal yang dipilih tidak sesuai, maka munculkan pesan

III.4.3 Activity Diagram

Activity diagram pada penelitian ini menjelaskan tentang alur kerja tahapan-tahapan aktivitas dari *use case* yang akan dibangun. Berikut adalah masing-masing *activity diagram* analisis sentimen:

1. *Activity Diagram Crawling*

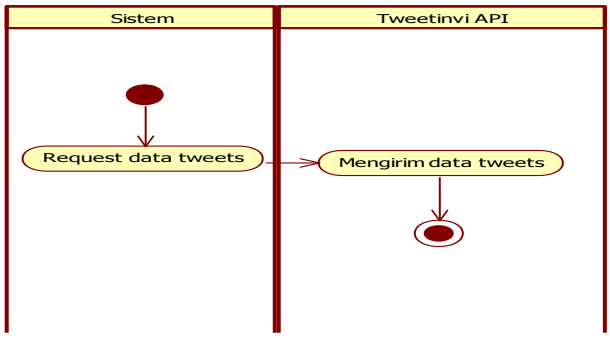
Activity ini menjelaskan tentang proses *crawling* pada sistem yang akan dibangun. Pada *activity* ini terdapat *sub activity request tweets*. *Activity diagram crawling* dapat dilihat pada Gambar III.16.



Gambar III.16 *Activity Diagram Crawling*

1.1. *Sub Activity Diagram Request Tweets*

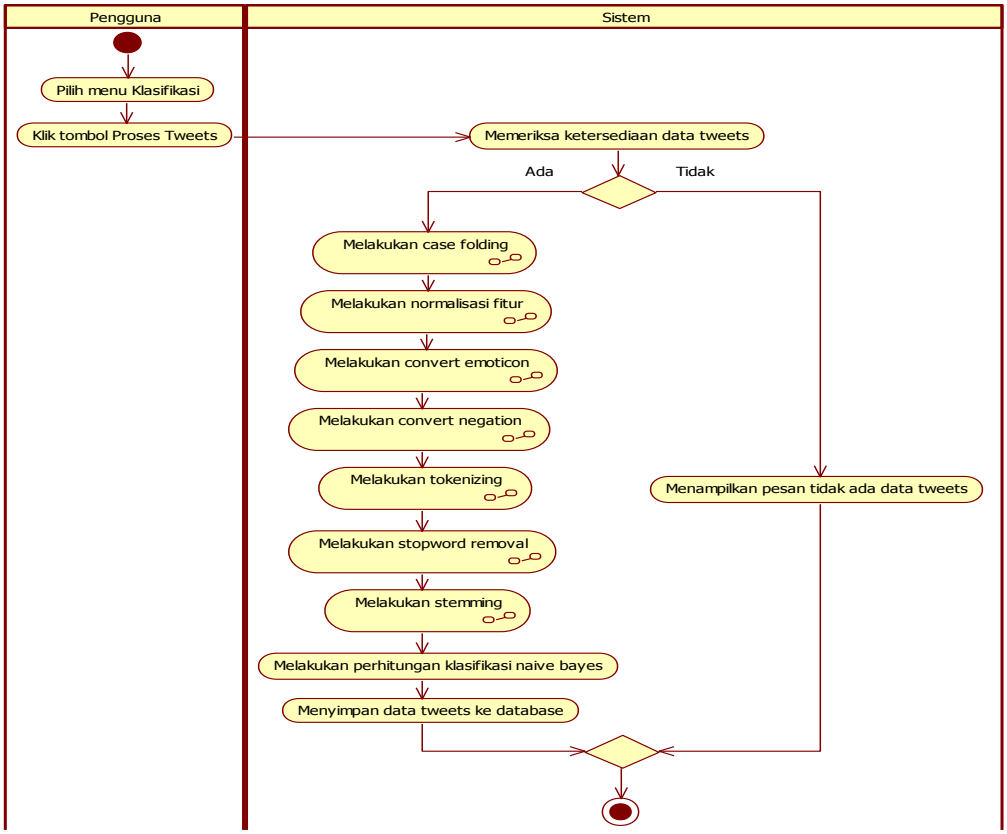
Sub activity ini menggambarkan *request tweets* ke Tweetinvi API. *Sub activity diagram request tweets* dapat dilihat pada Gambar III.17.



Gambar III.17 *Sub Activity Diagram Request Tweets*

2. *Activity Diagram* Klasifikasi *Tweets*

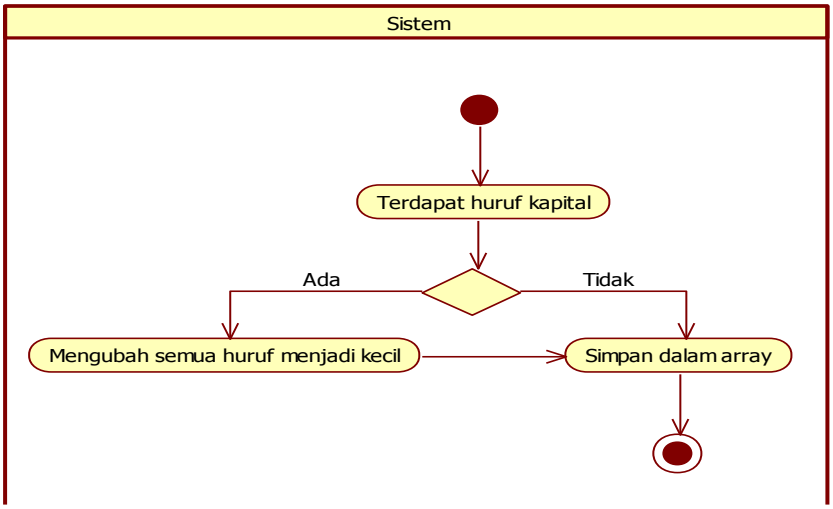
Activity ini menjelaskan tentang tahapan-tahapan yang dilakukan dalam proses klasifikasi *tweets*. Pada *activity* ini terdapat *sub activity preprocessing* seperti *tokenizing*, normalisasi fitur, *case folding*, *convert emoticon*, *convert negation*, *stemming*, dan *stopword removal*. *Activity diagram* klasifikasi *tweets* digambarkan pada Gambar III.18.



Gambar III.18 *Activity Diagram* Klasifikasi *Tweets*

2.1. *Sub Activity Diagram Case Folding*

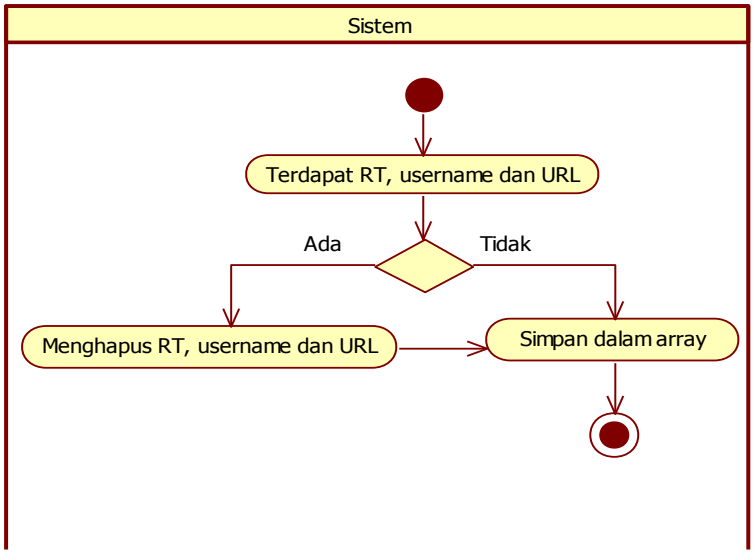
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *case folding*. *Sub activity diagram case folding* dapat dilihat pada Gambar III.19.



Gambar III.19 *Sub Activity Diagram Case Folding*

2.2. *Sub Activity Diagram* Normalisasi Fitur

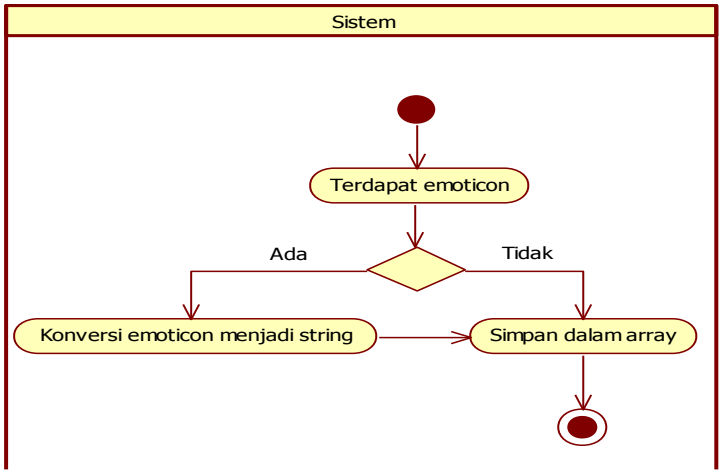
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan normalisasi fitur. *Sub activity diagram* normalisasi fitur dapat dilihat pada Gambar III.20.



Gambar III.20 *Sub Activity* Diagram Normalisasi Fitur

2.3. *Sub Activity Diagram Convert Emoticon*

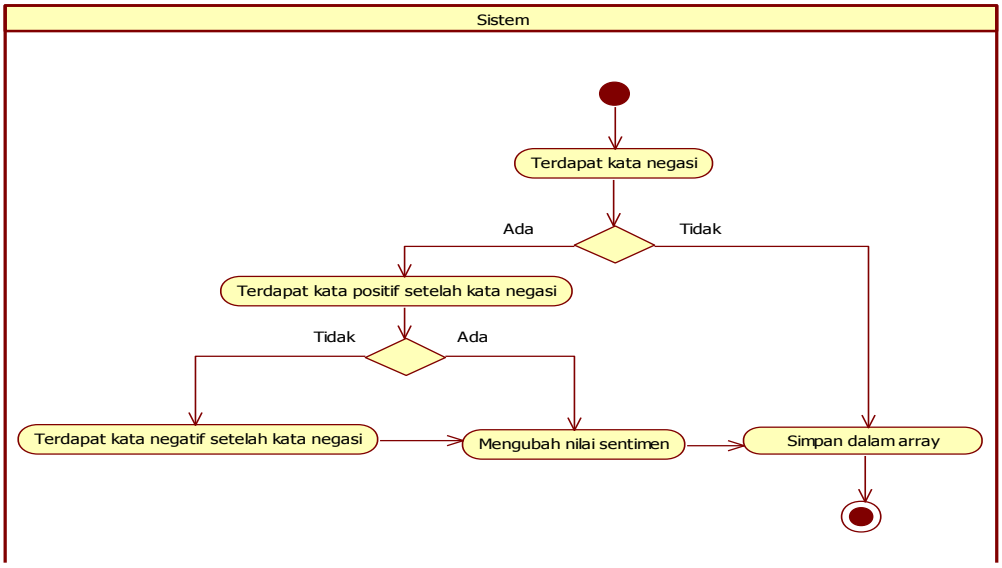
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *convert emoticon*. *Sub activity diagram convert emoticon* dapat dilihat pada Gambar III.21.



Gambar III.21 *Sub Activity Diagram Convert Emoticon*

2.4. *Sub Activity Diagram Convert Negation*

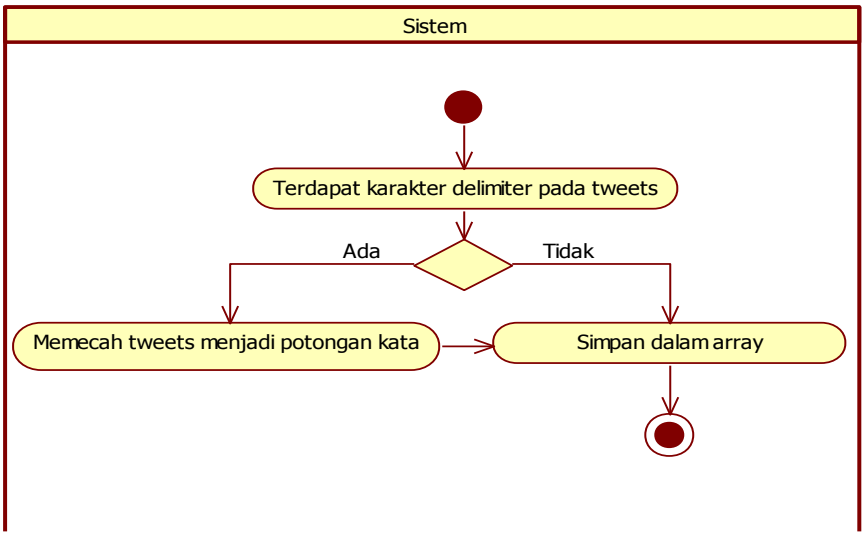
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *convert negation*. *Sub activity diagram convert negation* dapat dilihat pada Gambar III.22.



Gambar III.22 *Sub Activity Diagram Convert Negation*

2.5. *Sub Activity Diagram Tokenizing*

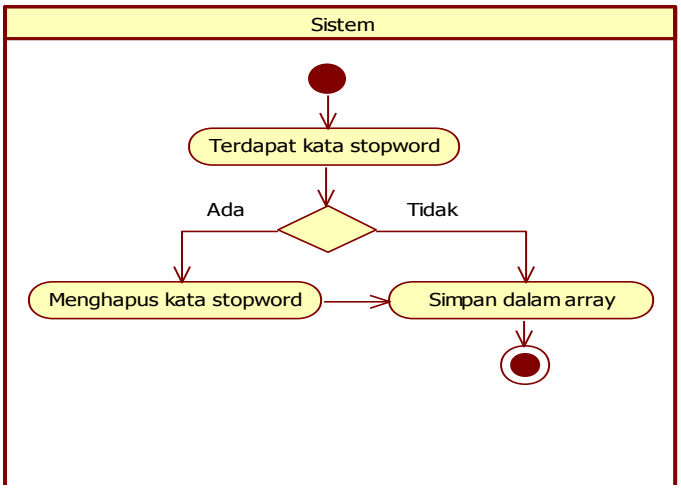
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *tokenizing*. *Sub activity diagram tokenizing* dapat dilihat pada Gambar III.23.



Gambar III.23 *Sub Activity Diagram Tokenizing*

2.6. *Sub Activity Diagram Stopword Removal*

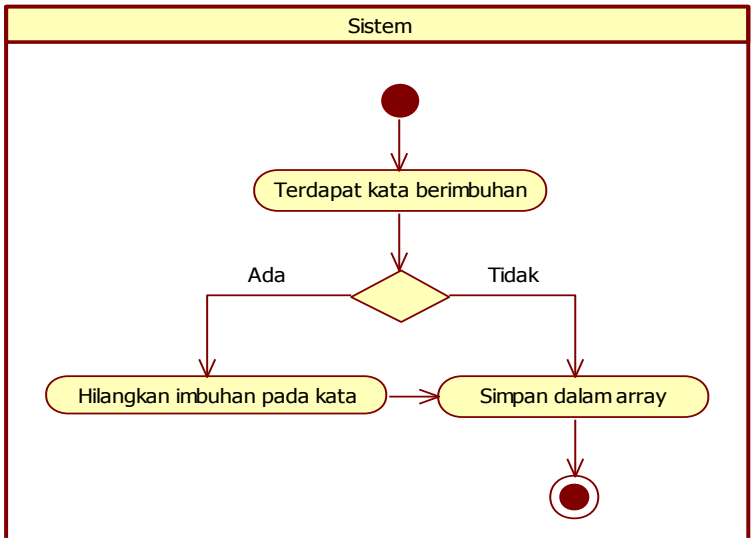
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *stopword removal*. *Sub activity diagram stopword removal* dapat dilihat pada Gambar III.24.



Gambar III.24 *Sub Activity Diagram Stopword Removal*

2.7. *Sub Activity Diagram Stemming*

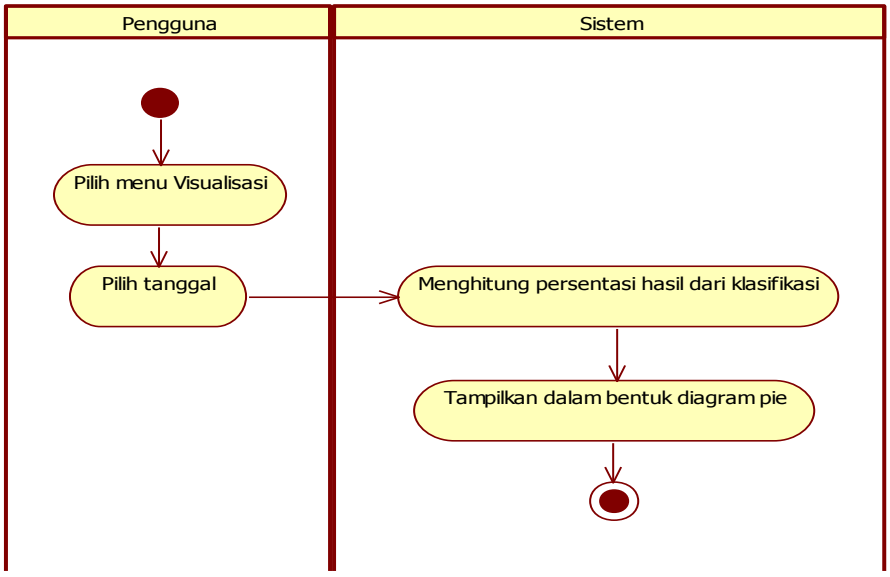
Sub activity ini menjelaskan tentang aturan-aturan yang ada pada tahapan *stemming*. *Sub activity diagram stemming* dapat dilihat pada Gambar III.25.



Gambar III.25 *Sub Activity Diagram Stemming*

3. *Activity Diagram* Visualisasi

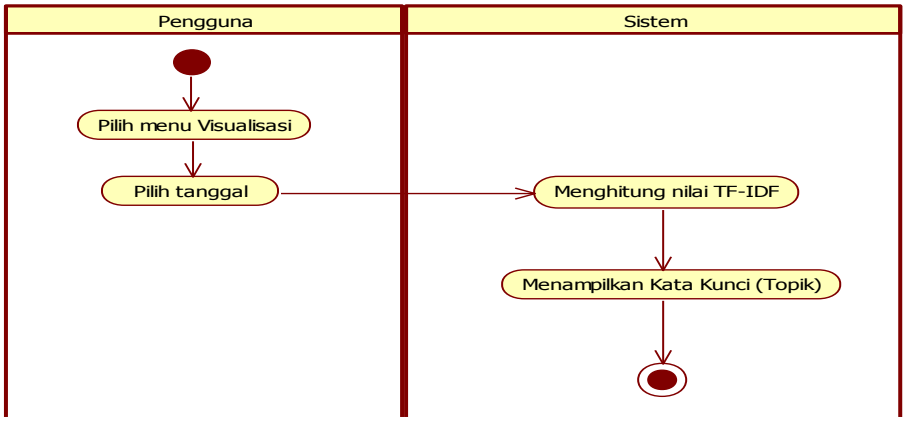
Activity ini menjelaskan tentang proses visualisasi pada sistem yang akan dibangun. *Activity diagram* visualisasi dapat dilihat pada Gambar III.26.



Gambar III.26 *Activity Diagram* Visualisasi

4. *Activity Diagram* Ekstraksi *Keyword*

Activity ini menjelaskan tentang proses ekstraksi *keyword* pada sistem yang akan dibangun. *Activity diagram* ekstraksi *keyword* dapat dilihat pada Gambar III.27.



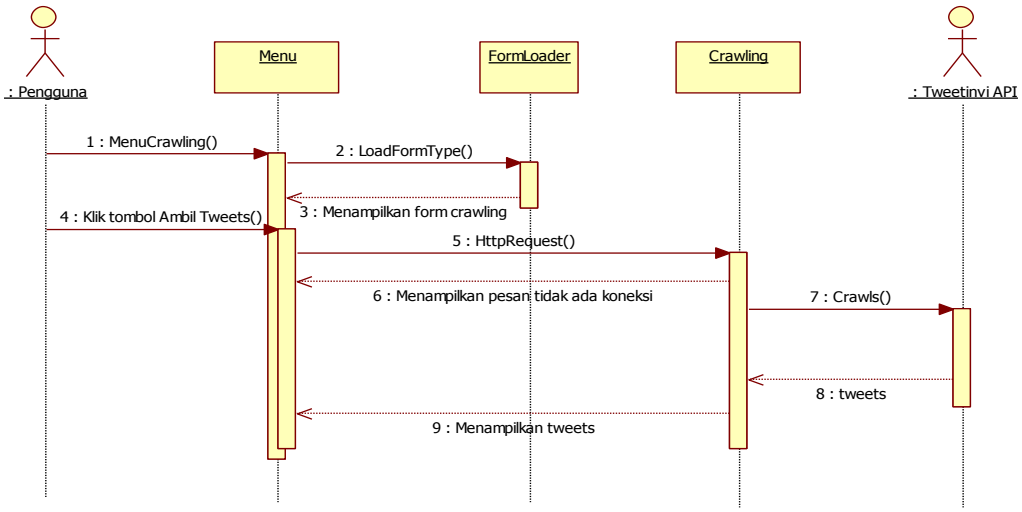
Gambar III.27 *Activity Diagram* Ekstraksi *Keyword*

III.4.4 Sequence Diagram

Sequence diagram menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah *sequence diagram* pada sistem analisis sentimen:

1. *Sequence Diagram Crawling*

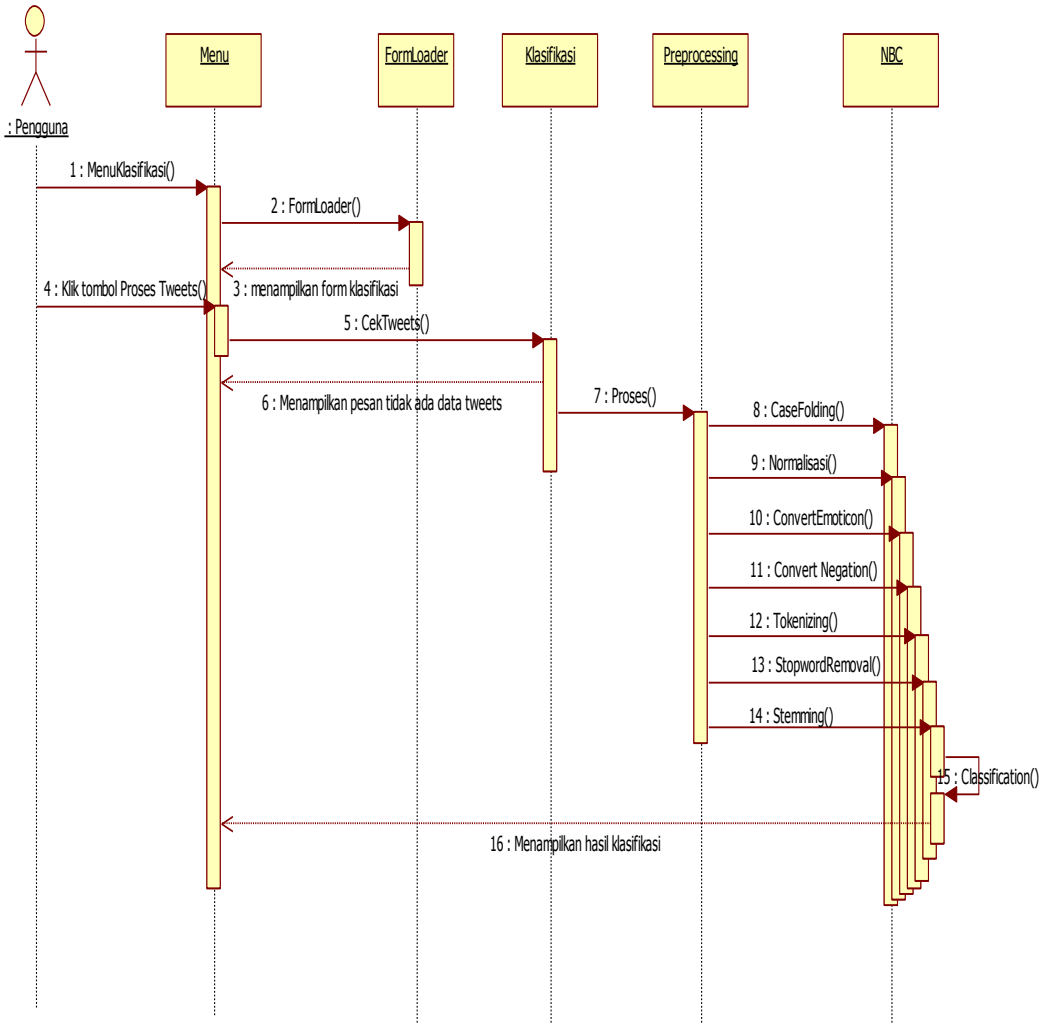
Sequence diagram crawling dapat dilihat pada berikut:



Gambar III.28 *Sequence Diagram Crawling*

2. *Sequence Diagram* Klasifikasi *Tweets*

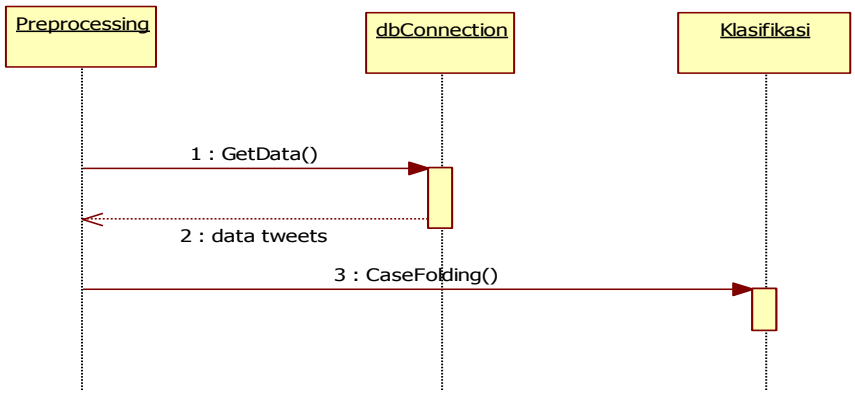
   Sequence diagram klasifikasi *tweets* dapat dilihat pada Gambar III.29 berikut:



Gambar III.29 *Sequence Diagram* Klasifikasi *Tweets*

3. *Sequence Diagram Case Folding*

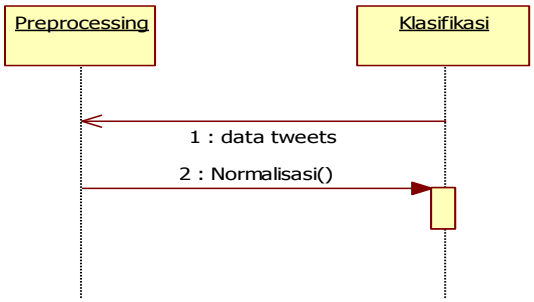
   Sequence diagram case folding dapat dilihat pada gambar berikut:



Gambar III.30 *Sequence Diagram Case Folding*

4. *Sequence Diagram* Normalisasi Fitur

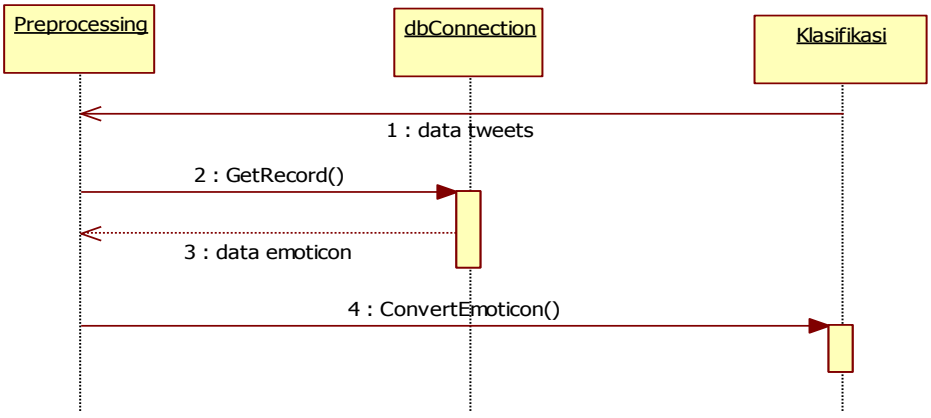
   Sequence diagram normalisasi fitur dapat dilihat pada gambar berikut:



Gambar III.31 *Sequence Diagram* Normalisasi Fitur

5. *Sequence Diagram Convert Emoticon*

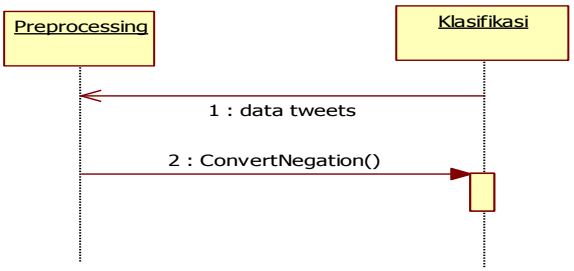
   Sequence diagram convert emoticon dapat dilihat pada gambar berikut:



Gambar III.32 *Sequence Diagram Convert Emoticon*

6. *Sequence Diagram Convert Negation*

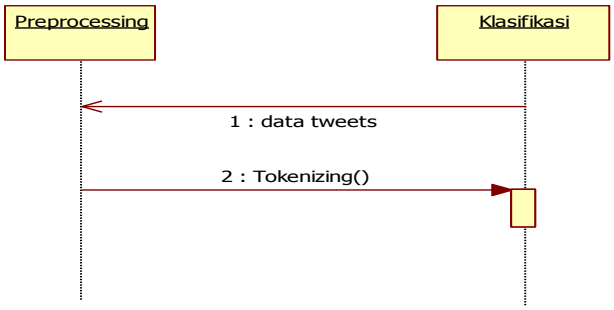
   Sequence diagram convert negation dapat dilihat pada gambar berikut:



Gambar III.33 *Sequence Diagram Convert Negation*

7. *Sequence Diagram Tokenizing*

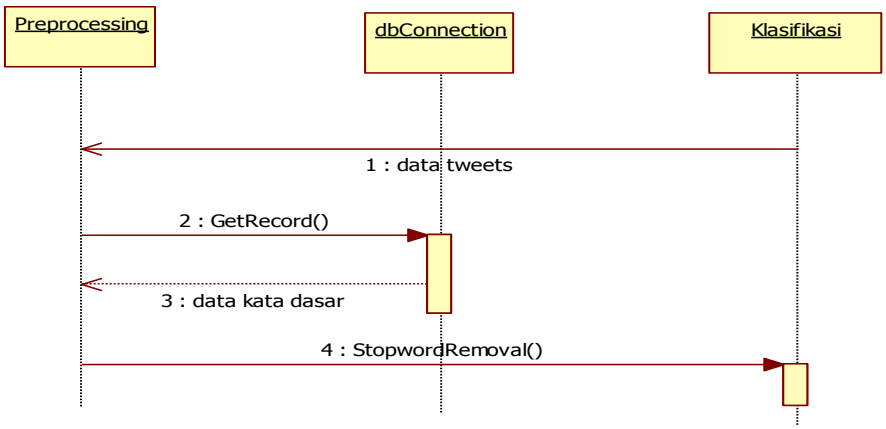
   Sequence diagram tokenizing dapat dilihat pada gambar berikut:



Gambar III.34 *Sequence Diagram Tokenizing*

8. *Sequence Diagram Stopword Removal*

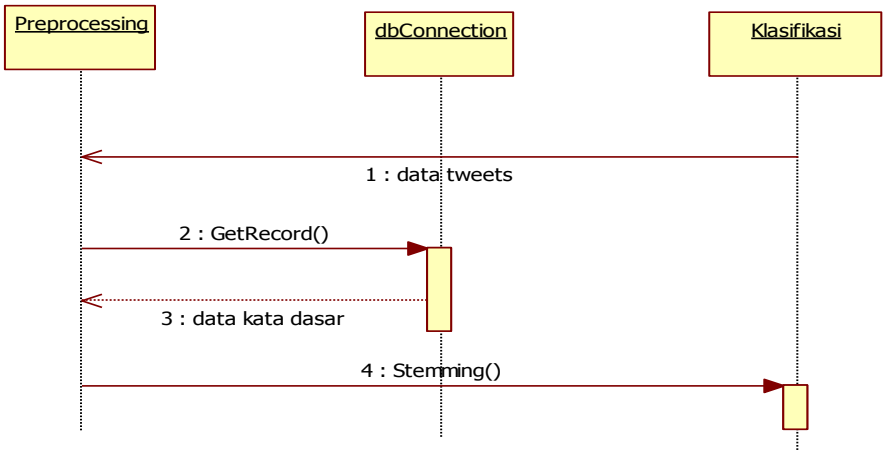
   Sequence diagram stopword removal dapat dilihat pada gambar berikut:



Gambar III.35 *Sequence Diagram Stopword Removal*

9. *Sequence Diagram Stemming*

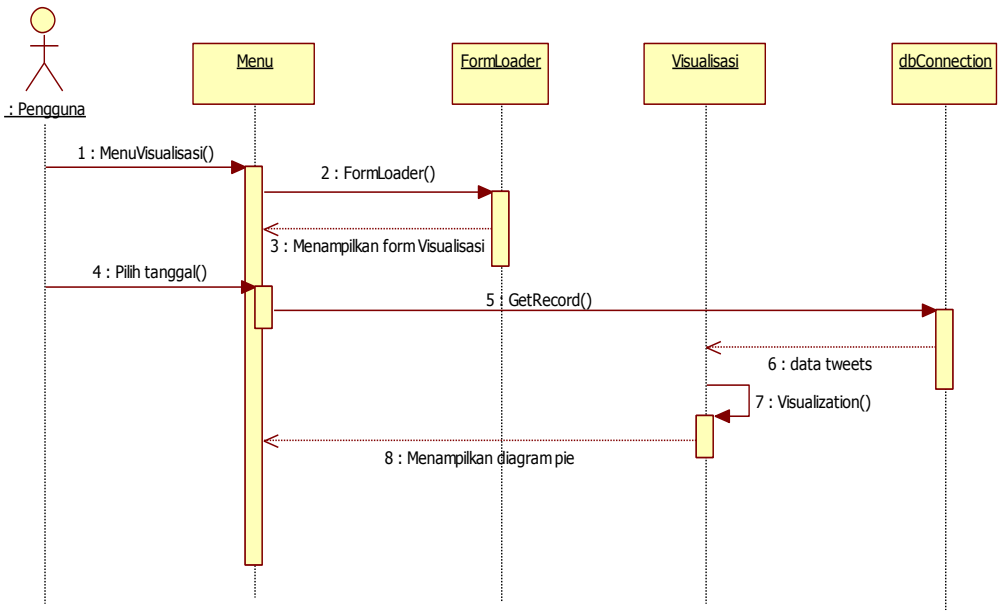
   Sequence diagram stemming dapat dilihat pada berikut:



Gambar III.36 *Sequence Diagram Stemming*

10. *Sequence Diagram* Visualisasi

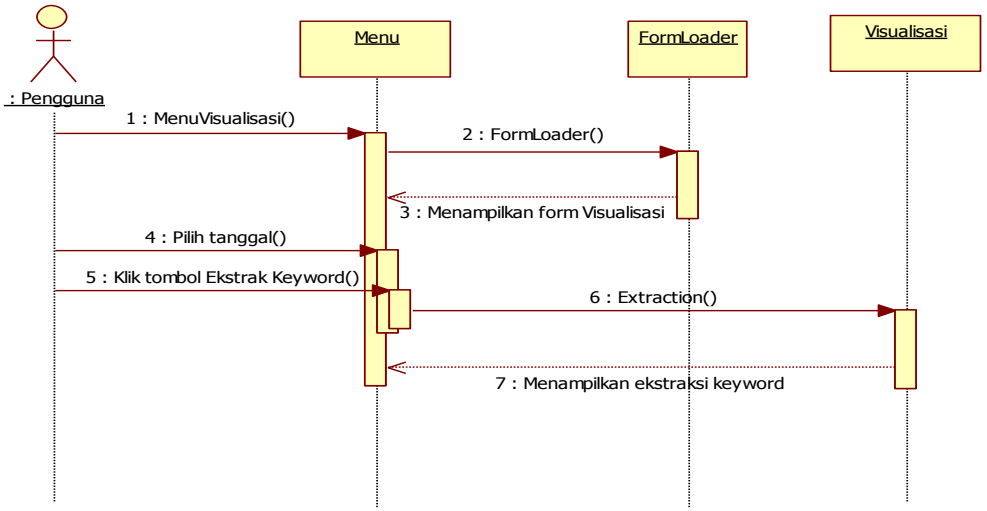
    Sequence diagram visualisasi dapat dilihat pada Gambar III.37 berikut:



Gambar III.37 *Sequence Diagram* Visualisasi

11. *Sequence Diagram* Ekstraksi *Keyword*

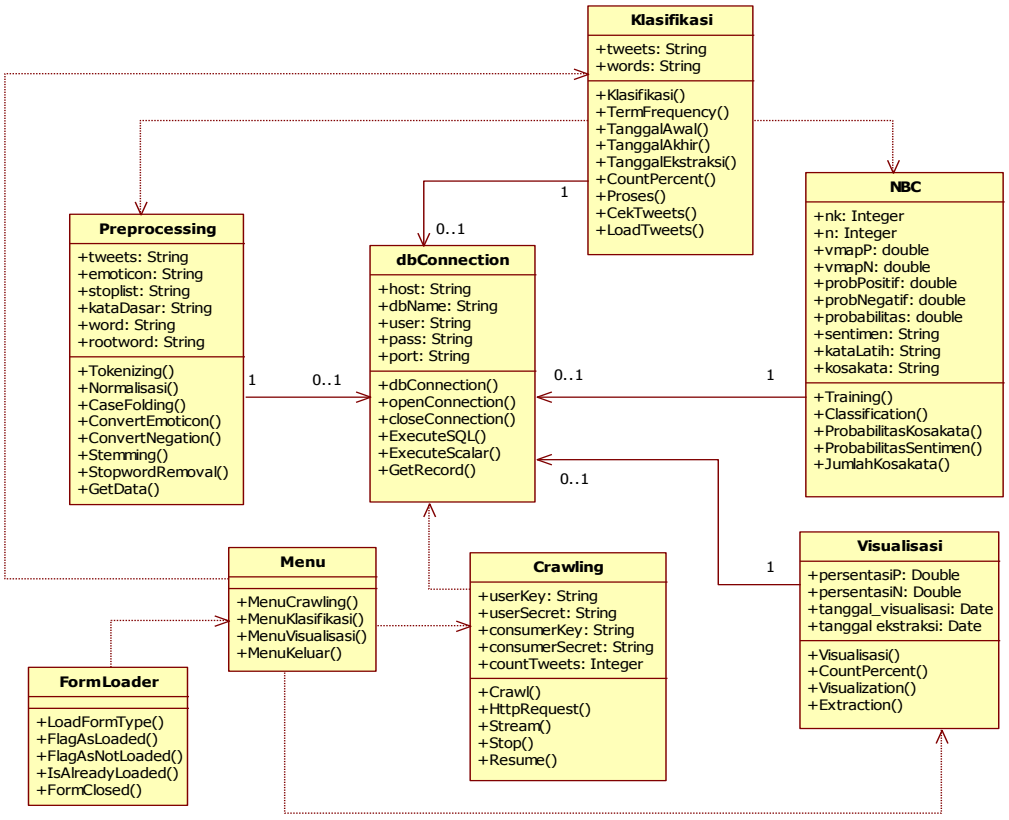
    Sequence diagram ekstraksi *keyword* dapat dilihat pada gambar berikut:



Gambar III.38 *Sequence Diagram* Ekstraksi *Keyword*

III.4.5 Class Diagram

Diagram ini menggambarkan kelas-kelas dan ditujukan untuk mendefinisikan proses yang berjalan pada sistem analisis sentimen.



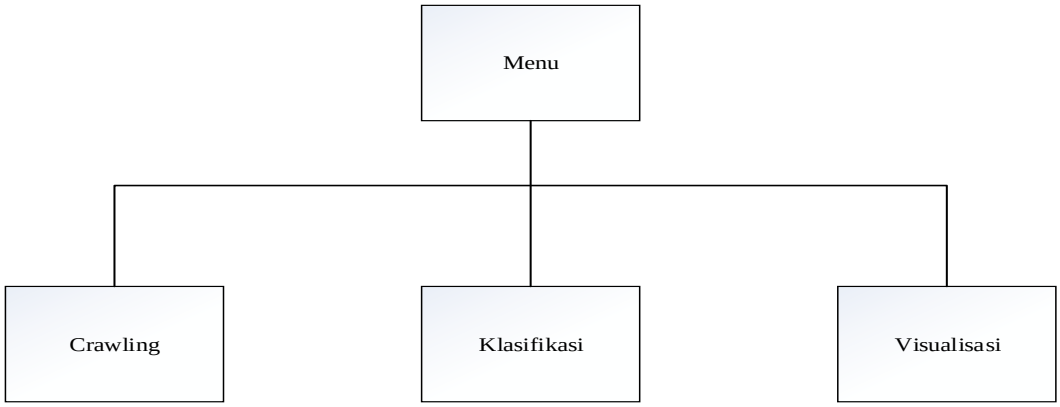
Gambar III.39 Class Diagram Analisis Sentimen

III.5 Perancangan Arsitektur Sistem

Tahap perancangan merupakan tahapan selanjutnya setelah tahapan analisis. Perancangan dalam membangun sistem ini meliputi perancangan struktur menu, perancangan antarmuka, perancangan pesan dan perancangan prosedural.

III.5.1 Perancangan Struktur Menu

Berikut ini adalah struktur menu yang memaparkan mengenai menu yang akan dibangun pada sistem analisis sentimen.

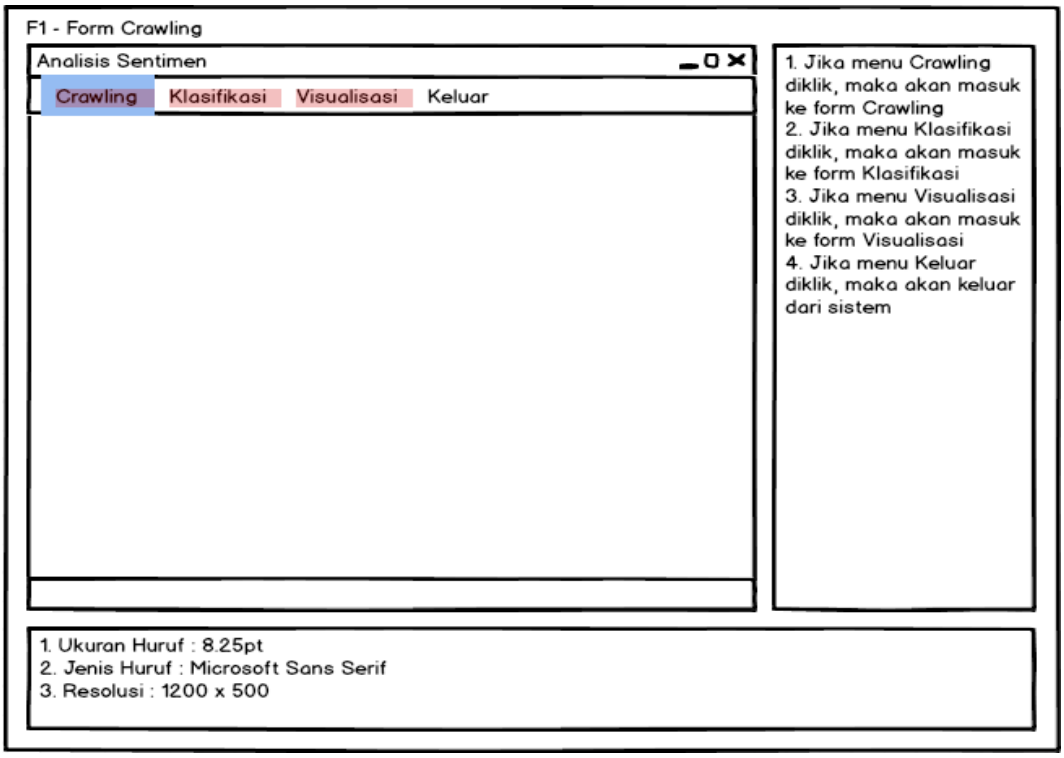


Gambar III.40 Struktur Menu

III.5.2 Perancangan Antarmuka

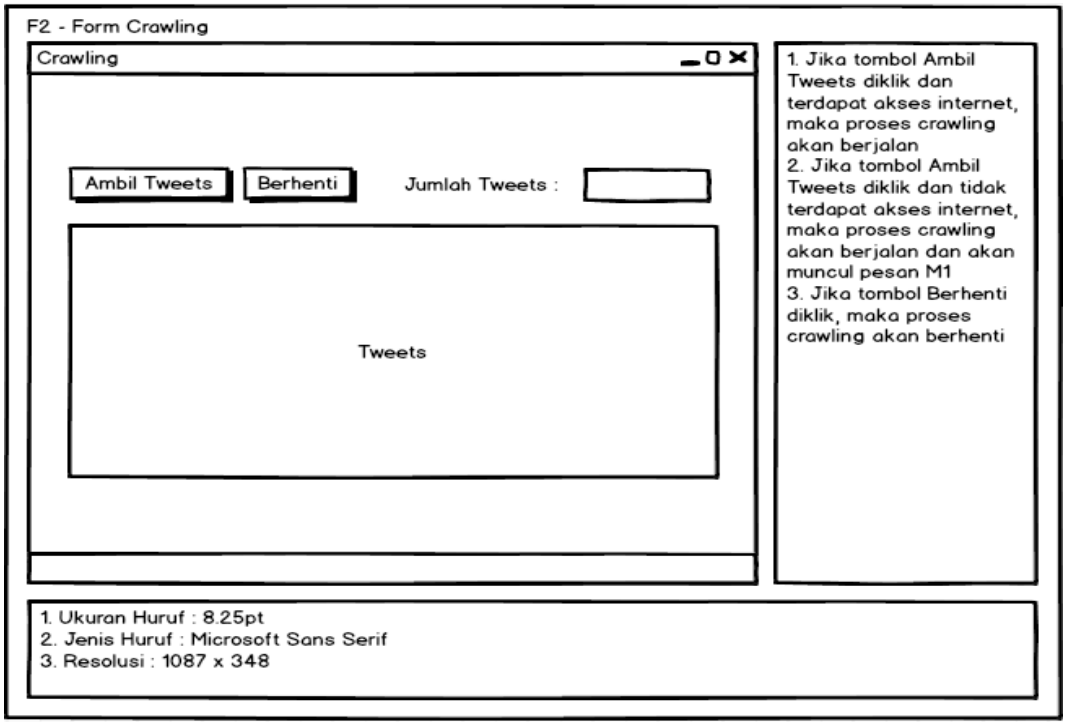
Perancangan antarmuka sendiri adalah proses pembuatan perancangan *form-form* (bentuk-bentuk) yang akan disajikan pada layar. Hasil akhir dari proses perancangan antarmuka ini disebut tampilan layar. Berikut merupakan gambar hasil akhir proses perancangan antarmuka, yaitu tampilan layar pada sistem perangkat lunak yang dibangun:

1. *Form Menu*



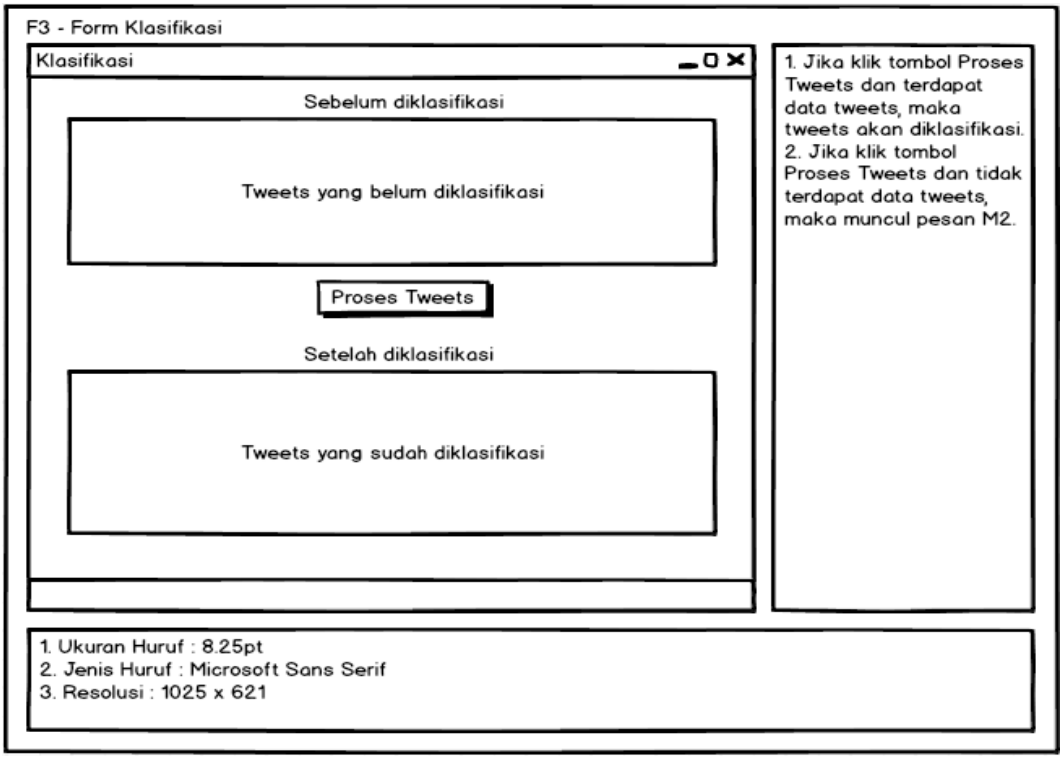
Gambar III.41 *Form Menu*

2. *Form Crawling*



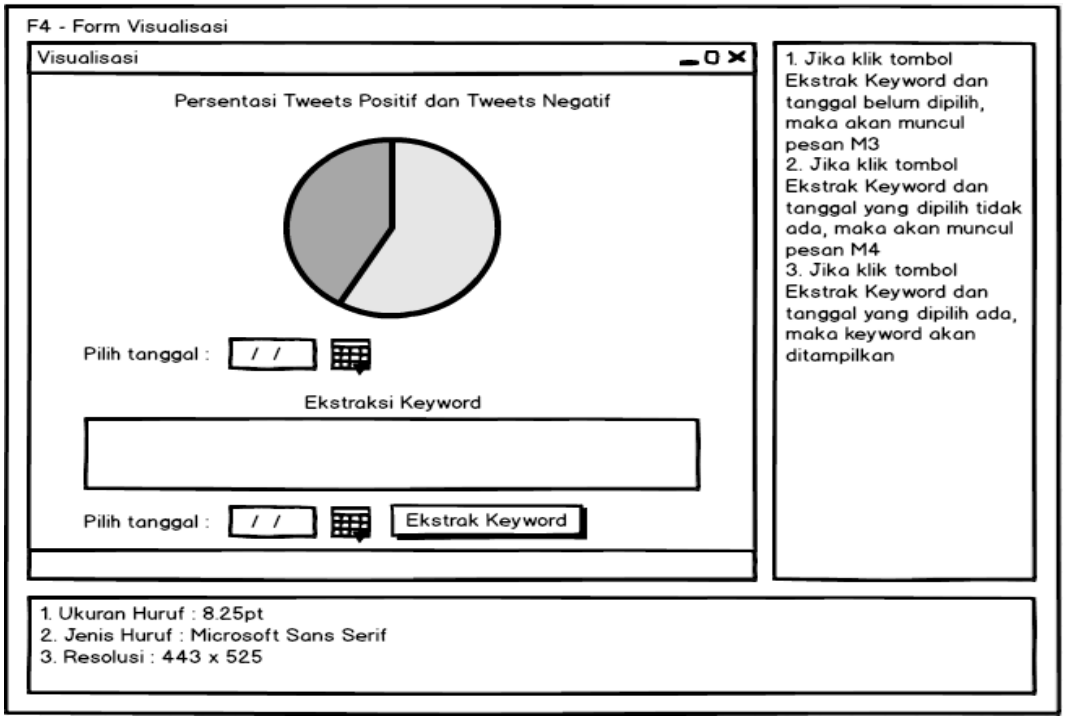
Gambar III.42 *Form Crawling*

3. *Form* Klasifikasi *Tweets*



Gambar III.43 *Form* Klasifikasi *Tweets*

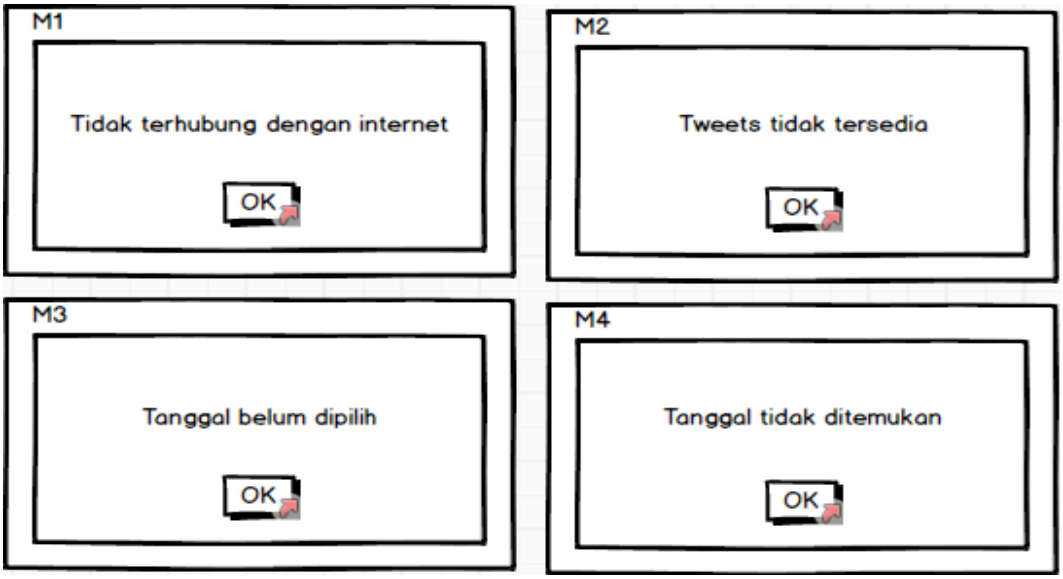
4. *Form* Visualisasi



Gambar III.44 *Form* Visualisasi

III.5.3 Perancangan Pesan

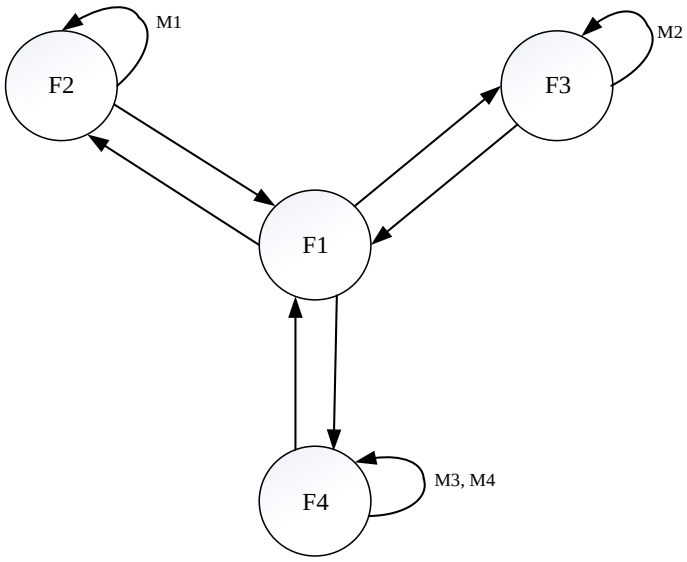
Berikut ini adalah perancangan pesan yang digunakan pada analisis sentimen:



Gambar III.45 Perancangan Pesan

III.5.4 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan dari navigasi menu dari saru *form* ke *form* lain. Jaringan semantik yang terbentuk pada sistem ini dapat dilihat pada Gambar III.46.



Gambar III.46 Jaringan Semantik