BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang akan dibangun. Analisis tersebut meliputi analisis masalah, analisis sistem yang sedang berjalan, analisis data masukan, analisis preprocessing, analisis metode/ algoritma, analisis kebutuhan non fungsional dan analisis kebutuhan fungsional.

III.1.1 Analisis Masalah

PT. Maicih Inti Sinergi (MIS) merupakan sebuah perusahaan yang menggunakan t*witter* sebagai salah satu sarana untuk mempromosikan produknya. Dengan adanya akun *twitter* PT. MIS dan jumlah postingan yang banyak setiap harinya memungkinkan *feedback* dari konsumen atau *follower* PT. MIS sebagai ketersediaan sentimen yang dapat dimanfaatkan untuk evaluasi produk PT.MIS.

Namun permasalahannya adalah bagaimana membangun sebuah perangkat lunak yang dapat melakukan pengklasifikasian sentimen pada data *twitter* yang nantinya dapat dimanfaatkan untuk evaluasi produk PT.MIS.

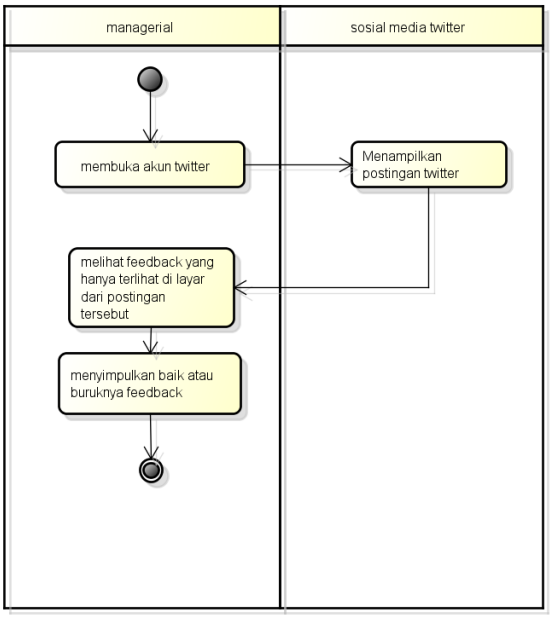
III.1.2 Analisis Sistem Berjalan

Pada penelitian ini, analisis sistem yang sedang berjalan dilakukan dengan metode wawancara untuk mendapatkan gambaran lengkap tentang bagaimana sistem yang sedang berjalan saat ini yang kemudian hasilnya akan digambarkan kedalam bentuk *activity diagram* sesuai dengan masalah yang sedang terjadi.

Berikut adaah sistem yang berjalan di PT. Maicih Inti Sinergi dalam mengetahui bagaimana sentimen para konsumen terhadap produknya.

1. Managerial membuka akun twitter PT. MIS (@maicih)
2. Managerial melihat *feedback* yang hanya terlihat dilayar.
3. Managerial menyimpulkan baik atau buruk nya *feedback* yang diberikan konsumen.

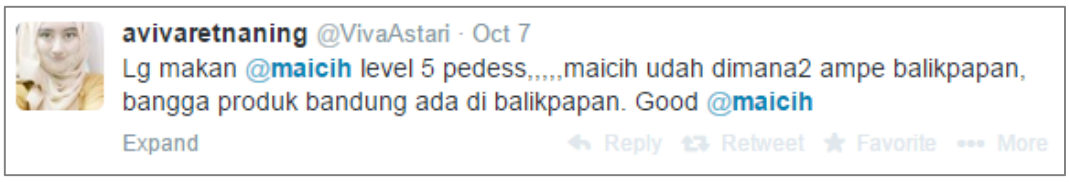
Activity diagram sistem yang sedang berjalan di PT. Maicih Inti Sinergi dapat dilihat pada Gambar III-1



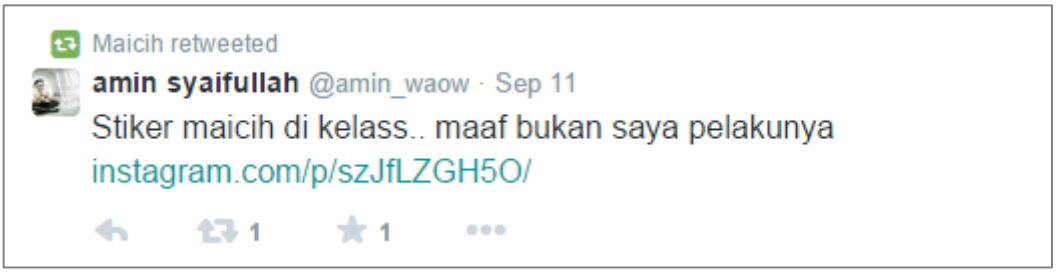
Gambar III-1 sistem yang sedang berjalan

III.1.3 Analisis Data Masukan

Data masukan yang digunakan adalah data *tweet* dari akun twitter PT. Maicih Inti Sinergi (@maicih). Data *tweet* tersebut didapat dengan memanfaatkan fitur API (*Aplication Interface*) yang telah disediakan oleh *twitter*. Data yang diambil adalah data *tweet* yang mengandung kata "maicih" atau data *tweet* yang terdapat pada akun PT. Maicih Inti Sinergi "@maicih". Data yang didapat berupa sebuah kalimat dengan panjang maksimal 140 karakter. Contoh data *tweet* yang didapat dari *twitter* dapat dilihat pada Gambar III-2 dan Gambar III-3



Gambar III-2 contoh tweet positif



Gambar III-3 Contoh tweet negatif

III.1.4 Analisis Preprocessing

Agar data *tweets* tersebut dapat dimanfaatkan dengan baik untuk mengklasifikasikan sentimen, maka diperlukan proses *preprocessing.* Pada proses ini data *tweet* yang digunakan untuk data *training* dan data *testing* dibersihkan dari *noise* seperti link, "RT", "@", *stopword* dan lain sebagainya. Gambaran proses *preprocessing* dapat dilihat pada Gambar III-4



Gambar III-4 Gambaran Tahapan Preprocessing

Proses *preprocessing* terdiri dari beberapa tahapan yaitu *case folding, cleansing, stopword removal, convert emoticon, convert negation, tokenizer* dan *stemming.* Berikut adalah penjelasan dari masing-masing tahapan:

1. **Case Folding**

Case Folding ialah proses merubah huruf kapital (*uppercase*) menjadi huruf kecil (*lowercase*). Hal ini dilakukan agar semua huruf pada *tweet* menjadi seragam. Gambaran proses dari tahap *case folding* dapat dilihat pada Gambar III-5

@maicih **M**aicih level 10 pedasss #kebakaranmulut ga kuat lagi makannya :'(
@maicih **K**abar menggembirakan **U**ntuk kita semua. **M**aicih kini ada **L**evel 0 :)
RT @maicih **M**aicih kayaknya **E**nak **YES**. :) :)



@maicih maicih level 10 pedasss #kebakaranmulut ga kuat lagi makannya :'(
@maicih kabar menggembirakan untuk kita semua. maicih kini ada level 0 :)
rt @maicih maicih kayaknya enak yes. :) :)

Gambar III-5 Gambaran Proses *Case Folding*

2. Tokenizing

Pada proses *tokenizing* setiap kata pada *tweet* dipisahkan, pada proses ini tahap yang dilakukan adalah memisahkan setiap kata yang dipisahkan oleh spasi. Hal ini dilakukan agar tahap preproses selanjutnya dapat berjalan. Gambaran proses dari tahap *tokenizing* dapat dilihat pada Gambar III-6

maicih level **10** pedasss gakuat makannya emotsedih
kabar menggembirakan**.** maicih level **0** emotsenang
maicih kayaknya enak yes**.** emotsenang emotsenang

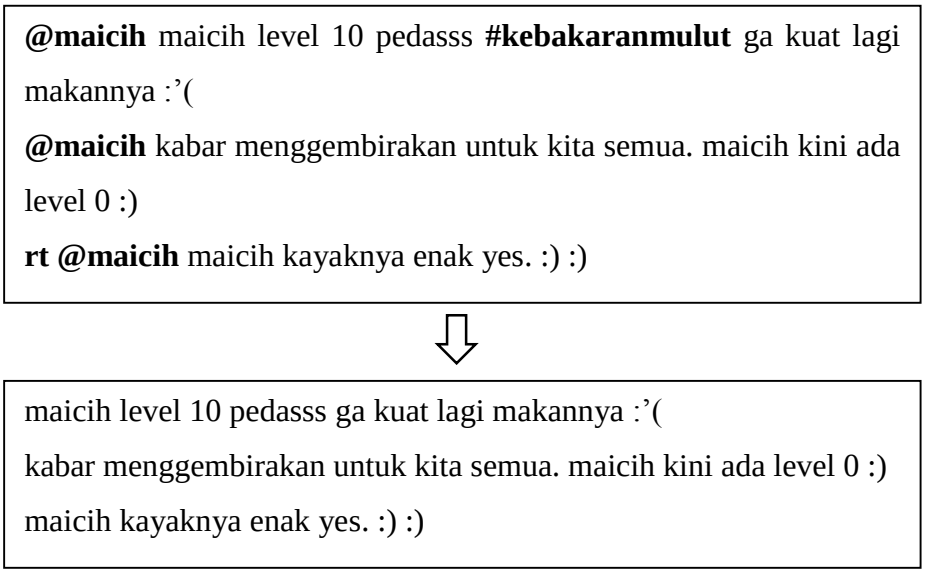


maicih level pedasss gakuat makannya emotsedih
kabar menggembirakan maicih level emotsenang
maicih kayaknya enak yes emotsenang emotsenang

Gambar III-6 Gambaran Proses *Tokenizer*

3. Cleansing

Tweet yang terdapat pada akun PT. Maicih Inti Sinergi memiliki berbagai komponen atau karateristik *tweet* yang khas seperti "@", link, "#" dan RT. Komponen-komponen tersebut tidak memiliki pengaruh apapun terhadap sentimen, maka akan dibuang. Gambaran proses dari tahap *cleansing* dapat dilihat pada Gambar III-7



Gambar III-7 Gambaran Proses Cleansing

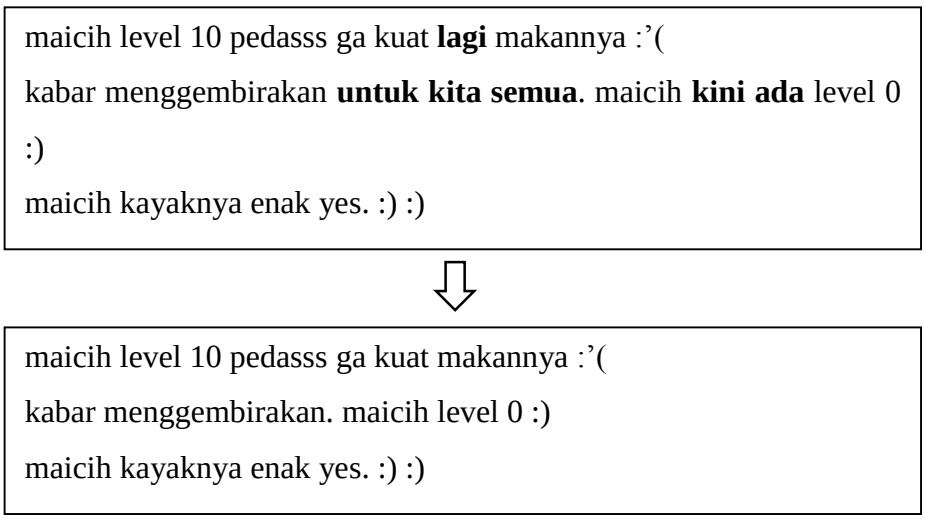
Berikut aturan produksi *regular expression* yang digunakan pada proses *cleansing* :

```
1  // untuk menghilangkan mention
2  $tweet = preg_replace('/@[-A-Z0-9+&@#\/%?=~_|$!:,.;]*[A-Z0-
3  9+&@#\/%=~_|$]/i','', $tweet);
4  //untuk menghilangkan hashtag
5  $tweet=  preg_replace('/#[-A-Z0-9+&@#\/%?=~_|$!:,.;]*[A-Z0-
6  9+&@#\/%=~_|$]/i','', $tweet);
7  // untuk menghilangkan link
8  $tweet = preg_replace('/\b(https?|ftp|file|http):\/\/[-A-Z0-
9  9+&@#\/%?=~_|$!:,.;]*[A-Z0-9+&@#\/%=~_|$]/i','', $tweet);
```

4. Stopword Removal

Pada tahap ini *tweet* masih mengandung kata yang dianggap tidak dapat memerikan pengaruh dalam menentukan suatu kategori sentimen. Kata-kata tersebut dimasukkan ke dalam daftar *stopword* yang biasa berupa kata ganti orang, kata ganti penghubung, paranomial petunjuk dan lain sebagainya. Jika pada *tweet*

yang sudah melalui proses *cleansing* masih terdapat kata yang tercantum pada daftar *stopword* maka kata tersebut dihilangkan. Untuk *list* lengkap *stopword* yang digunakan dapat dilihat pada Lampiran A. Gambaran proses dari tahap *stopword removal* dapat dilihat pada Gambar III-8



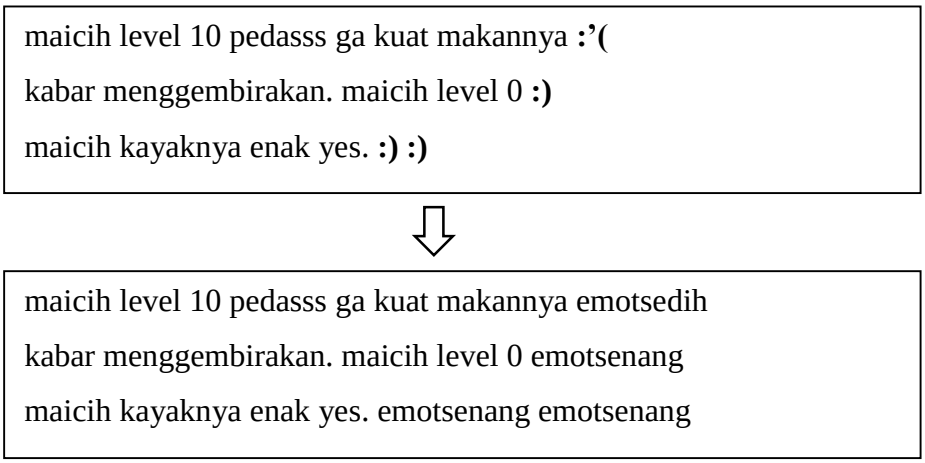
Gambar III-8 Gambaran Proses *Stopword Removal*

Berikut aturan produksi *regular expression* yang digunakan pada proses *stopword Removal* :

```
1  foreach ($hasil as $key) {
2         $stoplist[]= $key['stopword'];
3  }
4
5  $tweet = preg_replace(array_map( function($stopword){
6                   return'/\b'.$stopword.'\b/';
7           }, $stoplist), '',$tweet);
```

5. Convert Emoticon

Convert Emoticon adalah proses mengkonversikan *emoticon* kedalam string yang sesuai dengan ekspresi *emoticon* itu sendiri. *Convert emoticon* dilakukan karena pada *tweet* yang diambil terdapat *emoticon* yang merupakan salah satu cara *follower* mengekspresikan persetujuan atau pertidaksetujuan dalam suatu *tweet*. Hal ini dirasa mempunyai pengaruh terhadap pengklasifikasian sentimen, oleh karena itu *convert emoticon* digunakan. Gambaran proses dari tahap *convert emoticon* dapat dilihat pada Gambar III-9



Gambar III-9 Gambaran Proses *Convert Emoticon*

Untuk *standard* emoticon yang digunakan dalam penelitian ini dapat dilihat pada Tabel III-1

Tabel III-1 standard emoticon

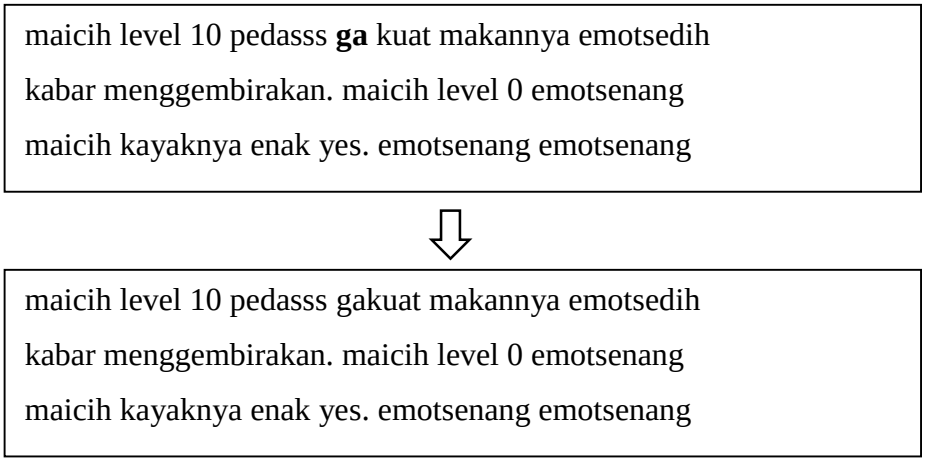
Emoticon	**Konversi**
>:]:-):):o):]:3:c):>=]8)=):}:>)	emotsenang
>:[:-(:(:'(:-c:c:-<:-[:[:{>.><.<>.<	emotsedih

Berikut aturan produksi *regular expression* yang digunakan pada proses *convert emoticon* :

```
1   $esenang=array(">:]",":)",":)",":o)",":]",":3",":c)",":>","
2   =]","8)","=)",":}",":>)",":d",":-d");
3   $esedih = array(">:[",":-(",":(",":'(",":-c",":c",":-<",":-
4   [",":[",":{",">.>","<.<",">.<","t_t");
5
6   //regex senang
7   foreach ($esenang as $item)
8   {
9     $quotedSenang[] = preg_quote($item,'#');
10  }
11  $regexSenang = implode('|', $quotedSenang);
12  $fullSenang = '#(^|\W)('.$regexSenang.')($|\W)#';
13
14  //regex sedih
15  foreach ($esedih as $item)
16  {
17    $quotedSedih[] = preg_quote($item,'#');
18  }
19  $regexSedih = implode('|', $quotedSedih);
20  $fullSedih = '#(^|\W)('.$regexSedih.')($|\W)#';
21
22  $tweet = preg_replace($fullSenang, ' emotsenang ', $tweet);
23  $tweet = preg_replace($fullSedih, ' emotsedih ', $tweet);
```

6. Convert Negation

Convert Negation merupakan proses konversi kata-kata negasi yang terdapat pada suatu *tweet*, karena kata negasi mempunyai pengaruh dalam merubah nilai sentimen pada suatu *tweet*. Jika terdapat kata negasi pada suatu *tweet* makan kata tersebut akan disatukan dengan kata setelahnya. Kata-kata negasi tersebut meliputi "bukan", "bkn", "tidak", "enggak", "g", "ga", "jangan", "nggak", "tak", "tdk", dan "gak". Gambaran proses dari tahap *convert negation* dapat dilihat pada Gambar III-10



Gambar III-10 Gambaran Proses *Convert Negation*

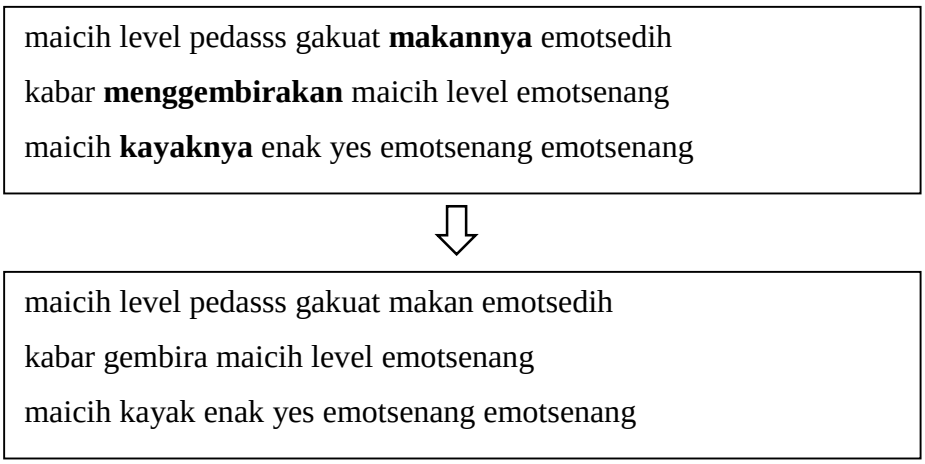
Berikut aturan produksi *regular expression* yang digunakan pada proses *convert negation* :

```
1   $list = array('gak ' => 'gak',
2            'ga ' => 'ga',
3            'ngga ' => 'ngga',
4            'tidak '  => 'tidak',
5            'bkn '=>'bkn',
6            'tida '=>'tida',
7            'tak '=>'tak',
8            'jangan '=>'jangan',
9            'enggak '=>'enggak',
10           'gak  ' => 'gak',
11           'ga  ' => 'ga',
12           'ngga  ' => 'ngga',
13           'tidak  '  => 'tidak',
14           'bkn  '=>'bkn',
15           'tida  '=>'tida',
16           'tak  '=>'tak',
17           'jangan  '=>'jangan',
18           'enggak  '=>'enggak'
19       );
20  $patterns = array();
21  $replacement = array();
22  foreach ($list as $from => $to)
```

```
{
   $from = '/\b' . $from . '\b/';
   $patterns[] = $from;
   $replacement[] = $to;
}
$tweet = preg_replace($patterns, $replacement, $tweet);
```

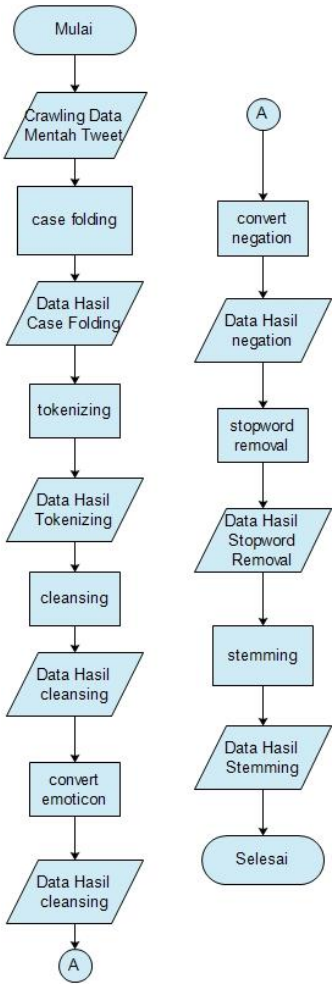
7. Stemming

Stemming adalah proses mengubah kata berimbuhan kebentuk asalnya (kata dasar). Algoritma yang digunakan untuk proses *stemming* berbahasa Indonesia adalah Nazief dan Andriani. Proses *Stemming* juga termasuk salah satu proses dalam *Information Retrieval*. Gambaran proses dari tahap *stemming* dapat dilihat pada Gambar III-11



Gambar III-11 Gambaran Proses *Stemming*

Berikut adalah diagram alir dari proses preprocessing dapat dilihat pada Gambar III-12

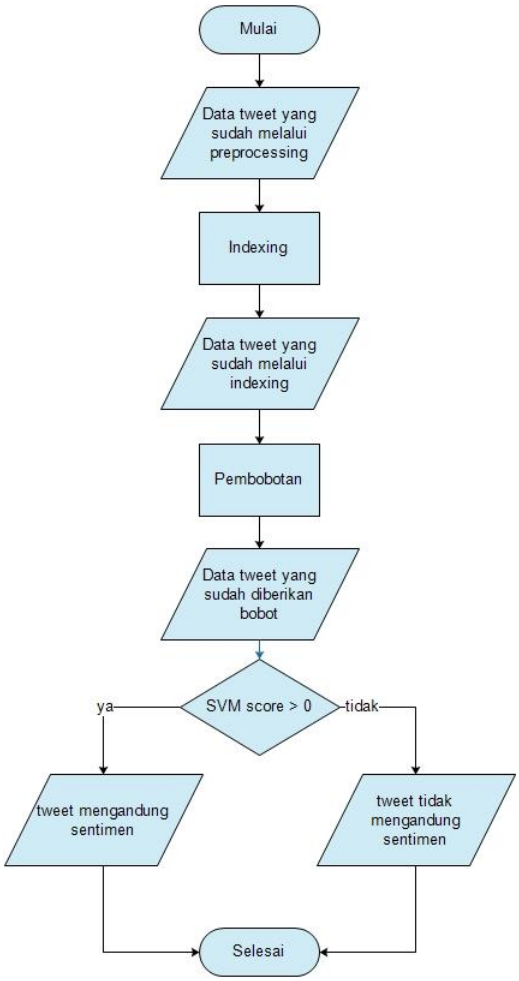


Gambar III-12 Flowchart Preprocessing

III.1.5 Analisis Metode / Algoritma

Analisis Metode yang dilakukan pada penelitian ini memiliki 2 tahap diantaranya *Support Vector Machine* sebagai metode untuk retrieval dan Naïve Bayes sebagai metode untuk *extraction* berupa klasifikasi.

Berikut adalah *flow chart* proses retrieval menggunakan *Support Vector Machine* dapat dilihat pada Gambar III-13



Gambar III-13 flowchart retrieval menggunakan SVM

Dikarenakan SVM merupakan metode *machine learning* yang hanya peduli tentang titik dalam ruang daripada email atau dokumen. Maka dalam penelitian ini untuk format representasi data SVM adalah sebagai berikut.

Relevan : “maicih enak pisan”

menjadi

1 1:0.764 2:0.847 3:0.173

Angka 1 pada karkater pertama menyatakan data tersebut masuk dalam kelas data relevan. Jika data adalah dari kelas tidak relevan maka angka tersebut diganti dengan -1. Kata “maicih” pada kalimat di atas digantikan dengan 1:0.764 pada data vektor yang berarti 1 sebagai id kata untuk “maicih” dan 0.764 sebagai bobot tf-idf untuk kata “maicih” dalam tweet. Begitu seterusnya sehingga semua kata dalam tweet terwakili oleh data vektor.

1. Menghitung Data *Training*

Pada tahap ini data *tweet* training yang sudah melalui tahap preprocessing selanjutnya di index dan dihitung masing-masing bobot per katanya menggunakan persamaan (II.3).

Tabel III-2 Contoh kasus

Tweet	Kategori	Fitur
Tweet1	Relevan	maicih level pedasss gakuat makan emotsedih
Tweet2	Relevan	kabar gembira maicih level emotsenang
Tweet3	Relevan	maicih kayak enak yes emotsenang emotsenang
Tweet4	Tidak relevan	jual produk

Kemudian dihitung bobot masing-masing kata menggunakan TF-IDF.

Tabel III-3 Pembobotan TF-IDF

term	TF				df	idf	TF-IDF			
	T1	T2	T3	T4			T1	T2	T3	T4
maicih	1	1	1	0	3	0.1249387	0.1249387	0.1249387	0.1249387	0
level	1	1	0	0	2	0.30103	0.30103	0.30103	0	0
pedasss	1	0	0	0	1	0.60206	0.60206	0	0	0
gakuat	1	0	0	0	1	0.60206	0.60206	0	0	0
makan	1	0	0	0	1	0.60206	0.60206	0	0	0
emotsedih	1	0	0	0	1	0.60206	0.60206	0	0	0
kabar	0	1	0	0	1	0.60206	0	0.60206	0	0
gembira	0	1	0	0	1	0.60206	0	0.60206	0	0
emotsenang	0	1	2	0	2	0.30103	0	0.30103	0.60206	0
kayak	0	0	1	0	1	0.60206	0	0	0.60206	0
enak	0	0	1	0	1	0.60206	0	0	0.60206	0
yes	0	0	1	0	1	0.60206	0	0	0.60206	0
jual	0	0	0	1	1	0.60206	0	0	0	0.60206
produk	0	0	0	1	1	0.60206	0	0	0	0.60206

Dari Tabel III-3 diketahui bobot masing-masing kata pada semua data tweet training. Selanjutnya data *tweet* tersebut diubah format nya kedalam format SVM. Untuk dokumen relevan di beri nilai 1 dan dokumen tidak relevan di beri label -1.

Tabel III-4 format data svm

term	tweet1 (x_1)	tweet2 (x_2)	tweet3 (x_3)	tweet4 (x4)
maicih	0.1249387	0.1249387	0.1249387	0
level	0.30103	0.30103	0	0
pedasss	0.60206	0	0	0
gakuat	0.60206	0	0	0

term	tweet1 (x_1)	tweet2 (x_2)	tweet3 (x_3)	tweet4 (x4)
makan	0.60206	0	0	0
emotsedih	0.60206	0	0	0
kabar	0	0.60206	0	0
gembira	0	0.60206	0	0
emotsenang	0	0.30103	0.60206	0
kayak	0	0	0.60206	0
enak	0	0	0.60206	0
yes	0	0	0.60206	0
jual	0	0	0	0.60206
produk	0	0	0	0.60206
y (kelas)	**1 (relevan)**	**1 (relevan)**	**1 (relevan)**	**-1 (non relevan)**

Kemudian tahap selanjutnya adalah melakukan kernelisasi menggunakan fungsi polynomial *kernel* pangkat 2 yang didefiniskan sebagai $K(x_i, x_j) = (x_i + x_i^T + 1)^2$. Matriks kernel K dihitung dengan dimensi l x l, dimana l adalah banyaknya data (*tweet*).

$x_1 = [0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]$

$x_2 = [0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]$

$x_3 = [0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]$

$x_4 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602]$

$x_1 x_1^T = [0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]$
$[0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]^T$
$= 0.016 + 0.091 + 0.362 + 0.362 + 0.362 + 0.362$
$= 1.555$

$x_1 x_2^T = [0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]$
$[0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]^T$
$= 0.016 + 0.091$
$= 0.107$

$x_1 x_3^T = [0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]$
$[0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]^T$
$= 0.016$

$x_1 x_4^T = [0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]$
$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602]^T$

$$= 0$$

$$x_2 x_1^T = [0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]$$
$$[0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}$$
$$= 0.016 + 0.091$$
$$= 0.107$$

$$x_2 x_2^T = [0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]$$
$$[0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]^{\mathrm{T}}$$
$$= 0.016 + 0.091 + 0.362 + 0.362 + 0.091$$
$$= 0.922$$

$$x_2 x_3^T = [0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]$$
$$[0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]^{\mathrm{T}}$$
$$= 0.016 + 0.181$$
$$= 0.197$$

$$x_2 x_4^T = [0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]$$
$$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602]^{\mathrm{T}}$$
$$= 0$$

$$x_3 x_1^T = [0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]$$
$$[0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}$$
$$= 0.016$$
$$= 0.016$$

$$x_3 x_2^T = [0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]$$
$$[0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]^{\mathrm{T}}$$
$$= 0.016 + 0.181$$
$$= 0.197$$

$$x_3 x_3^T = [0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]$$
$$[0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]^{\mathrm{T}}$$
$$= 0.016 + 0.362 + 0.362 + 0.362 + 0.362$$
$$= 1.826$$

$$x_3 x_4^T = [0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]$$
$$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602]^{\mathrm{T}}$$

$= 0$

$x_4 x_1^T = [0,0,0,0,0,0,0,0,0,0,0,0,0.602,0.602]$

$[0.125, 0.301, 0.602, 0.602, 0.602, 0.602, 0, 0, 0, 0, 0, 0, 0, 0]^T$

$= 0$

$x_4 x_2^T = [0,0,0,0,0,0,0,0,0,0,0,0,0.602,0.602]$

$[0.125, 0.301, 0, 0, 0, 0, 0.602, 0.602, 0.301, 0, 0, 0, 0, 0]^T$

$= 0$

$x_4 x_3^T = [0,0,0,0,0,0,0,0,0,0,0,0,0.602,0.602]$

$[0.125, 0, 0, 0, 0, 0, 0, 0, 0.602, 0.602, 0.602, 0.602, 0, 0]^T$

$= 0$

$x_4 x_4^T = [0,0,0,0,0,0,0,0,0,0,0,0,0.602,0.602]$

$[0,0,0,0,0,0,0,0,0,0,0,0,0.602,0.602]^T$

$= 0.362 + 0.362$

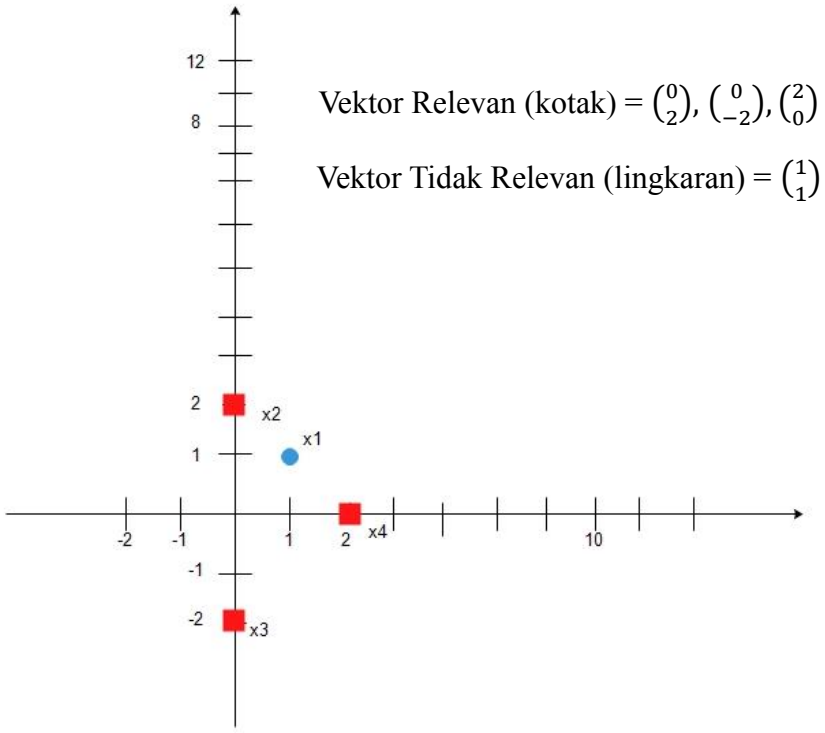
$= 0.725$

Maka matriks K yang terbentuk adalah sebagai berikut:

$$\begin{bmatrix} 1.555 & 0.107 & 0.016 & 0 \\ 0.107 & 0.922 & 0.197 & 0 \\ 0.016 & 0.197 & 1.826 & 0 \\ 0 & 0 & 0 & 0.725 \end{bmatrix}$$

Selanjutnya hasil kernelisasi disubtitusikan ke persamaan **(II-13)**, untuk setiap $\alpha_1 + \alpha_2 - \alpha_3 = 0$ dimana $\alpha_1 \geq 0$.

Didapatkanlah data set sebagai berikut:



Gambar III-14 data set

Kernel Trick Phi:

$$\phi\binom{x}{y} = \begin{cases} \binom{6-x+(x-y)^2}{6-y+(x-y)^2} jika \sqrt{x^2+y^2} \geq 2 \\ \binom{x}{y} \; jika \; tidak \end{cases} \quad \textbf{..(III. 1)}$$

Masukan Vektor-vektor sebaran kedalam persamaa **(III.1)**

Vektor Tidak Relevan = $\binom{1}{1}$

$x_1 = \phi\binom{1}{1} = \sqrt{x^2+y^2} < 2$

Vektor Relevan = $\binom{0}{2}, \binom{0}{-2}, \binom{2}{0}$

$x_2 = \phi\binom{0}{2} = \sqrt{x^2+y^2} \geq 2$

$x_3 = \phi\binom{0}{-2} = \sqrt{x^2+y^2} \geq 2$

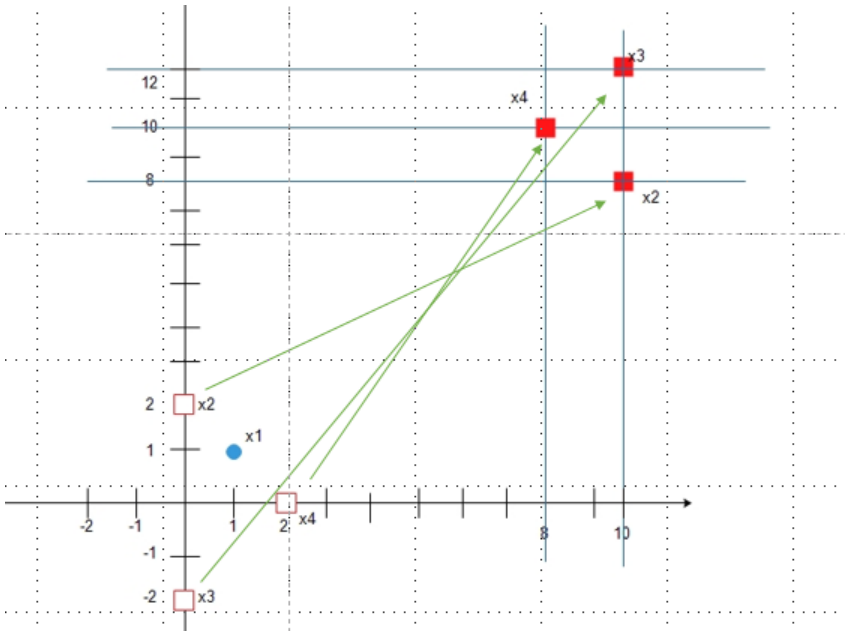
$x_4 = \phi\binom{2}{0} = \sqrt{x^2+y^2} \geq 2$

Disini yang memenuhi hanyalah vektor-vektor relevan, maka setelah dilakukan kernel trick, akan didapatkan data set baru seperti berikut:

$x_2 = \phi\binom{0}{2} = \binom{6-0+(0-2)^2}{6-2+(0-2)^2} = \binom{10}{8}$

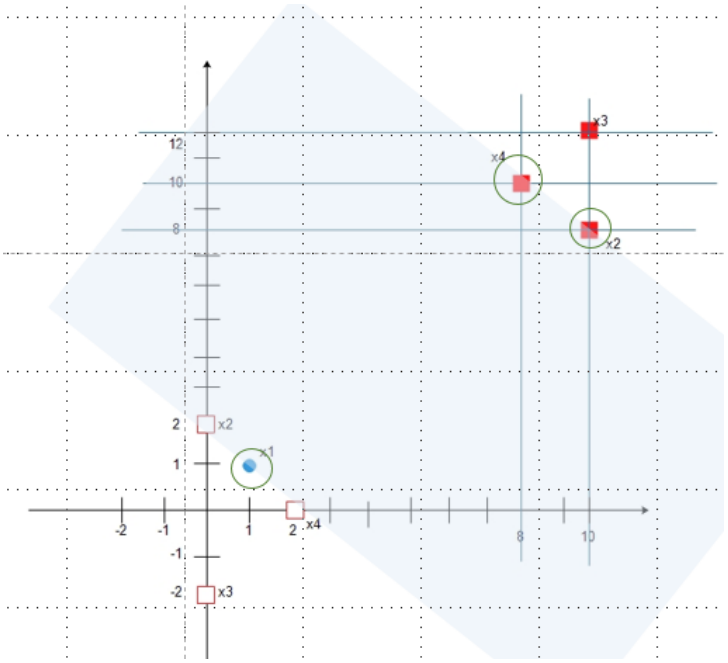
$$x_3 = \phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}6-0+(0-2)^2\\6-(-2)+(0-(-2))^2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$$

$$x_4 = \phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}6-2+(2-0)^2\\6-0+(2-0)^2\end{pmatrix} = \begin{pmatrix}8\\10\end{pmatrix}$$



Gambar III-15 data set baru

Setelah Data set terpisah secara linier, barulah kita dapat menentukan support vector pada data set tersebut:



Gambar III-16 support vector

$S_1 = \binom{8}{10}, S_2 = \binom{10}{8}, S_3 = \binom{1}{1}$

Setelah itu masing-masing *support vector* diberi nilai bias 1, menjadi:

$$S_1 = \begin{pmatrix} 8 \\ 10 \end{pmatrix} => \widetilde{S_1} = \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 10 \\ 8 \end{pmatrix} => \widetilde{S_2} = \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} => \widetilde{S_3} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Sekarang cari 3 parameter α_1, α_2 dan α_3 menggunakan persamaan linear:

$$\alpha_1\widetilde{S_1}.\widetilde{S_1} + \alpha_2\widetilde{S_2}.\widetilde{S_1} + \alpha_3\widetilde{S_3}.\widetilde{S_1} = +1\ (Relevan)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_2} + \alpha_2\widetilde{S_2}.\widetilde{S_2} + \alpha_3\widetilde{S_3}.\widetilde{S_2} = +1\ (Relevan)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_3} + \alpha_2\widetilde{S_2}.\widetilde{S_3} + \alpha_3\widetilde{S_3}.\widetilde{S_3} = -1\ (Tidak\ Relevan)$$

Subtitusikan menjadi:

$$\alpha1\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} = +1$$

$$\alpha1\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} = +1$$

$$\alpha1\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha2\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1$$

Setelah dikalikan

$$165\alpha1 + 161\alpha2 + 19\alpha3 = +1$$

$$161\alpha1 + 16152 + 19\alpha3 = +1$$

$$19\alpha1 + 19\alpha2 + 3\alpha3 = -1$$

Didapatlah:

α1 = 0.859, α2 = 0.859, α3 = -1.4219

Setelah didapatkan nilai α masukan persamaan:

$$\widetilde{W} = \sum_i \alpha_i \tilde{S}_i$$

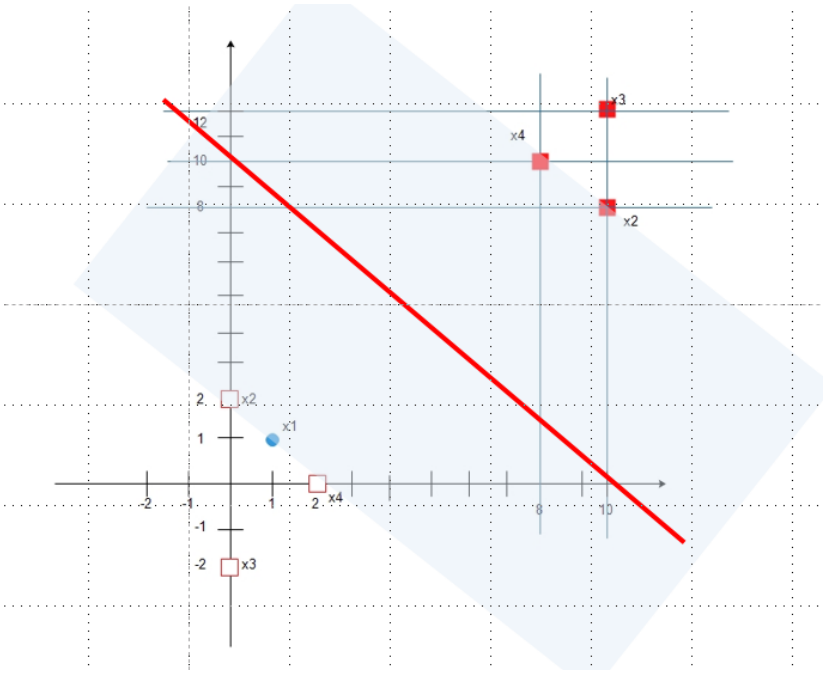
$$\widetilde{W} = \alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\widetilde{W} = (0.0859) \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + (0.0859) \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + (-1.4219) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.1243 \\ 0.1243 \\ -1.2501 \end{pmatrix}$$

y = w x + b dengan w = $\begin{pmatrix} 0.1243/0.1243 \\ 0.1243/01243 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Dan dengan offset b = $-\frac{1.2501}{0.1243} = -10.057$

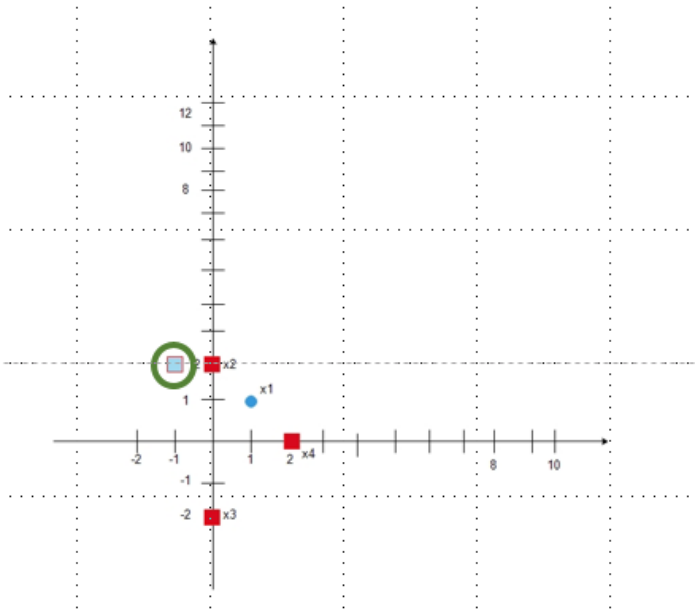
Didapatkan lah nilai hyperplane untuk mengklasifikasikan kedau kelas, yaitu 10.057



Gambar III-17 hyperplane

2. Menghitung Data Testing

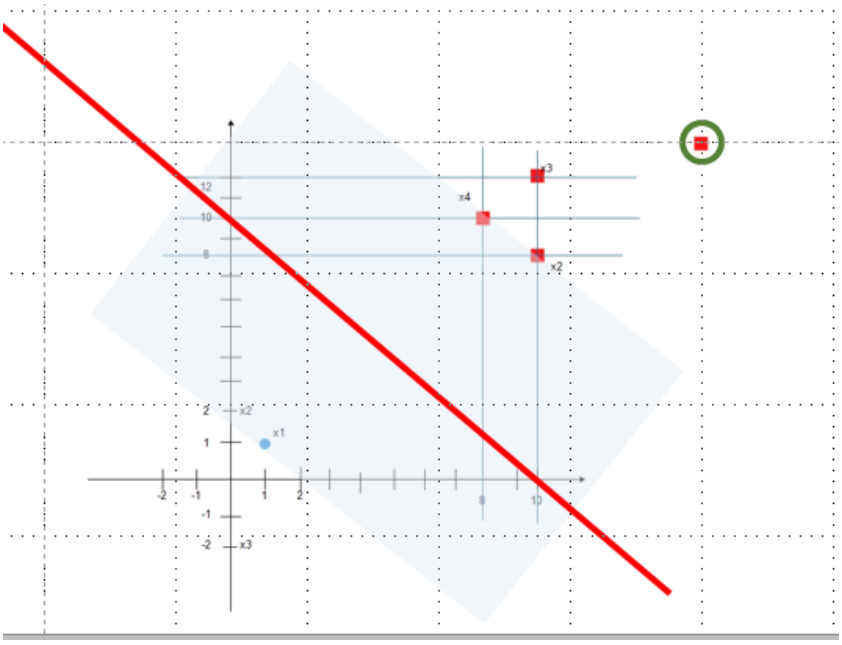
Setelah mendapatkan nilai hyperplane, selanjutnya kita dapat menentukan tweet yang belum memiliki kategori dengan nilai hyperplane tersebut. Misalnya kita memiliki data uji dengan vektor (-1,2).



Masukan vektor tersebut kedalam persamaan (**III.1**)

$$\phi\begin{pmatrix} x \\ y \end{pmatrix} = \phi\begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 6+1+(-1-2)^2 \\ 6-2+(-1-2)^2 \end{pmatrix} = \begin{pmatrix} 16 \\ 13 \end{pmatrix}$$

$$\text{w} \,.\, \phi(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 16 \\ 13 \end{pmatrix} = 29 > 10$$



Gambar III-18 hasil retrieval

Nilai vektor (-1,2) adalah 29 dimana 29 > 10. Berarti vektor tersebut masuk kedalam kategori relevan.

Didalam aplikasi yang dibuat untuk menghitung nilai-nilai di atas dalam penelitian ini menggunakan sebuah library bernama LibSVM yang di buat oleh IanBarber yang selanjutnya di modifikasi sesuai dengan kebutuhan. *Function* yang digunakan adalah sebagai berikut:

1. Function Train

Function ini berfungsi untuk membuat data latih pada metode SVM.

```
1   public function train($dataFile, $modelFile, $testDataFile =
2   null) {
3           $this->recordCount = $this->loadData($dataFile);
4
5           // if we're using the external dotcache, precalc it
6           if($this->useExternal) {
7             $this->precalculateDots();
8           }
9
10          // Initialise lagrange multipliers - there will be
11  one for each example
12          for($i = 0; $i < $this->recordCount; $i++) {
13            $this->lagrangeMults[$i] = 0;
14          }
15
16          // Loop over the records with examineExample, until
17  we're
18          // not seeing changes any more.
19          $numChanged = 0; $examined = 0;
20          while($numChanged > 0 || $examined == 0) {
21            $numChanged = 0;
22            if($examined == 0) {
23              for($i = 0; $i < $this->recordCount; $i++) {
24                $numChanged += $this->examineExample($i);
25              }
26            } else {
27              foreach($this->lagrangeMults as $id => $val) {
28                if($val != 0 && $val != self::UPPER_BOUND) {
29                  $numChanged += $this->examineExample($id);
30                }
31              }
32            }
33
34            if($examined == 0) {
35              $examined = 1;
36            } else if($numChanged == 0) {
37              $examined = 0;
38            }
39          }
40
41          if(isset($modelFile) && is_string($modelFile)) {
42            $this->writeSVM($modelFile);
43          }
44
45          // close the file if we're precalcing
```

```
46          if(isset($this->fh)) {
47            fclose($this->fh);
48            unset($this->fh);
49          }
50
51          if(isset($testDataFile) && is_string($testDataFile))
52  {
53            $this->test($testDataFile, null, null);
54          }
55        }
56
```

2. Function Predict

Function ini berfungsi untuk mendapatkan hasil klasifikasi dengan nilai -1 atau 1.

```
1   public function predict($dataFile, $modelFile = null,
2   $outputFile = null)
3   {
4     $right = $wrong = 0;
5   // If we're already in a valid SVM, we don't need the model
6     if($modelFile) {
7       $vectorCount = $this->readSVM($modelFile);
8         } else if(!count($this->lagrangeMults)) {
9           if(!$this->quiet) { echo 'No model supplied'; }
10          return false;
11        } else {
12          $vectorCount= $this->recordCount;
13        }
14
15        $this->recordCount    =    $this->loadData($dataFile,
16  $vectorCount);
17        if($outputFile) {
18          $fh = fopen($outputFile, 'w');
19        }
20
21        if($this->useExternal) {
22          $this->precalculateDots();
23        }
24
25        // classify the example
26        for($i = $vectorCount; $i < $this->recordCount; $i++)
27  {
28          $classification = $this->classify($i);
29          if($outputFile) {
30            fwrite($fh, $classification . "\n");
31          }
32          if(isset($this->targets[$i]) && $this->targets[$i]
33  != 0) {
34            if(($this->targets[$i] > 0) == ($classification
35  > 0)) {
36              $right++;
37            } else {
38              $wrong++;
```

```
39              }
40            }
41          }
42          if($outputFile) {
43            fclose($fh);
44          }
45          // close the file if we're precalcing
46          if(isset($this->fh)) {
47            fclose($this->fh);
48            unset($this->fh);
49          }
50
51          if(($right + $wrong) > 0 && !$this->quiet) {
52            echo "\nAccuracy: " . ($right/($right+$wrong)) . "
53  over " . ($right+$wrong) . " examples.\n";
54          }
55      }
```

Setelah di modifikasi kedua fungsi di atas diubah menjadi:

1. Function Train

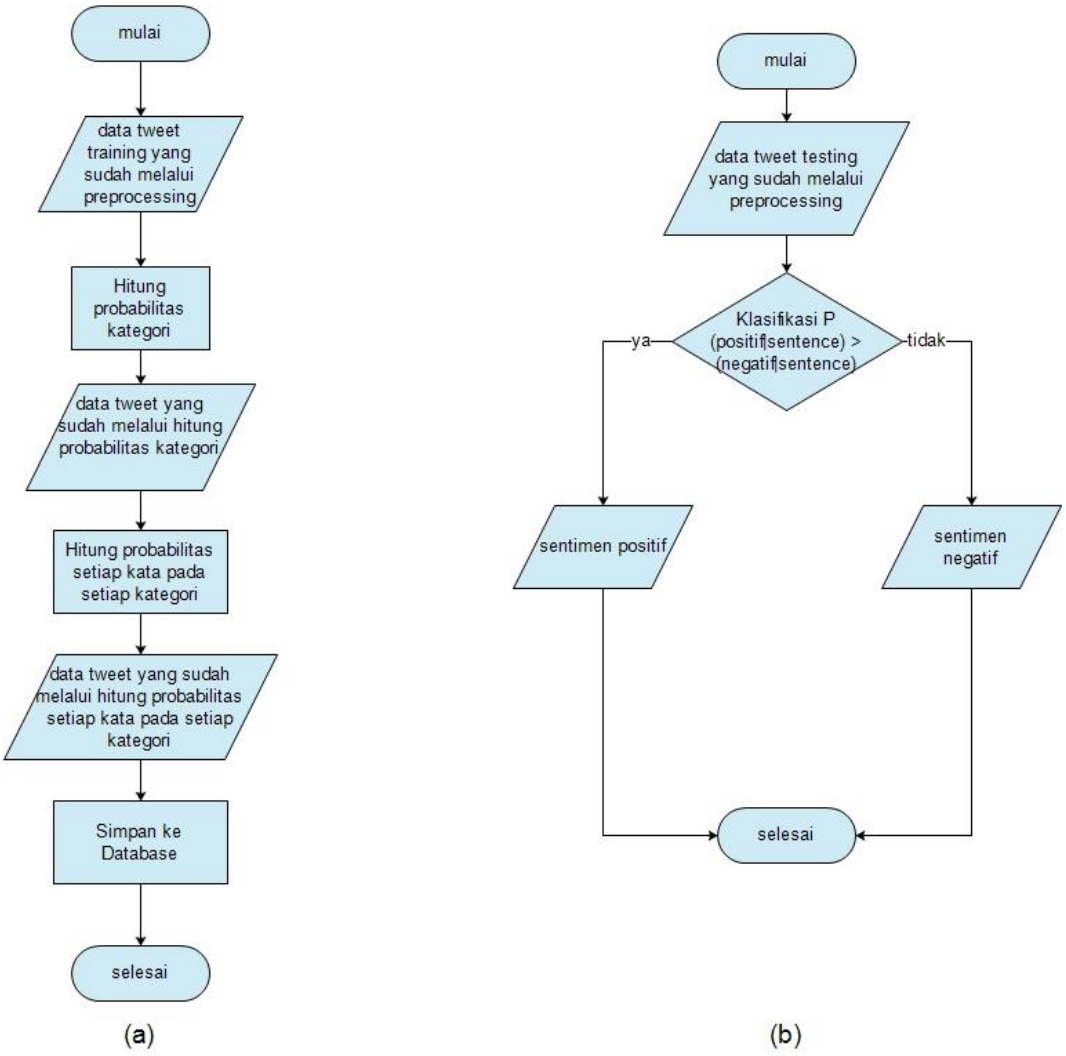
```
1   public function train($data) {
2           $this->recordCount = $this->loadData($data);
3
4
5           if($this->useExternal) {
6             $this->precalculateDots();
7           }
8
9           // Inisialisasi lagrange multipliers
10          for($i = 0; $i < $this->recordCount; $i++) {
11            $this->lagrangeMults[$i] = 0;
12          }
13
14          // Looping records dengan examineExample
15          $numChanged = 0; $examined = 0;
16          while($numChanged > 0 || $examined == 0) {
17            $numChanged = 0;
18            if($examined == 0) {
19              for($i = 0; $i < $this->recordCount; $i++) {
20                $numChanged += $this->examineExample($i);
21              }
22            } else {
23              foreach($this->lagrangeMults as $id => $val) {
24                if($val != 0 && $val != self::UPPER_BOUND) {
25                  $numChanged += $this->examineExample($id);
26                }
27              }
28            }
29
30            if($examined == 0) {
31              $examined = 1;
32            } else if($numChanged == 0) {
33              $examined = 0;
```

```
34              }
35          }
36
37      }
```

2. Function Predict

```
1   public function predict($data) {
2           $right = $wrong = 0;
3
4           if($data) {
5              $vectorCount = $this->readSVM($modelFile);
6           } else if(!count($this->lagrangeMults))
7           else {
8              $vectorCount= $this->recordCount;
9           }
10
11    $this->recordCount=$this->loadData($data, $vectorCount);
12
13          // Klasifikasian
14       for($i = $vectorCount; $i < $this->recordCount; $i++){
15          $classification = $this->classify($i);
16          }
17          return $classification;
18      }
```

Setelah dilakukan pemisahan data testing antara dokumen relevan dengan dokumen tidak relevan. Maka didapatlah sekumpulan data relevan dari hasil retrievel menggunakan SVM ini kemudian data tersebut diklasifikasi menggunakan metode Naïve Bayes. Berikut adalah *flowchart* pada proses klasifikasi menggunakan metode Naïve Bayes dapat dilihat pada Gambar III-19



Gambar III-19 flowchart klasifikasi Naive Bayes (a) Proses Training (b) Proses Testing

1. Menghitung Data Training

Tabel III-5 Tabel contoh Data Training

Tweet	Kategori	Fitur
Tweet1	negatif	maicih level pedasss gakuat makan emotsedih
Tweet2	Positif	kabar gembira maicih level emotsenang
Tweet3	Positif	maicih kayak enak yes emotsenang emotsenang

Dibentuk sebuah model probabilistik, dengan mengacu pada persamaan (**II.18**) dan persamaan (**II.19**):

$$P(w_{maicih}|c_{positif}) = \frac{\#words(w_{maicih}, c_{positif}) + 1}{\#words(c_{positif}) + |V|} = \frac{2+1}{11+12} = \frac{3}{23}$$

$$P(w_{maicih}|c_{negatif}) = \frac{\#words(w_{bagus}, c_{positif}) + 1}{\#words(c_{positif}) + |V|} = \frac{1+1}{6+12} = \frac{2}{18}$$

Jika dibentuk sebuah tabel, maka hasil dari perhitungan probabilistik setiap kata pada data training terdapat pada Tabel III-6 dan Tabel III-7

Tabel III-6 Hasil perhitungan probabilitas data training (1)

Kategori	$P(c)$	$P(w_{kj}\|$ c)							
		maicih	level	pedasss	gakuat	makan	emotsedih	kabar	gembira
Positif	$\frac{2}{3}$	$\frac{3}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{2}{23}$	$\frac{2}{23}$
Negatif	$\frac{1}{3}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{1}{18}$	$\frac{1}{18}$

Tabel III-7 hasil perhitungan probabilitas data training (2)

Kategori	$P(c)$	$P(w_{kj}\|$ c)			
		emotsenang	kayak	enak	yes
Positif	$\frac{2}{3}$	$\frac{3}{23}$	$\frac{2}{23}$	$\frac{2}{23}$	$\frac{2}{23}$
Negatif	$\frac{1}{3}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$

Hasil perhitungan probabilitas tersebut digunakan sebagai model probabilistik yang nantinya digunakan sebagai penentuan kategori data testing. Contoh kasus data testing yang sudah melalui tahap *preprocessing* adalah sebagai berikut:

Tweet1 = pedasss emotsenang emotsenang

Tweet2 = gakuat makan maicih emotsedih

2. Menghitung Data Testing

Gunakan persamaan (**II.16**) untuk mengetahui kedalam kategori apa kedua data testing tersebut masuk.

$$P(\text{Tweet1}|C_{positif})$$

$$= P(w_{pedasss}|C_{positif}) \times P(w_{esenang}|C_{positif}) \times P(w_{esenang}|C_{positif})$$

$$= \frac{1}{23} \times \frac{3}{23} \times \frac{3}{23} \times \frac{2}{3}$$

$$= 0.00004931371$$

$$P(\text{Tweet1}|C_{negatif})$$

$$= P(w_{pedasss}|C_{negatif}) \times P(w_{esenang}|C_{negatif}) \times P(w_{esenang}|C_{negatif})$$
$$= \frac{2}{18} \times \frac{1}{18} \times \frac{1}{18} \times \frac{1}{3}$$
$$= 0.00001143118$$

Sedangkan untuk perhitungan tweet2 adalah sebagai berikut:

$$P(\text{Tweet1}|C_{positif})$$
$$= P(w_{gakuat}|C_{positif}) \times P(w_{makan}|C_{positif}) \times P(w_{maicih}|C_{positif}) \times P(w_{emotsedih}|C_{positif})$$
$$= \frac{1}{23} \times \frac{1}{23} \times \frac{3}{23} \times \frac{1}{23} \times \frac{2}{3}$$
$$= 0.00000071469$$

$$P(\text{Tweet1}|C_{negatif})$$
$$= P(w_{gakuat}|C_{negatif}) \times P(w_{makan}|C_{negatif}) \times P(w_{maicih}|C_{negatif}) \times P(w_{emotsedih}|C_{negatif})$$
$$= \frac{2}{18} \times \frac{2}{18} \times \frac{2}{18} \times \frac{2}{18} \times \frac{1}{3}$$
$$= 0.00000508052$$

Setelah menghitung probabilitas dari setiap tweet yang digunakan sebagai data testing didapatkan hasil yang terdapat pada Tabel III-8

Tabel III-8 Nilai probabilitas pada data testing

Tweet	Positif	Negatif
Tweet1	**0.00004931371**	0.00001143118
Tweet2	0.00000071469	**0.00000508052**

Dapat disimpulkan bawah tweet1 masuk kedalam kategori positif dikarenakan nilai probabilitas positifnya lebih besar, sedangkan tweet2 masuk kedalam kategori negatif karena porbabilitas negatif nya lebih besar.

III.1.6 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak yang dibangun dibagi menjadi dua kebutuhan, yaitu kebutuhan nonfungsional dan fungsional. Spesifikasi kebutuhan

nonfungsional dapat dilihat pada Tabel III-9 sedangkan spesifikasi kebutuhan fungsional dapat dilihat pada Tabel III-10.

Tabel III-9 Spesifikasi Kebutuhan Non Fungsional

Nomor	Spesifikasi Kebutuhan NonFungsional
SKPL-NF-001	Sistem yang dibangun berbasis web.
SKPL-NF-002	Aplikasi yang dibangun dapat berjalan optimal di browser google chrome dan mozilla firefox.
SKPL-NF-003	Aplikasi ini dibangun dengan spesifikasi hardware yang memenuhi standar minimum kebutuhan.
SKPL-NF-004	Aplikasi ini dapat berjalan jika ada koneksi internet.
SKPL-NF-005	Pengguna yang akan menggunakan aplikasi ini harus memenuhi lima kategori yaitu *user knowledge and experience*, *user jobs and tasks*, *user physical characteristic* dan *user tools*.
SKPL-NF-006	Data yang dipakai untuk di klasifikasi adalah data yang diambil langsung dari twitter.

Tabel III-10 Spesifikasi Kebuthan Fungsional

Nomor	Spesifikasi Kebutuhan Fungsional
SKPL-F-001	Aplikasi ini dapat membantu Mangerial melakukan crawling data tweet yang didapatkan langsung dari twitter menggunakan API yang disediakan twitter.
SKPL-F-002	Aplikasi ini dapat melakukan preprocessing untuk mengubah data mentah menjadi data siap pakai.
SKPL-F-003	Aplikasi ini dapat membantu Mangerial melakukan retrieval terhadap tweet yang di crawling. Sehingga data yang diklasifikasi benar-benar tweet yang mengandung sentimen.
SKPL-F-004	Aplikasi ini dapat membantu Mangerial melakukan klasifikasi terhadap sentimen yang diberikan konsumen
SKPL-F-005	Aplikasi ini dapat melakukan pengujian untuk menguji akurasi klasifikasi.
SKPL-F-006	Aplikasi ini dapat membantu Mangerial melihat hasil klasifikasi yang di visualisasikan berupa grafik

III.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan *Non Fungsional* menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan perangkat lunak yang dibangun. Analisis kebutuhan non fungsional yang dilakukan diantaranya Analisis Kebutuhan Perangkat Lunak (*Software*), Analisis Kebutuhan Perangkat Keras (*Hardware*) dam Analisis Kebutuhan Perangkat Pikir (*Brainware*).

III.1.7.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak (*software*) pada penelitian ini merupakan kebutuhan perangkat lunak yang digunakan dalam pembangunan perangkat lunak ini. Adapun perangkat lunak yang digunakan sebagai berikut:

1. Sistem Operasi Window 8.1 Profesional 64 bit.
2. Bahasa Pemrograman PHP
3. Web server XAMPP v3.2.1
4. *Code Editor* SublimeText 3
5. DBMS : MongoDB
6. Adobe Photoshop CS6
7. Web Browser Google Chrome 38.0.2125.101 m

III.1.7.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras (*hardware*) pada penelitian ini merupakan kebutuhan perangkat keras yang digunakan dalam pembangunan perangkat lunak ini. Adapun perangkat keras yang digunakan sebagai berikut:

1. *Processor* intel i3-2305 2.3GHz
2. Hardisk 340GB
3. Memory 6GB (RAM)
4. Monitor dengan resolusi 1366x768 *pixels*
5. QWERTY Creative
6. Mouse Wireless Logitec
7. VGA intel(R) HD graphics 3000 1696MB

III.1.7.3 Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir (*brainware*) yang dilakukan pada penelitian ini dikelompokkan menjadi empat kategori yaitu *user knowledge and experience, user jobs and tasks, user physical characteristic* dan *user tools*. Perangkat pikir yang terlibat dalam pembangunan perangkat lunak pada penelitian ini adalah bagian managerial dari PT. Maicih Inti Sinergi.

1. Pengetahuan dan Pengalaman Pengguna (*User Knowledge and Experience*)

Pengetahuan dan pengalaman pengguna yang di targetkan dalam pembangunan perangkat lunak ini dapat dilihat pada Tabel III-11

Tabel III-11 Pengetahuan dan Pengalaman Pengguna

Aspek	Pengguna
Education Level	SMA sampai Perguruan tinggai (D3,S1,S2,S3)
Reading Level	Sedang Sampai Tinggi
Computer Literacy	*moderate experience till highly technical*
Task Experience	Rendah sampai Tinggi
System Experience	Rendah sampai Tinggi
Application Experience	Rendah sampai Tinggi
Use Of Other System	No
Native Language or Culture	Dominan Indonesia

2. Tugas dan Kebutuhan Pengguna (*User Jobs and Tasks*)

Tugas dan pekerjaan pengguna yang ditargetkan dalam pembangunan perangkat lunak ini dapat dilihat pada Tabel III-12

Tabel III-12 Tugas dan Kebutuhan Pengguna

Aspek	Pengguna
Frequency Of Use	*Continual*
Primary Training	Disediakan menu petunjuk
Task or Need Importance	Sedang sampai Tinggi
Task Structure	Sedang sampai Tinggi
Job Categories	No
Life Style	No

3. Karakteristik Pengguna (*User Physical Characteristic*)

Karakter fisik pengguna yang ditargetkan dalam pembangunan perangkat lunak ini dapat dilihat pada Tabel III-13

Tabel III-13 Karakteristik Pengguna

Aspek	Pengguna
Handedness	Keduanyadalam kondisi baik
Gender	Laki-laki dan perempuan
Age	20 tahun ke atas
Disabilities	Tidak cacat tubuh

4. *User Tools*

Ketika pengguna menggunakan perangkat lunak ini peralatan yang dominan digunakan adalah mouse.

III.1.8 Analisis Kebutuhan Fungsional

Analisis Kebutuhan Fungsional pada penelitian ini menggunakan pendekatan *Object Oriented Analysis and Design* (OOAD). Diagram yang akan digambarkan pada bagian ini adalah *Use Case Diagram*, *activity diagram*, *Class Diagram* dan *Sequence Diagram*.

1. *Use Case Diagram*

Use Case Diagram dari pembangunan perangkat lunak ini dapat dilihat pada Gambar III-20.

Definisi aktor dari pembangunan perangkat lunak ini dapat dilihat pada Tabel III-14

Tabel III-14 Definisi Aktor

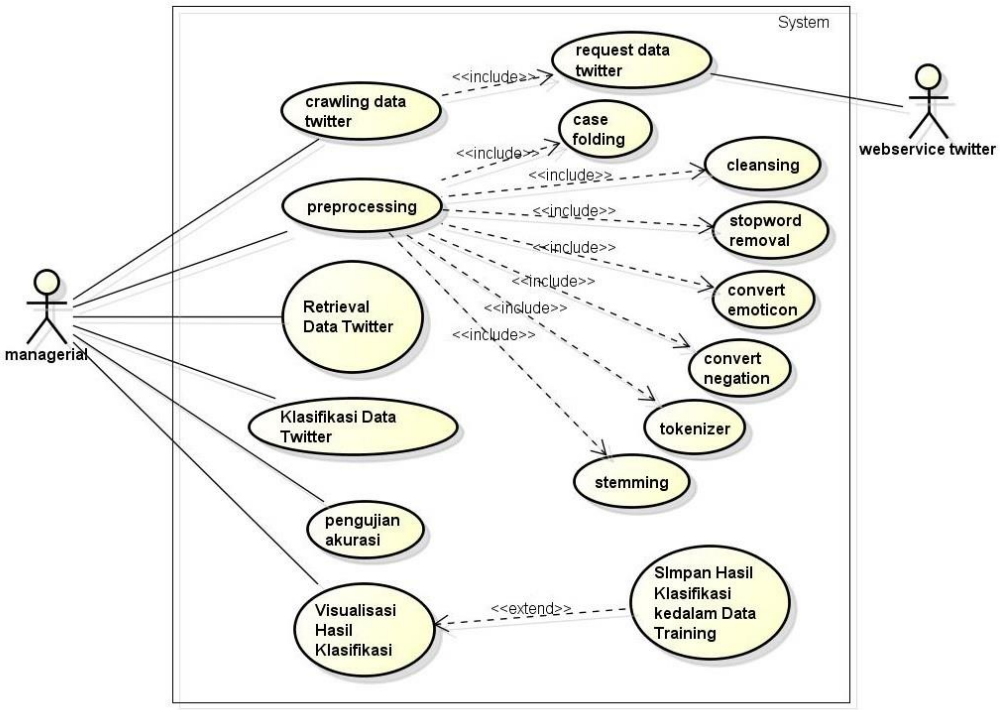
No	Aktor	Deskripsi
1	Mangerial	Managerial dapat melakukan crawling data twitter, preprocessing, Retrieve Data, klasifikasi, menguji akurasi dan melihat hasil klasifikasi.
2	Webservice Twitter	Webservice twitter dapat melakukan pengiriman data twitter.

Definisi *Use Case* dari pembangunan perangkat lunak ini dapat dilihat pada Tabel III-15

Tabel III-15 Definisi Use Case

No	Usecase	Deskripsi
1	Crawling Data Twitter	Fungsionalitas ini digunakan oleh Mangerial untuk mengambil data mentah langsung dari twitter.
2	Preprocessing	Fungsionalitas ini digunakan oleh Managerial untuk merubah data mentah twitter menjadi layak untuk di klasifikasi
3	Retrieve Data Twitter	Fungsionalitas ini digunakan oleh Managerial untuk meretrieve data twitter yang layak untuk diklasifikasi.
4	Klasifikasi Data Twitter	Fungsionalitas ini digunakan oleh Managerial untuk mengklasifikasi data tweet
5	Pengujian Akurasi	Fungsionalitas ini digunakan oleh Managerial untuk menguji akurasi klasifikasi yang dilakukan
6	Visualisasi Hasil Klasifikasi	Fungsionalitas ini digunakan oleh Managerial untuk melihat hasil klasifikasi
7	Case Folding	Fungsionalitas ini digunakan oleh Managerial untuk merubah tweet yang masih memiliki huruf kapital menjadi huruf kecil

No	Usecase	Deskripsi
8	Cleansing	Fungsionalitas ini digunakan oleh Managerial untuk menghapus atribut-atribut twitter seperti hashtag, username dll
9	Stopword Removal	Fungsionalitas ini digunakan oleh Managerial untuk menghilangka kata-kata yang masuk kedalam stoplist.
10	Convert Emoticon	Fungsionalitas ini digunakan oleh Managerial untuk merubah emoticon menjadi sebuah kata yang sudah didefinisikan sebelumnya
11	Convert Negation	Fungsionalitas ini digunakan oleh Managerial untuk merubah kata negasi menjadi sebuah kata yang sudah didefinisikan sebelumnya
12	Tokenizing	Fungsionalitas ini digunakan oleh Managerial untuk memisahkan setiap kata.
13	Stemming	Fungsionalitas ini digunakan oleh Managerial untuk merubah kata berimbuhan kedalam kata dasar.
14	Request Data Twitter	Fungsional ini digunakan oleh Webservice Twitter untuk mengirim data twitter.
15	Simpan Hasil Klasifikasi kedalam Data Training	Fungsional ini digunakan oleh Managerial untuk menambahkan data hasil klasifikasi kedalam data training.



Gambar III-20 usecase diagram

2. *Usecase Skenario*

Pada bagian ini akan di isi dengan skenario untuk tiap *use case* yang menggambarkan urutan interaksi *actor* dengan *use case* tersebut dari awal sampai akhir.

Tabel III-16 usecase scenario crawling data tweet

Use Case Name	Crawling Data Tweet	
Related Requirement	Requirement SKPL-F-001	
Goal In Context	Mendapatkan data dari hasil crawling *tweet* pada *tweeter*	
Description	Fungsionalitas ini digunakan oleh Managerial untuk mendapatkan data testing yang akan di klasifikasi	
Related Use Case	-	
Successful End Condition	Data hasil *crawling* tersimpan di *database*	
Failed End Condition	Data hasil *crawling* tidak tersimpan di *database* (*error* saat melakukan *crawling*)	
Actors	Managerial	
Trigger	Menekan tombol menu *crawling data*.	
Included Cases	**Request data twitter**	
Main Flow	Step	Action
	1.	Managerial menekan tombol menu *crawling*
	2. **Include::Request data twitter**	Sistem melakukan *crawling data*
	3.	Data hasil *crawling* tersimpan di *database*
Extension	Step	Branching Action
	3.1	Data hasil *crawling* tidak tersimpan di *database* (*error* saat melakukan *crawling*)

Tabel III-17 usecase skenario request data twitter

Use Case Name	*Request Data Twitter*	
Related Requirement	Requirement SKPL-F-001	
Goal In Context	Mengirim data *twitter* dari *server*	
Description	Fungsionalitas ini digunakan oleh *Webservice Twitter* Untuk mengirimkan *data twitter*	
Related Use Case	Crawling data tweet	
Successful End Condition	Data *twitter* berhasil dikirim	
Failed End Condition	Data *twitter* gagal dikirim	
Actors	*Webservice* Twitter	
Trigger	*Credential* dari *crawling* data sesuai dengan yang terdapat pada *webservice twitter*	
Main Flow	Step	Action
	1.	*Credential* dari *crawling* data sesuai dengan yang terdapat pada *webservice twitter*.
	2.	*Credential* crawling data di verifikasi didatabase
	3.	Data twitter yang di minta di kirim.
Extension	Step	Branching Action
	1.1	*Credential* dari *crawling* data tidak sesuai.

Tabel III-18 usecase skenario preprocessing

Use Case Name	Preprocessing	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses	
Description	Fungsionalitas ini digunakan oleh Managerial untuk memproses data hasil *crawling*	
Related Use Case	-	
Successful End Condition	Data hasil *crawling* berhasil di preproses	
Failed End Condition	Data hasil *crawling* gagal di preproses	
Actors	Managerial	
Trigger	Menekan tombol menu *preproses*	
Included Cases	**Case Folding, Cleansing, Stopword removal, convert emoticon, convert negation, tokenizer, stemming**	
Main Flow	Step	Action
	1.	Managerial menekan tombol *preproses*
	2. Include:: Case Folding	Mengubah huruf kapital menjadi huruf kecil
	3. Include:: Cleansing	Membersihkan tweet dari hashtag,username dan lain-lain (atribut bawaan twitter)
	4. Include:: Stopword removal	Menghilangkan kata yang terdapat pada kamus stoplist
	5. Include:: convert emoticon	Merubah karakter spesial yang sering digunakan sebagai emoticon kedalam kata yang sudah didefinisikan
	6. Include:: convert negation	Menggabungkan kalimat negasi sehingga menjadi kata baru.
	7. Include:: tokenizer	Memisahkan kalimat kedalam token kata
	8. Include:: stemming	Merubah kata berimbuhan menjadi kata dasar
	3.	Data tweet hasil *crawling* berhasil di preproses
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal di preproses.

Tabel III-19 usecase skenario case folding

Use Case Name	Case folding
Related Requirement	Requirement SKPL-F-002
Goal In Context	Data hasil crawling di preproses pada bagian case folding

Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui proses case folding	
Failed End Condition	Data hasil *crawling* gagal melalui proses case folding	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing case folding*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing case folding*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing case folding*

Tabel III-20 usecase skenario cleansing

Use Case Name	Cleansing	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *cleansing*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses cleansing*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses cleansing*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing cleansing*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing cleansing*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing cleansing*

Tabel III-21 usecase skenario stopword removal

Use Case Name	*stopword removal*	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *stopword removal*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses stopword removal*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses stopword removal*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing stopword removal*

	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing stopword removal*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing stopword removal*

Tabel III-22 usecase skenario convert emoticon

Use Case Name	Convert emoticon	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *convert emoticon*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses convert emoticon*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses convert emoticon*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing convert emoticon*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing convert emoticon*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing convert emoticon*

Tabel III-23 usecase skenario convert negation

Use Case Name	Convert Negation	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *convert negation*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses convert negation*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses convert negation*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing convert negation*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing convert negation*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing convert negation*

Tabel III-24 usecase skenario tokenizer

Use Case Name	Tokenizing	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *tokenizing*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses tokenizing*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses tokenizing*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing tokenizing*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing tokenizing*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing tokenizing*

Tabel III-25 usecase skenario stemming

Use Case Name	Stemming	
Related Requirement	Requirement SKPL-F-002	
Goal In Context	Data hasil crawling di preproses pada bagian *stemming*	
Related Use Case	Preprocessing	
Successful End Condition	Data hasil *crawling* berhasil melalui *proses stemming*	
Failed End Condition	Data hasil *crawling* gagal melalui *proses stemming*	
Actors	Managerial	
Trigger	Terdapat data crawling yang akan di preproses	
Main Flow	Step	Action
	1.	Data Crawling masuk ke bagian preprocessing
	2.	Sistem melakukan *preprocessing stemming*
	3.	Data tweet hasil *crawling* berhasil melewati proses *preprocessing stemming*
Extension	Step	Branching Action
	3.1	Data tweet hasil *crawling* gagal melewati proses *preprocessing stemming*

Tabel III-26 usecase skenario penerapan metode information retrieval

Use Case Name	Penerapan Metode *Information Retrieval*
Related Requirement	Requirement SKPL-F-003
Goal In Context	Data hasil *Preprocessing* berhasil di *retrive*
Description	Fungsionalitas ini digunakan oleh Managerial untuk me *retrieve* data hasil *preprocessing*
Related Use Case	-
Successful End Condition	Data hasil *preprocessing* berhasil di retrieve
Failed End Condition	Data hasil *preprocessing* gagal di retrieve
Actors	Managerial

Trigger	Menekan tombol *Retrieve Data*.	
Main Flow	Step	Action
	1.	Managerial menekan tombol *Retrieve Data*
	2.	Sistem melakukan *Retrieval* terhadap data tweet
	3.	Data berhasil di *Retrieve*
Extension	Step	Branching Action
	3.1	Data gagal di *Retrieve*

Tabel III-27 usecase skenario penerapan metode extraction

Use Case Name	Penerapan Metode *Extraction*	
Related Requirement	Requirement SKPL-F-004	
Goal In Context	Data hasil Retrieval berhasil di ekstrak	
Description	Fungsionalitas ini digunakan oleh Managerial untuk mengekstrak data hasil retrieval	
Related Use Case	-	
Successful End Condition	Data hasil retrieval berhasil di ekstrak	
Failed End Condition	Data hasil retrieval gagal di ekstrak	
Actors	Managerial	
Trigger	Menekan tombol Klasifikasi *Data*.	
Main Flow	Step	Action
	1.	Managerial menekan tombol Klasifikasi *Data*
	2.	Sistem melakukan Klasifikasi terhadap data tweet
	3.	Data berhasil di Klasifikasi
Extension	Step	Branching Action
	3.1	Data gagal di Klasifikasi

Tabel III-28 usecase skenario pengujian akurasi

Use Case Name	Pengujian Akurasi	
Related Requirement	Requirement SKPL-F-005	
Goal In Context	Data hasil Klasifikasi berhasil di uji dengan tingkat akurasi yang besar	
Description	Fungsionalitas ini digunakan oleh Managerial untuk mengetahui tingkat akurasi hasil klasfikasi	
Related Use Case	-	
Successful End Condition	Data hasil klasifikasi berhasil di uji.	
Failed End Condition	Data hasil klasifikasi gagal di uji	
Actors	Managerial	
Trigger	Menekan tombol Pengujian	
Main Flow	Step	Action
	1.	Managerial menekan tombol pengujian
	2.	Sistem melakukan pengujian terhadap data tweet yang sudah di klasifikasi
	3.	Data berhasil di uji
Extension	Step	Branching Action

Tabel III-29 usecase Penyajian Hasil klasifikasi

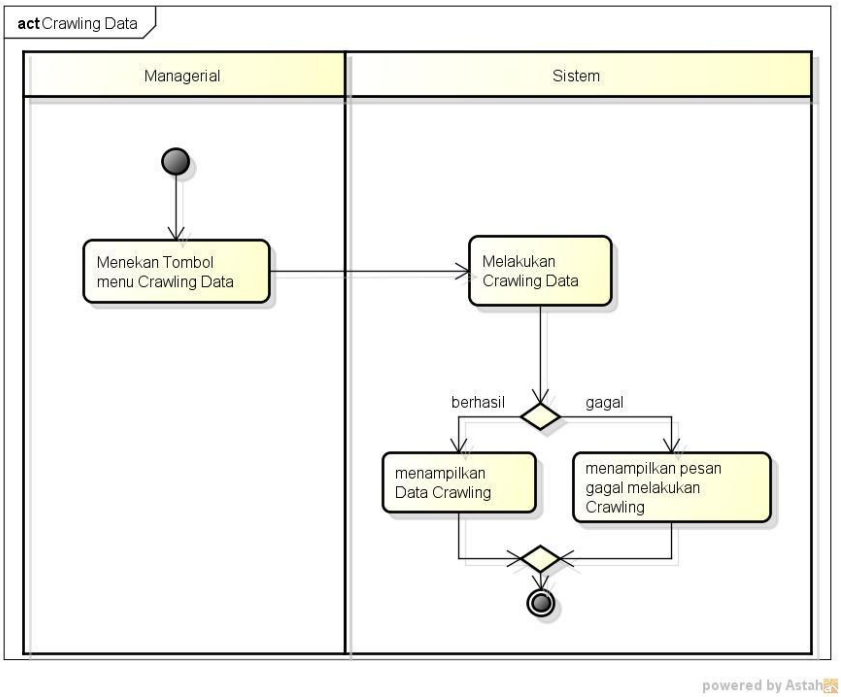
Use Case Name	Visualisasi Hasil klasifikasi	
Goal In Context	Melihat Data tweet yang sudah diklasifikasi	
Related Requirement	Requirement SKPL-F-006	
Description	Fungsionalitas ini digunakan oleh Managerial untuk melihat hasil klasifikasi	
Related Use Case	-	
Successful End Condition	Sistem berhasil menampilkan data hasil klasifikasi	
Failed End Condition	Sistem gagal menampilkan data hasil klasifikasi	
Actors	Managerial	
Trigger	Menekan tombol hasil Klasifikasi	
Main Flow	Step	Action
	1.	Managerial menekan tombol hasil Klasifikasi
	2.	Sistem Menampilkan Data hasil klasifikasi
Extension	Step	Branching Action

Tabel III-30 use case skenario Simpan Hasil Klasifikasi ke Data Training

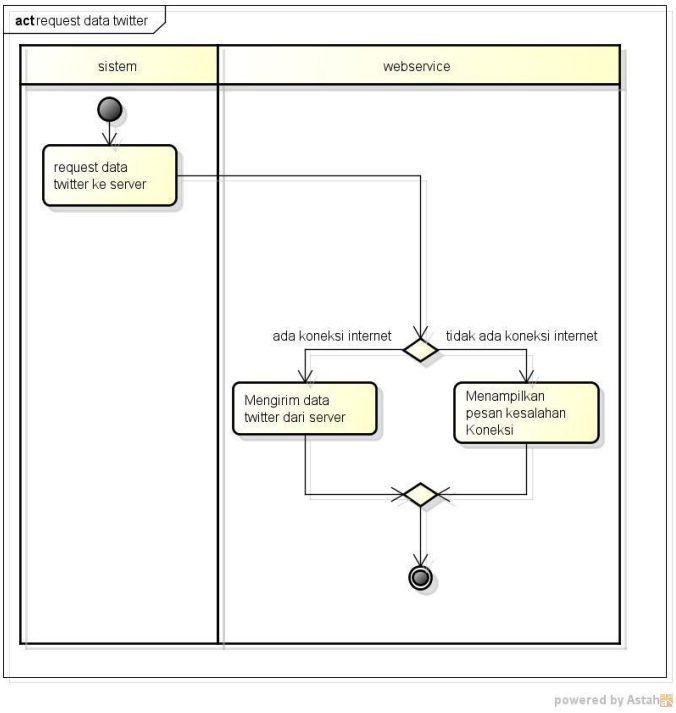
Use Case Name	Simpan Hasil Klasifikasi ke dalam Data Training	
Goal In Context	Hasil Klasifikasi masuk ke dalam Data training untuk menambah Data Training	
Related Requirement	Requirement SKPL-F-006	
Description	Fungsionalitas ini digunakan oleh Managerial untuk menambahkan data hasil klasifikasi kedalam data training	
Related Use Case	-	
Successful End Condition	Sistem berhasil memasukkan data hasil klasifikasi kedalam data training	
Failed End Condition	Sistem gagal menyimpan data hasil klasifikasi kedalam data training	
Actors	Managerial	
Trigger	Menekan tombol simpan hasil Klasifikasi	
Main Flow	Step	Action
	1.	Managerial menekan tombol simpan hasil Klasifikasi
	2.	Sistem Menampilkan Berhasil menyimpan Data hasil klasifikasi
Extension	Step	Branching Action
	2.1	Sistem Menampilkan Gagal Menyimpan Data Hasil Klasifikasi

3. Activity Diagram

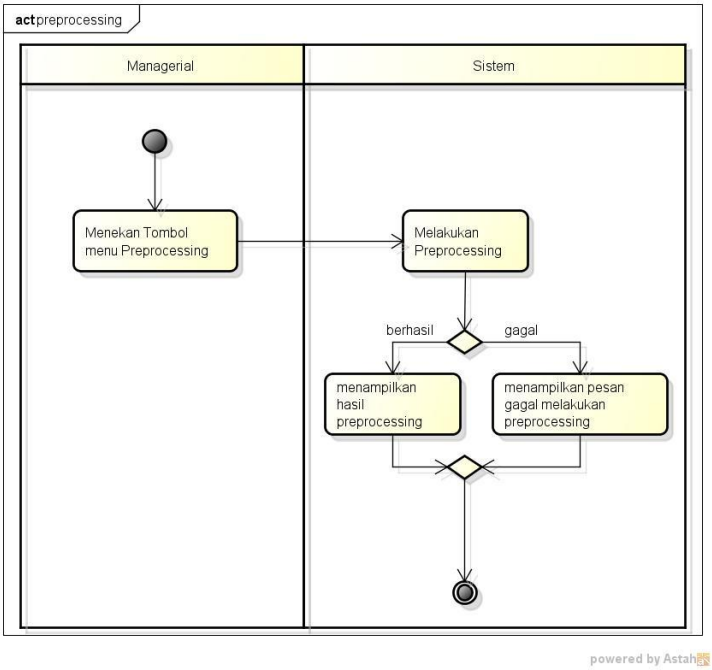
Activity Diagram dari pembangunan perangkat lunak pada penelitian ini adalah sebagai berikut:



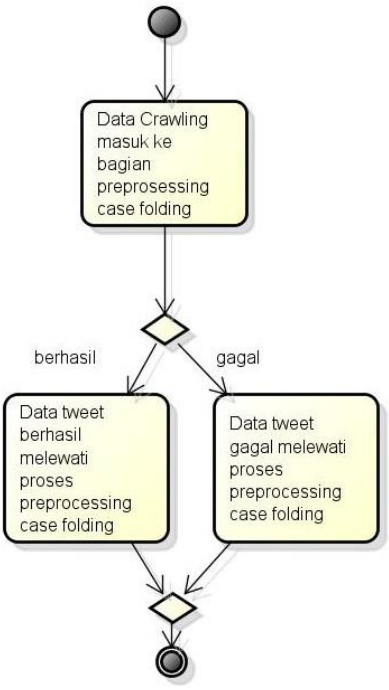
Gambar III-21 Acitivity diagram crawling data



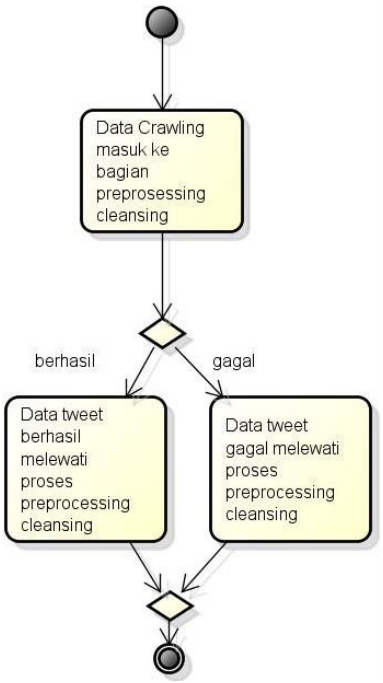
Gambar III-22 activity diagram request data twitter



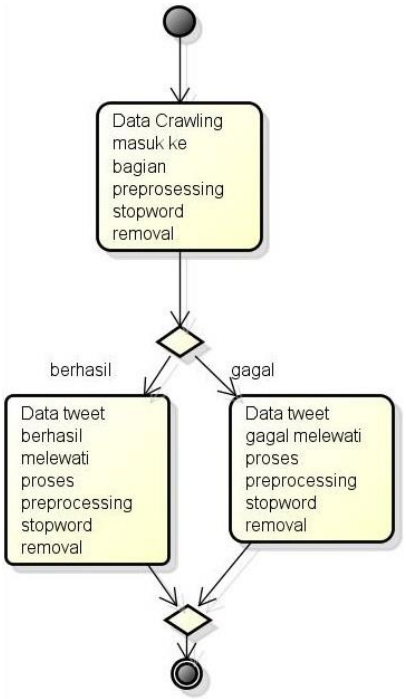
Gambar III-23 activity diagram preprocessing



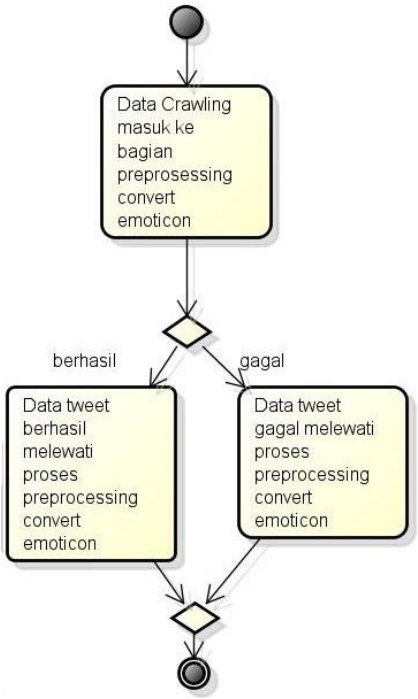
Gambar III-24 activity diagram case folding



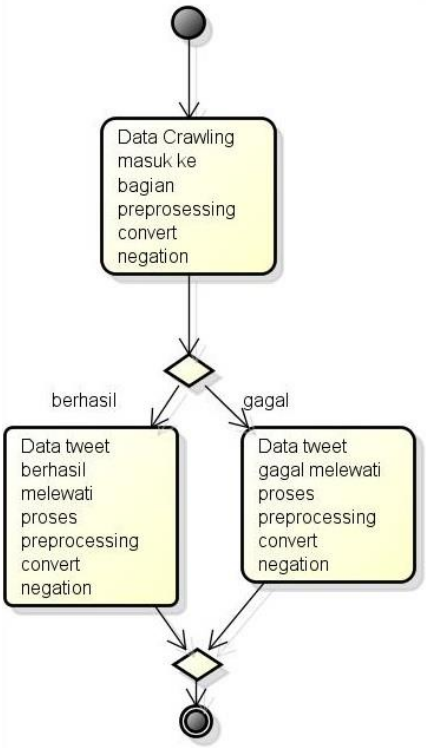
Gambar III-25 activity diagram cleansing



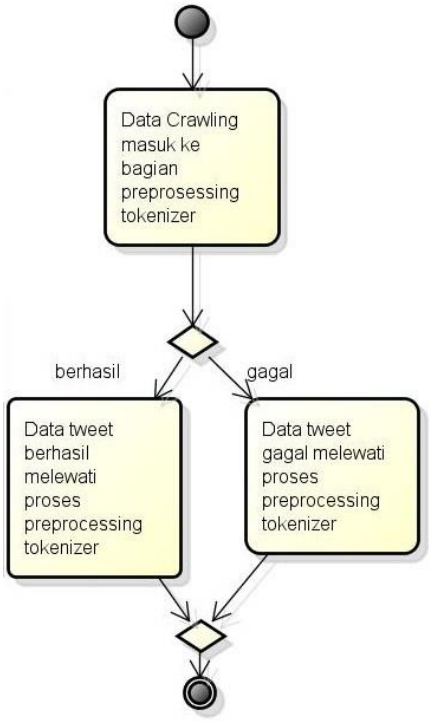
Gambar III-26 activity diagram stopword removal



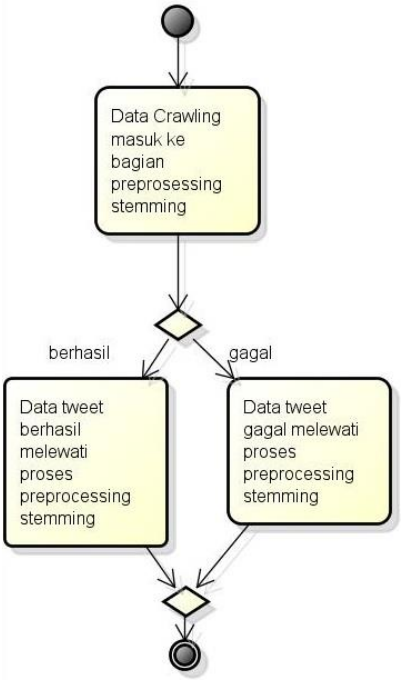
Gambar III-27 activity diagram convert emoticon



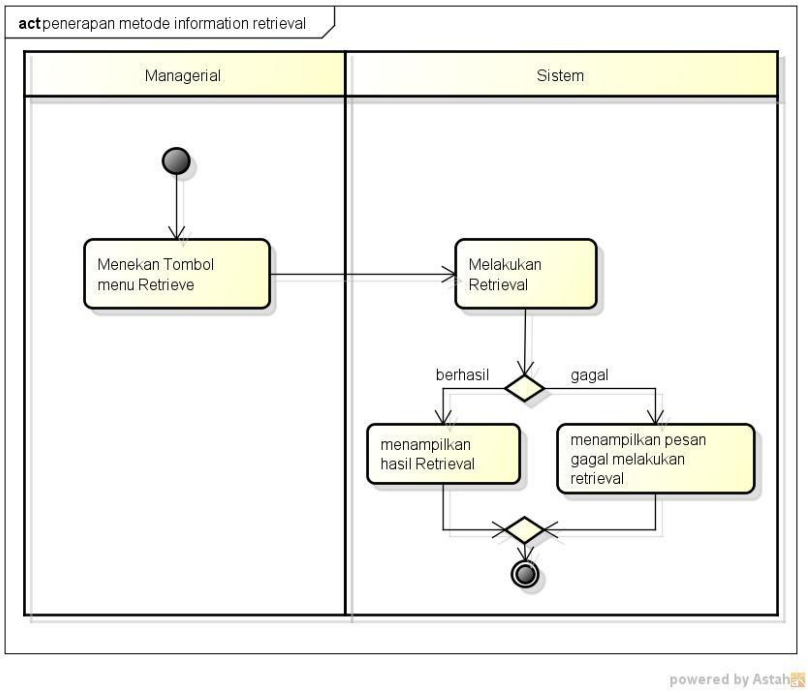
Gambar III-28 activity diagram convert negation



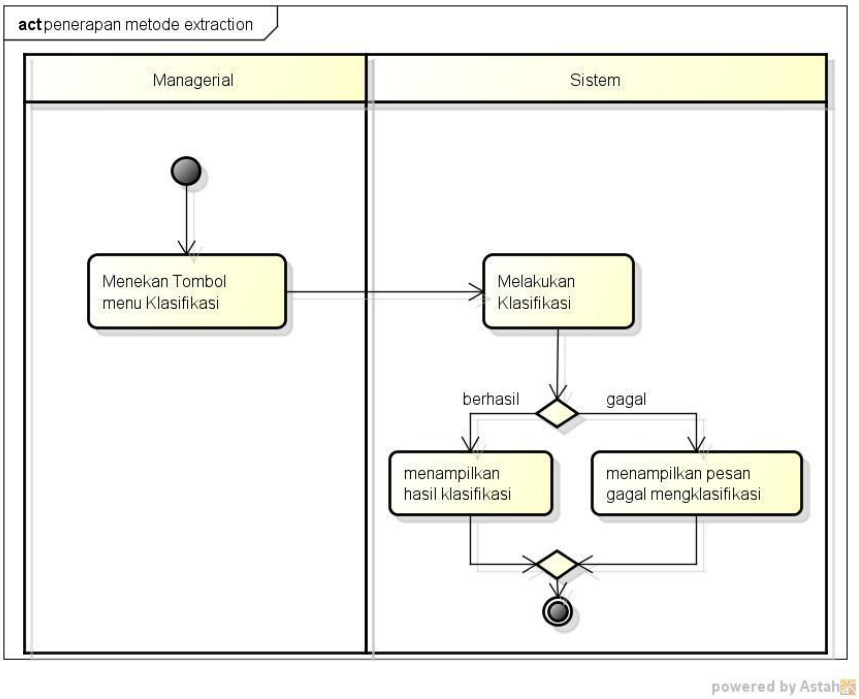
Gambar III-29 activity diagram tokenizing



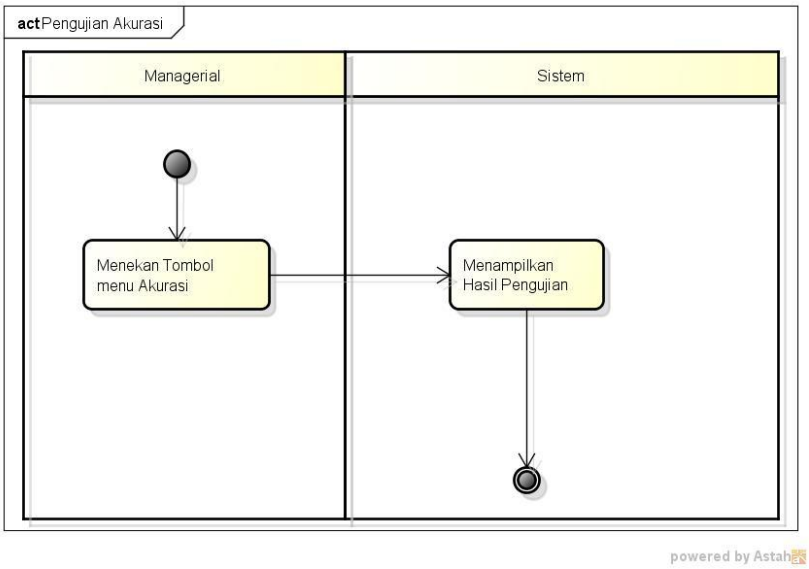
Gambar III-30 activity diagram stemming



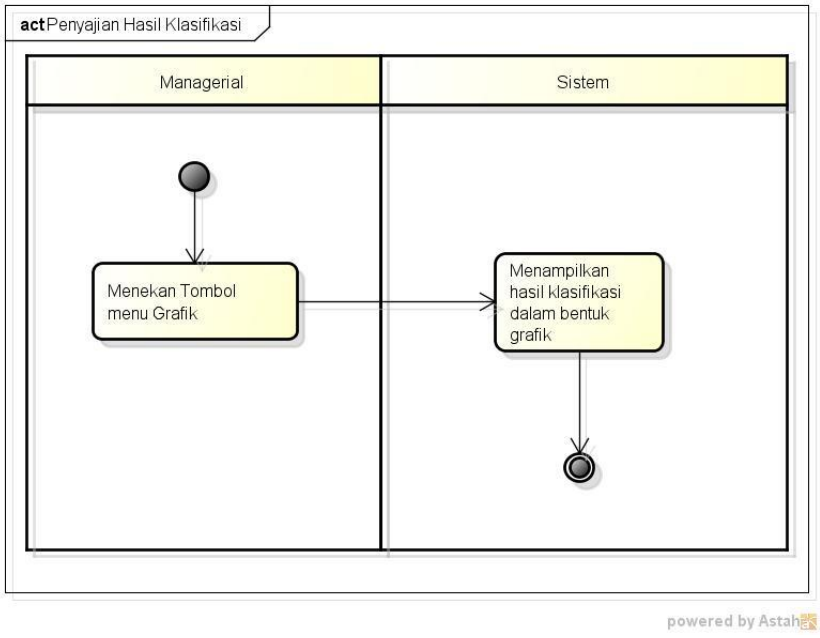
Gambar III-31 activity diagram retrieve data tweet



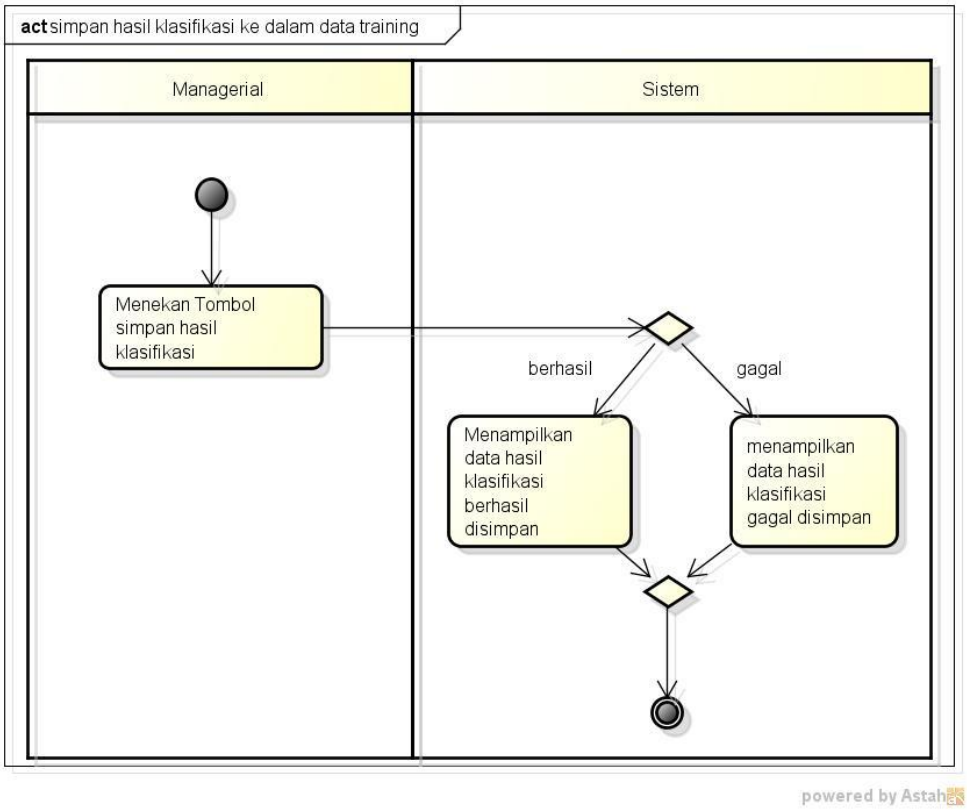
Gambar III-32 activity diagram klasifikasi data tweet



Gambar III-33 activity diagram pengujian akurasi



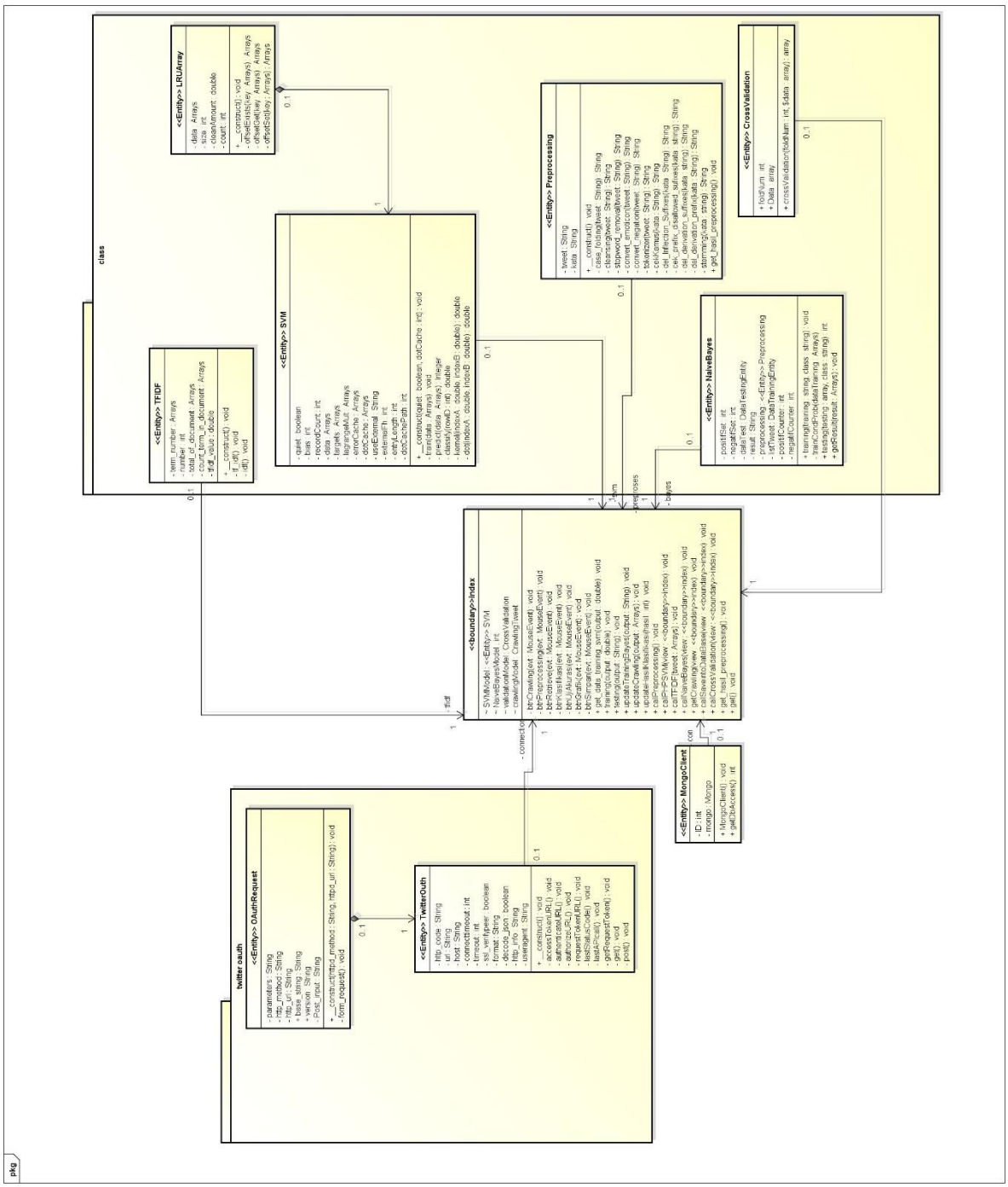
Gambar III-34 activity diagram penyajian hasil klasifikasi



Gambar III-35 Activity diagram simpan hasil klasifikasi ke dalam data training

4. *Class Diagram*

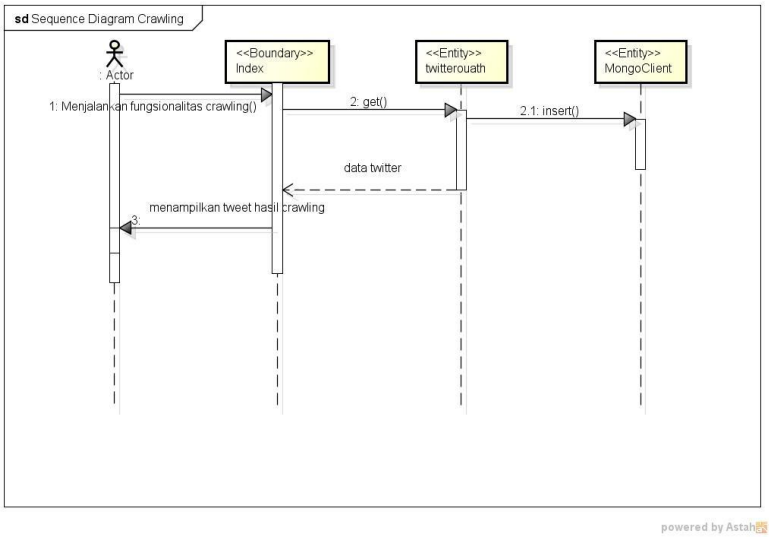
   Berikut adalah class diagram dari pembangunan perangkat lunak ini:



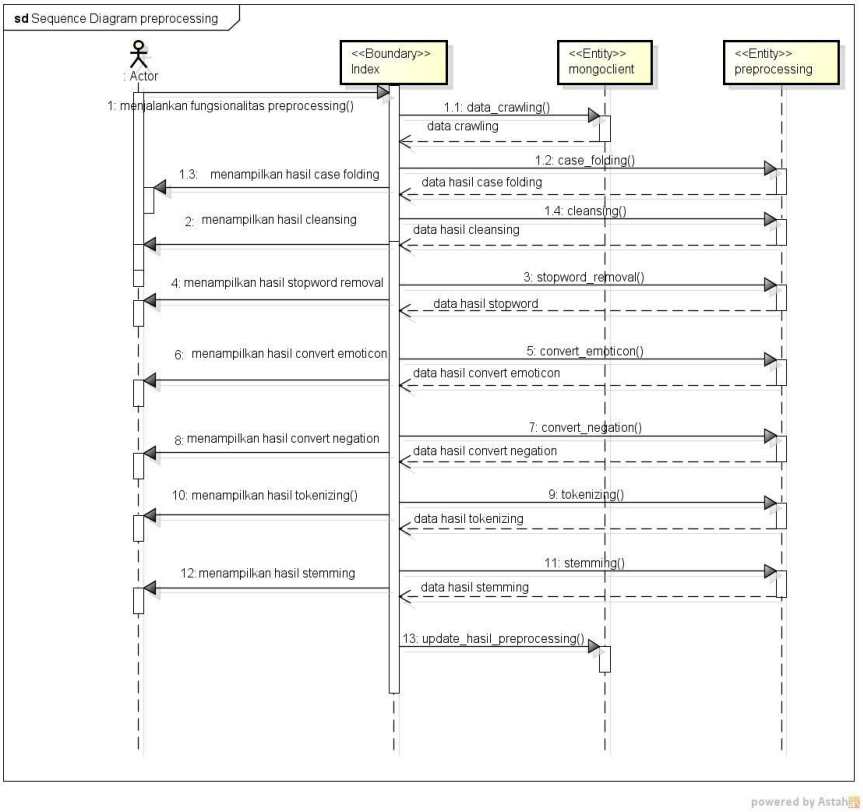
Gambar III-36 Class Diagram

5. *Sequence Diagram*

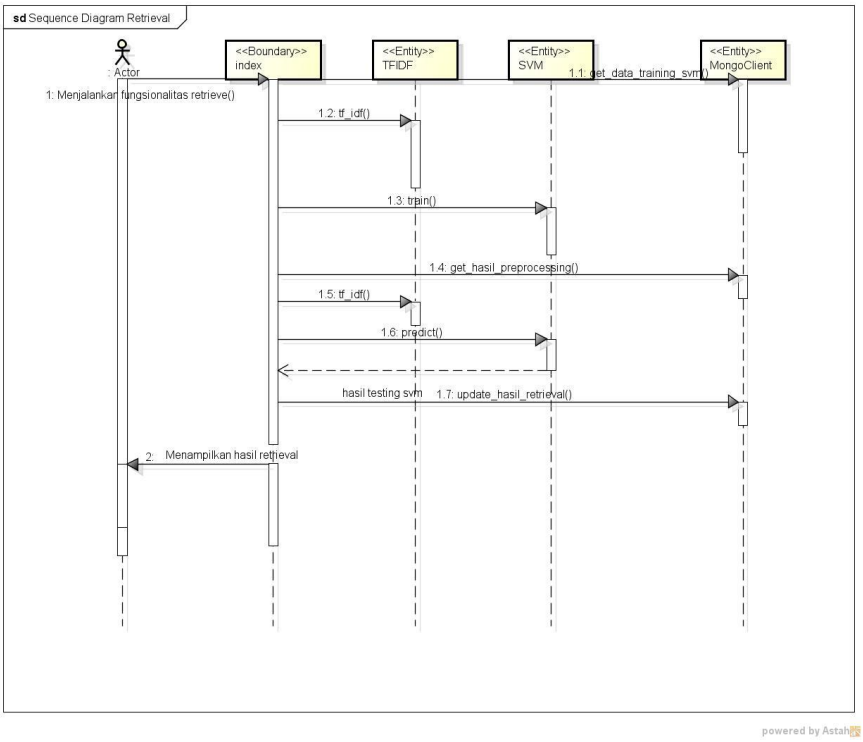
Di asumsikan index.php adalah *Boundary Class* pada aplikasi ini yang menjalankan method pada class lainnya. Berikut adalah sequence diagram yang terdapat pada pembangunan perangkat lunak ini:



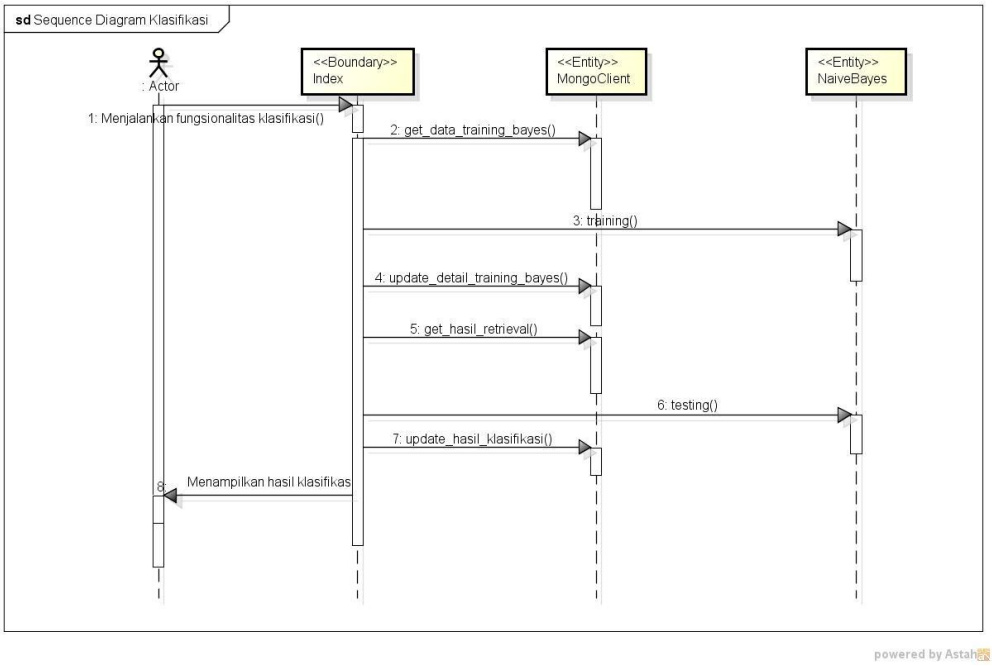
Gambar III-37 Sequence Diagram Crawling



Gambar III-38 Sequence Diagram Preprocessing



Gambar III-39 Sequence Diagram Retrieve



Gambar III-40 Sequence Diagram Klasifikasi

III.2 Perancangan Sistem

Pada perancangan Sistem ini, hal yang akan dibahas adalah Perancangan Data, Perancangan Arsitektural Menu, Perancangan Antarmuka, Perancangan Pesan dan Jaringan Semantik.

III.2.1 Perancangan Data

Pada penelitian ini *database* yang digunakan adalah NoSQL maka akan ditampilkan struktur dari masing-masing collection yang akan digunakan:

Tabel III-31 Struktur Collection crawling

Format Document	Keterangan

{
_id : ObjectID,
tweet_text :string
}	

Tabel III-32 Struktur Collection data_training_bayes

Format Document	Keterangan

{
_id : ObjectID,
tweet_text : string,
kategori: string
}	

Tabel III-33 Struktur Collection detail_data_training_bayes

Nama Document	Keterangan

{
_id : ObjectID,
kata : string,
probNeg: double,
probPos: double
}	

Tabel III-34 Struktur Collection data_training_svm

Format Document	Keterangan
{	

Format Document	Keterangan
_id : ObjectID tweet_text : string, kategori: string }	

Tabel III-35 Struktur Collection grafik

Nama Document	Keterangan

{
_id : ObjectID,
positif : integer,
negatif: integer
}	

Tabel III-36 Struktur Collection hasil_preprocessing

Nama Document	Keterangan

{
_id : ObjectID,
tweet_text : string
}	

Tabel III-37 Struktur Collection kata_dasar

Nama Document	Keterangan

{
_id : ObjectID,
kata: string
}	

Tabel III-38 Struktur Collection kata_stopword

Nama Document	Keterangan

{
_id : ObjectID,
stopword : string
}	

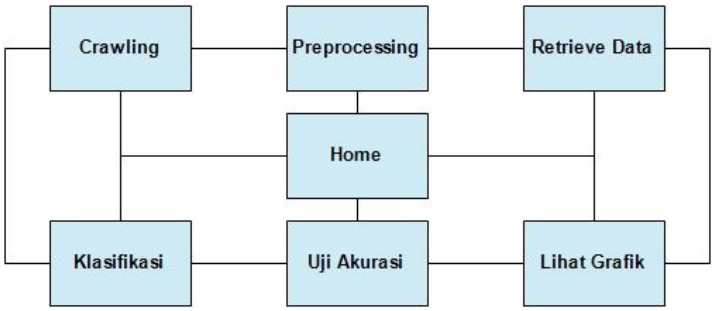
Tabel III-39 Struktur Collection hasil_klasifikasi

Nama Document	Keterangan
{ _id : ObjectID, Tweet_text : string,	

Nama Document	Keterangan
Kategori: string }	

III.2.2 Perancangan Arsitektural Menu

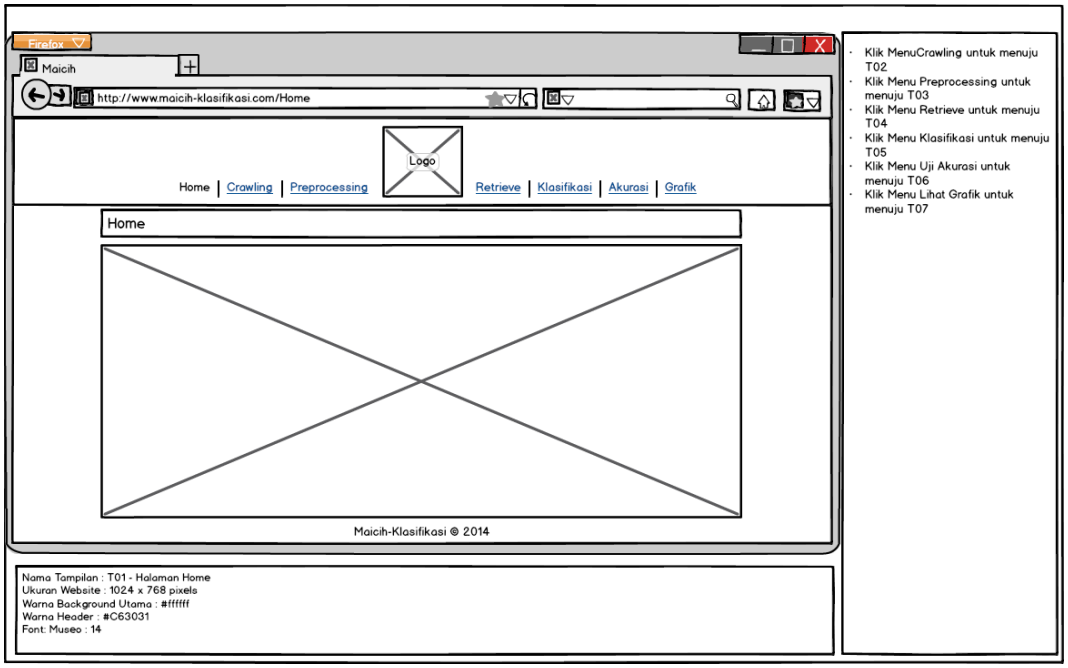
Perancangan Menu pada pembangunan perangkat lunak ini dapat dilihat pada Gambar III-41



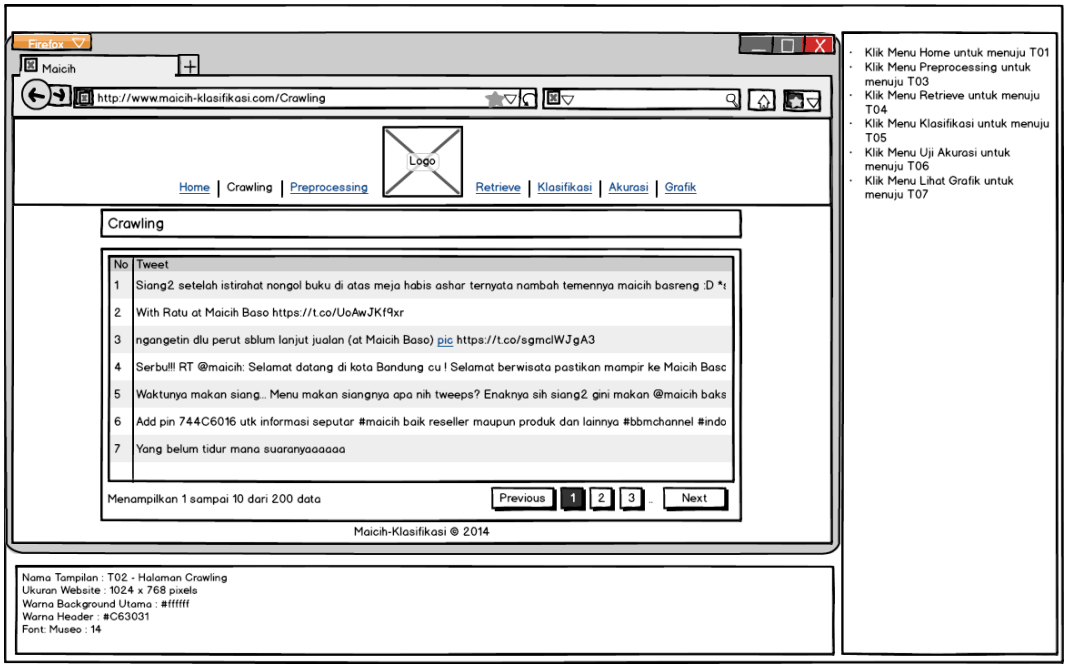
Gambar III-41 Struktur Menu

III.2.3 Perancangan Antarmuka

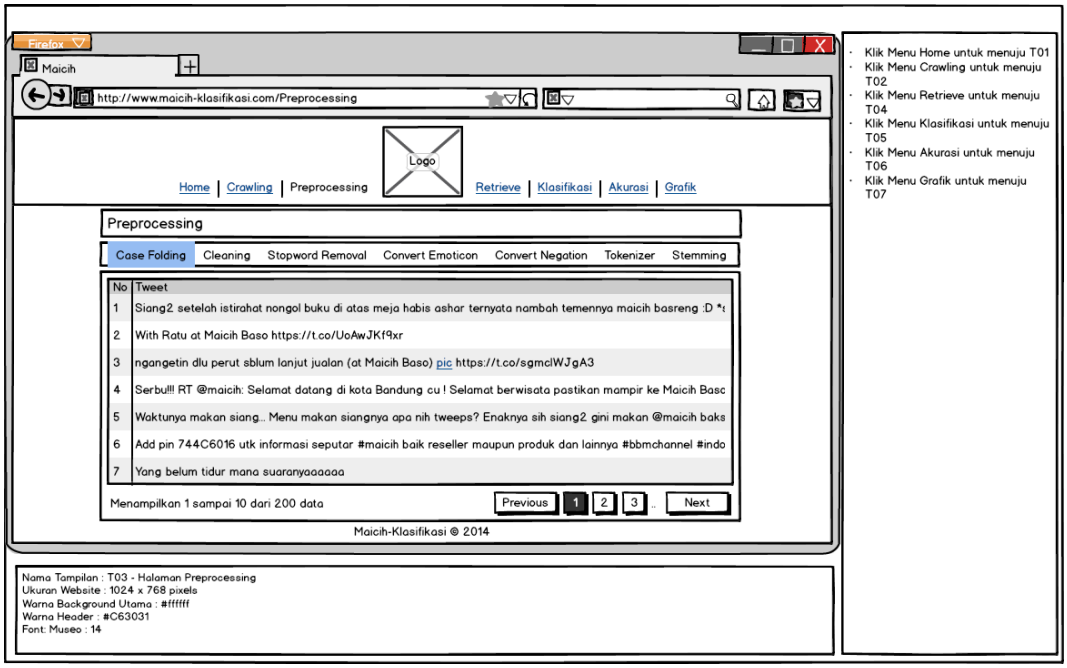
Berikut adalah Perancangan Antarmuka pada penelitian ini:



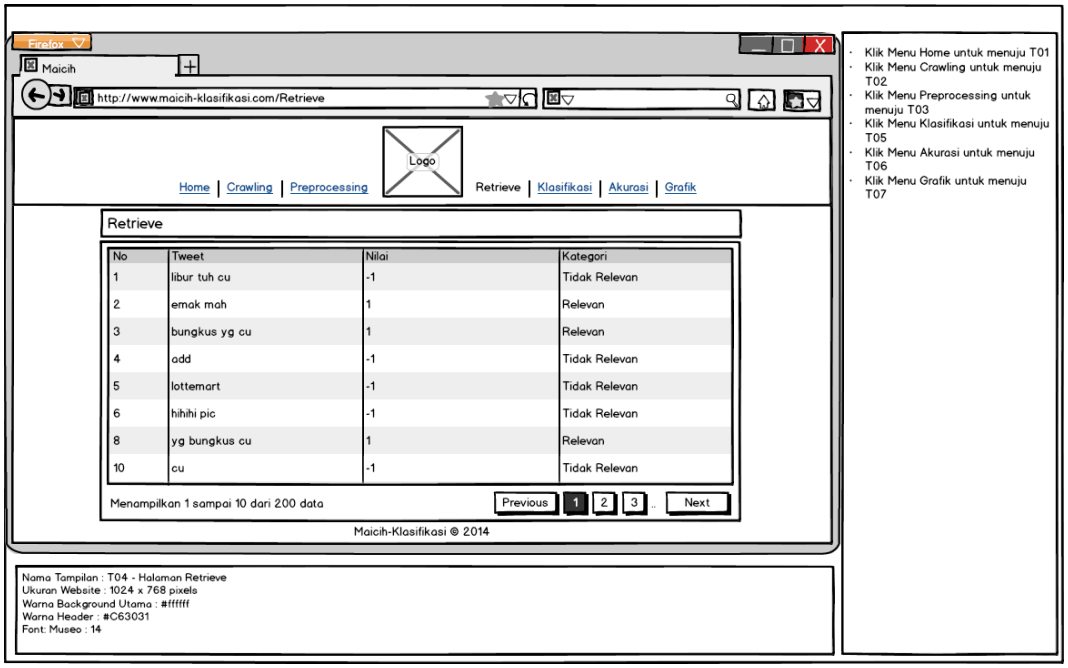
Gambar III-42 Halaman Home



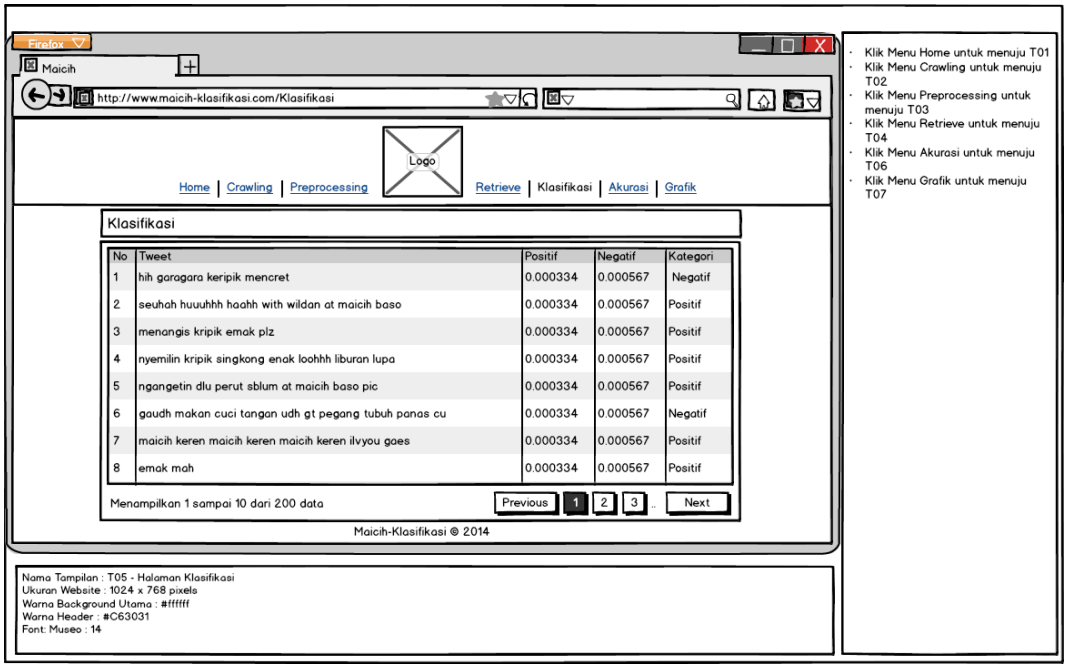
Gambar III-43 Halaman Crawling



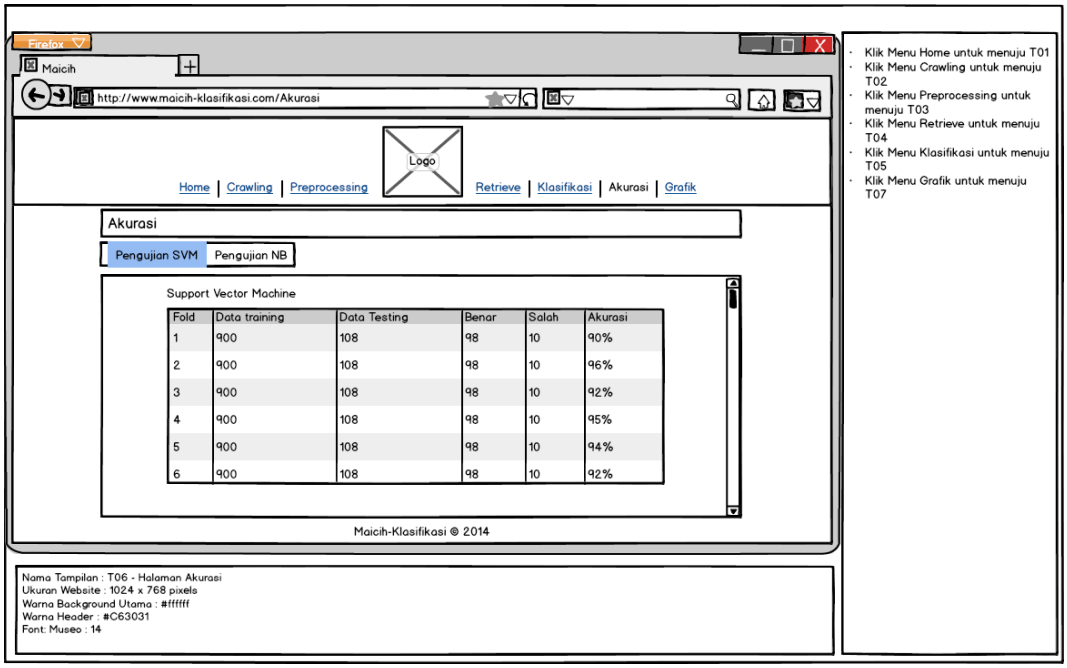
Gambar III-44 Halaman Preprocessing



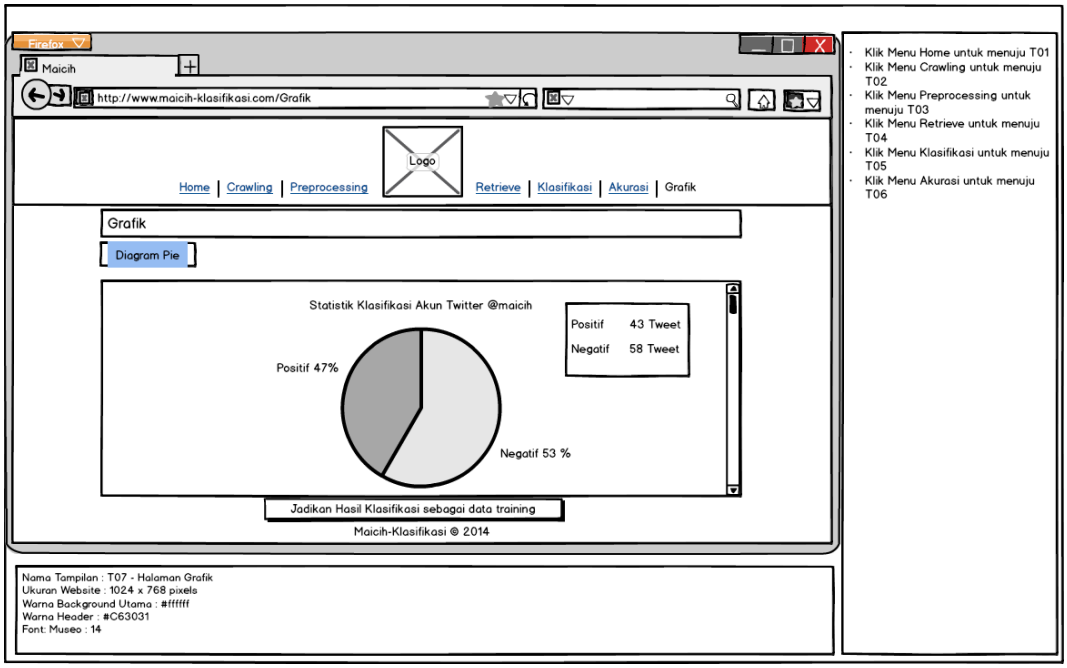
Gambar III-45 Halaman Retrieve Data



Gambar III-46 Halaman Klasifikasi



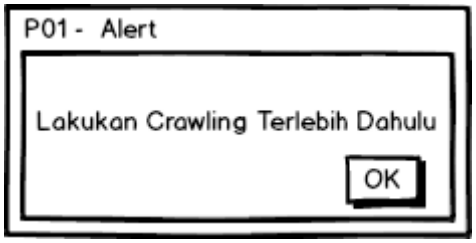
Gambar III-47 Halaman Uji Akurasi



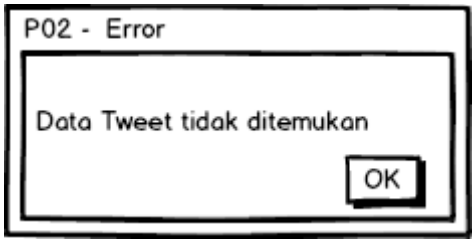
Gambar III-48 Halaman Lihat Grafik

III.2.4 Perancangan Pesan

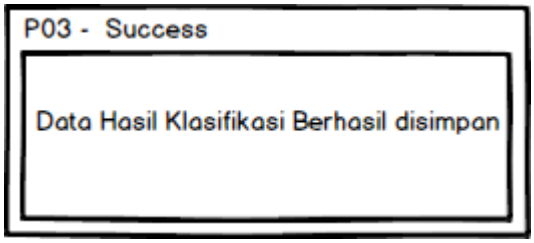
Berikut adalah perancangan pesan pada perangkat lunak dalam penelitian ini:



Gambar III-49 Alert



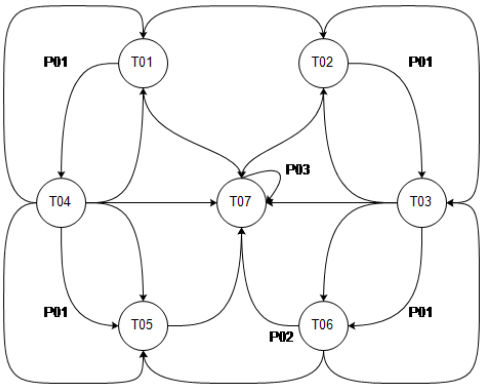
Gambar III-50 Error



Gambar III-51 Success

III.2.5 Jaringan Semantik

Berikut adalah jaringan semantik pada perangkat lunak dalam penelitian ini:



Gambar III-52 Jaringan Semantik