BAB 3
ANALISIS DAN PERANCANGAN

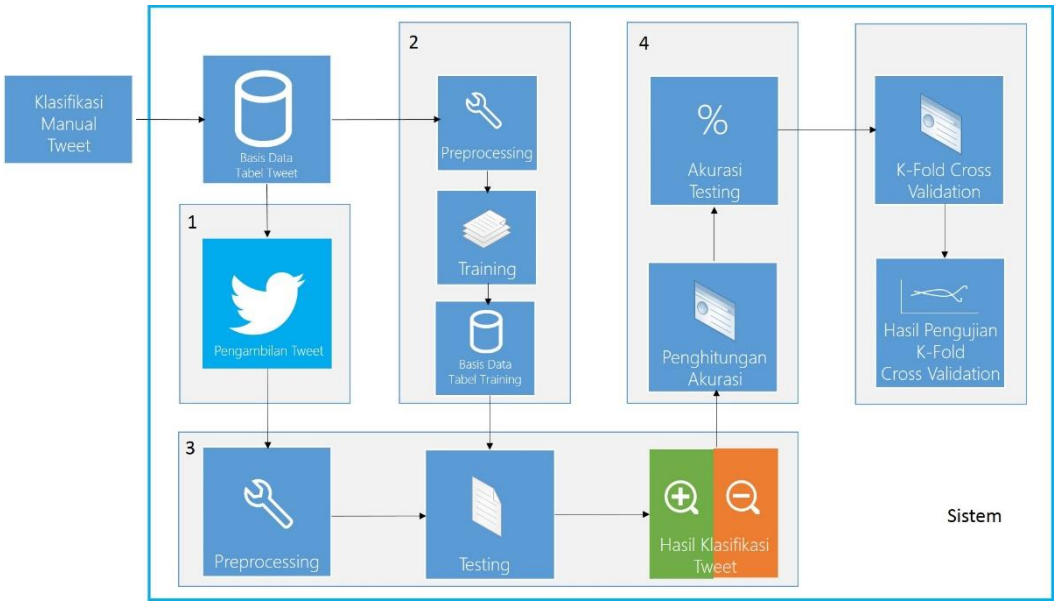
3.1 Analisis Masalah

Analisis masalah dapat didefinisikan suatu gambaran masalah yang diangkat dalam penulisan skripsi tentang optimasi akurasi analisis sentimen pada Twitter menggunakan metode n-gram. Berdasarkan hasil pengamatan terhadap *tweet-tweet* kepada akun Twitter salah satu acara pencarian bakat di Indonesia, ditemukan beberapa penggunaan kata dengan variasi penulisan kata seperti “kereeeen”, “seruuuu”, ”kerenzz”, ”baguss”, ”kren”, ”kecee”, dan sebagainya. Keberadaan kata yang tidak baku dapat mempengaruhi akurasi hasil penghitungan dengan Naive Bayes [2], karena setiap kata pada *tweet* berperan pada proses klasifikasi. Jika dilihat kata-kata tersebut merepresentasikan sebuah kata baku, namun ditulis dengan berbagai variasi penulisan. Oleh sebab itu, diperlukan suatu proses untuk mengakomodir penggunaan kata-kata tersebut, yaitu dengan menggunakan metode n-gram untuk mengambil fitur pada *tweet*.

3.2 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari sistem utama ke dalam sub-sub sistem dengan tujuan untuk mengidentifikasikan kebutuhan-kebutuhan yang diperlukan agar dapat dibangun sebuah aplikasi untuk melakukan pengujian untuk mengetahui pengaruh metode yang digunakan.

Aplikasi yang akan dibangun memiliki beberapa tahapan. Tahapan-tahapan tersebut yaitu proses pengambilan *tweet*, proses *training*, proses *testing*, proses penghitungan akurasi. Gambaran umum dari aplikasi yang akan dibangun ditampilkan pada gambar 3.1 berikut ini



Gambar 3.1 Gambaran umum sistem yang akan dibangun

Berikut ini penjelasan dari gambar 3.1.

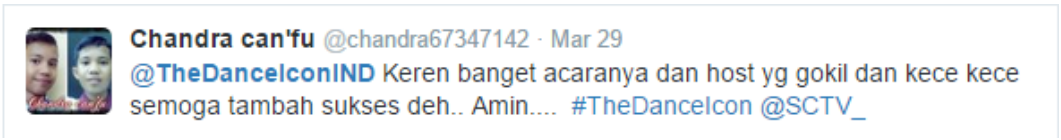
1. Bagian pertama adalah proses pengambilan *tweet* dari database tabel *tweet* yang akan digunakan untuk proses *testing*. Namun sebelumnya, data *tweet* pada tabel *tweet*, sudah dilakukan proses klasifikasi secara manual atau sudah diberi label secara manual yaitu kelas sentimen positif atau negatif..
2. Bagian kedua adalah proses *training*. Data *tweet* diambil dari database tabel *tweet*, kemudian dilakukan proses *preprocessing*. Tahapan preprocessing yang dilakukan adalah *convert emoticon*, *cleansing*, *case folding*, *tokenizing*, *convert negation*, *remove stopword*, dan *stemming*. Setelah itu dilakukan proses *training*, hasil pada proses *training* adalah model probabilistik berupa fitur beserta probabilitas terhadap kategori positif dan negatif. Jenis fitur yang digunakan dalam penelitian ini yaitu fitur kata dan fitur dengan n-gram dengan nilai n=2,3,4,5. Model probabilistik tersebut disimpan ke dalam tabel *training*.
3. Bagian ketiga adalah proses *testing*. Data masukan untuk proses *testing* adalah data *tweet* yang sebelumnya telah diambil pada bagian pertama. *Tweet* tersebut terlebih dahulu melewati tahap *preprocessing* (sama seperti tahap *preprocessing* pada bagian kedua), setelah itu dilakukan proses pencocokan token hasil tahap *preprocessing* dan fitur pada database dan penghitungan

dengan menggunakan algoritma *naive bayes* untuk menentukan klasifikasi *tweet* (positif atau negatif).

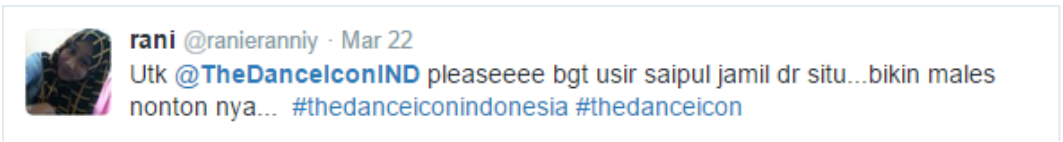
4. Bagian keempat adalah proses penghitungan akurasi. Pada bagian ini, *tweet* yang sudah dilakukan proses *testing*, dihitung jumlah *tweet* yang terklasifikasi sesuai kategori sentimennya dibagi dengan seluruh jumlah *tweet* yang diproses pada proses *testing*, lalu menghasilkan nilai akurasinya.
5. Bagian kelima adalah proses pengujian menggunakan *k-fold cross validation* dengan nilai k=10. Pengujian dilakukan terhadap masing-masing fitur. Hasil dari proses pengujian ini disajikan dalam bentuk grafik beserta tabel rata-rata akurasi dari hasil pengujian terhadap setiap fitur.

3.3 Analisis Data Masukan

Data masukan yang dimaksudkan adalah *tweet* yang diambil dari akun Twitter The Dance Icon Indonesia (@TheDanceIconIND) atau yang mengandung keyword TheDanceIconIND. Contoh dari *tweet* yang mengandung sentimen pada ditampilkan pada gambar 3.2 dan gambar 3.3.



Gambar 3.2 Contoh *tweet* sentimen positif



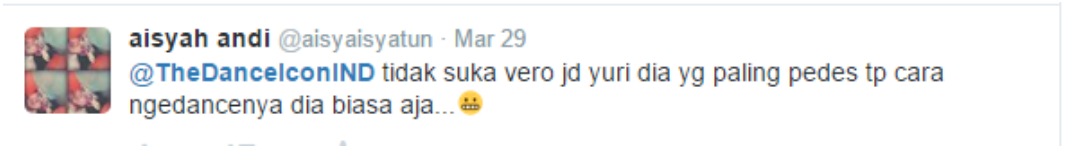
Gambar 3.3 Contoh *tweet* sentimen negatif

Data *tweet* yang diambil masih merupa data mentah dan terdapat *noise* atau yang tidak memiliki pengaruh pada klasifikasi sentimen seperti URL, *hashtag* (#), *username* (@). Contoh dari *tweet* mentah tersebut ditampilkan pada gambar 3.4 berikut ini:



Gambar 3.4 Contoh *tweet* yang mengandung *URL*, *hashtag*, dan *username*

Karakteristik *tweet* selanjutnya yaitu terdapat kata negasi seperti "tidak", "ga", "ngga", dan sebagainya. Contohnya kata "suka", jika terdapat kata negasi "tidak" di depannya, maka akan membalikan sentimen menjadi negatif. Contoh *tweet* yang terdapat kata negasi terdapat pada gambar 3.5 berikut ini:



Gambar 3.5 Contoh *tweet* yang mengandung negasi

Selanjutnya, pada sebuah *tweet* memungkinkan terdapat sebuah *emoticon*. *Emoticon* merupakan salah satu cara dalam mengungkapkan ekspresi dalam teks, dengan kata lain dapat mempengaruhi dalam penentuan kelas sentimen. Contoh *tweet* yang terdapat penggunaan *emoticon* pada gambar 3.6 berikut ini:



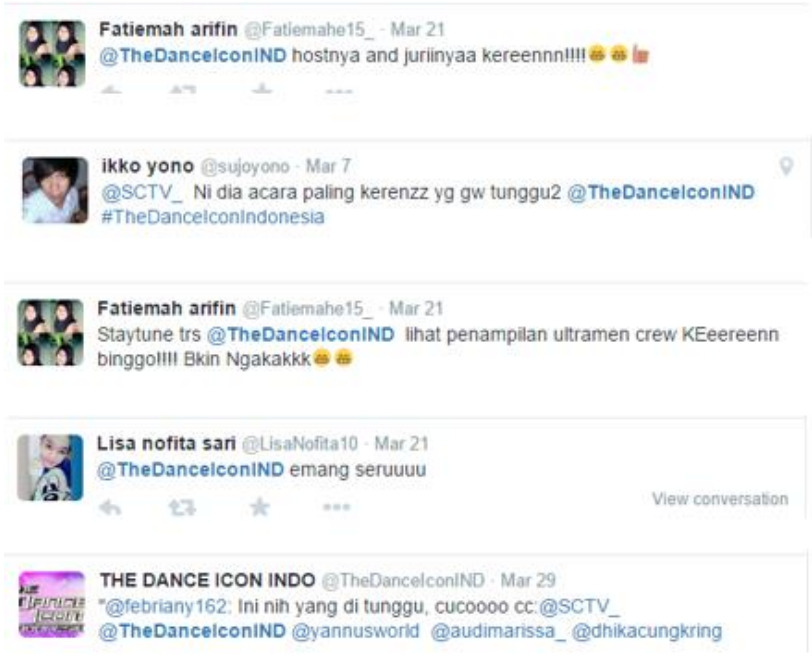
Gambar 3.6 Contoh *tweet* yang mengandung *emoticon*

Selanjutnya, pada sebuah *tweet* terdapat beberapa penggunaan imbuhan. Sebagai contoh seperti kata "dukungan" dan "mendukung" yang memiliki satu kata dasar yaitu "dukung". Berikut ini contoh *tweet* yang terdapat penggunaan imbuhan pada gambar 3.7 berikut ini:



Gambar 3.7 Contoh *tweet* yang mengandung imbuhan

Selain itu juga penggunaan kata-kata tidak baku dengan berbagai variasinya banyak ditemukan pada *tweet* yang dibuat pengguna. Sebagai contoh kata "keren" terdapat berbagai variasnya seperti "kerennn", "kerenzz", "keeereenn", dan sebagainya. Berikut ini beberapa contoh *tweet* yang menggunakan kata-kata dengan berbagai variasinya pada gambar 3.8.

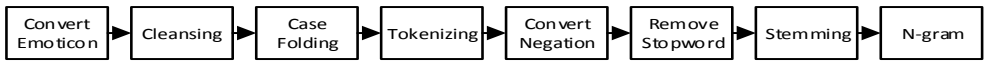


Gambar 3.8 Contoh *tweet* dengan penggunaan variasi kata

Berdasarkan pengamatan dari data tweet yang ada, maka perlu dilakukan tahap *preprocessing* sebelum melakukan klasifikasi.

3.4 Analisis Tahap Preprocessing

Berdasarkan dari analisis data masukan yang telah dijelaskan, maka perlu untuk dilakukan tahap *preprocessing*. Tahap *preprocessing* yang akan dilakukan yaitu *convert emoticon, cleansing, case folding, tokenizing, convert negation, remove stopword, stemming* dan n-gram (jika menggunakan fitur n-gram). Gambaran dari tahap *preprocessing* dapat dilihat pada gambar 3.9 berikut:

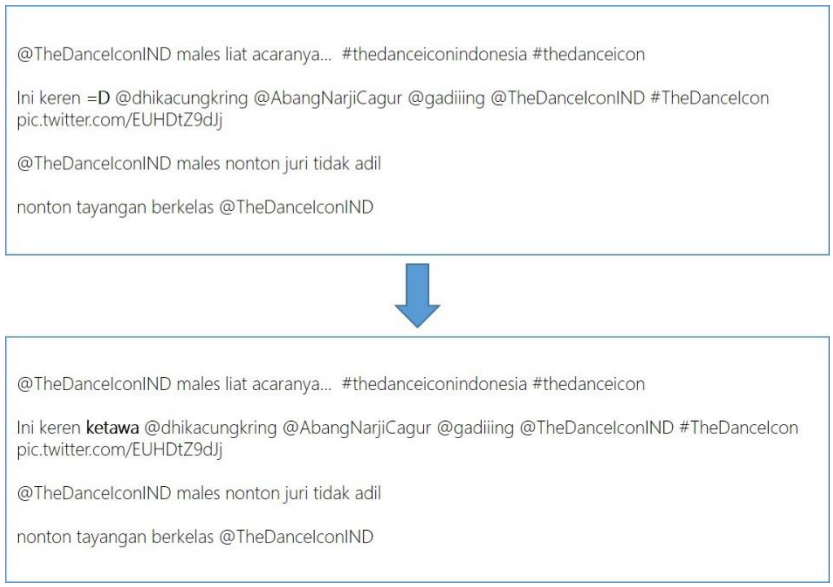


Gambar 3.9 Tahapan *preprocessing*

Berikut ini dijelaskan masing-masing dari tahap *preprocessing* yang dilakukan tersebut pada sub bab berikut ini:

3.4.1 Convert Emoticon

Emoticon merupakan salah satu cara pengungkapan ekspresi perasaan secara tekstual. Hal tersebut akan membantu dalam menentukan sentimen dalam suatu kalimat atau *tweet.* Setiap *emoticon* akan dikonversikan ke dalam string yang bersesuaian. Jika pada suatu *tweet* mengandung *emoticon* yang berada pada *list emoticon* (dapat dilihat pada tabel 2.1), maka konversikan ke dalam string yang bersesuaian. Sebagai gambaran dari tahap *convert emoticon* pada beberapa *tweet* ditampilkan pada gambar 3.10 berikut ini:



Gambar 3.10 Tahap *convert emoticon*

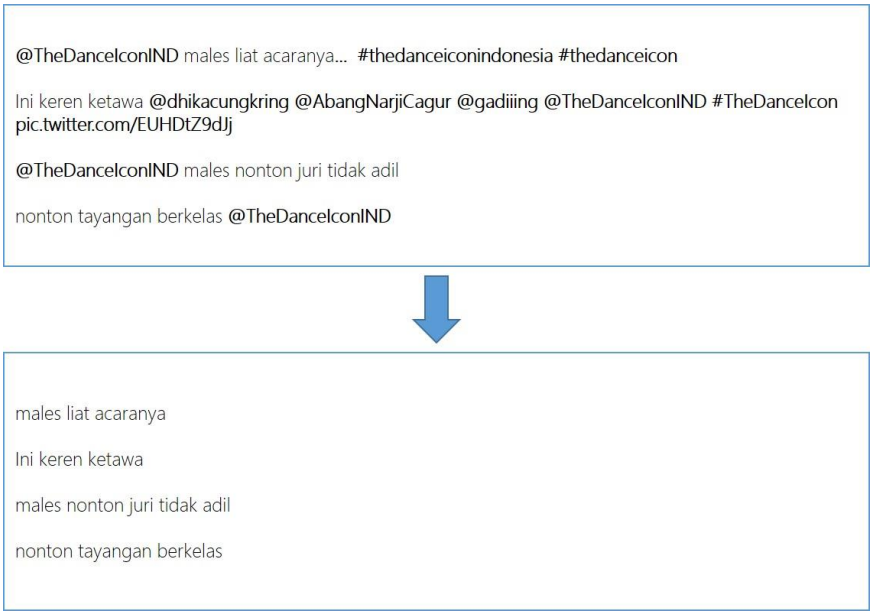
3.4.2 Cleansing

Pada *tweet* yang dibuat oleh pengguna Twitter memiliki berbagai komponen khas seperti *hashtag*, *username*, URL (tautan) dan RT (tanda *retweet*) serta simbol tanda baca seperti titik, koma, dan sebagainya. Komponen *hashtag* ditandai dengan kemunculan simbol "#". Komponen username ditandai dengan kemunculan simbol "@". Komponen URL dikenali melalui ekpresi regular seperti http dan www. Selain itu juga, pada *tweet* memungkinkan terdapat angka dan spasi lebih dari satu. Komponen-komponen tersebut tidak memiliki pengaruh terhadap sentimen, maka

perlu untuk dibersihkan (*cleansing*). Berikut langkah-langkah pada tahap *cleansing*:

1. *Tweet* hasil dari tahap convert *emoticon* akan diperiksa apakah mengandung hashtag (#), username (@), URL, RT, simbol-simbol, angka, dan spasi lebih dari satu.
2. Jika ada komponen di atas, maka hilangkan.

Sebagai gambaran dari tahap *cleansing* pada beberapa *tweet* ditampilkan pada gambar 3.11 berikut ini:



Gambar 3.11 Tahap *cleansing*

3.4.3 Case Folding

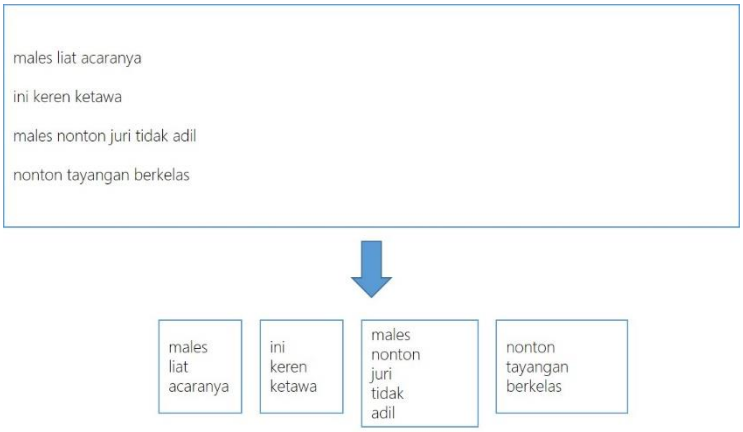
C*ase folding* yaitu menyeragamkan huruf dari *tweet*, dalam tahap ini huruf besar (*uppercase*) pada *tweet* diubah menjadi huruf kecil (*lowercase*). Jika ditemukan karakter yang menggunakan huruf kapital (*uppercase*) pada suatu *tweet*, maka ubah ke bentuk kecil (*lowercase*). Sebagai gambaran dari tahap *case folding* pada beberapa *tweet* ditampilkan pada gambar 3.12 berikut ini:



Gambar 3.12 Tahap *case folding*

3.4.4 Tokenizing

Proses *tokenizing* yaitu memisahkan kata pada *tweet* menjadi kata-kata tunggal. Pemisahan tersebut ditandai dengan tanda pemisah yaitu spasi. Sebagai gambaran dari tahap *tokenizing* pada beberapa *tweet* ditampilkan pada gambar 3.13 berikut:



Gambar 3.13 Tahap *tokenizing* kata

3.4.5 Convert Negation

Pada sebuah *tweet* yang dibuat pengguna, memungkinkan terdapat negasi yang dapat membalikan arti dari suatu kata dan akan membalikan sentimen yang tadinya negatif menjadi positif dan sebaliknya. Jika pada *tweet* mengandung kata negasi, maka gabungkan dengan kata setelahnya. Sebagai gambaran dari tahap *convert negation* pada beberapa *tweet* ditampilkan pada gambar 3.14 berikut ini:



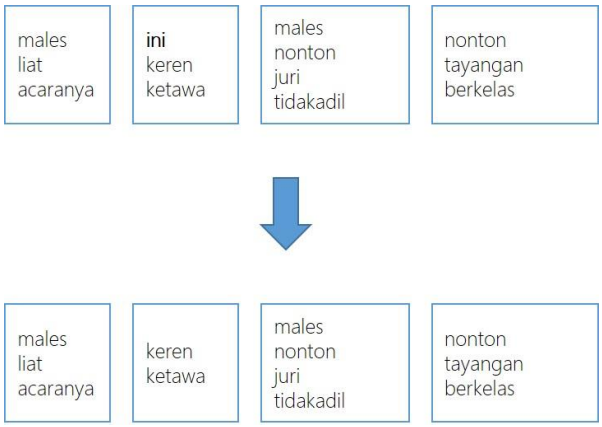
Gambar 3.14 Tahap *convert negation*

3.4.6 Remove Stopword

Remove stopword yaitu proses membuang kata-kata yang terdapat pada daftar *stopword*. Kata-kata yang berada dalam daftar *stopword* berupa kata ganti orang, kata ganti penghubung dan penunjuk. Berikut adalah tahapan-tahapan pada proses *remove stopword*:

1. Setiap kata diperiksa apakah ditemukan pada daftar kata *stopword*.
2. Jika ditemukan, maka hilangkan kata tersebut.

Sebagai gambaran dari tahap *remove stopword* pada beberapa *tweet* ditampilkan pada gambar 3.15 berikut ini:

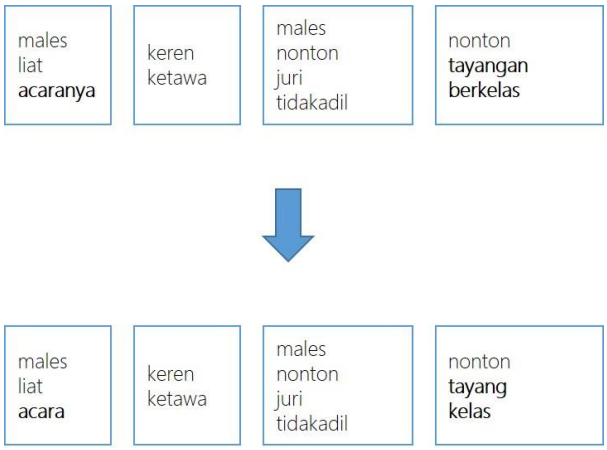


Gambar 3.15 Tahap *remove stopword*

3.4.7 Stemming

Pada sebuah *tweet* terdapat beberapa penggunaan imbuhan. Sebagai contoh seperti kata “dukungan” dan “mendukung” yang memiliki satu kata dasar yaitu

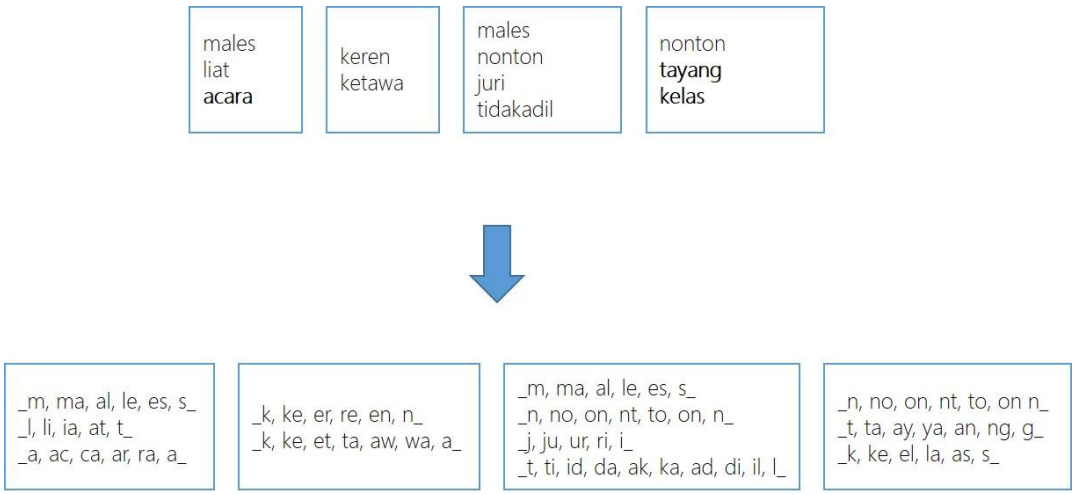
“dukung. Salah satu algoritma yang dapat digunakan untuk proses *stemming* yaitu algoritma porter bahasa Indonesia. Gambaran dari tahap *stemming* pada beberapa *tweet* ditampilkan pada gambar 3.16 berikut ini:



Gambar 3.16 Tahap *stemming*

3.4.8 N-gram

Pada tahap ini yaitu membangkitkan n-gram karakter dari hasil tahap preprocessing. Pada gambar 3.17 berikut ditampilkan n-gram karakter yang dibangkitkan dari *tweet* hasil tahap *stemming* dengan nilai n=2.



Gambar 3.17 Tahap membangkitkan n-gram (n=2)

3.5 Analisis Klasifikasi Menggunakan Naive Bayes

Tahap klasifikasi menggunakan *naive bayes* terbagi kedalam 2 proses, yaitu *training* dan *testing*. Proses *training* dilakukan untuk menghasilkan model probabilistik dari fitur yang nantinya akan digunkan sebagai acuan perhitungan untuk mengklasifikasikan *tweet* data *testing* ke dalam dua kelas positif atau negatif. Berikut adalah penjelasan dari tahapan klasifikasi sentimen menggunakan *naive bayes*.

1. Proses *training*
   a. Hitung $P(c_i)$
   b. Hitung $P(w_k|c_i)$ untuk setiap w_k pada model
2. Proses *testing*
   a. Hitung $P(c_i) \prod_k P(w_k|c_i)$ untuk setiap kategori
   b. Tentukan c^*, yaitu kategori dengan nilai $P(c_i) \prod_k P(w_k|c_i)$ tertinggi

Tweet yang dipergunakan sebagai data *training* yaitu hasil pada tahap *preprocessing* sebelumnya. *Tweet-tweet* tersebut sebelumnya sudah dilabeli kategorinya secara manual seperti ditunjukan pada tabel 3.1 berikut:

Tabel 3.1 Data *tweet training*

Nama *Tweet*	Tweet	Kategori
Tweet 1	males liat acara	Negatif
Tweet 2	keren ketawa	Positif
Tweet 3	males nonton juri tidakadil	Negatif
Tweet 4	nonton tayang kelas	Positif

Namun, untuk menyederhanakan penjelasan pada tahap selanjutnya, hanya *Tweet* 1 dan *Tweet* 2 saja yang dipergunakan sebagai data *training*. Pada penelitian ini menggunakan metode n-gram karakter untuk mengambil fitur pada *tweet*. Sebagai perbandingan, akan dijelaskan pengambilan fitur tanpa metode n-gram (fitur berupa kata) dan dengan menggunakan metode n-gram. Selanjutnya akan dijelaskan proses klasifikasi dari masing-masing jenis fitur.

3.5.1 Proses Training dengan Fitur Kata

Tweet yang akan digunakan sebagai data *training* dengan fitur kata ditunjukan pada tabel 3.2 di atas, dimana pengambilan fitur tanpa menggunakan n-gram(fitur berupa kata), ditampilkan pada tabel berikut:

Tabel 3.2 Data *training* dengan fitur kata

Nama Tweet	Fitur hasil seleksi	Kategori
Tweet 1a	males liat acara	negatif
Tweet 1b	keren ketawa	positif

Pertama-tama hitung $P(c_i)$, dimana c_1 = positif dan c_2 = negatif. Dengan mengacu pada persamaan 2.5 maka dapat dihitung sebagai berikut:

$$P(c_{positif}) = \frac{1}{2}$$

$$P(c_{negatif}) = \frac{1}{2}$$

Setelah itu dibentuk model probabilistik, dengan mengacu pada persamaan 2.6 sebagai berikut:

$$P(w_{males}|c_{positif}) = \frac{0+1}{2+4} = \frac{1}{6}$$

$$P(w_{males}|c_{negatif}) = \frac{1+1}{2+4} = \frac{2}{6}$$

Jika keseluruhan $P(w_k|c_i)$ dihitung, maka hasil perhitungannya dapat dilihat pada tabel 3.3 berikut ini

Tabel 3.3 Penghitungan probabilitas fitur kata

k	Fitur (w_k)	Frekuensi positif	Frekuensi negatif	Probabilitas positif $P(w_k \mid c_{positif})$	Probabilitas negatif $P(w_k \mid c_{negatif})$
1	males	0	1	0,142857143	0,25
2	liat	0	1	0,142857143	0,25
3	acara	0	1	0,142857143	0,25
4	keren	1	0	0,285714286	0,125
5	ketawa	1	0	0,285714286	0,125

Hasil perhitungan tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan kelas sentimen dari *tweet testing* dengan fitur kata.

3.5.2 Proses Training dengan Fitur N-gram (n=2)

Tweet yang akan digunakan sebagai data *training* dengan fitur n-gram (n=2) ditunjukan pada tabel 3.4, dimana pengambilan fitur menggunakan n-gram (n=2).

Tabel 3.4 Data *training* dengan fitur n-gram (n=2)

Nama Tweet	Fitur hasil seleksi n-gram	Kategori
Tweet 1a	_m, ma, al, le, es, s_ _l, li, ia, at, t_ _a, ac, ca, ar, ra, a_	negatif
Tweet 1b	_k, ke, er, re, en, n_ _k, ke, et, ta, aw, wa, a_	positif

Pertama-tama hitung $P(c_i)$, dimana c_1 = positif dan c_2 = negatif. Dengan mengacu pada persamaan 2.5 maka dapat dihitung sebagai berikut:

$$P(c_{positif}) = \frac{1}{2}$$

$$P(c_{negatif}) = \frac{1}{2}$$

Setelah itu dibentuk model probabilistik, dengan mengacu pada persamaan 2.6 sebagai berikut:

$$P(w_{_m}|c_{positif}) = \frac{0+1}{13+27} = \frac{1}{40} = 0{,}025$$

$$P(w_{_m}|c_{negatif}) = \frac{1+1}{17+27} = \frac{2}{44} = 0.045454545$$

Jika keseluruhan $P(w_k|c_i)$ dihitung, maka hasil perhitungannya dapat dilihat pada tabel berikut 3.5 ini.

Tabel 3.5 Penghitungan probabilitas fitur dengan n-gram (n=2)

k	Fitur (w_k)	Frekuensi n_k Positif	Frekuensi n_k Negatif	Probabilitas positif $P(w_k\|c_{positif})$	Probabilitas negatif $P(w_k\|c_{negatif})$
1	_m	0	1	0.025	0.045454545
2	ma	0	1	0.025	0.045454545
3	al	0	1	0.025	0.045454545
4	le	0	1	0.025	0.045454545
5	es	0	1	0.025	0.045454545
6	s_	0	1	0.025	0.045454545
7	_l	0	1	0.025	0.045454545
8	li	0	1	0.025	0.045454545
9	ia	0	1	0.025	0.045454545
10	at	0	1	0.025	0.045454545
11	t_	0	1	0.025	0.045454545
12	_a	0	1	0.025	0.045454545
13	ac	0	1	0.025	0.045454545
14	ca	0	1	0.025	0.045454545
15	ar	0	1	0.025	0.045454545
16	ra	0	1	0.025	0.045454545
17	a_	1	1	0.05	0.045454545
18	_k	2	0	0.075	0.022727272
19	ke	2	0	0.075	0.022727272

20	er	1	0	0.05	0.022727272
21	re	1	0	0.05	0.022727272
22	en	1	0	0.05	0.022727272
23	n_	1	0	0.05	0.022727272
24	et	1	0	0.05	0.022727272
25	ta	1	0	0.05	0.022727272
26	aw	1	0	0.05	0.022727272
27	wa	1	0	0.05	0.022727272

Hasil perhitungan tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan kelas sentimen dari *tweet testing* dengan fitur n-gram (n=2).

3.5.3 Proses Training dengan Fitur N-gram (n=3)

Tweet yang akan digunakan sebagai data *training* dengan fitur n-gram (n=2) ditunjukan pada tabel 3.6, dimana pengambilan fitur menggunakan n-gram (n=3).

Tabel 3.6 Data *training* dengan fitur n-gram (n=3)

Nama Tweet	Fitur hasil seleksi n-gram	Kategori
Tweet 1a	_ma, mal, ale, les, es_, s__ _li, lia, iat, at_, t__ _ac, aca, car, ara, ra_, a__	negatif
Tweet 1b	_ke, ker, ere, ren, en_, n__ _ke, ket, eta, taw, awa, wa_, a__	positif

Pertama-tama hitung $P(c_i)$, dimana c_1 = positif dan c_2 = negatif. Dengan mengacu pada persamaan 2.5 maka dapat dihitung sebagai berikut:

$$P(c_{positif}) = \frac{1}{2}$$

$$P(c_{negatif}) = \frac{1}{2}$$

Setelah itu dibentuk model probabilistik, dengan mengacu pada persamaan 2.6 sebagai berikut:

$$P(w_{_ma}|c_{positif}) = \frac{0+1}{13+28} = \frac{1}{40} = 0.024390244$$

$$P(w_{_ma}|c_{negatif}) = \frac{1+1}{17+28} = \frac{2}{45} = 0.044444444$$

Jika keseluruhan $P(w_k|c_i)$ dihitung, maka hasil perhitungannya dapat dilihat pada tabel berikut 3.7 ini.

Tabel 3.7 Penghitungan probabilitas fitur dengan n-gram (n=3)

k	Fitur (w_k)	Frekuensi n_k Positif	Frekuensi n_k Negatif	Probabilitas positif $P(w_k\|c_{positif})$	Probabilitas negatif $P(w_k\|c_{negatif})$
1	_ma	0	1	0.024390244	0.044444444
2	mal	0	1	0.024390244	0.044444444
3	ale	0	1	0.024390244	0.044444444
4	les	0	1	0.024390244	0.044444444
5	es_	0	1	0.024390244	0.044444444
6	s__	0	1	0.024390244	0.044444444
7	_li	0	1	0.024390244	0.044444444
8	lia	0	1	0.024390244	0.044444444
9	iat	0	1	0.024390244	0.044444444
10	at_	0	1	0.024390244	0.044444444
11	t__	0	1	0.024390244	0.044444444
12	_ac	0	1	0.024390244	0.044444444
13	aca	0	1	0.024390244	0.044444444
14	car	0	1	0.024390244	0.044444444
15	ara	0	1	0.024390244	0.044444444
16	ra_	0	1	0.024390244	0.044444444
17	a__	1	1	0.048780488	0.044444444
18	_ke	2	0	0.073170732	0.022222222
19	ker	1	0	0.048780488	0.022222222

20	ere	1	0	0.048780488	0.022222222
21	ren	1	0	0.048780488	0.022222222
22	en_	1	0	0.048780488	0.022222222
23	n__	1	0	0.048780488	0.022222222
24	ket	1	0	0.048780488	0.022222222
25	eta	1	0	0.048780488	0.022222222
26	taw	1	0	0.048780488	0.022222222
27	awa	1	0	0.048780488	0.022222222
28	wa_	1	0	0.048780488	0.022222222

Hasil perhitungan tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan kelas sentimen dari *tweet testing* dengan fitur n-gram (n=3).

3.5.4 Proses Training dengan Fitur N-gram (n=4)

Tweet yang akan digunakan sebagai data *training* dengan fitur n-gram (n=4) ditunjukan pada tabel 3.8, dimana pengambilan fitur menggunakan n-gram (n=4).

Tabel 3.8 Data *training* dengan fitur n-gram (n=4)

Nama Tweet	Fitur hasil seleksi n-gram	Kategori
Tweet 1a	_mal, male, ales, les_, es__, s___ _lia, liat, iat_, at__, t___ _aca, acar, cara, ara_, ra__, a___	negatif
Tweet 1b	_ker, kere, eren, ren_, en__, n___ _ket, keta, etaw, tawa, awa_, wa__, a___	positif

Pertama-tama hitung $P(c_i)$, dimana c_1 = positif dan c_2 = negatif. Dengan mengacu pada persamaan 2.5 maka dapat dihitung sebagai berikut:

$$P(c_{positif}) = \frac{1}{2}$$

$$P(c_{negatif}) = \frac{1}{2}$$

Setelah itu dibentuk model probabilistik, dengan mengacu pada persamaan 2.6 sebagai berikut:

$$P(w_{_mal}|c_{positif}) = \frac{0+1}{13+29} = \frac{1}{42} = 0.023809524$$

$$P(w_{_mal}|c_{negatif}) = \frac{1+1}{17+29} = \frac{2}{46} = 0.043478261$$

Jika keseluruhan $P(w_k|c_i)$ dihitung, maka hasil perhitungannya dapat dilihat pada tabel berikut 3.9 ini.

Tabel 3.9 Penghitungan probabilitas fitur dengan n-gram (n=3)

k	Fitur (w_k)	Frekuensi n_k Positif	Frekuensi n_k Negatif	Probabilitas positif $P(w_k\|c_{positif})$	Probabilitas negatif $P(w_k\|c_{negatif})$
1	_mal	0	1	0.023809524	0.043478261
2	male	0	1	0.023809524	0.043478261
3	ales	0	1	0.023809524	0.043478261
4	les_	0	1	0.023809524	0.043478261
5	es__	0	1	0.023809524	0.043478261
6	s___	0	1	0.023809524	0.043478261
7	_lia	0	1	0.023809524	0.043478261
8	liat	0	1	0.023809524	0.043478261
9	iat_	0	1	0.023809524	0.043478261
10	at__	0	1	0.023809524	0.043478261
11	t___	0	1	0.023809524	0.043478261
12	_aca	0	1	0.023809524	0.043478261
13	acar	0	1	0.023809524	0.043478261
14	cara	0	1	0.023809524	0.043478261
15	ara_	0	1	0.023809524	0.043478261
16	ra__	0	1	0.023809524	0.043478261
17	a___	1	1	0.047619048	0.043478261
18	_ker	1	0	0.047619048	0.02173913
19	kere	1	0	0.047619048	0.02173913

20	eren	1	0	0.047619048	0.02173913
21	ren_	1	0	0.047619048	0.02173913
22	en__	1	0	0.047619048	0.02173913
23	n___	1	0	0.047619048	0.02173913
24	_ket	1	0	0.047619048	0.02173913
25	keta	1	0	0.047619048	0.02173913
26	etaw	1	0	0.047619048	0.02173913
27	tawa	1	0	0.047619048	0.02173913
28	awa_	1	0	0.047619048	0.02173913
29	wa__	1	0	0.047619048	0.02173913

Hasil perhitungan tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan kelas sentimen dari *tweet testing* dengan fitur n-gram (n=4).

3.5.5 Proses Training dengan Fitur N-gram (n=5)

Tweet yang akan digunakan sebagai data *training* dengan fitur n-gram (n=5) ditunjukan pada tabel 3.10, dimana pengambilan fitur menggunakan n-gram (n=5).

Tabel 3.10 Data *training* dengan fitur n-gram (n=5)

Nama Tweet	Fitur hasil seleksi n-gram	Kategori
Tweet 1a	_male, males, ales_, les__, es___, s____ _liat, liat_, iat__, at___, t____ _acar, acara, cara_, ara__, ra___, a____	negatif
Tweet 1b	_kere, keren, eren_, ren__, en___, n____ _keta, ketaw, etawa, tawa_, awa__, wa___ a____	positif

Pertama-tama hitung $P(c_i)$, dimana c_1 = positif dan c_2 = negatif. Dengan mengacu pada persamaan 2.5 maka dapat dihitung sebagai berikut:

$$P(c_{positif}) = \frac{1}{2}$$

$$P(c_{negatif}) = \frac{1}{2}$$

Setelah itu dibentuk model probabilistik, dengan mengacu pada persamaan 2.6 sebagai berikut:

$$P(w_{_male}|c_{positif}) = \frac{0+1}{13+29} = \frac{1}{42} = 0.023809524$$

$$P(w_{_male}|c_{negatif}) = \frac{1+1}{17+29} = \frac{2}{46} = 0.043478261$$

Jika keseluruhan $P(w_k|c_i)$ dihitung, maka hasil perhitungannya dapat dilihat pada tabel berikut 3.11 ini.

Tabel 3.11 Penghitungan probabilitas fitur dengan n-gram (n=3)

k	Fitur (w_k)	Frekuensi n_k Positif	Frekuensi n_k Negatif	Probabilitas positif $P(w_k\|c_{positif})$	Probabilitas negatif $P(w_k\|c_{negatif})$
1	_male	0	1	0.023809524	0.043478261
2	males	0	1	0.023809524	0.043478261
3	ales_	0	1	0.023809524	0.043478261
4	les__	0	1	0.023809524	0.043478261
5	es___	0	1	0.023809524	0.043478261
6	s____	0	1	0.023809524	0.043478261
7	_liat	0	1	0.023809524	0.043478261
8	liat_	0	1	0.023809524	0.043478261
9	iat__	0	1	0.023809524	0.043478261
10	at___	0	1	0.023809524	0.043478261
11	t____	0	1	0.023809524	0.043478261
12	_acar	0	1	0.023809524	0.043478261
13	acara	0	1	0.023809524	0.043478261
14	cara_	0	1	0.023809524	0.043478261
15	ara__	0	1	0.023809524	0.043478261
16	ra___	0	1	0.023809524	0.043478261
17	a____	1	1	0.047619048	0.043478261
18	_kere	1	0	0.047619048	0.02173913
19	keren	1	0	0.047619048	0.02173913

20	eren_	1	0	0.047619048	0.02173913
21	ren__	1	0	0.047619048	0.02173913
22	en___	1	0	0.047619048	0.02173913
23	n____	1	0	0.047619048	0.02173913
24	_keta	1	0	0.047619048	0.02173913
25	ketaw	1	0	0.047619048	0.02173913
26	etawa	1	0	0.047619048	0.02173913
27	tawa_	1	0	0.047619048	0.02173913
28	awa__	1	0	0.047619048	0.02173913
29	wa___	1	0	0.047619048	0.02173913

Hasil perhitungan tersebut digunakan sebagai model probabilistik yang nantinya akan digunakan sebagai penentuan kelas sentimen dari *tweet testing* dengan fitur n-gram (n=5).

3.5.6 Proses Testing dengan Fitur Kata

Setelah proses *training* dengan fitur kata dilakukan, selanjutnya akan dilakukan proses *testing*. Berikut ini pada tabel 3.12 ditampilkan *tweet* yang akan dilakukan proses *testing* dengan fitur kata.

Tabel 3.12 *Tweet testing* dengan fitur kata

Tweet	**Setelah *Preprocessing* (fitur kata)**
liat kren acara ketawa	liat kren acara ketawa

Untuk penentuan kategori *tweet* pada tabel 3.12 di atas, digunakan persamaan 2.4. untuk penghitungan probabilitas dengan fitur kata ditampilkan pada tabel 3.13 berikut ini:

Tabel 3.13 Penghitungan probabilitas *tweet* dengan fitur kata

k	Fitur (w_k)	$P(w_k \mid positif)$	$P(w_k \mid negatif)$
1	liat	0,142857143	0,25
2	kren	1	1
3	acara	0,142857143	0,25
4	ketawa	0,285714286	0,125
$\prod_k P(w_k \mid c_i)$		0,005830904	0,0078125

$$P(c_{positif} \mid tweet)$$
$$= P(c_{positf}) \prod_k P(w_k \mid c_{positif})$$
$$= 0.5 * 0{,}005830904$$
$$= 0{,}002915452$$

$$P(c_{negatif} \mid tweet)$$
$$= P(c_{negatif}) \prod_k P(w_k \mid c_{negatif})$$
$$= 0.5 * 0{,}0078125$$
$$= 0{,}00390625$$

3.5.7 Proses Testing dengan fitur n-gram (n=2)

Setelah proses *training* dengan fitur n-gram (n=2) dilakukan, selanjutnya akan dilakukan proses *testing*. Berikut ini pada tabel 3.14 ditampilkan *tweet* yang akan dilakukan proses *testing*.

Tabel 3.14 Tweet *testing* dengan fitur n-gram (n=2)

Tweet	Setelah *Preprocessing* (fitur n-gram (n=2))
liat kren acara ketawa	_l, li, ia, at, t_ _k, kr, re, en, n_ _a, ac, ca, ar, ra, a_ _k, ke, et, ta, aw, wa, a_

Untuk penentuan kategori *tweet* pada tabel 3.14 di atas, digunakan persamaan 2.4. untuk penghitungan probabilitas dengan fitur kata ditampilkan pada tabel 3.15 berikut ini:

Tabel 3.15 Penghitungan probabilitas *tweet* dengan fitur dengan n-gram

k	n-gram dari *tweet* yang akan diklasifikasi (w_k)	$P(w_k \mid positif)$	$P(w_k \mid negatif)$
1	_l	0.025	0.045454545
2	li	0.025	0.045454545
3	ia	0.025	0.045454545
4	at	0.025	0.045454545
5	t_	0.025	0.045454545
6	_k	0.075	0.022727272
7	kr	1	1
8	re	0.05	0.022727272
9	en	0.05	0.022727272
10	n_	0.05	0.022727272
11	_a	0.025	0.045454545
12	ac	0.025	0.045454545
13	ca	0.025	0.045454545
14	ar	0.025	0.045454545
15	ra	0.025	0.045454545
16	a_	0.05	0.045454545
17	_k	0.075	0.022727272
18	ke	0.075	0.022727272
19	et	0.05	0.022727272
20	ta	0.05	0.022727272
21	aw	0.05	0.022727272
22	wa	0.05	0.022727272
23	a_	0.05	0.045454545
$\prod_k P(w_k \mid c_i)$		$8x10^{-32}$	$3x10^{-33}$

$P(c_{positif}|tweet)$
$= P(c_{positf}) \prod_k P(w_k|c_{positif})$
$= 0.5 * 8x10^{-32}$
$= 4x10^{-32}$

$P(c_{negatif}|tweet)$
$= P(c_{negatif}) \prod_k P(w_k|c_{negatif})$
$= 0.5 * 3x10^{-33}$
$= 1x10^{-33}$

3.5.8 Proses Testing dengan fitur n-gram (n=3)

Setelah proses *training* dengan fitur n-gram (n=3) dilakukan, selanjutnya akan dilakukan proses *testing*. Berikut ini pada tabel 3.7 ditampilkan *tweet* yang akan dilakukan proses *testing*.

Tabel 3.16 Tweet *testing* dengan fitur n-gram (n=3)

Tweet	**Setelah *Preprocessing* (fitur n-gram (n=3))**
liat kren acara ketawa	_li, lia, iat, at_, t__ _kr, kre, ren, en_, n__ _ac, aca, car, ara, ra_, a__ _ke, ket, eta, taw, awa, wa_, a__

Untuk penentuan kategori *tweet* pada tabel 3.16 di atas, digunakan persamaan 2.4. untuk penghitungan probabilitas dengan fitur kata ditampilkan pada tabel 3.17 berikut ini:

Tabel 3.17 Penghitungan probabilitas *tweet* dengan fitur dengan n-gram

k	n-gram dari *tweet* yang akan diklasifikasi (w_k)	$P(w_k \mid positif)$	$P(w_k \mid negatif)$
1	_li	0.024390244	0.044444444
2	lia	0.024390244	0.044444444
3	iat	0.024390244	0.044444444
4	at_	0.024390244	0.044444444
5	t__	0.024390244	0.044444444
6	_kr	1	1
7	kre	1	1
8	ren	0.048780488	0.022222222
9	en_	0.048780488	0.022222222
10	n__	0.048780488	0.022222222
11	_ac	0.024390244	0.044444444
12	aca	0.024390244	0.044444444
13	car	0.024390244	0.044444444
14	ara	0.024390244	0.044444444
15	ra_	0.024390244	0.044444444
16	a__	0.048780488	0.044444444
17	_ke	0.073170732	0.022222222
18	ket	0.048780488	0.022222222
19	eta	0.048780488	0.022222222
20	taw	0.048780488	0.022222222
21	awa	0.048780488	0.022222222
22	wa_	0.048780488	0.022222222
23	a__	0.048780488	0.044444444
$\prod_k P(w_k \mid c_i)$		$4x10^{-31}$	$8x10^{-32}$

$$P(c_{positif} \mid tweet)$$
$$= P(c_{positf}) \prod_k P(w_k \mid c_{positif})$$
$$= 0.5 * 4x10^{-31}$$
$$= 2x10^{-31}$$

$P(c_{negatif}|tweet)$

$= P(c_{negatif}) \prod_k P(w_k|c_{negatif})$

$= 0.5 * 8x10^{-32}$

$= 4x10^{-32}$

3.5.9 Proses Testing dengan fitur n-gram (n=4)

Setelah proses *training* dengan fitur n-gram (n=4) dilakukan, selanjutnya akan dilakukan proses *testing*. Berikut ini pada tabel 3.18 ditampilkan *tweet* yang akan dilakukan proses *testing*.

Tabel 3.18 Tweet *testing* dengan fitur n-gram (n=4)

Tweet	**Setelah *Preprocessing* (fitur n-gram (n=4))**
liat kren acara ketawa	_lia, liat, iat_, at__, t___ _kre, kren, ren_, en__, n___ _aca, acar, cara, ara_, ra__, a___ _ket, keta, etaw, tawa, awa_, wa__, a___

Untuk penentuan kategori *tweet* pada tabel 3.18 di atas, digunakan persamaan 2.4. untuk penghitungan probabilitas dengan fitur kata ditampilkan pada tabel 3.19 berikut ini:

Tabel 3.19 Penghitungan probabilitas *tweet* dengan fitur dengan n-gram(n=4)

k	**n-gram dari *tweet* yang akan diklasifikasi (w_k)**	$P(w_k\|positif)$	$P(w_k\|negatif)$
1	_lia	0.023809524	0.043478261
2	liat	0.023809524	0.043478261
3	iat_	0.023809524	0.043478261
4	at__	0.023809524	0.043478261
5	t___	0.023809524	0.043478261
6	_kre	1	1
7	kren	1	1

8	ren_	0.047619048	0.02173913
9	en__	0.047619048	0.02173913
10	n___	0.047619048	0.02173913
11	_aca	0.023809524	0.043478261
12	acar	0.023809524	0.043478261
13	cara	0.023809524	0.043478261
14	ara_	0.023809524	0.043478261
15	ra__	0.023809524	0.043478261
16	a___	0.047619048	0.043478261
17	_ket	0.047619048	0.02173913
18	keta	0.047619048	0.02173913
19	etaw	0.047619048	0.02173913
20	tawa	0.047619048	0.02173913
21	awa_	0.047619048	0.02173913
22	wa__	0.047619048	0.02173913
23	a__	0.047619048	0.04347826
$\prod_k \boldsymbol{P}(\boldsymbol{w_k}\|\boldsymbol{c_i})$		$2x10^{-31}$	$5x10^{-32}$

$$P(c_{positif}|tweet)$$
$$= P(c_{positf}) \prod_k P(w_k|c_{positif})$$
$$= 0.5 * 2x10^{-31}$$
$$= 8x10^{-32}$$

$$P(c_{negatif}|tweet)$$
$$= P(c_{negatif}) \prod_k P(w_k|c_{negatif})$$
$$= 0.5 * 5x10^{-32}$$
$$= 2x10^{-32}$$

3.5.10 Proses Testing dengan fitur n-gram (n=5)

Setelah proses *training* dengan fitur n-gram (n=5) dilakukan, selanjutnya akan dilakukan proses *testing*. Berikut ini pada tabel 3.20 ditampilkan *tweet* yang akan dilakukan proses *testing*.

Tabel 3.20 Tweet *testing* dengan fitur n-gram (n=5)

Tweet	Setelah *Preprocessing* (fitur n-gram (n=5))
liat kren acara ketawa	_liat, liat_, iat__, at___, t____ _kren, kren_, ren__, en___, n____ _acar, acara, cara_, ara__, ra___, a____ _keta, ketaw, etawa, tawa_, awa__, wa___, a____

Untuk penentuan kategori *tweet* pada tabel 3.20 di atas, digunakan persamaan 2.4. untuk penghitungan probabilitas dengan fitur kata ditampilkan pada tabel 3.21 berikut ini:

Tabel 3.21 Penghitungan probabilitas *tweet* dengan fitur dengan n-gram(n=4)

k	n-gram dari *tweet* yang akan diklasifikasi (w_k)	$P(w_k \mid positif)$	$P(w_k \mid negatif)$
1	_liat	0.023809524	0.043478261
2	liat_	0.023809524	0.043478261
3	iat__	0.023809524	0.043478261
4	at___	0.023809524	0.043478261
5	t____	0.023809524	0.043478261
6	_kren	1	1
7	kren_	1	1
8	ren__	0.047619048	0.02173913
9	en___	0.047619048	0.02173913
10	n____	0.047619048	0.02173913
11	_acar	0.023809524	0.043478261
12	acara	0.023809524	0.043478261
13	cara_	0.023809524	0.043478261
14	ara__	0.023809524	0.043478261
15	ra___	0.023809524	0.043478261
16	a____	0.047619048	0.043478261
17	_keta	0.047619048	0.02173913
18	ketaw	0.047619048	0.02173913
19	etawa	0.047619048	0.02173913

20	tawa_	0.047619048	0.02173913
21	awa__	0.047619048	0.02173913
22	wa___	0.047619048	0.02173913
23	a____	0.047619048	0.043478261
$\prod_k \boldsymbol{P}(\boldsymbol{w_k}\|\boldsymbol{c_i})$		$2x10^{-31}$	$5x10^{-32}$

$$P(c_{positif}|tweet)$$
$$= P(c_{positf}) \prod_k P(w_k|c_{positif})$$
$$= 0.5 * 2x10^{-31}$$
$$= 8x10^{-32}$$

$$P(c_{negatif}|tweet)$$
$$= P(c_{negatif}) \prod_k P(w_k|c_{negatif})$$
$$= 0.5 * 5x10^{-32}$$
$$= 2x10^{-32}$$

3.5.11 Hasil Proses Testing

Hasil dari perhitungan yang telah dilakukan pada proses *testing* untuk setiap fitur ditampilkan pada tabel 3.22 berikut ini:

Tabel 3.22 Hasil proses testing menggunakan *naive bayes*

Tweet	**Jenis Fitur**	$\boldsymbol{P(c_{positif}\|tweet)}$	$\boldsymbol{P(c_{negatif}\|tweet)}$	**Terkategori**
liat kren acara ketawa	Kata	0,002915452	0,00390625	Negatif
	n-gram(n=2)	$4x10^{-32}$	$1x10^{-33}$	Positif
	n-gram(n=3)	$2x10^{-31}$	$4x10^{-32}$	Positif
	n-gram(n=4)	$8x10^{-32}$	$2x10^{-32}$	Positif
	n-gram(n=5)	$8x10^{-32}$	$2x10^{-32}$	Positif

Berdasarkan hasil proses *testing* yang ditampilkan pada tabel 3.22, dengan fitur kata, nilai tertinggi adalah $P(c_{negatif}|tweet)$, maka *tweet* tersebut termasuk ke dalam sentimen negatif. Pada fitur n-gram (n=2), nilai tertinggi yaitu $P(c_{positif}|tweet)$, maka *tweet* tersebut termasuk ke dalam sentimen positif. Pada

fitur n-gram (n=3), nilai tertinggi yaitu $P(c_{positif}|tweet)$,maka *tweet* tersebut termasuk ke dalam sentimen positif. Pada fitur n-gram (n=4), nilai tertinggi yaitu $P(c_{positif}|tweet)$, maka *tweet* tersebut termasuk ke dalam sentimen positif. Pada fitur n-gram (n=5), nilai tertinggi yaitu $P(c_{positif}|tweet)$,maka *tweet* tersebut termasuk ke dalam sentimen positif. Jika dilihat, sebetulnya *tweet* tersebut termasuk ke dalam sentimen positif. Namun, pada hasil yang ditampilkan pada tabel 3.10 terlihat bahwa terdapat perbedaan hasil klasifikasi antara kedua jenis fitur tersebut. Hal ini disebabkan pada data *training* yang menggunakan fitur kata, kata "kren" (variasi penulisan dari kata "keren") tidak terdapat pada data *training*, yang ada hanya ada kata "keren" (bentuk bakunya), sehingga mempengaruhi dalam perhitungan. Sedangkan untuk fitur n-gram, karena fitur diambil berdasarkan n-karakter katanya, maka terdapat kedekatan n-gram karakter yang sama antara "kren" dan "keren" sehingga mempengarhui pula dalam perhitungan.

3.6 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dilakukan berdasarkan kebutuhan perangkat lunak atau aplikasi analisis sentimen yang akan dibangun. Spesifikasi kebutuhan perangkat lunak dibagi menjadi dua bagian yaitu Spesifikasi Kebutuhan Perangkat Lunak-Fungsional (SKPL-F) dan Spesifikasi Kebutuhan Perangkat Lunak-Non Fungsional (SKPL-NF). Berikut ini adalah tabel spesifikasi beutuhan perangkat lunak fungsional dan non fungsional pada tabel 3.23 dan 3.24.

Tabel 3.23 Spesifikasi kebutuhan perangkat lunak fungsional

Kode	Kebutuhan
SKPL-F01	Aplikasi dapat mengambil *tweet* dari tabel *tweet*
SKPL-F02	Aplikasi dapat melakukan proses *training* dan menampilkan hasilnya
SKPL-F03	Aplikasi dapat melalukan proses *testing* dan menampilkan hasilnya
SKPL-F04	Aplikasi dapat menghitung akurasi dari pengujian (proses *testing*) dan menampilkannya
SKPL-F05	Aplikasi dapat melakukan proses pengujian menggunakan *k-fold cross validation*

Tabel 3.24 Spesifikasi kebutuhan perangkat lunak non fungsional

Kode	Kebutuhan
SKPL-NF01	Perangkat keras yang digunakan adalah komputer dengan spesifikasi Processor 2,27 GHz, RAM 4 GB, Layar dengan resolusi 1366 x 768 pixel, Keyboard standar, Mouse standar
SKPL-NF02	Perangkat lunak yang digunakan adalah sistem operasi Windows 8.1 Professional 32 Bit, bahasa pemrograman PHP, web server XAMPP 1.8.3, editor Sublime Text 3, DBMS MySQL 5.5, framework Twitter Bootstrap, web browser Google Chrome

3.7 Analisis Kebutuhan Non Fungsional

Analisis non fungsional yang dilakukan dibagi ke dalam 2 tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

3.7.1 Analisis Kebutuhan Perangkat Keras

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

1. *Processor* 2,27 GHz
2. RAM 2 GB
3. Layar dengan resolusi 1366 x 768 pixel
4. *Keyboard* standar
5. *Mouse* standar

3.7.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak (software) pada penelitian ini merupakan tools yang digunakan untuk pembangunan aplikasi ini. Adapun tools yang digunakan adalah sebagai berikut:

1. Sistem operasi Windows 8.1 Professional 32 Bit
2. Bahasa pemrograman PHP

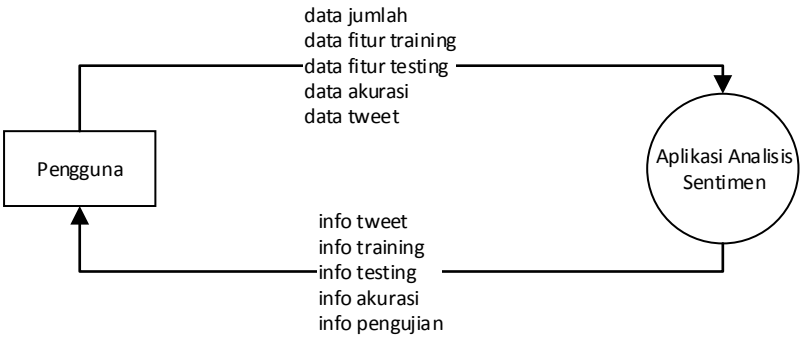
3. Web server XAMPP 1.8.3
4. Editor Sublime Text 3
5. DBMS MySQL 5.5
6. Framework Twitter Bootstrap
7. Web Browser Google Chrome

3.8 Analisis Kebutuhan Fungsional

Dalam bagian ini dilakukan penentuan entitas-entitas yang diperlukan untuk implementasi metode dalam penelitian ini, baik entitas eksternal/internal, data yang mengalir, serta prosedur-prosedur yang bisa dilakukan masing-masing entitas seperti diagram konteks, diagram alir data (*data flow diagram*), spesifikasi proses dan kamus data.

3.8.1 Diagram Konteks

Diagram Konteks atau sering disebut *Context Diagram* (CD) adalah bagian dari *data flow diagram* yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem [7]. Adapun diagram konteks dari aplikasi yang akan dibangun ditampilkan pada gambar 3.18 berikut ini:

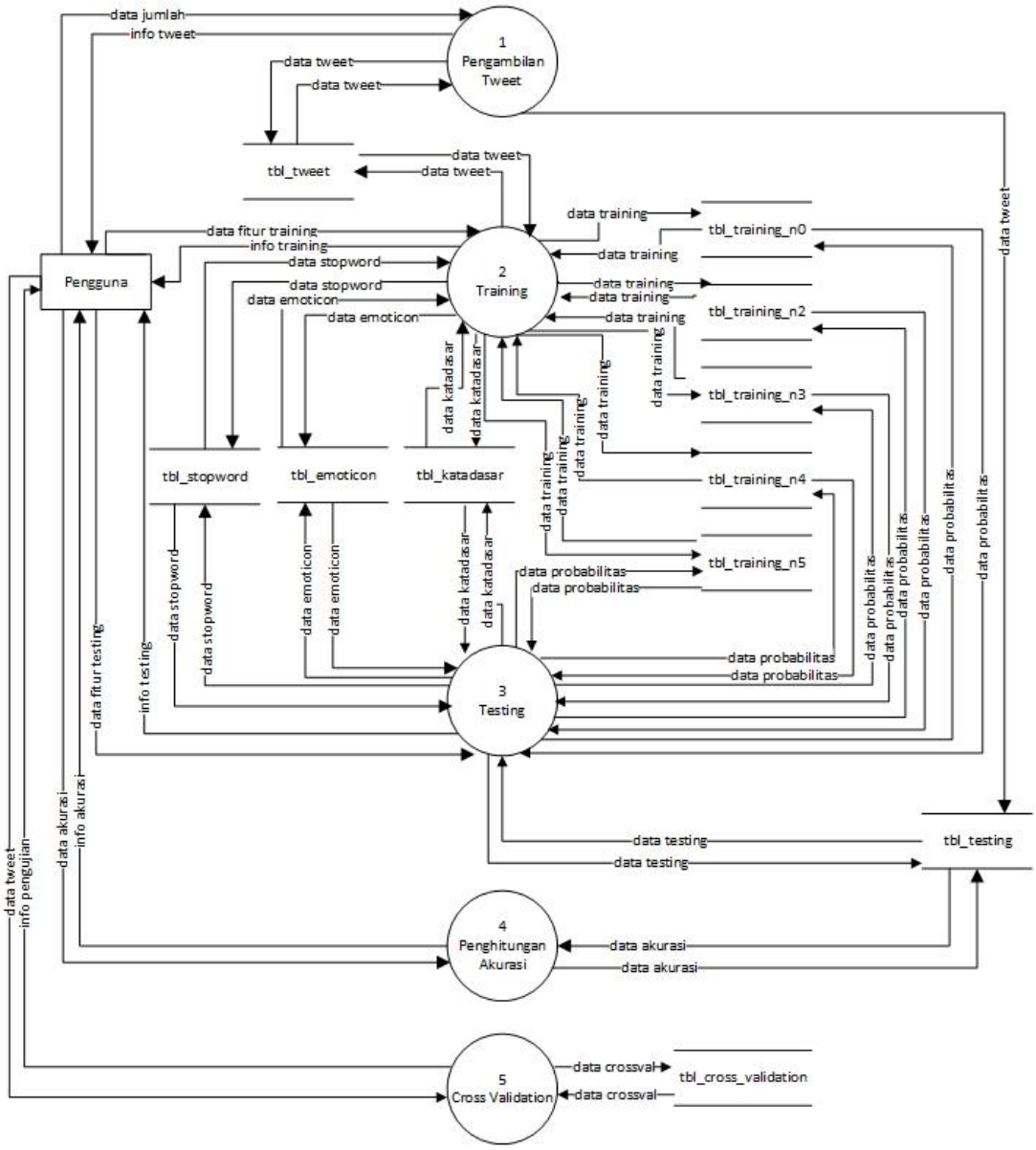


Gambar 3.18 Diagram konteks aplikasi

3.8.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data

tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Adapun Data Flow Diagram dari aplikasi yang akan dibangun ditampilkan pada gambar 3.19 berikut:



Gambar 3.19 *Data flow diagram* aplikasi

3.8.3 Spesifikasi Proses

Proses-proses yang terdapat pada *Data Flow Diagram* akan dijelaskan lebih terperinci pada spesifikasi proses, sebagaimana diperlihatkan pada tabel 3.25 berikut ini:

Tabel 3.25 Spesifikasi proses

No Urut.	Proses	Keterangan
1	No. Proses	1
	Nama Proses	Pengambilan *Tweet*
	Source (sumber)	Pengguna
	Input	Data *tweet*
	Output	Info *tweet*
	Destination	Pengguna, tbl_testing
	Logika Proses	1. Pengguna memilih jumlah *tweet* yang akan diambil, setelah itu menekan tombol "Ambil Tweet". 2. Aplikasi mengambil data *tweet* yang berada pada tbl_tweet sebanyak yang dipilih pengguna. 3. Aplikasi menyimpan *tweet* yang diambil tersebut ke tbl_testing 4. Aplikasi menampilkan semua *tweet* yang diambil tersebut
2	No. Proses	2
	Nama Proses	*Training*
	Source (sumber)	Pengguna
	Input	Data fitur *training*
	Output	info *training*
	Destination	Pengguna
	Logika Proses	1. Pengguna memilih jenis fitur yang digunakan. Setelah itu pengguna menekan tombol "Mulai Training" 2. Aplikasi mengambil semua *tweet* yang berada pada tbl_tweet, lalu melakukan proses *preprocessing* yaitu *convert emoticon*, *cleansing*, *case folding*, *tokenizing*, *convert negation*, *remove stopword* dan *stemming* terhadap *tweet* tersebut.

		3. Pada tahap *convert emoticon*, aplikasi melakukan pengecekan terhadap simbol *emoticon* yang terdapat pada tbl_emoticon, jika pada tweet terdapat simbol tersebut maka ubah ke string yang bersesuaian. Jika tidak, maka tidak dilakukan apa-apa. 4. Pada tahap *cleansing*, aplikasi melakukan pengecekan *tweet* apakah mengandung username (@), URL (http), Retweet (RT), tanda baca dan angka. Jika ada, maka hapus elemen-elemen tersebut. Jika tidak maka tidak dilakukan apa-apa. 5. Pada tahap *case folding*, aplikasi melakukan pengecekan apakah terdapat huruf kapital pada *tweet*. Jika ada, maka ubah menjadi huruf kecil (lowercase). Jika tidak, maka tidak dilakukan apa-apa. 6. Pada tahap *tokenizing*, *tweet* dipecah menjadi kata (token). 7. Pada tahap *convert negation*, aplikasi melakukan pengecekan apakah *tweet* mengandung kata negasi seperti "tidak", "tdk", "nggak", "ga", "gak", "gag", "gk". Jika ada, maka kata negasi tersebut digabungkan dengan kata setelahnya. Jika tidak, maka tidak dilakukan apa-apa. 8. Pada tahap *remove stopword*, aplikasi melakukan pengecekan *tweet* apakah mengandung *stopword* yang terdapat pada tbl_stopword. Jika ada, maka hapus kata tersebut pada *tweet*. Jika tidak, maka tidak dilakukan apa-apa. 9. Pada tahap *stemming*, aplikasi melakukan penyeragaman bentuk kata yang berimbuhan menjadi kata dasarnya dengan menggunakan algoritma porter bahasa Indonesia. 10. Hasil dari tahap *preprocessing* yaitu token. 11. Jika seleksi fitur yang digunakan fitur n-gram, pecah lagi token tersebut menjadi n-karakter token. Jika tidak tidak dilakukan pemecahan lagi. 12. Aplikasi melakukan pencocokan token dengan fitur di database (tbl_training) sesuai dengan jenis fitur yang sedang digunakan.

		13. Jika cocok, maka akan ditambah jumlah frekuensi kemunculannya ke database. Jika tidak, tambah token tersebut sebagai fitur baru. 14. Jika semua token telah dicocokan, hitung jumlah frekuensi semua fitur setiap kategori dan jumlah semua fitur. Jika tidak, kembali ke langkah no 12. 15. Aplikasi menghitung jumlah frekuensi setiap fitur setiap kategori dan jumlah semua fitur. 16. Aplikasi mengambil fitur dan frekuensinya dari database 17. Aplikasi menghitung nilai $P(w_k\|c_i)$ 18. Jika semua sudah dihitung, hitung $P(c_i)$. Jika tidak, kembali ke langkah no 15. 19. Aplikasi menampilkan setiap fitur beserta nilai probabilitas $P(w_k\|c_i)$
3	No. Proses	3
	Nama Proses	*Testing*
	Source (sumber)	Pengguna
	Input	Data fitur *testing*
	Output	Info *testing*
	Destination	Pengguna
	Logika Proses	1. Pengguna menekan tombol "Mulai Testing" untuk melakukan *testing* untuk semua jenis fitur. 2. Aplikasi mengambil semua *tweet* yang berada pada tbl_testing, lalu melakukan proses *preprocessing* yaitu *convert emoticon*, *cleansing*, *case folding*, *tokenizing*, *convert negation*, *remove stopword* dan *stemming* terhadap *tweet* tersebut. 3. Pada tahap *convert emoticon*, aplikasi melakukan pengecekan terhadap simbol *emoticon* yang terdapat pada tbl_emoticon, jika pada tweet terdapat simbol tersebut maka ubah ke string yang bersesuaian. Jika tidak, maka tidak dilakukan apa-apa. 4. Pada tahap *cleansing*, aplikasi melakukan pengecekan *tweet* apakah mengandung username (@), URL (http), Retweet (RT), tanda baca dan angka. Jika ada, maka

		hapus elemen-elemen tersebut. Jika tidak maka tidak dilakukan apa-apa. 5. Pada tahap *case folding*, aplikasi melakukan pengecekan apakah terdapat huruf kapital pada *tweet*. Jika ada, maka ubah menjadi huruf kecil (lowercase). Jika tidak, maka tidak dilakukan apa-apa. 6. Pada tahap *tokenizing*, *tweet* dipecah menjadi kata (token). 7. Pada tahap *convert negation*, aplikasi melakukan pengecekan apakah *tweet* mengandung kata negasi seperti "tidak", "tdk", "nggak", "ga", "gak", "gag", "gk". Jika ada, maka kata negasi tersebut digabungkan dengan kata setelahnya. Jika tidak, maka tidak dilakukan apa-apa. 8. Pada tahap *remove stopword*, aplikasi melakukan pengecekan *tweet* apakah mengandung stopword yang terdapat pada tbl_stopword. Jika ada, maka hapus kata tersebut pada *tweet*. Jika tidak, maka tidak dilakukan apa-apa. 9. Pada tahap *stemming*, aplikasi melakukan penyeragaman bentuk kata yang berimbuhan menjadi kata dasarnya dengan menggunakan algoritma porter bahasa Indonesia. 10. Hasil dari tahap *preprocessing* yaitu token. 11. Jika seleksi fitur yang sedang diproses yaitu fitur n-gram, pecah lagi token tersebut menjadi n-karakter token. Jika tidak tidak dilakukan pemecahan lagi. 12. Aplikasi melakukan pencocokan token dengan fitur di database (tbl_training) sesuai dengan jenis fitur yang sedang digunakan. 13. Jika cocok, ambil nilai $P(w_k\|c_i)$ nya. Jika tidak, nilai $P(w_k\|c_i) = 1$. 14. Jika semua token telah dicocokan, maka hitung nilai hitung nilai c* dari setiap kategori. Jika tidak, maka kembali ke no 21. 15. Jika nilai c* positif > c* negatif maka beri label positif pada *tweet*. Jika tidak, maka beri label negatif pada *tweet* 16. Aplikasi menyimpan hasil pengujian ke tbl_testing.

		17. Aplikasi menampilkan *tweet* hasil klasifikasi sistem.
4.	No. Proses	4
	Nama Proses	Penghitungan Akurasi
	Source (sumber)	Pengguna
	Input	data akurasi
	Output	info akurasi
	Destination	Pengguna
	Logika Proses	1. Pengguna menekan tombol "Tampilkan Akurasi. 2. Aplikasi melakukan pengecekan terhadap hasil pengujian di tbl_testing 3. Jika hasil pengujian ada, maka ke langkah no 4. Jika tidak, muncul pesan gagal (kode pesan P2) 4. Aplikasi melakukan penghitungan jumlah *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN) dari setiap pengujian yang dilakukan (pengujian dengan fitur kata, fitur n-gram (n=2), fitur n-gram (n=3), fitur n-gram (n=4), fitur n-gram (n=5). 5. Aplikasi menghitung akurasi dari setiap pengujian yang dilakukan dengan persemaan 2.7 6. Aplikasi menampilkan tabel informasi hasil pengujian yaitu nilai TP, FP, TN, FN dan akurasinya
5.	No. Proses	5
	Nama Proses	Cross Validation
	Source (sumber)	Pengguna
	Input	Data tweet
	Output	Info akurasi
	Destination	Pengguna
	Logika Proses	1. Pengguna menekan tombol "Cross Validation" 2. Aplikasi melakukan pengecekan data hasil pengujian pada tbl_cross_validation 3. Jika ada data hasil pengujian, maka ke langkah 9. 4. Jika tidak ada, aplikasi membagi dataset (data pada tbl_tweet) menjadi 10 bagian (karena menggunakan 10-fold cross validation) 5. Aplikasi melakukan proses *training* pada 9 bagian dataset

		6. Aplikasi melakukan proses *testing* pada 1 bagian dataset lainnya (selain 9 dataset yang digunakan sebagai data *training*) 7. Aplikasi melakukan penghitungan jumlah *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN) dan akurasi dari pengujian yang dilakukan, lalu disimpan pada tbl_cross_validation 8. Proses 5 sampai 7 dilakukan berulang sampai semua bagian dataset telah dilakukan proses *testing* untuk masing-masing jenis fitur. 9. Aplikasi menampilkan grafik dan tabel akurasi hasil pengujian

3.8.4 Kamus data

Kamus data dapat mendefinisikan dengan lengkap data yang mengalir diantara proses, menyimpan data, dan entitas. Data yang mengalir tersebut dapat berupa masukan untuk sistem atau hasil dari proses sistem. Kamus data dibuat berdasarkan arus data yang mengalir pada Diagram Konteks dan *Data Flow Diagram*, sebagaimana diperlihatkan pada Tabel 3.26 di bawah ini:

Tabel 3.26 Kamus data

Nama	Data jumlah
Where used/how used	Proses 1
Deskripsi	Berisi jumlah *tweet* yang akan diambil untuk dilakukan *testing*
Struktur data	jml_tweet
Penjelasan struktur data	jml_tweet : int
Nama	Data tweet
Where used/how used	Proses 1, Proses 2, Proses 5
Deskripsi	Berisi data *tweet* yang akan diproses baik itu dalam proses *training* maupun proses *testing*
Struktur data	id_tweet+ tweet+sentimen
Penjelasan struktur data	id_tweet : char(18) tweet : string(140) sentimen : enum(0,1) dimana 0 adalah negatif, 1 adalah positif.
Nama	Data fitur training

Where used/how used	Proses 2
Deskripsi	Berisi data jenis fitur *training* yang akan digunakan
Struktur data	jenis_fitur
Penjelasan struktur data	jenis_fitur (0,2,3,4,5,6,7,8,9,10) keterangan : 0 = fitur kata 2 = fitur n-gram (n=2) 3 = fitur n-gram (n=3) 4 = fitur n-gram (n=4) 5 = fitur n-gram (n=5) 6 = fitur n-gram (n=6) 7 = fitur n-gram (n=7) 8 = fitur n-gram (n=8) 9 = fitur n-gram (n=9) 10 = fitur n-gram (n=10)
Nama	Data training
Where used/how used	Proses 2
Deskripsi	Berisi fitur beserta nilai probabilitas setiap fitur tweet
Struktur data	id_fitur+fitur+jml_positif+jml_negatif+prob_positif+prob_negatif
Penjelasan struktur data	id_fitur : int(5) fitur : string(10) jml_positif : int(4) jml_negatif : int(4) prob_positif : double prob_negatif : double
Nama	Data fitur testing
Where used/how used	Proses 3
Deskripsi	Berisi data jenis fitur *testing* yang akan digunakan
Struktur data	jenis_fitur
Penjelasan struktur data	jenis_fitur (0,2,3,4,5,6,7,8,9,10) keterangan : 0 = fitur kata 2 = fitur n-gram (n=2) 3 = fitur n-gram (n=3) 4 = fitur n-gram (n=4) 5 = fitur n-gram (n=5)

	6 = fitur n-gram (n=6) 7 = fitur n-gram (n=7) 8 = fitur n-gram (n=8) 9 = fitur n-gram (n=9) 10 = fitur n-gram (n=10)
Nama	Data testing
Where used/how used	Proses 3
Deskripsi	Berisi data *tweet* yang telah diklasifikasikan oleh aplikasi
Struktur data	id_tweet+ tweet
Penjelasan struktur data	id_tweet : char(18) tweet : string(140)
Nama	Data probabilitas
Where used/how used	Proses 3
Deskripsi	Berisi data probabilitas dari data *training*
Struktur data	prob_positif+prob_negatif
Penjelasan struktur data	prob_negatif : double prob_positif : double
Nama	Data stopword
Where used/how used	Proses 2, Proses 3
Deskripsi	Berisi data stopword
Struktur data	stopword
Penjelasan struktur data	stopword : string(10)
Nama	Data emoticon
Where used/how used	Proses 2 dan proses 3
Deskripsi	Berisi data *emoticon* yaitu simbol *emoticon* dan *string* yang bersesuaian dengan *emoticon* tersebut
Struktur data	emoticon+text
Penjelasan struktur data	emoticon : simbol pada tabel *emoticon* text : string(8)
Nama	Data katadasar
Where used/how used	Proses 2, Proses 3
Deskripsi	Berisi data katadasar
Struktur data	katadasar
Penjelasan struktur data	katadasar : string(20)
Nama	Data akurasi
Where used/how used	Proses 4

Deskripsi	Berisi data hasil klasifikasi *tweet* yang akan dihitung dengan rumus pada persamaan 2.7
Struktur data	sentimen_sistem0+sentimen_sistem2 +sentimen_sistem3+sentimen_sistem4+sentimen_sistem5
Penjelasan struktur data	sentimen_sistem0 : enum(0,1) sentimen_sistem2 : enum(0,1) sentimen_sistem3 : enum(0,1) sentimen_sistem4 : enum(0,1) sentimen_sistem5 : enum(0,1) dimana 0 adalah negatif, 1 adalah positif

3.9 Perancangan Sistem

Pada bagian ini akan dijelaskan perancangan sistem yang untuk mengimplementasikan hasil analisis sebelumnya, dimana sistem yang dibuat akan digunakan untuk mensimulasikan metode yang digunakan. Perancangan sistem yang dibuat terdiri dari perancangan struktur tabel, perancangan menu, perancangan antarmuka, dan jaringan semantik.

3.9.1 Perancangan Struktur Tabel

Struktur tabel menggambarkan detail tabel yang berisi field, tipe data, dan keterangan lainnya. Tabel yang terdapat pada basis data aplikasi yang akan dibangun adalah tabel *tweet* (tbl_tweet), tabel *training* (tbl_training_n0, tbl_training_n2, tbl_training_n3, tbl_training_n4, tbl_training_n5), tabel *testing* (tbl_testing), tabel *stopword* (tbl_stopword), tabel katadasar (tbl_katadasar), tabel *emoticon* (tbl_emoticon) dan tabel *cross validation* (tbl_cross_validation). Berikut ini adalah perancangan dari masing-masing tabel.

1. Tabel Tweet

   Tabel tweet merupakan tabel untuk menyimpan tweet yang akan digunakan sebagai data *training*. Berikut pada tabel 3.27 ditampilkan informasinya.

Tabel 3.27 Tabel *tweet*

Nama field	Tipe data	Ukuran	Keterangan
id_tweet	char	18	Primary key

tweet	varchar	140	tweet
sentimen	enum(0,1)		0 : sentimen negatif 1 : sentimen positif
dataset	int	2	-

2. Tabel Training

Tabel *training* merupakan tabel untuk menyimpan hasil *training* dengan fitur kata. Berikut pada tabel 3.28 ditampilkan informasi dari tabel *training*.

Tabel 3.28 Tabel *training*

Nama field	Tipe data	Ukuran	Keterangan
id_fitur	int	5	Primary key
fitur	varchar	10	fitur
jml_positif	int	4	Frekuensi fitur pada kategori positif
jml_negatif	int	4	Frekuensi fitur pada kategori negatif
prob_positif	double		Probabilitas fitur terhadap kategori positif
prob_negatif	double		Probabilitas fitur terhadap kategori negatif
jenis_fitur	int	2	Jenis fitur: fitur kata dan fitur n-gram

3. Tabel Testing

Tabel *testing* merupakan tabel yang digunakan untuk menyimpan hasil *testing* oleh sistem. Berikut pada tabel 3.33 ditampilkan informasi dari tabel *testing*.

Tabel 3.29 Tabel *testing*

Nama field	Tipe data	Ukuran	Keterangan
id_tweet	char	18	*Primary key*
tweet	varchar	140	Tweet hasil *testing*

sentimen_manual	enum(0,1)		Menyimpan sentimen *tweet* yang diklasifikasikan secara manual 0 : sentimen negatif 1 : sentimen positif
sentimen_sistem0	enum(0,1,2)		Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, *testing* menggunakan fitur kata 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem2	enum(0,1,2)		Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, *testing* menggunakan fitur n-gram (n=2) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem3	enum(0,1,2)		Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, *testing* menggunakan fitur n-gram (n=3) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem4	enum(0,1,2)		Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, *testing* menggunakan fitur n-gram (n=4) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem5	enum(0,1,2)		Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=5)

			0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem6			Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=6) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem7			Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=7) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem8			Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=8) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem9			Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=9) 0 : sentimen negatif 1 : sentimen positif 2 : belum *ditesting*
sentimen_sistem10			Menyimpan sentimen *tweet* yang diklasifikasikan oleh sistem, testing menggunakan fitur n-gram (n=10) 0 : sentimen negatif 1 : sentimen positif

			2 : belum *ditesting*

4. Tabel Stopword

   Tabel *stopword* merupakan tabel yang digunakan untuk menyimpan list *stopword*. Berikut pada tabel 3.34 ditampilkan informasi dari tabel *stopword*.

Tabel 3.30 Tabel *stopword*

Nama field	Tipe data	Ukuran	Keterangan
id	int	5	Primary key
stopword	varchar	10	stopword

5. Tabel Katadasar

   Tabel katadasar merupakan tabel yang digunakan untuk menyimpan list kata dasar. Berikut pada tabel 3.35 ditampilkan informasi dari tabel katadasar.

Tabel 3.31 Tabel katadasar

Nama field	Tipe data	Ukuran	Keterangan
id_katadasar	int	5	Primary key
katadasar	varchar	20	Kata dasar

6. Tabel Emoticon

   Tabel *emoticon* merupakan tabel yang digunakan untuk menyimpan list *emoticon* beserta bentuk *string* yang bersesuaian. Berikut pada tabel 3.36 ditampilkan informasi dari tabel *emoticon*.

Tabel 3.32 Tabel *emoticon*

Nama field	Tipe data	Ukuran	Keterangan
id_emoticon	int	3	Primary key
emoticon	varchar	3	Simbol *emoticon*
text	varchar	6	Bentuk string dari *emoticon*

7. Tabel Cross Validation

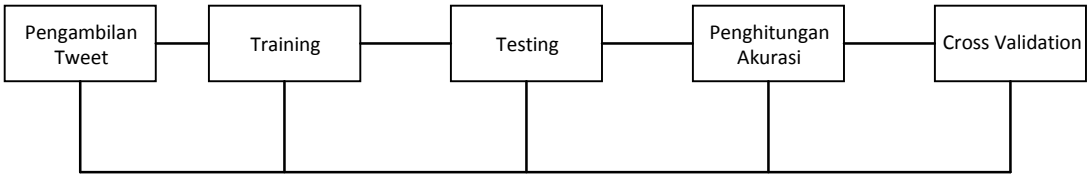
   Tabel *cross validation* merupakan tabel yang digunakan untuk menyimpan hasil proses pengujian dengan *k-fold cross validation*. Berikut pada tabel 3.37 ditampilkan informasi dari tabel *cross validation*.

Tabel 3.33 Tabel *cross validation*

Nama field	Tipe data	Ukuran	Keterangan
id_cross	char	3	*Primary key*
tp	int	3	*True positive*
tn	int	3	*True negative*
fp	int	3	*False positive*
fn	int	3	*False negative*
akurasi	int	3	Persentase akurasi

3.9.2 Perancangan Menu

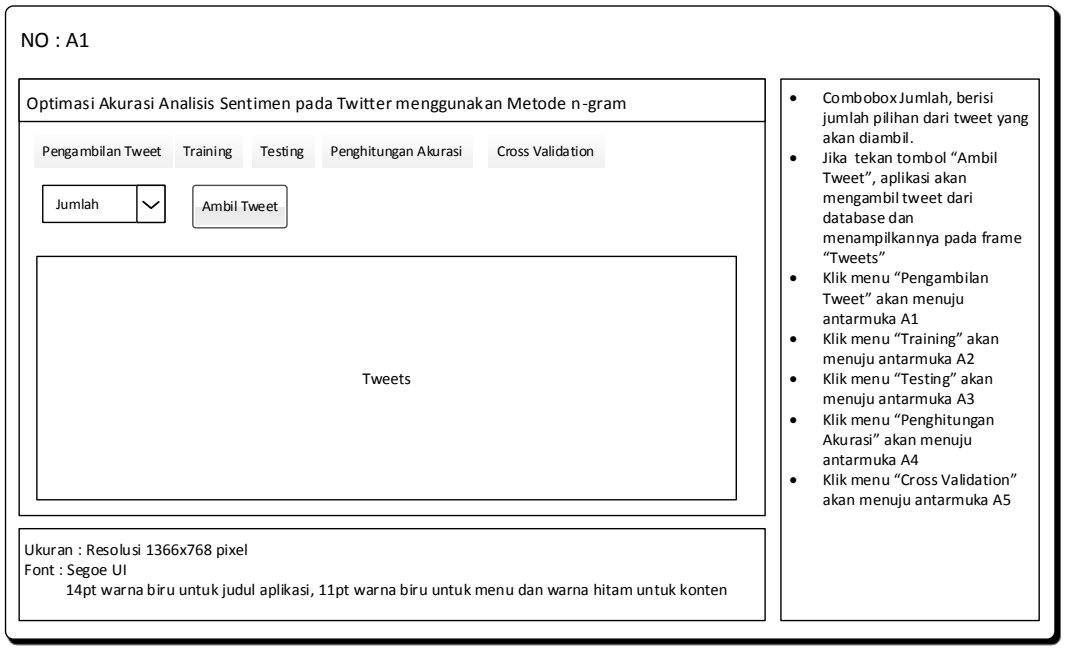
Berikut ini adalah perancangan struktur menu yang akan diterapkan pada aplikasi. Adapun perancangan struktur menu dapat dilihat pada gambar 3.20 di bawah ini.



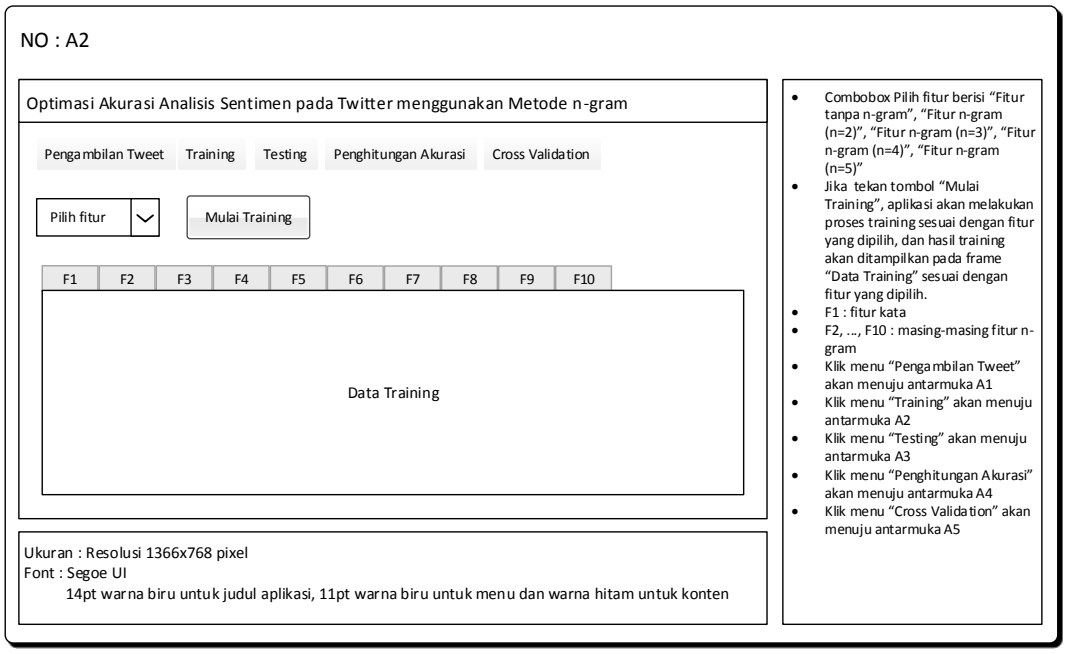
Gambar 3.20 Perancangan menu

3.9.3 Perancangan Antarmuka

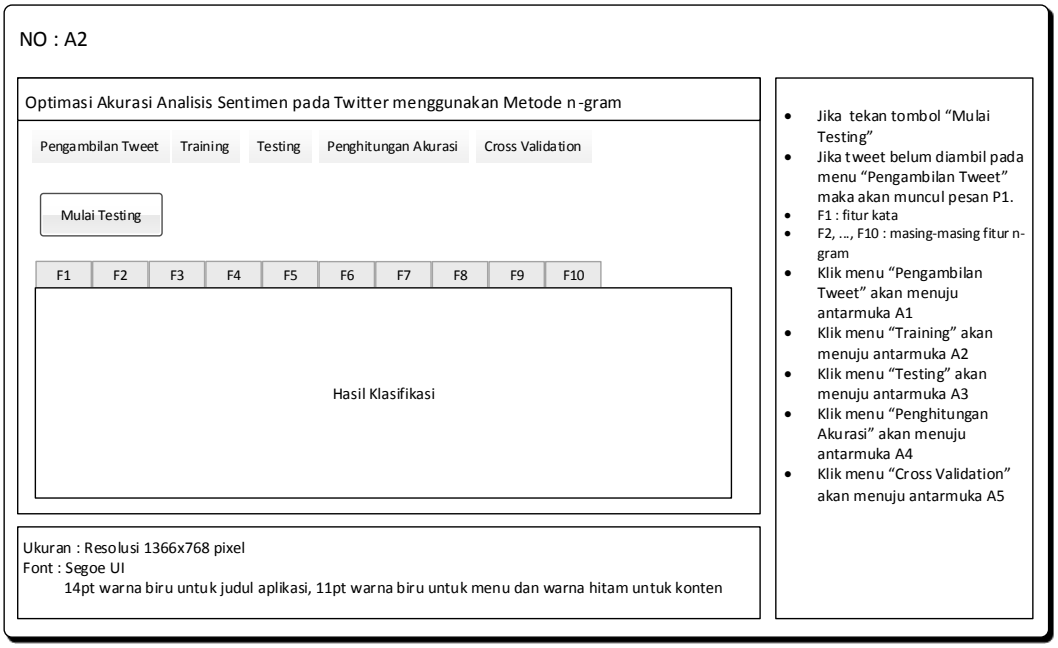
Perancangan antarmuka merupakan suatu bentuk tampilan dari aplikasi yang akan dibuat untuk kebutuhan *interface* dengan pengguna. Adapun perancangan antarmuka dapat dilihat pada gambar 3.21 sampai 3.25 di bawah ini.



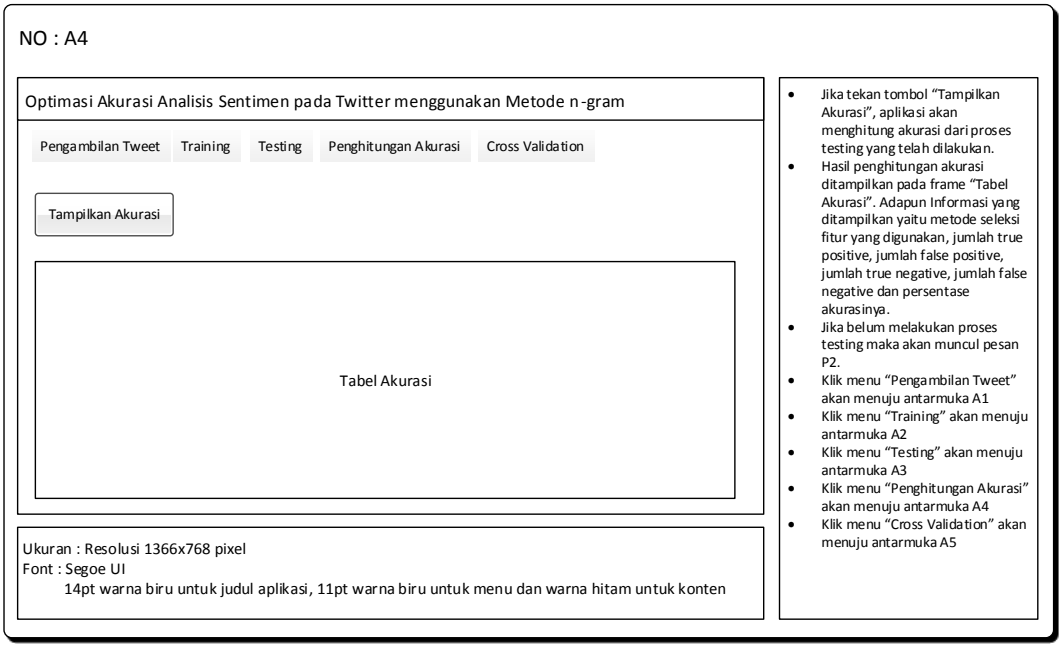
Gambar 3.21 Antarmuka Pengambilan *Tweet*



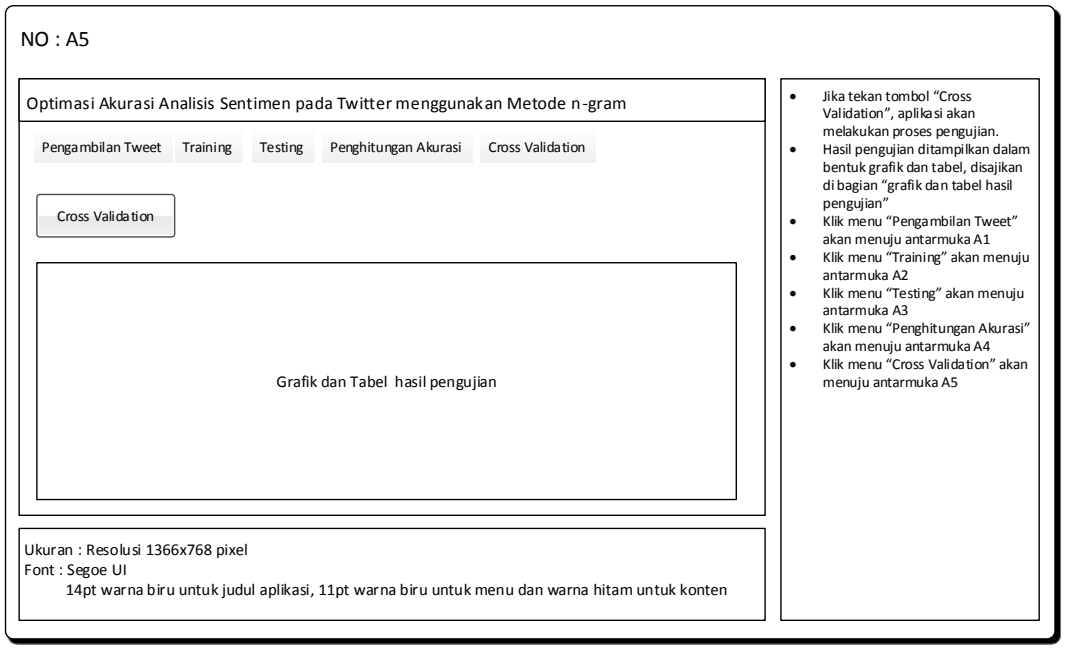
Gambar 3.22 Antarmuka *Training*



Gambar 3.23 Antarmuka *Testing*



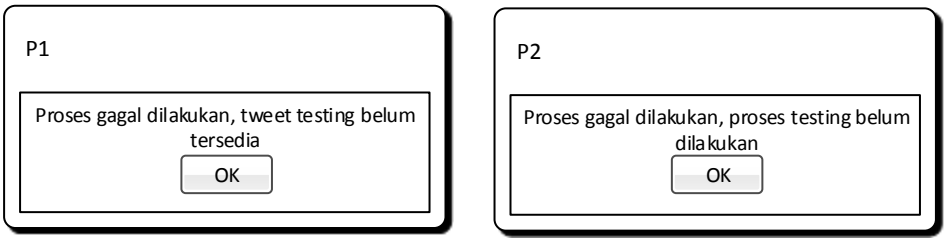
Gambar 3.24 Antarmuka Penghitungan Akurasi



Gambar 3.25 Antarmuka *Cross Validation*

3.9.4 Perancangan Pesan

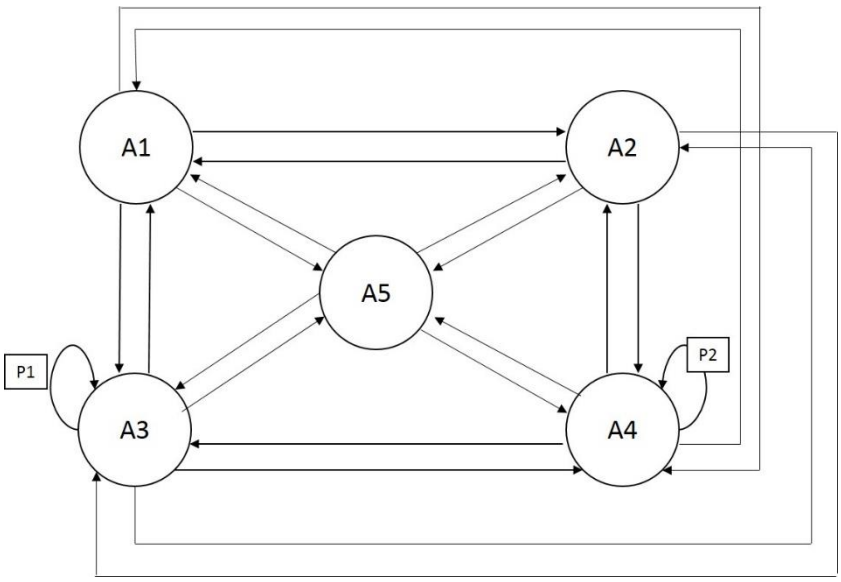
Berikut ini adalah perancangan pesan yang terdapat pada aplikasi ini dtiampilkan pada gambar 3.26:



Gambar 3.26 Perancangan pesan

3.9.5 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan dari navigasi menu dari saru antarmuka ke antarmuka lain. Jaringan semantik yang terbentuk pada sistem ini dapat dilihat pada Gambar 3.27.



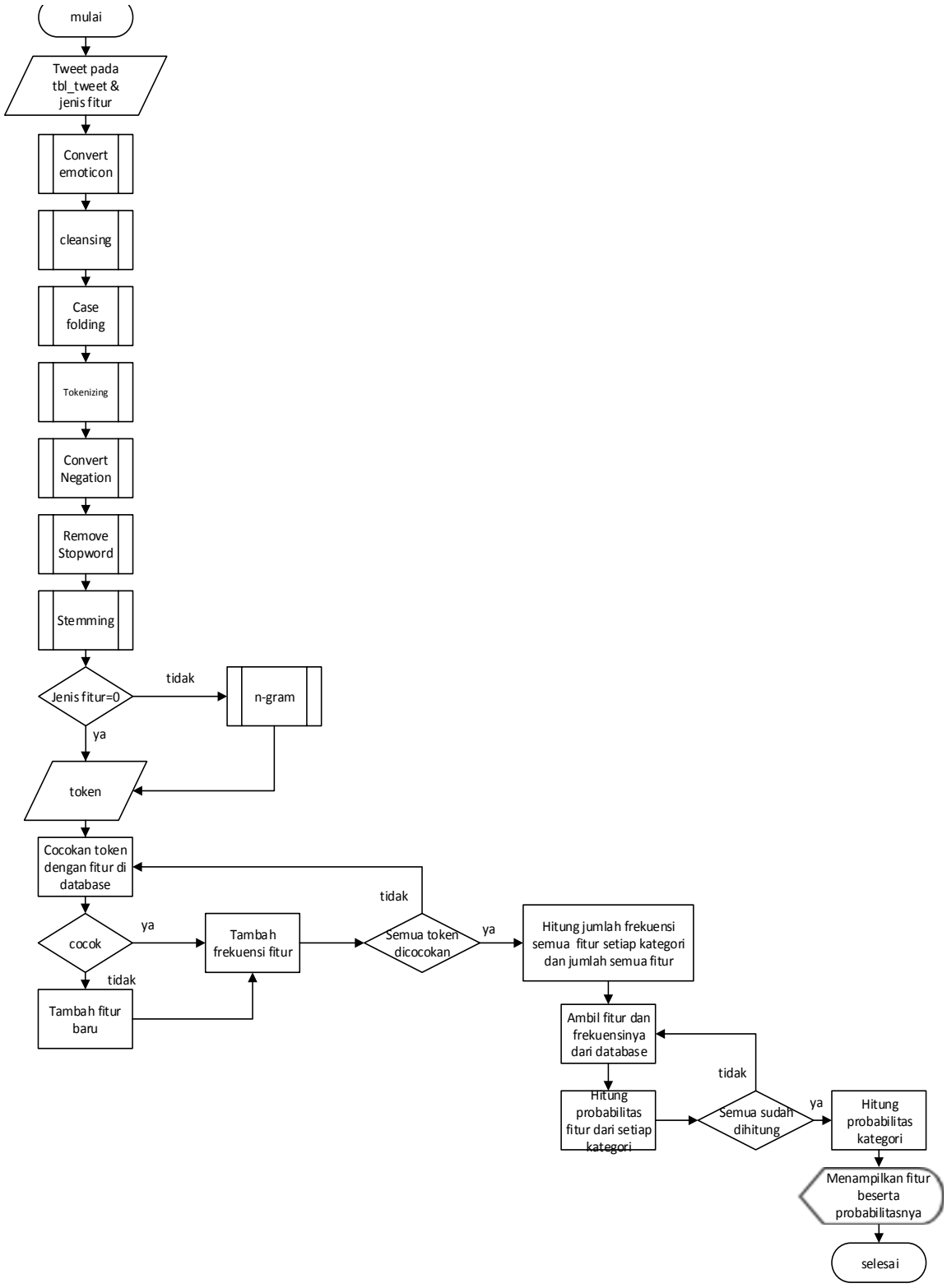
Gambar 3.27 Jaringan semantik

3.9.6 Perancangan Prosedural

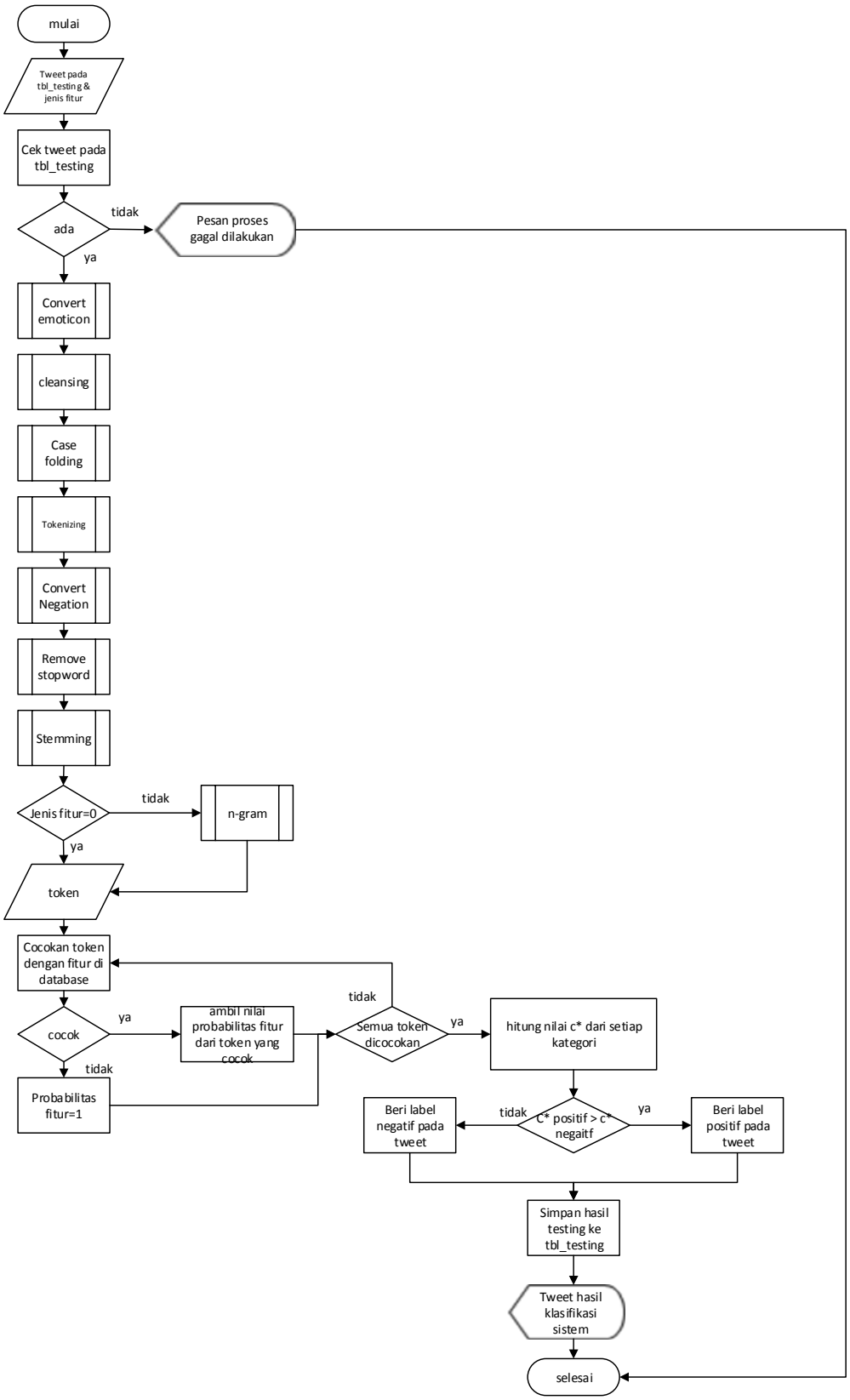
Perancangan prosedural adalah *flowchart* dari tahapan setiap fungsional pada aplikasi. Perancangan prosedural ditampilkan pada gambar 3.28, 3.29, 3.30 dan 3.31 berikut ini:



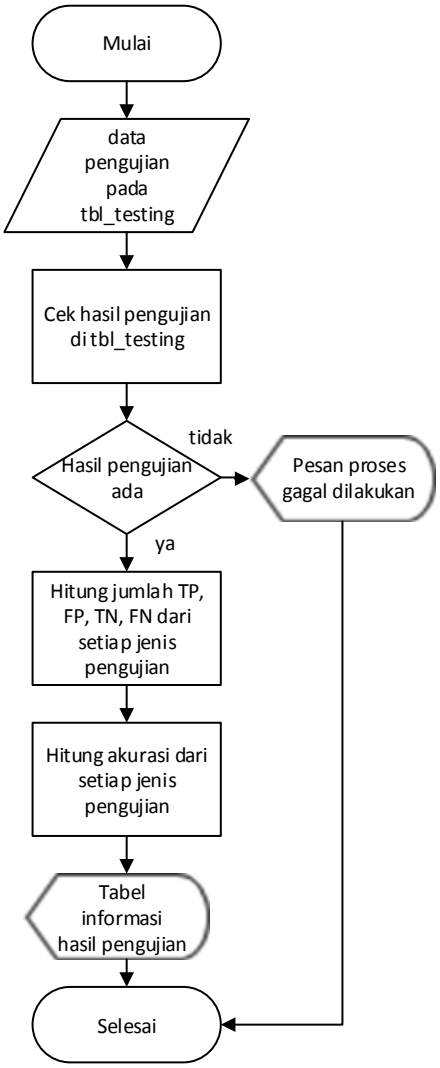
Gambar 3.28 Perancangan prosedural pengambilan *tweet*



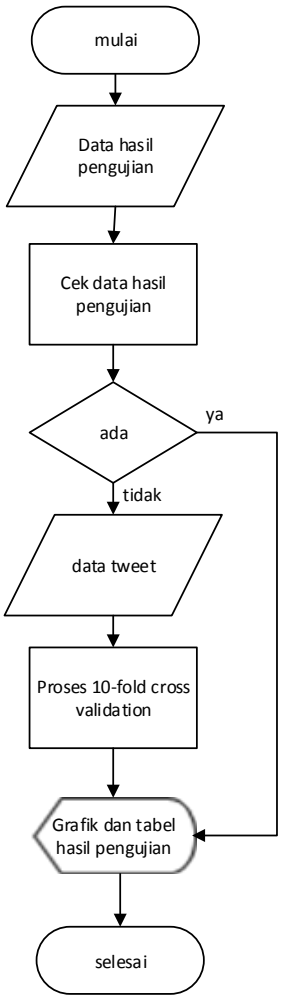
Gambar 3.29 Perancangan prosedural *Training*



Gambar 3.30 Perancangan prosedural *testing*

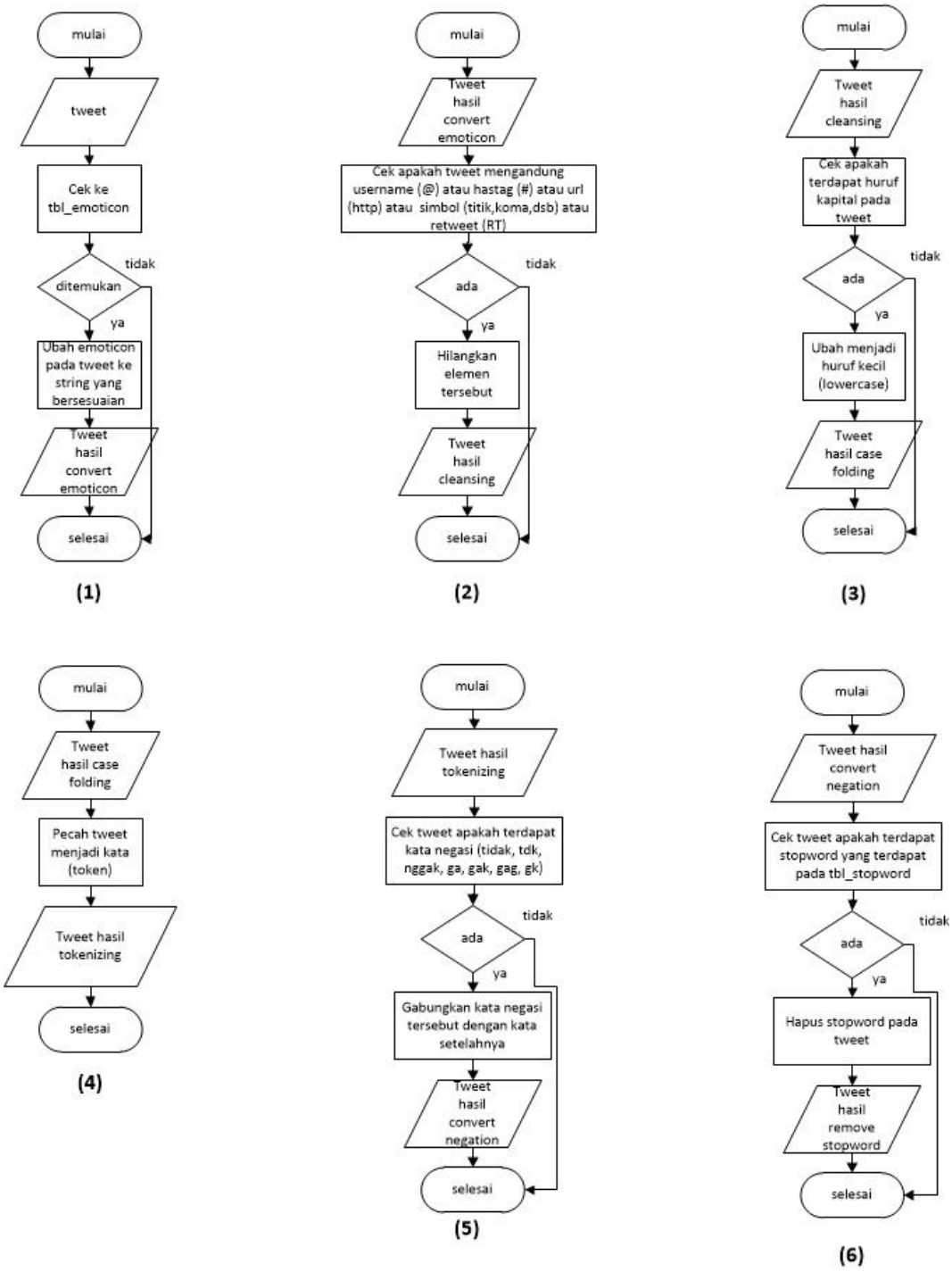


Gambar 3.31 Perancangan prosedural penghitungan akurasi

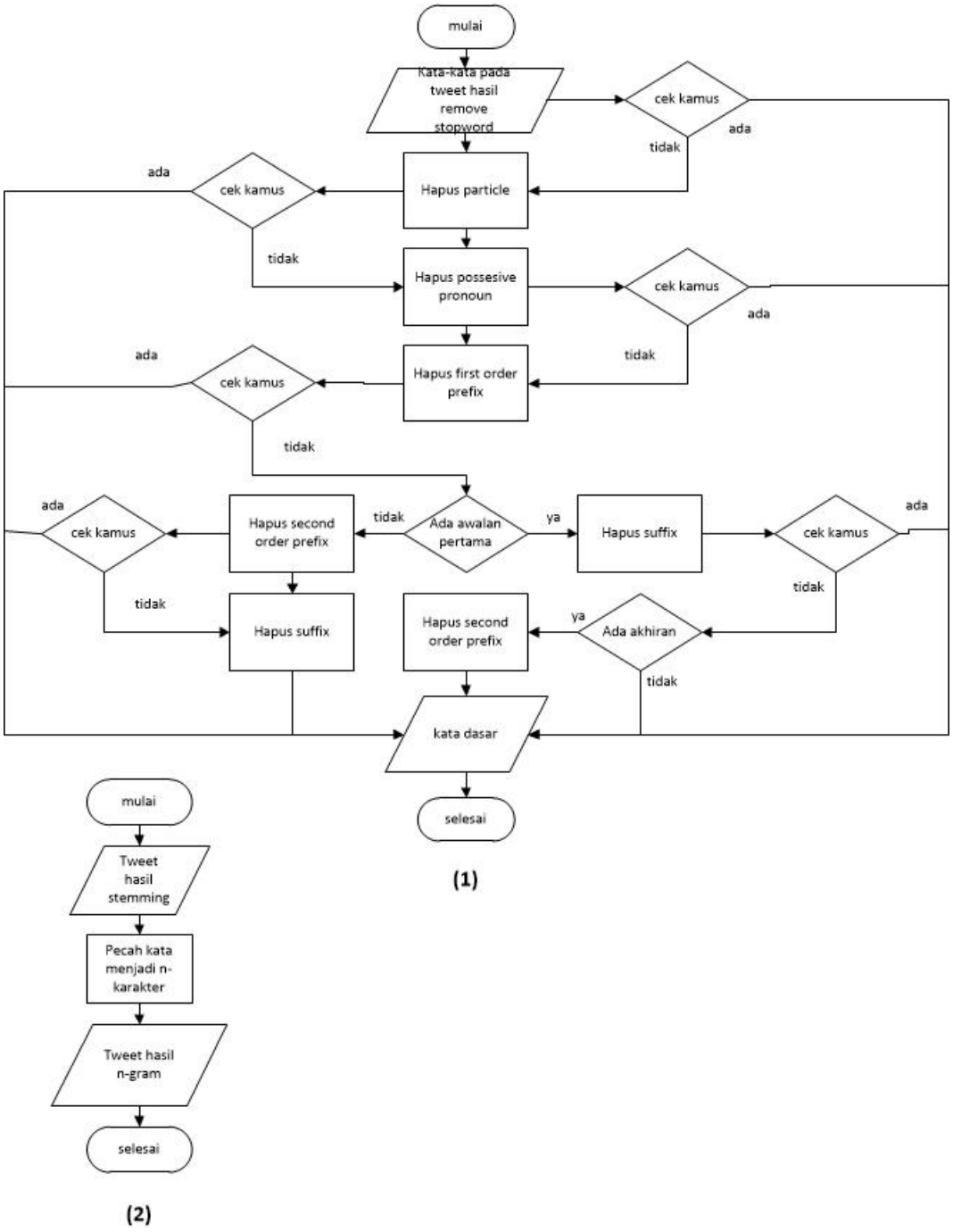


Gambar 3.32 Perancangan prosedural *cross validation*

Untuk sub proses tahapan *preprocessing* (*convert emoticon*, *cleansing*, *case folding*, *tokenizing*, *convert negation*, *remove stopword*, *stemming* dan ditampilkan pada gambar 3.33 dan 3.34 berikut:



Gambar 3.33 Perancangan prosedural *convert emoticon* (1), *cleansing* (2), *case folding* (3), *tokenizing* (4), *convert negation*(5), *remove stopword* (6)



Gambar 3.34 Perancangan prosedural *stemming*(1), dan n-gram (2)