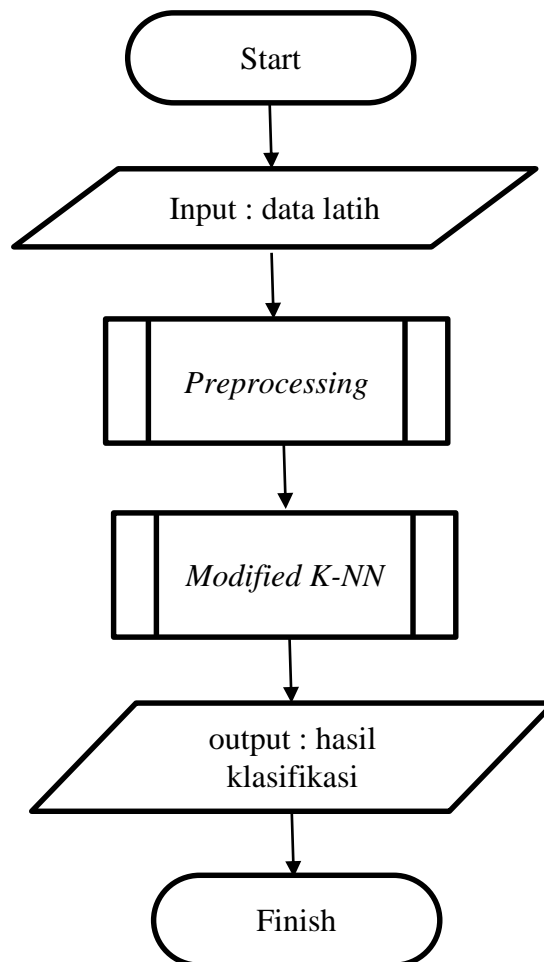


BAB 5 PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perancangan sistem yang akan dikembangkan. Dalam bab Perancangan dijelaskan diagram *flowchart*, proses perhitungan manualisasi, perancangan antarmuka pengguna, dan rancangan skenario pengujian yang dilakukan terhadap sistem.

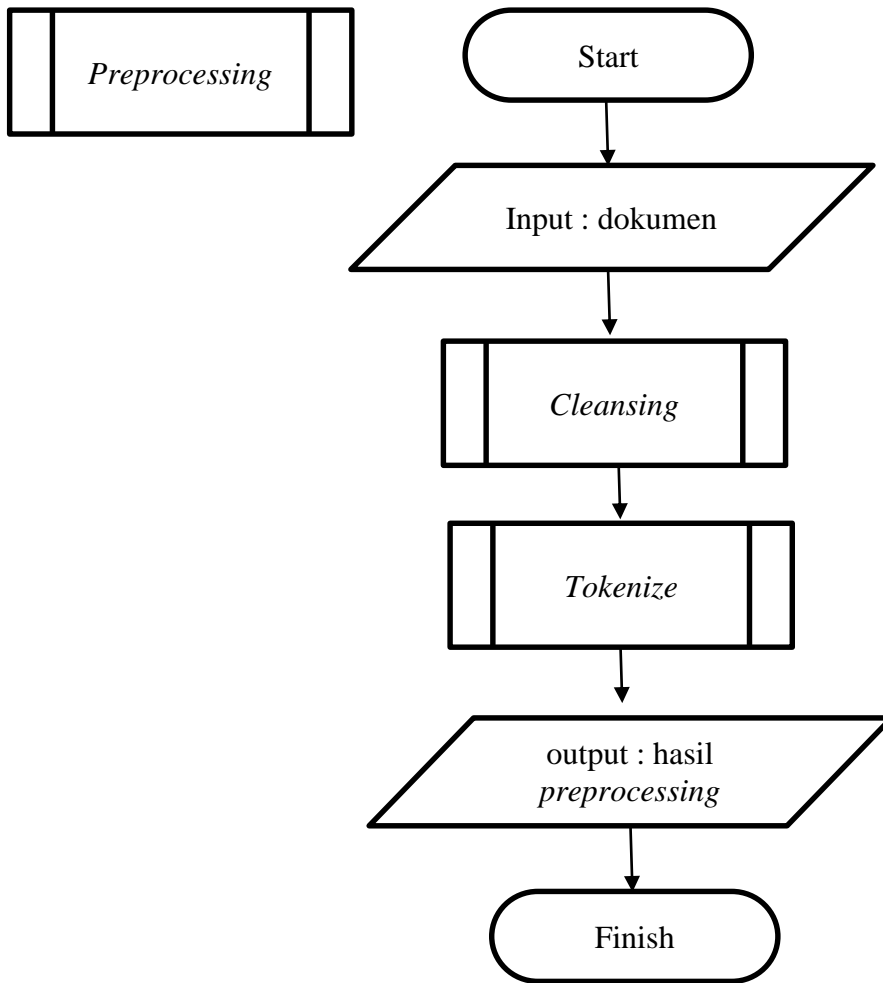
1.1 Perancangan Sistem

Berikut adalah diagram flowchart dari sistem secara keseluruhan



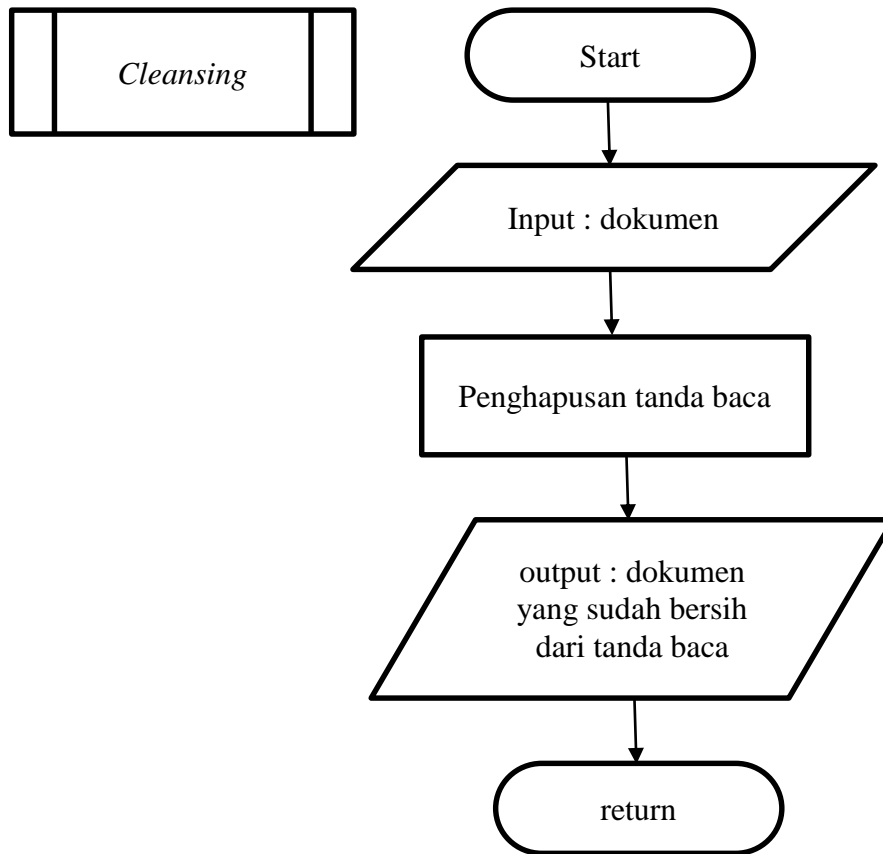
Gambar 4.1 Rancangan sistem keseluruhan

Sistem dimulai dengan inputan berupa data latih atau data uji, yang kemudian data tersebut akan mengalami proses *preprocessing* sebelum masuk dalam metode klasifikasi *Modified K Nearest Neighbour*. Hasil *output* akan berupa klasifikasi data.



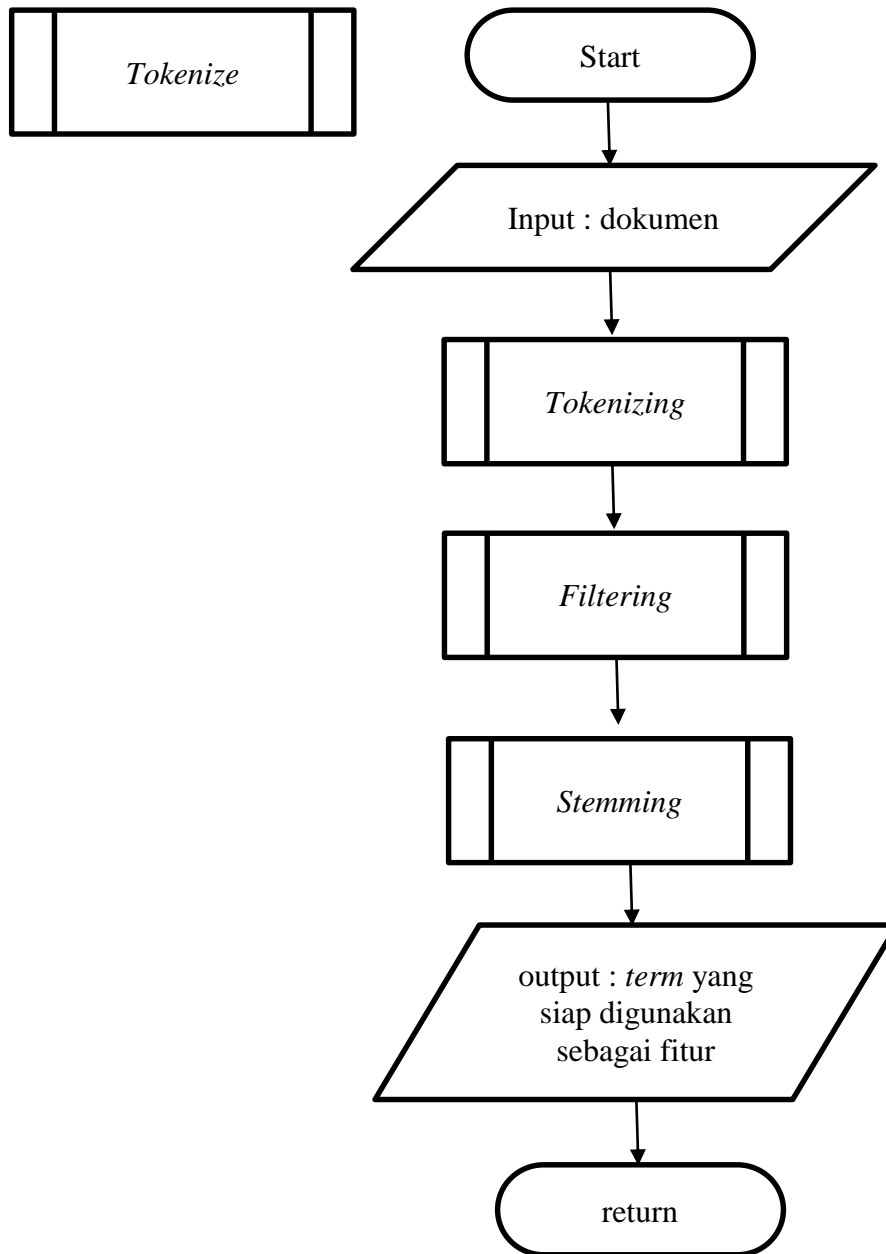
Gambar 4.2 Rancangan Proses Preprocessing

Gambar 4.2 menjelaskan tentang diagram alur proses *Preprocessing* yang dilakukan pada tahap awal proses klasifikasi sentimen. Proses *preprocessing* dimulai dengan *input* dokumen. Dokumen akan mengalami beberapa tahap untuk didapatkan kata yang akan digunakan sebagai fitur. Tahapan tersebut diantaranya adalah proses *Cleansing*, dan *Tokenize*.



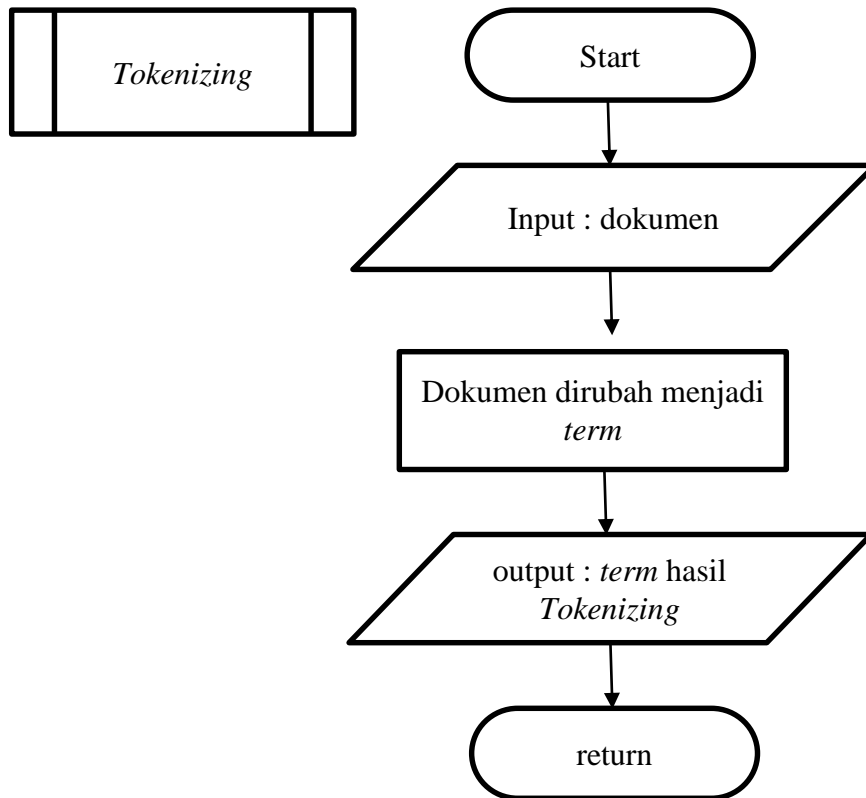
Gambar 4.3 Rancangan Proses Cleansing

Gambar 4.3 merupakan proses *cleansing* atau pembersihan dokumen dari macam-macam tanda baca yang ada. Contoh dari tanda baca yang akan dihilangkan adalah seperti koma (,), titik (.), tanda tanya (?), tanda seru (!), dan lain sebagainya. Sehingga yang akan digunakan untuk penelitian hanyalah dokumen yang sudah bersih dari tanda baca tersebut untuk masuk pada proses selanjutnya yakni *Tokenize*.



Gambar 4.3 Rancangan proses Tokenize

Gambar 4.3 merupakan kumpulan proses yang didalamnya terdapat beberapa proses yakni *Tokenizing*, *Filtering*, dan *Stemming*. Hasil dari proses Tokenize adalah kumpulan term yang telah terseleksi untuk digunakan sebagai fitur pada tahap selanjutnya, yakni perhitungan nilai TF-IDF.



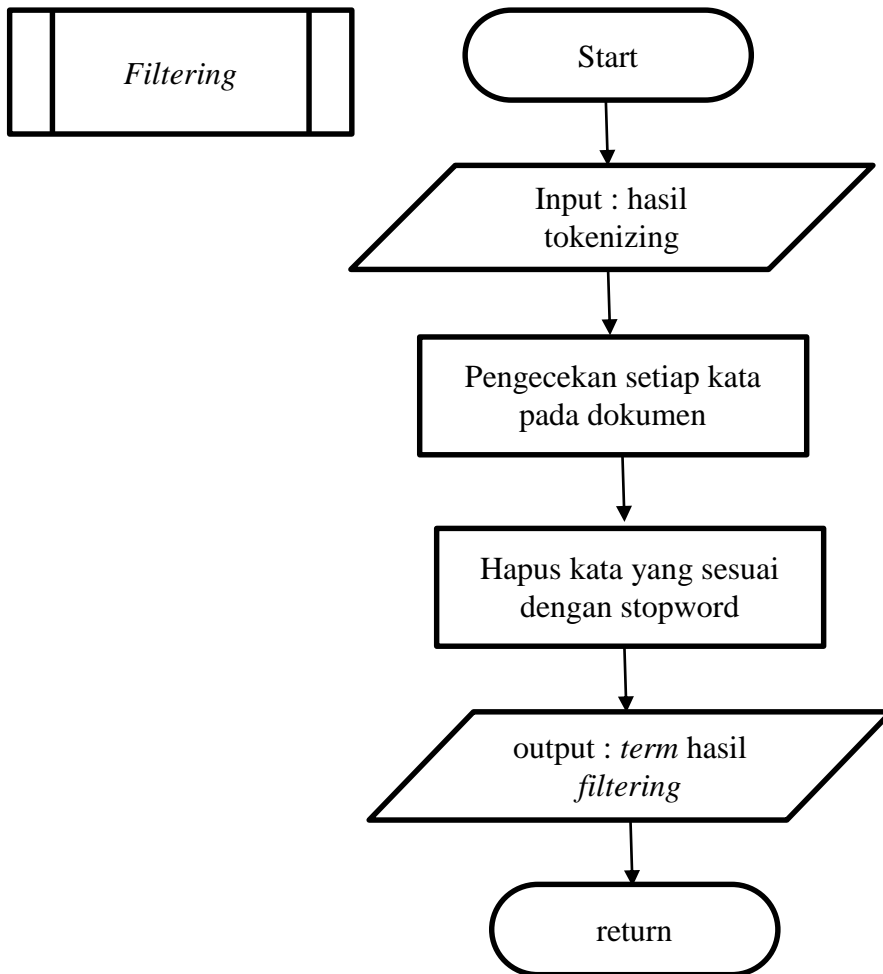
Gambar 4.4 Rancangan proses Tokenizing

Gambar 4.4 menjelaskan proses tokenizing. Proses *Tokenizing* merupakan proses mengubah dokumen menjadi sebuah kata (*term*) yang menyusun dokumen tersebut. dimulai dengan *input* dokumen, kemudian dokumen dirubah menjadi *term* penyusun dokumen dan akan menghasilkan keluaran berupa *term*. Dalam proses ini, inputan berupa dokumen asli atau berupa suatu kalimat. Dari kalimat tersebut akan dipecah menjadi beberapa kata atau *term* berdasarkan spasi yang ada pada dokumen tersebut. kata inilah yang nantinya akan digunakan oleh sistem sebagai fitur untuk melakukan proses selanjutnya. Contoh :

Dokumen : “Ini merupakan suatu aplikasi yang sangat membantu dalam kehidupan sehari-hari saya.”

Hasil tokenizing : “ini”, “merupakan”, “suatu”, “aplikasi”, “ yang”, “sangat”, “membantu”, “dalam”, “kehidupan”, “sehari-hari”, “saya”.

Terdapat 10 spasi pada dokumen, sehingga jumlah kata yang dihasilkan ada 11 kata atau *term*.



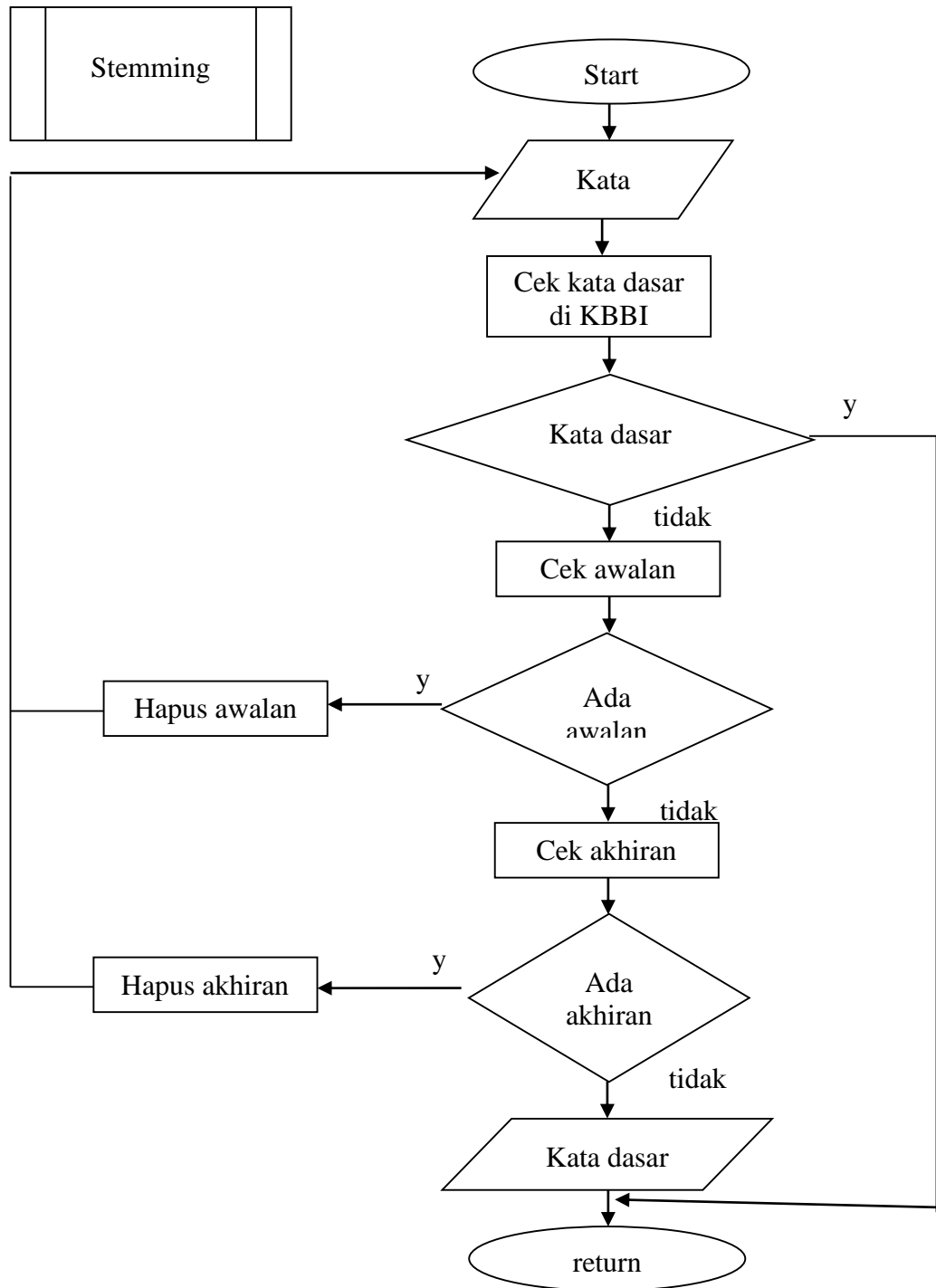
Gambar 4.5 Rancangan Proses Filtering

Gambar 4.5 merupakan Rancangan Proses Filtering. Proses Filtering adalah proses menghilangkan kata yang termasuk dalam stopwords/stoplist sehingga diperoleh term yang memiliki ciri khusus dari kelas hasil sentimen. Contoh dari kata yang termasuk dalam Stopword/stoplist adalah kata sambung, artikel. Proses ini berfungsi untuk menghilangkan kata yang dianggap tidak penting oleh sistem. Dimana kata-kata tersebut kurang berpengaruh apabila ada pada suatu sentimen atau dokumen. Kata-kata tersebut sudah terangkum dalam stoplist yang disediakan oleh pengembang sehingga sistem dapat mengaksesnya secara langsung untuk mempercepat proses perhitungan. Contoh :

Dokumen asli : hari ini merupakan hari yang cerah.

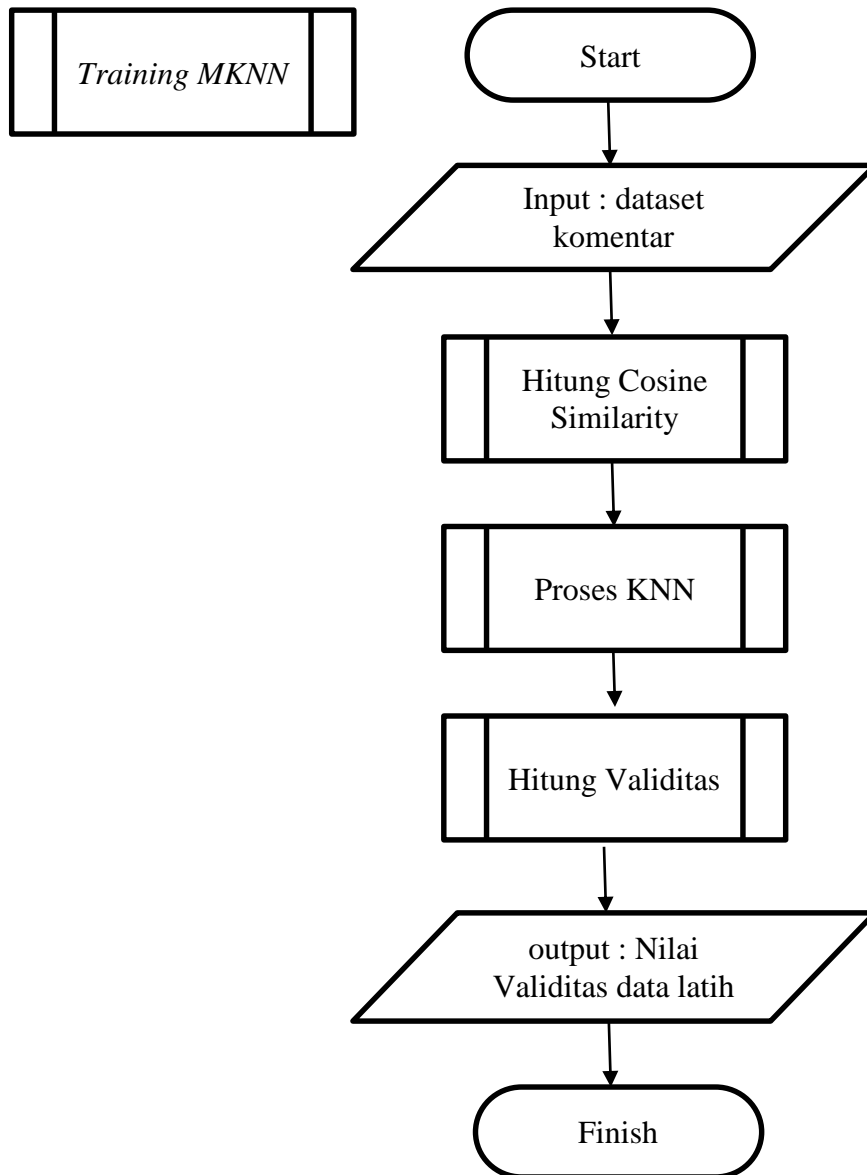
Dokumen hasil *filtering* : cerah.

Pada dokumen tersebut, terdapat beberapa kata yang sama dengan kata yang terdaftar dalam stoplist. Sehingga menyisakan 1 kata yang bisa digunakan untuk proses selanjutnya.



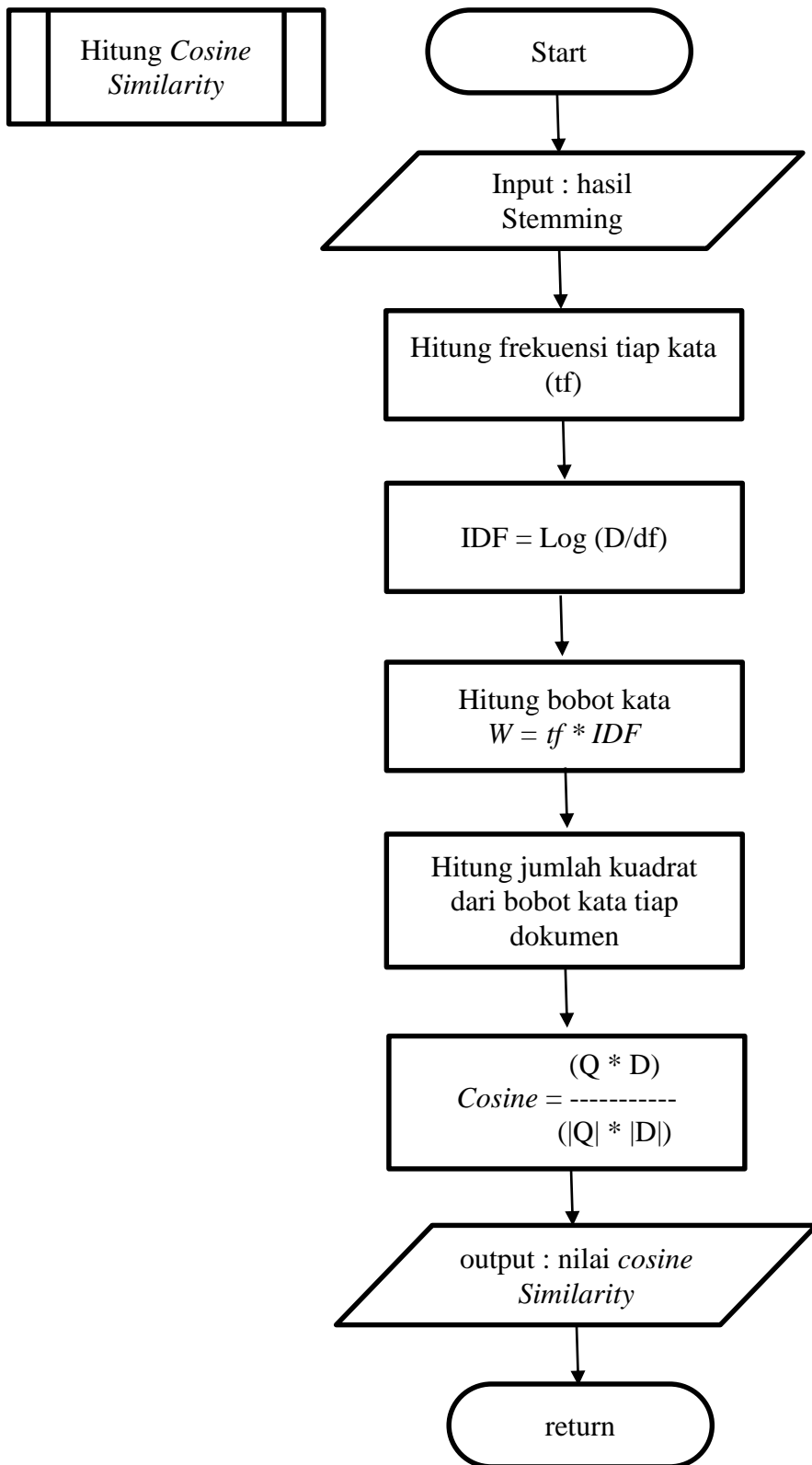
Gambar 4.6 Rancangan Proses Stemming

Gambar 4.6 merupakan rancangan proses *Stemming*. Proses *Stemming* adalah pengambilan kata dasar dari *term* yang sudah diperoleh pada proses *Filtering*. Dimulai dengan inputan hasil *filtering* kemudian *term* hasil *filtering* dirubah menjadi kata dasar yang merujuk pada Kamus Besar Bahasa Indonesia. Sehingga menghasilkan *output* berupa kata dasar yang siap digunakan sebagai fitur pada proses klasifikasi. Contoh proses *Stemming* adalah misalnya ‘permainan’ akan menjadi ‘main’.

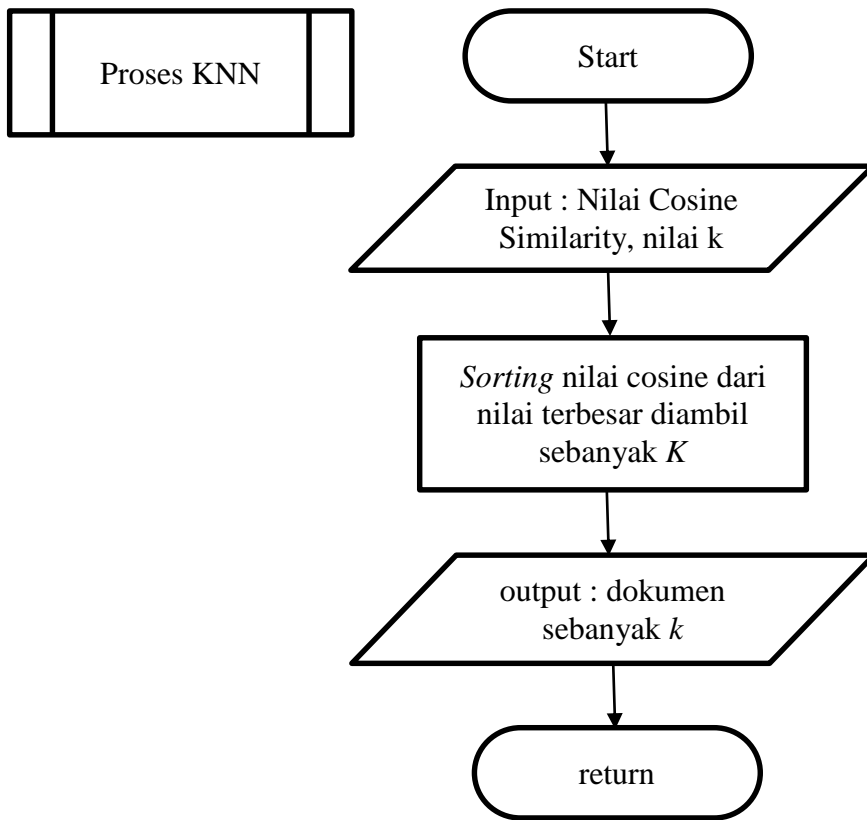


Gambar 4.7 Rancangan Proses Training MKNN

Gambar 4.7 Rancangan Proses MKNN. Proses tersebut adalah suatu proses yang digunakan untuk mencari nilai validitas dari setiap data latih yang ada. Setiap data latih dicari nilai *cosine similarity* nya terhadap semua data latih yang lain. Kemudian dilakukan proses *sorting* untuk mengurutkan data berdasarkan nilai *cosine similarity* yang terbesar. Setelah data terurut, dihitung nilai validitasnya berdasarkan data yang termasuk dalam k terdekat. dari sini diperoleh nilai validitas yang akan digunakan dalam proses perhitungan *weight voting* pada proses pengujian.

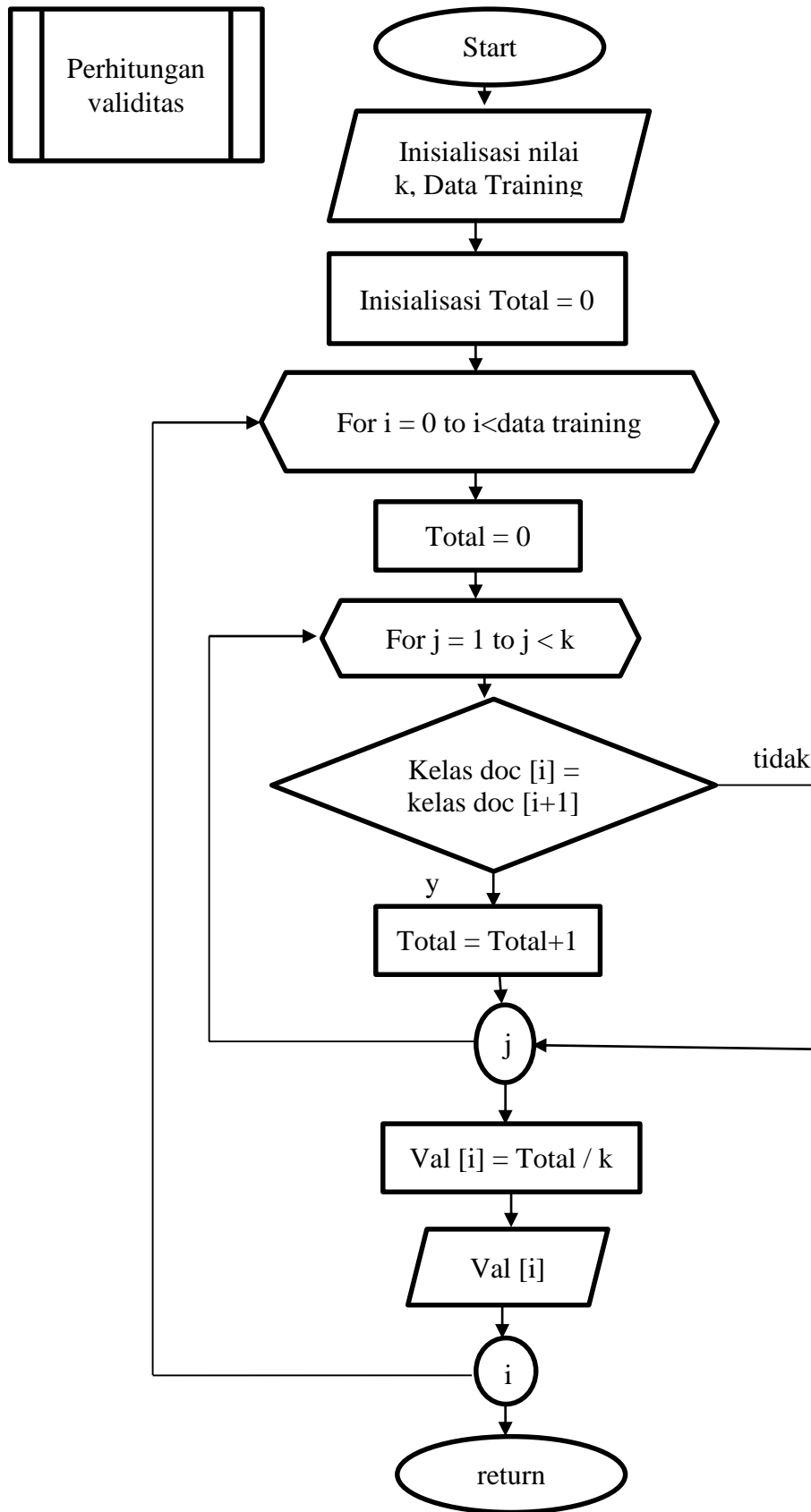


Gambar 4.8 Rancangan Proses Perhitungan Nilai Cosine Similarity

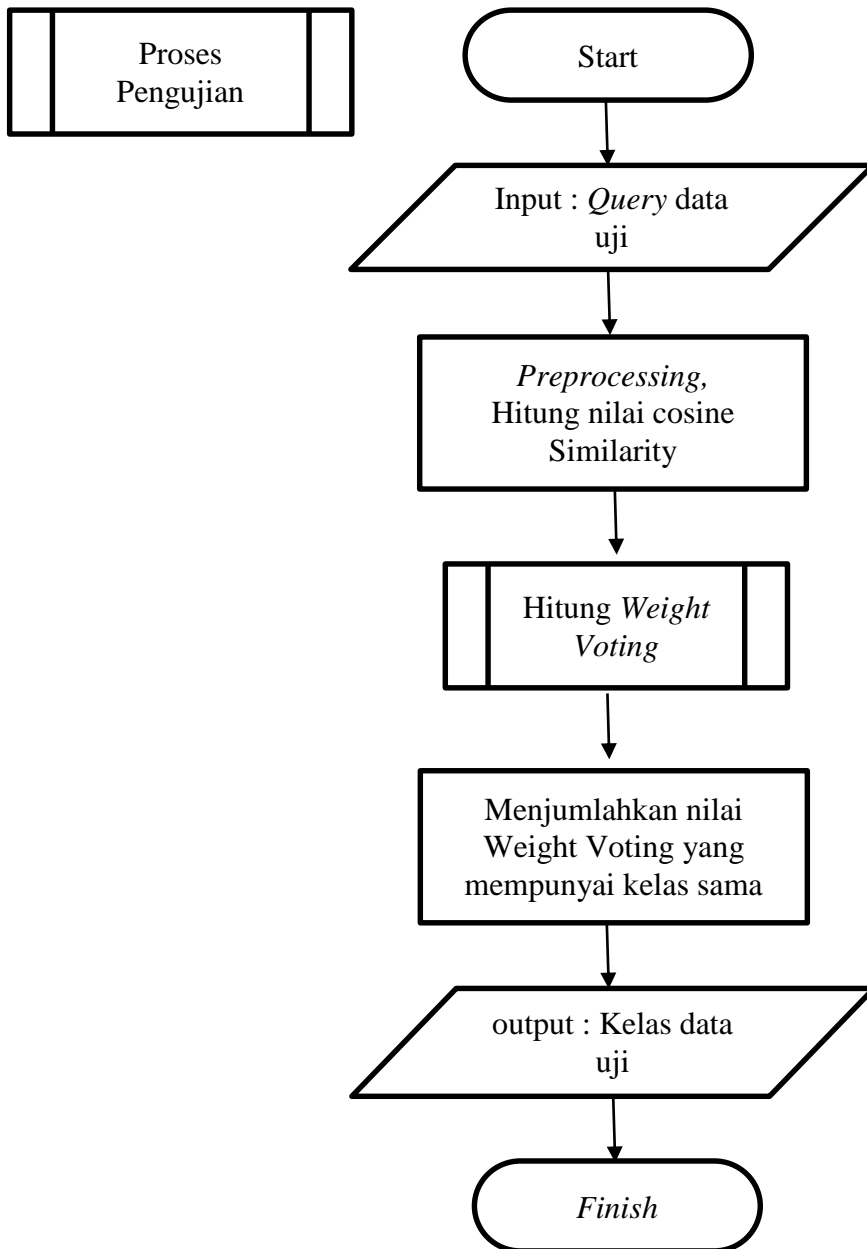


Gambar 4.9 Proses KNN

Gambar 4.9 merupakan alur dari rancangan proses *K Nearest Neighbour*. Pada proses ini terjadi pengurutan yang dilakukan pada dokumen berdasarkan nilai cosine similarity yang telah dihitung sebelumnya. Jumlah data yang diurutkan tergantung pada nilai k yang diinputkan. Proses pengurutan dilakukan secara terurut dari nilai cosine similarity terbesar sampai nilai cosine similarity terkecil. Inputan yang dibutuhkan oleh sistem berupa nilai cosine similarity dan nilai k . Keluaran yang dihasilkan adalah suatu dokumen yang telah terurutkan. Output dari proses ini nantinya akan digunakan sebagai acuan pada proses perhitungan validasi. Karena dalam proses perhitungan validasi tersebut perlu dicek apakah kelas yang terdekat dari suatu dokumen memiliki kelas yang sama dengan kelas dari dokumen itu sendiri.

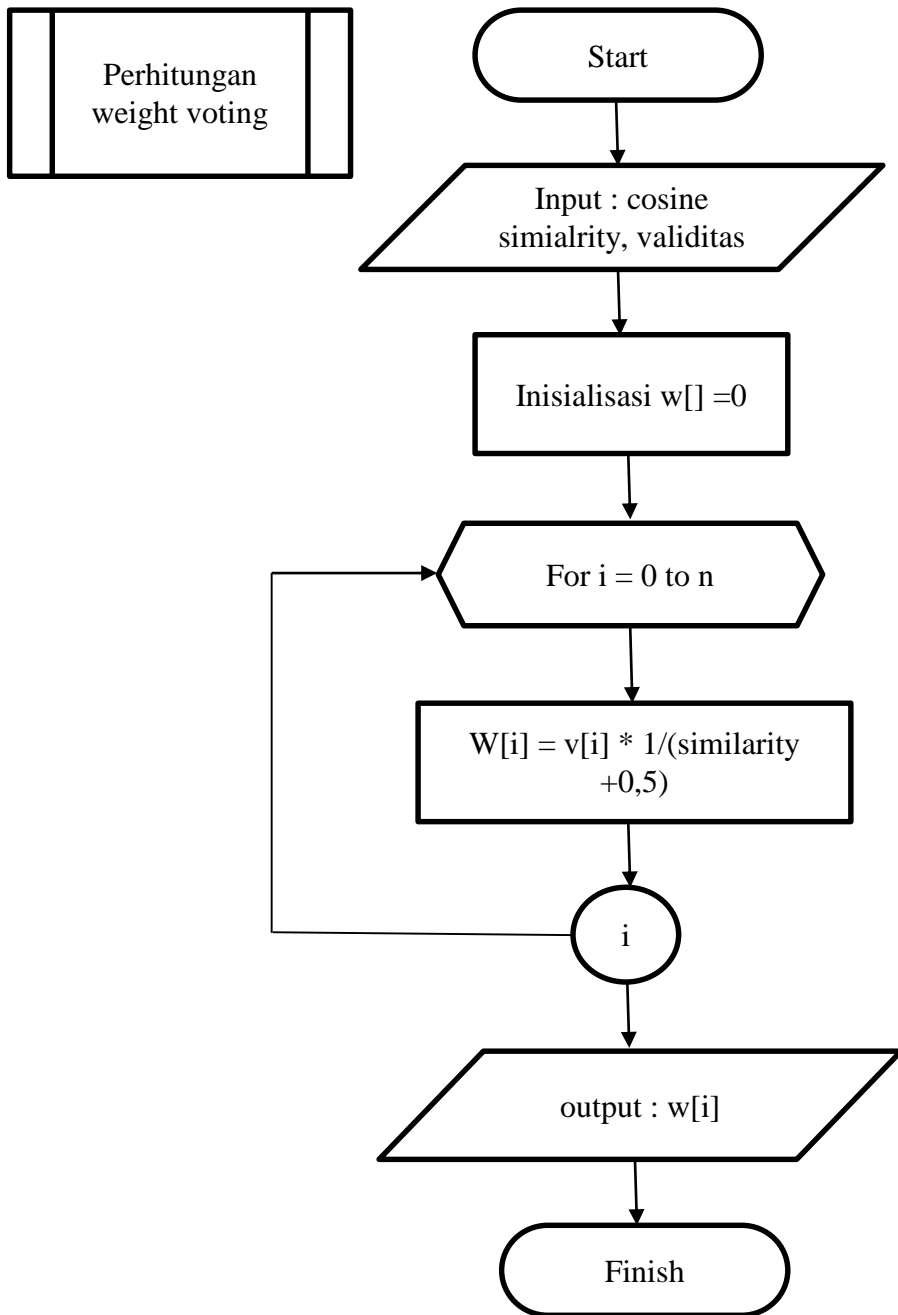


Gambar 4.10 Rancangan proses perhitungan validitas



Gambar 4.11 Proses Pengujian

Gambar 4.11 merupakan alur dari rancangan proses pengujian. Dalam proses ini, langkah awal adalah dokumen uji mengalami preprocessing yang kemudian dihitung nilai cosine similarity nya. Kemudian dari nilai cosine similarity tersebut akan dihitung nilai weight voting yang akan digunakan sebagai nilai dalam menentukan kelas dari dokumen uji.



Gambar 4.12 Rancangan Proses perhitungan weight voting

Pada perhitungan *weight voting* dibutuhkan *inputan* berupa nilai validitas, dan nilai cosine similarity dari dokumen. Kelas dari suatu dokumen ditentukan berdasarkan jumlah dari nilai *weight voting* terbesar dari jumlah k tetangga terdekat.

1.2 Perancangan User Interface

Pada sub bab ini dijelaskan bagaimana rancangan antarmuka program yang akan dibangun. Dalam penelitian ini, program akan dibangun menggunakan pemrograman *java* dengan antarmuka 2 tab. Tab yang pertama menunjukkan proses *training* data latih. Sedangkan tab kedua merepresentasikan proses pengujian. Berikut adalah rancangan antarmuka program

Training		Testing	
Load Data	Training	Nilai K	Nilai Akurasi
Dokumen Asli	Dokumen hasil preprocessing	Kelas Asli	Kelas Hasil

Gambar 3.8 Antarmuka Proses Training

Gambar 3.8 merupakan antarmuka proses pelatihan dari program yang akan dikembangkan. Tombol *Load* data berfungsi untuk *reload* data latih yang telah tersimpan pada *database*. Tombol *Training* memiliki fungsi sebagai proses untuk menghitung beberapa proses yang diperlukan, seperti *preprocessing*, perhitungan nilai *tf-IDF*, menghitung nilai *Validitas*, penentuan kelas hasil proses *training*, dan perhitungan nilai akurasi dari program yang dibangun. *Field* nilai *K* berguna untuk memasukkan jumlah tetangga terdekat yang akan digunakan sebagai proses klasifikasi. Tabel dibawahnya mempunyai fungsi sebagai tampilan hasil dari proses *training*.

Training	Testing																																											
Uji Data Latih	Uji Data Uji	Reset																																										
Input sentimen yang akan diuji		Test																																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Dokumen Asli</th> <th style="width: 33%;">Dokumen Hasil Proses</th> <th style="width: 33%;">Kelas Hasil Proses</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>			Dokumen Asli	Dokumen Hasil Proses	Kelas Hasil Proses																																							
Dokumen Asli	Dokumen Hasil Proses	Kelas Hasil Proses																																										

Gambar 3.9 Antarmuka Proses Testing

Gambar 3.9 merupakan antarmuka proses pengujian. Pengguna memasukkan Sentimen/dokumen yang akan diuji. Kemudian Tombol *Test* berfungsi sebagai proses untuk perhitungan metode yang digunakan. Dalam tombol *Test* terdapat beberapa proses diantaranya *preprocessing*, dan perhitungan nilai *Weight Voting*. Hasil dari proses klasifikasi akan ditampilkan pada tabel. Tabel akan menampilkan dokumen asli dari yang diujikan, dokumen hasil dari preprocessing, serta kelas dari hasil perhitungan metode yang diterapkan.

Uji data latih merupakan button yang akan digunakan untuk menguji kinerja sistem dari data yang telah dimiliki dan dilatihkan ke sistem. Hasil dari pengujian ini juga akan ditampilkan pada tabel yang ada dibawahnya.

Uji data uji merupakan pengujian yang dilakukan pada dokumen yang telah disiapkan oleh pengembang. Hal ini digunakan untuk menguji akurasi sistem dari program yang telah dibangun.

1.3 Perhitungan Manual

Pada sub bab ini digambarkan proses-proses perhitungan manual yang akan dilakukan oleh sistem yang dibangun. Sebagai contoh akan diambil 5 data latih. Kemudian data latih akan melalui proses *preprocessing*. setelah proses *preprocessing*, data akan masuk proses perhitungan *Modified K Nearest Neighbour*. Langkah terakhir adalah perhitungan nilai akurasi sistem.

Tabel 4.3 Data Latih

ID	Data Latih	Kelas
1	catur olahraga asah otak yang membangkitkan daya fikir banyak manfaatnya	Positif
2	mantap bikin otak berfikir	Positif
3	kok yang senior tidak mau kebuka, padahal juniornya sudah kalah	Negatif
4	tolong dibenarkan kalau ikut turnamen kok bidak saya hitam terus, nggak pernah putih	Negatif
5	Permainan catur online bisa untuk menambah daya fikir otak	Positif

Data latih tersebut akan dilakukan proses *preprocessing*. Tahap awal proses preprocessing adalah Tokenizing.

Tabel 4.4 Tokenizing

ID	Data Latih	Kelas
1	"Catur", "olahraga", "asah", "otak", "yang", "membangkitkan", "daya", "fikir", "banyak", "manfaatnya"	Positif
2	"mantap", "bikin", "otak", "berfikir"	Positif
3	"kok", "yang", "senior", "tidak", "mau", "kebuka", "padahal", "juniornya", "sudah", "kalah"	Negatif
4	"tolong", "dibenarkan", "kalau", "ikut", "turnamen", "kok", "bidak", "saya", "hitam", "terus", "nggak", "pernah", "putih"	Negatif
5	"permainan", "catur", "online", "bisa", "untuk", "menambah", "daya", "fikir", "otak"	Positif

Setelah selesai proses *tokenizing*, tahap selanjutnya adalah proses *filtering*.

Tabel 4.5 Filtering

ID	Data Latih	Kelas
1	"Catur", "olahraga", "asah", "otak", "membangkitkan", "daya", "fikir", "manfaatnya"	Positif
2	"mantap", "otak", "berfikir"	Positif
3	"senior", "kebuka", "juniornya", "kalah"	Negatif
4	"dibenarkan", "turnamen", "bidak", "hitam", "putih"	Negatif
5	"permainan", "catur", "online", "menambah", "daya", "fikir", "otak"	Positif

Kemudian dilakukan proses *stemming*.

Tabel 4.6 Stemming

ID	Data Latih	Kelas
1	"Catur", "olahraga", "asah", "otak", "bangkit", "daya", "fikir", "manfaat"	Positif
2	"mantap", "otak", "fikir"	Positif
3	"senior", "buka", "junior", "kalah"	Negatif
4	"benar", "turnamen", "bidak", "hitam", "putih"	Negatif
5	"main", "catur", "online", "tambah", "daya", "fikir", "otak"	Positif

Setelah diperoleh kata dasar dari hasil proses *Stemming* proses selanjutnya adalah perhitungan frekuensi kata, nilai *Inverse Document Frequency* (IDF) tiap kelas dengan rumus yang ada pada persamaan 2.1 di kajian pustaka. Hasil dari perhitungan sebagaimana dijelaskan pada tabel berikut:

Tabel 4.7 Perhitungan IDF

No	Term	ID1	ID2	ID3	ID4	ID5	DF	IDF
1	Catur	1	0	0	0	1	2	0,398
2	Olahraga	1	0	0	0	0	1	0,699
3	Asah	1	0	0	0	0	1	0,699
4	Otak	1	1	0	0	1	3	0,222
5	Bangkit	1	0	0	0	0	1	0,699
6	Daya	1	0	0	0	1	2	0,398
7	Fikir	1	1	0	0	1	3	0,222
8	Manfaat	1	0	0	0	0	1	0,699
9	Mantap	0	1	0	0	0	1	0,699
10	Senior	0	0	1	0	0	1	0,699
11	Buka	0	0	1	0	0	1	0,699
12	Junior	0	0	1	0	0	1	0,699
13	Kalah	0	0	1	0	0	1	0,699

14	Benar	0	0	0	1	0	1	0,699
15	Turnamen	0	0	0	1	0	1	0,699
16	Bidak	0	0	0	1	0	1	0,699
17	Hitam	0	0	0	1	0	1	0,699
18	Putih	0	0	0	1	0	1	0,699
19	Main	0	0	0	0	1	1	0,699
20	Online	0	0	0	0	1	1	0,699
21	Tambah	0	0	0	0	1	1	0,699

Setelah diperoleh nilai IDF, tahap selanjutnya adalah mencari nilai bobot dari setiap *term* dengan cara mengalikan antara frekuensi dokumen dengan nilai IDF

Tabel 4.8 Bobot Kata

No	Term	tf * IDF				
		t1	t2	t3	t4	t5
1	Catur	0,398	0	0	0	0,398
2	Olahraga	0,699	0	0	0	0
3	Asah	0,699	0	0	0	0
4	Otak	0,222	0,222	0	0	0,222
5	Bangkit	0,699	0	0	0	0
6	Daya	0,398	0	0	0	0,398
7	Fikir	0,222	0,222	0	0	0,222
8	Manfaat	0,699	0	0	0	0
9	Mantap	0	0,699	0	0	0
10	Senior	0	0	0,699	0	0
11	Buka	0	0	0,699	0	0
12	Junior	0	0	0,699	0	0
13	Kalah	0	0	0,699	0	0
14	Benar	0	0	0	0,699	0
15	Turnamen	0	0	0	0,699	0
16	Bidak	0	0	0	0,699	0
17	Hitam	0	0	0	0,699	0
18	Putih	0	0	0	0,699	0
19	Main	0	0	0	0	0,398
20	Online	0	0	0	0	0,398
21	Tambah	0	0	0	0	0,699

Tahap selanjutnya adalah menghitung panjang vektor dari masing-masing dokumen dengan cara kuadrat dari bobot tiap dokumen

Tabel 4.9 vektor dokumen

No	Term	(tf * IDF) ²				
		D1	D2	D3	D4	D5
1	Catur	0,158	0	0	0	0,158
2	Olahraga	0,489	0	0	0	0
3	Asah	0,489	0	0	0	0
4	Otak	0,049	0,049	0	0	0,049
5	Bangkit	0,489	0	0	0	0
6	Daya	0,158	0	0	0	0,158
7	Fikir	0,049	0,049	0	0	0,049
8	Manfaat	0,489	0	0	0	0
9	Mantap	0	0,489	0	0	0
10	Senior	0	0	0,489	0	0
11	Buka	0	0	0,489	0	0
12	Junior	0	0	0,489	0	0
13	Kalah	0	0	0,489	0	0
14	Benar	0	0	0	0,489	0
15	Turnamen	0	0	0	0,489	0
16	Bidak	0	0	0	0,489	0
17	Hitam	0	0	0	0,489	0
18	Putih	0	0	0	0,489	0
19	Main	0	0	0	0	0,489
20	online	0	0	0	0	0,489
21	tambah	0	0	0	0	0,489
SUM =		2,369	0,587	1,954	2,443	1,881
SQRT =		1,539	0,766	1,398	1,563	1,372
Panjang vektor		D1	D2	D3	D4	D5

Langkah selanjutnya adalah mencari nilai Cosine Similarity dari dokumen 5 yang digunakan sebagai data uji yang diperoleh dari hasil perkalian antara nilai tf, IDF, dan dokumen uji (dalam hal ini adalah D5).

Tabel 4.10 Perhitungan nilai cosine similarity

No	Term	tf * IDF * D5			
		D1	D2	D3	D4
1	Catur	0,158	0	0	0
2	Olahraga	0	0	0	0
3	Asah	0	0	0	0
4	Otak	0,049	0,049	0	0
5	Bangkit	0	0	0	0
6	Daya	0,158	0	0	0
7	Fikir	0,049	0,049	0	0
8	Manfaat	0	0	0	0
9	Mantap	0	0	0	0
10	Senior	0	0	0	0
11	Buka	0	0	0	0
12	Junior	0	0	0	0
13	Kalah	0	0	0	0
14	Benar	0	0	0	0
15	Turnamen	0	0	0	0
16	Bidak	0	0	0	0
17	Hitam	0	0	0	0
18	Putih	0	0	0	0
19	Main	0	0	0	0,158
20	Online	0	0	0	0,158
21	Tambah	0	0	0	0
SUM =		0,415	0,098	0	0,317

$$\begin{aligned}
 \text{Similarity D1} &= D1 / (|D1| * |D5|) \\
 &= 0,415 / (1,539 * 1,372) \\
 &= 0,415 / 2,112 \\
 &= 0,197
 \end{aligned}$$

Dengan cara yang sama dihitung nilai *Similarity* dari setiap dokumen, kemudian diurutkan dari nilai similarity terbesar sehingga diperoleh nilai *Similarity* sebagai berikut:

Tabel 4.11 Nilai similarity

Dokumen	D1	D2	D3	D4	D5
D1	1	0,019	0	0	0,197
D2		1	0	0	0,021
D3			1	0	0
D4				1	0
D5					1

Setelah diperoleh nilai similarity dari dokumen, dihitung nilai validitas data *training*.

$$\text{Validitas (x = 1)} = \frac{1}{k} \sum_{i=1}^k S(\text{label}(x), (\text{label}(N_i(x))))$$

$$\text{Validitas (x = 1)} = \frac{1}{3} \sum_{i=1}^k S(\text{label}(x = 1), (\text{label}(N_i(x = 2))))$$

$$\text{Validitas (x = 1)} = \frac{1}{3} (1 + 1 + 0)$$

$$\text{Validitas (x = 1)} = 0,667$$

Proses perhitungan yang sama dilakukan pada semua data training sehingga diperoleh nilai validitas seperti ditunjukkan pada tabel berikut:

Tabel 4.12 Perhitungan manual validitas data training

Dokumen	Tetangga 1	Tetangga 2	Tetangga 3	Sum S(a,b)	validitas
D1	1	1	0	2	0,667
D2	1	1	0	2	0,667
D3	1	0	0	1	0,333
D4	1	0	0	1	0,333
D5	1	1	0	2	0,667

Proses *Training* berakhir pada perhitungan nilai validitas data training.

Tahapan selanjutnya adalah proses *testing*. Pada proses *testing*, dibutuhkan inputan berupa nilai *cosine similarity* data uji dan nilai validitas dari k tetangga terdekat dari data uji. Setelah diperoleh nilai validitas dan *cosine similarity*, langkah berikutnya adalah menghitung nilai *weight voting*. Berikut adalah contoh proses pengujian.

Data uji : Permainan ini dapat membangkitkan daya fikir otak.

Hasil preprocessing : “main”, “bangkit”, “daya”, “fikir”, “otak”

Nilai cosine similarity :

D1 = 0,249, D2 = 0,026, D3 = 0, D4 = 0, D5 = 0,279

Setelah diperoleh nilai *cosine similarity*, dilakukan perhitungan untuk mendapatkan nilai *weight voting*. Sebagaimana contoh perhitungan sebagai berikut:

$$\begin{aligned}
 W(x,D5) &= \text{Validitas}(1) * \frac{1}{\text{similarity}+0,5} \\
 &= 0,667 * \frac{1}{0,279 + 0,5} \\
 &= 0,667 * 1,284 \\
 &= 0,856428
 \end{aligned}$$

Dengan cara yang sama, dihitung juga nilai *weight voting* untuk k dokumen tetangga yang lainnya. Pada kasus ini digunakan nilai k = 3 sehingga dari proses perhitungan nilai *weight voting* diperoleh tabel sebagai berikut.

Tabel 4.13 Perhitungan manual nilai *weight voting*

Dokumen	Weight voting	Kelas
D5	0,856428	Positif
D1	0,890445	Positif
D2	1,267967	Positif

Pada tabel tersebut didapatkan bahwa pada kasus ini data uji masuk dalam kelas positif karena mempunyai nilai total *weight voting* yang lebih besar yakni 3,01484. Dengan rincian 0,856428 untuk dokumen 5, 0,890445 untuk dokumen 1, dan 1,267967. Sedangkan untuk kelas negatif nilai *weight voting*nya adalah 0.

Hasil dari perhitungan nilai sangat tergantung pada kualitas data training, karena nilai cosine similarity, validitas, dan *weight voting* mengacu pada dokumen yang ada pada data training.

1.4 Perancangan Uji Coba

Proses pelatihan terhadap hasil analisis sentimen dilakukan untuk menguji dan mengetahui akurasi sistem. Setiap uji coba akan dihitung akurasinya. Setiap uji coba akan menggunakan nilai k yang sama. Dalam penelitian ini akan dilakukan 53 kali uji coba untuk setiap nilai k . Nilai k yang diujikan adalah antara $k=1$ sampai $k=20$. Perancangan skenario uji coba sistem ditunjukkan pada tabel 4.1

Tabel 4.1 Tabel Perancangan Uji Coba

Jumlah Dataset	Jumlah Data Training (%)	Nilai k	Nilai Akurasi (%)
		1	
		2	
		3	
		..	
		20	

Dalam melakukan uji coba akan mengambil jumlah dataset yang berbeda yaitu 200, 300, dan 400. Data yang akan diuji adalah 100 data. Pada setiap uji coba untuk nilai k yang sama, akan diambil akurasinya kemudian dihitung rata-ratanya sehingga akan ditemukan nilai akurasi yang tertinggi. Perancangan hasil uji coba akan ditunjukkan pada tabel 4.2

Tabel 4.2 Perancangan Hasil Uji Coba

Jumlah Dataset	Nilai k	Nilai Akurasi (%) untuk data Training
200	1	
	..	
	20	
300	1	
	..	
	20	
400	1	
	..	
	20	