

BAB 4 PERANCANGAN

4.1 Perancangan Proses

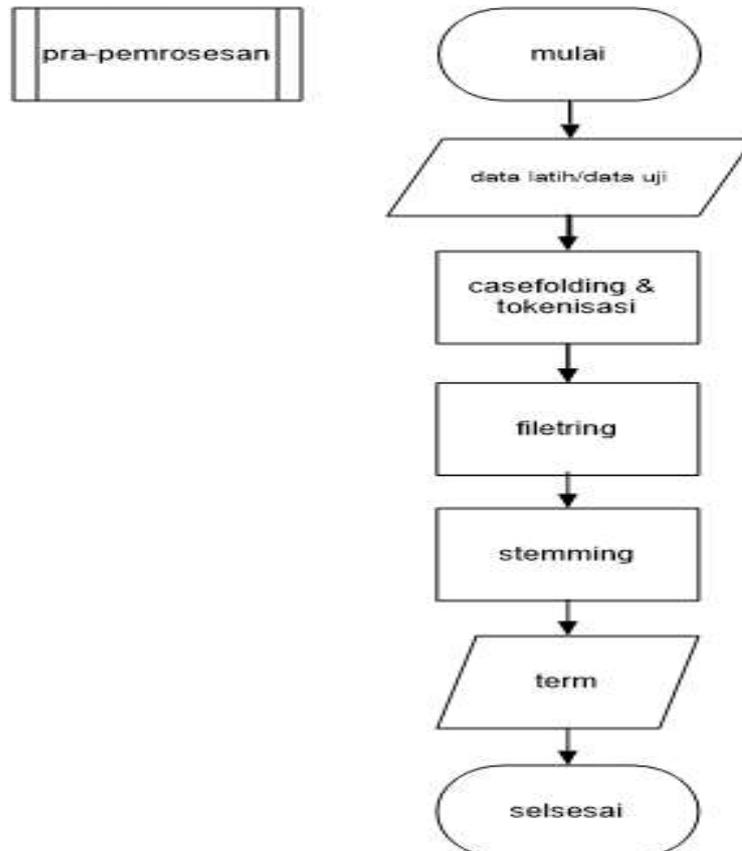
Pada perancangan proses akan dijelaskan tahapan-tahapan yang dilalui berdasarkan sistem yang akan dibuat. Proses disini dimulai dari pra-pemrosesan menggunakan metode pemrosesan teks yakni TF-IDF. Dilanjutkan menuju proses *fuzzy c-means* yang akan menghasilkan aturan.

4.1.1 Proses Pelatihan

Pada proses ini dilakukan pelatihan data yang nantinya akan membentuk aturan-aturan pengenalan emosi. Proses pelatihan dilakukan menggunakan data latih berjumlah 500 data yang telah didapatkan melalui dari Twitter. Proses pelatihan akan menghasilkan aturan-aturan yang nantinya digunakan untuk menguji suatu data uji untuk diketahui akurasi dari sistem yang dibuat.

4.1.1.1 Proses Pra-Pemrosesan

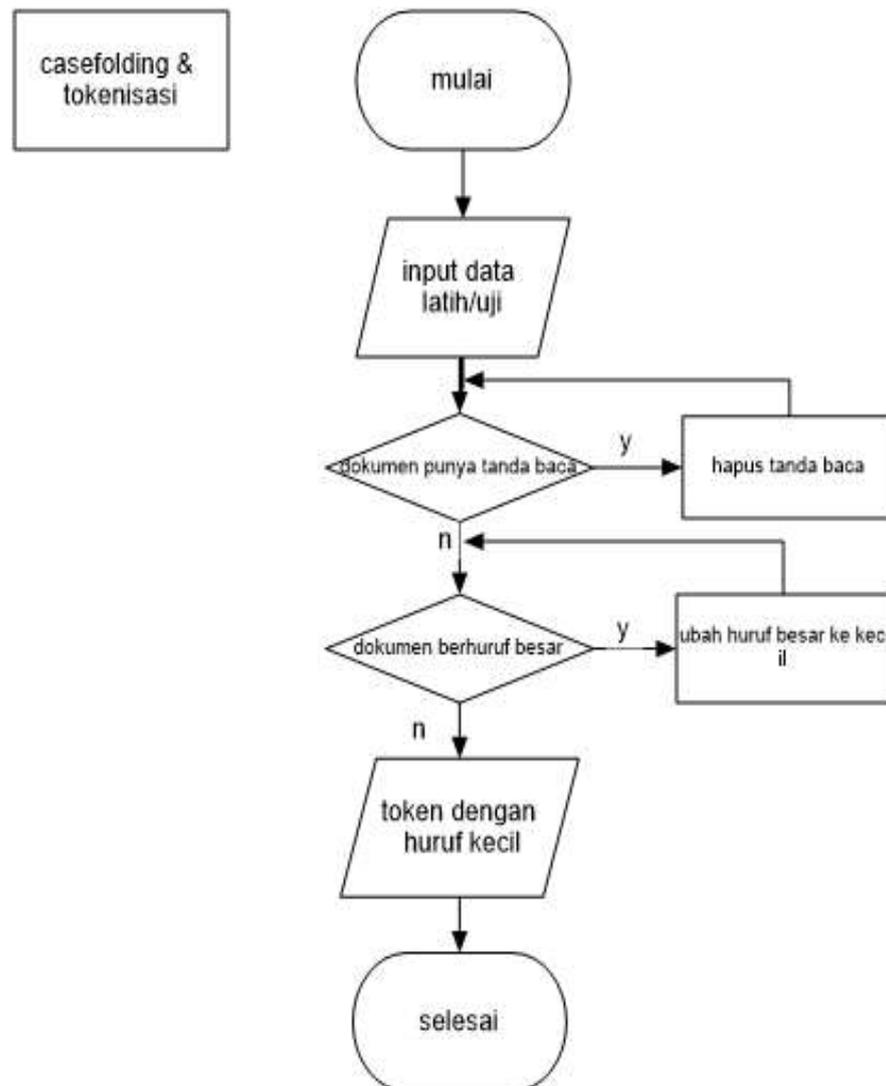
pada proses ini dilakukan perubahan jenis *input* dari data yang awalnya berupa teks menjadi angka. Pra-pemrosesan dilakukan menggunakan metode *text mining* dimana dilakukan proses *case folding*, *tokenisasi*, *filtering*, *Stemming*. Perancangan proses pra-pemrosesan diperlihatkan oleh Gambar 4.1.



Gambar 4.1 Diagram Alur Pra-Pemrosesan

1. Tokenisasi dan Case Folding

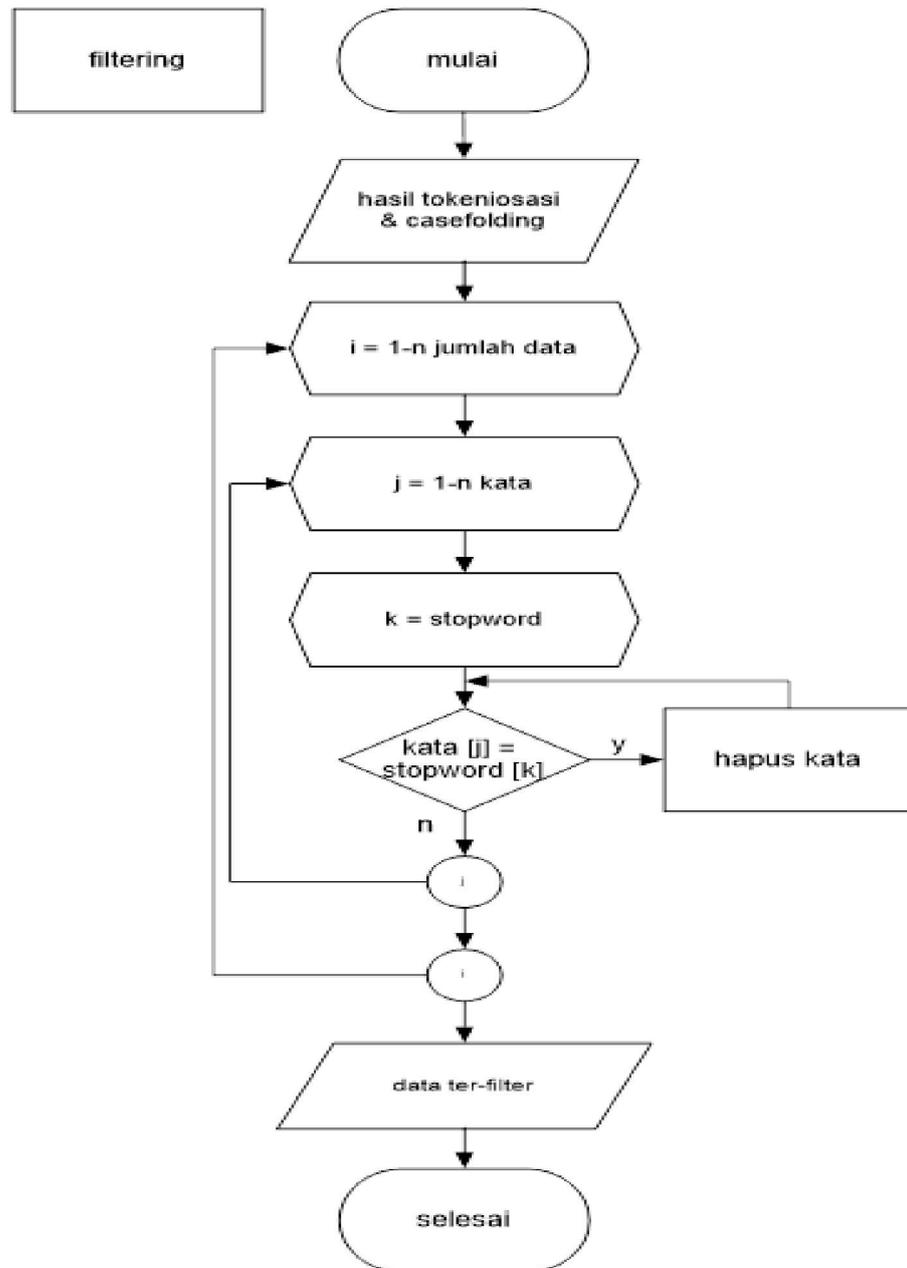
Tokenisasi merupakan proses menghapus tanda baca yang terdapat pada dokumen, angka dan karakter selain dari karakter huruf alfabet. Pada proses ini juga dilakukan proses *case folding* yaitu mengubah karakter huruf pada dokumen menjadi huruf kecil. Perancangan proses tokenisasi dan *case folding* ditunjukkan oleh Gambar 4.2.



Gambar 4.2 Diagram Alur Proses Tokenisasi & Case Folding

2. Filtering

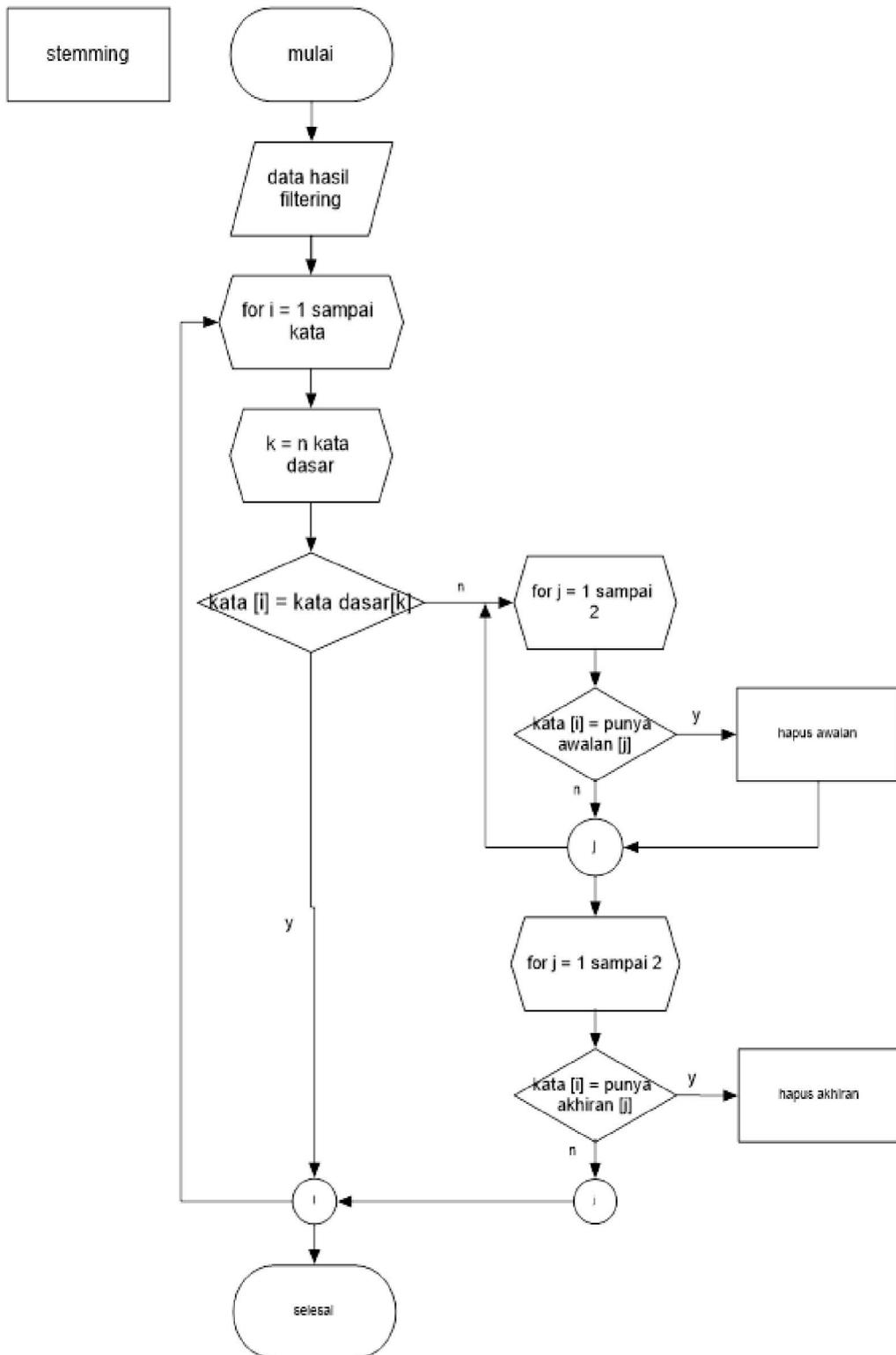
Filtering adalah proses menghapus kata-kata yang termasuk kategori dianggap tidak perlu untuk diolah oleh program. Pada tahapan ini dilakukan pembersihan kata yang dianggap tidak diperlukan. Kata-kata yang digunakan pada filtering disimpan pada suatu list bernama *stopword list*. Apabila pada dokumen terdapat kata yang persis dengan yang ada pada *stopword list* maka kata tersebut akan dihilangkan atau dihapus. Perancangan proses *filtering* ditunjukkan oleh Gambar 4.3.



Gambar 4.3 Diagram Alur Proses *Filtering*

3. *Stemming*

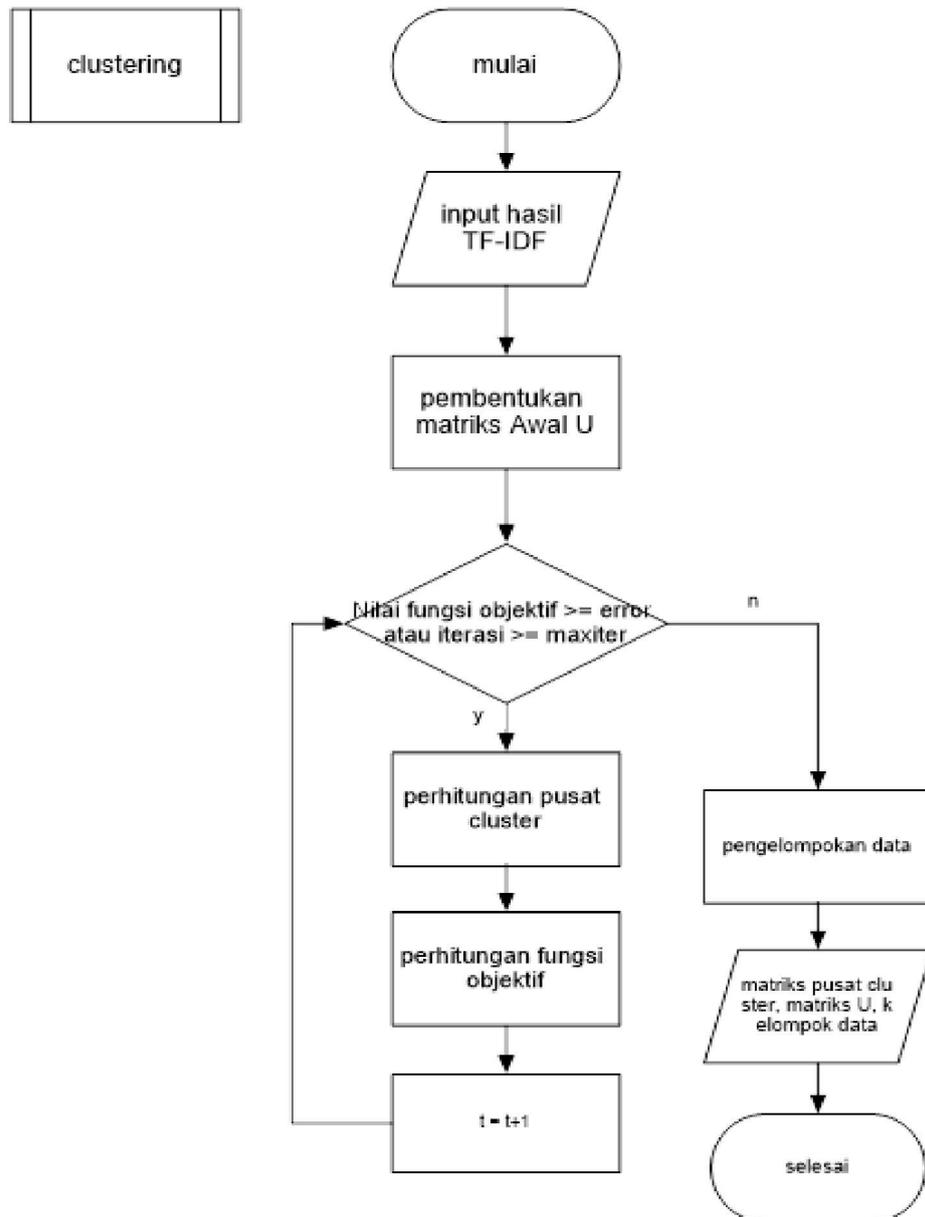
Stemming adalah proses mengubah hasil pada tahapan *filtering* menjadi kata dasar. Pada proses ini setelah didapat data dari *filtering* maka *stemming* akan menghapus suatu imbuhan dari kata baik itu awalan maupun akhiran. Pada penelitian ini akan menggunakan algoritme *stemmer* Nazief & Adriani. Algoritme tersebut menggunakan kamus untuk pencocokan dan pembentukan kata dasar sehingga bisa digunakan untuk melakukan *stemmer* Bahasa Indonesia. Perancangan proses *stemming* ditunjukkan oleh Gambar 4.4.



Gambar 4.4 Diagram Alur Proses *Stemming*

4.1.1.2 Proses *Clustering*

Proses *clustering* data input memakai algoritme *fuzzy c-means* ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Diagram Alur Proses Clustering

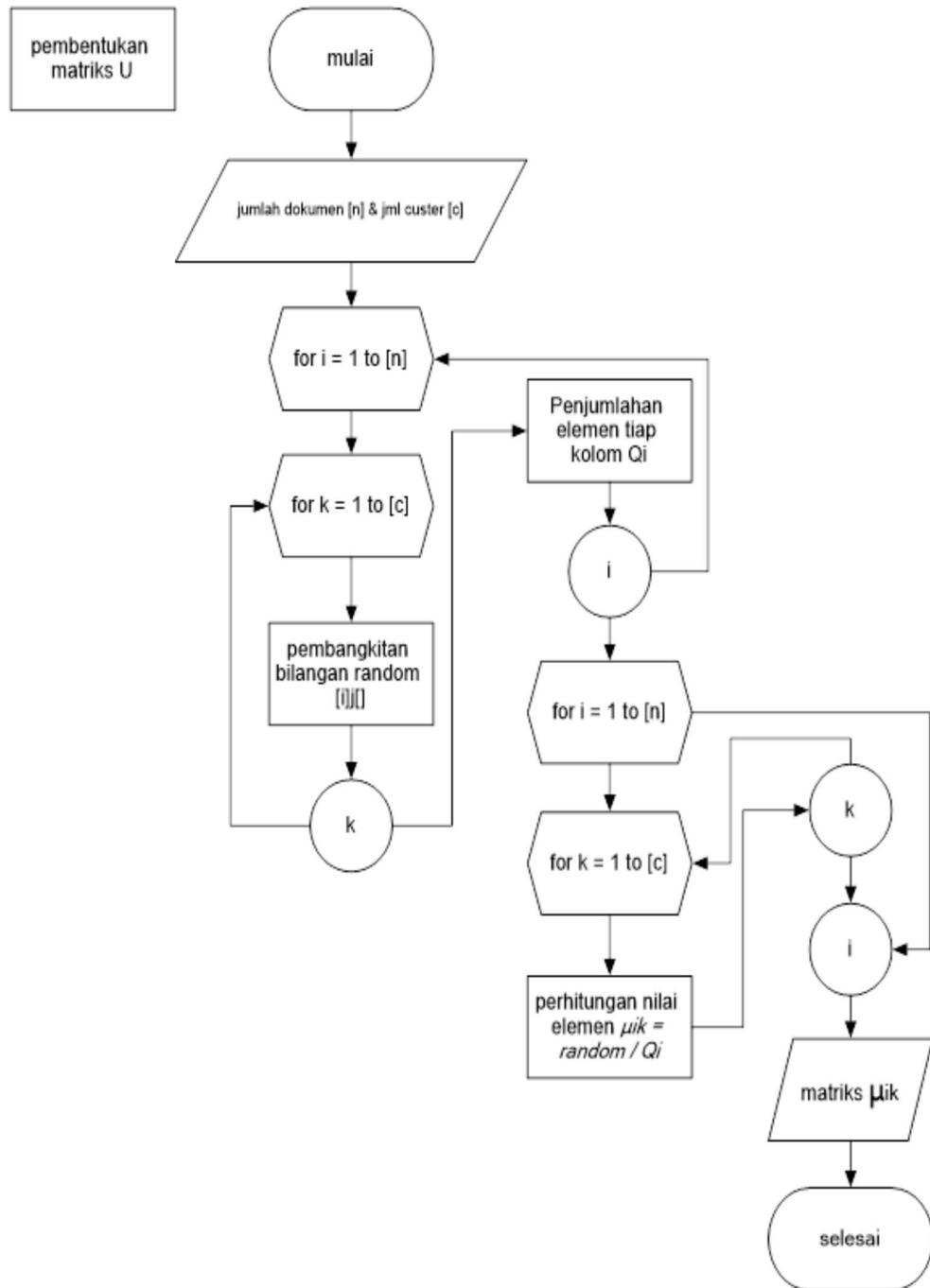
1. Proses Pembentukan Matriks Partisi Awal U

Proses pembuatan matriks U awal tidak melibatkan data. *Input* proses akan ini memerlukan paramter banyaknya jumlah data n dan juga jumlah *cluster* c . *Output* pada proses ini merupakan matriks yang merepresentasikan derajat keanggotaan awal data terhadap tiap *cluster* μ_{ik} . Dengan dimensi $n * c$. Rincian untuk proses untuk membent matriks U adalah sebagai berikut.

- Pembangkitkan bilangan *random* digunakan untuk menunjukkan derajat keanggotaan dari data ke- i ke dalam suatu *cluster* ke- k μ_{ik} . Rentang bilangan *random* adalah 0 hingga 1.

- Penjumlahan elemen tiap kolom Proses ini dari nilai *random* μ_{ik} setiap iterasi ke-*i* ($Q_i = \mu_{i1} + \mu_{i2} + \mu_{i3} + \dots + \mu_{ik}$) hingga didapatkan Q_i satu berdimensi *n*.
- memperbaiki nilai elemen matriks μ_{ik} . Jika perhitungan yang dilakukan benar, maka nilai jumlah μ_{ik} pada tiap iterasi ke-*i* merupakan 1 ($\mu_{i1} + \mu_{i2} + \mu_{i3} + \dots + \mu_{ik} = 1$).

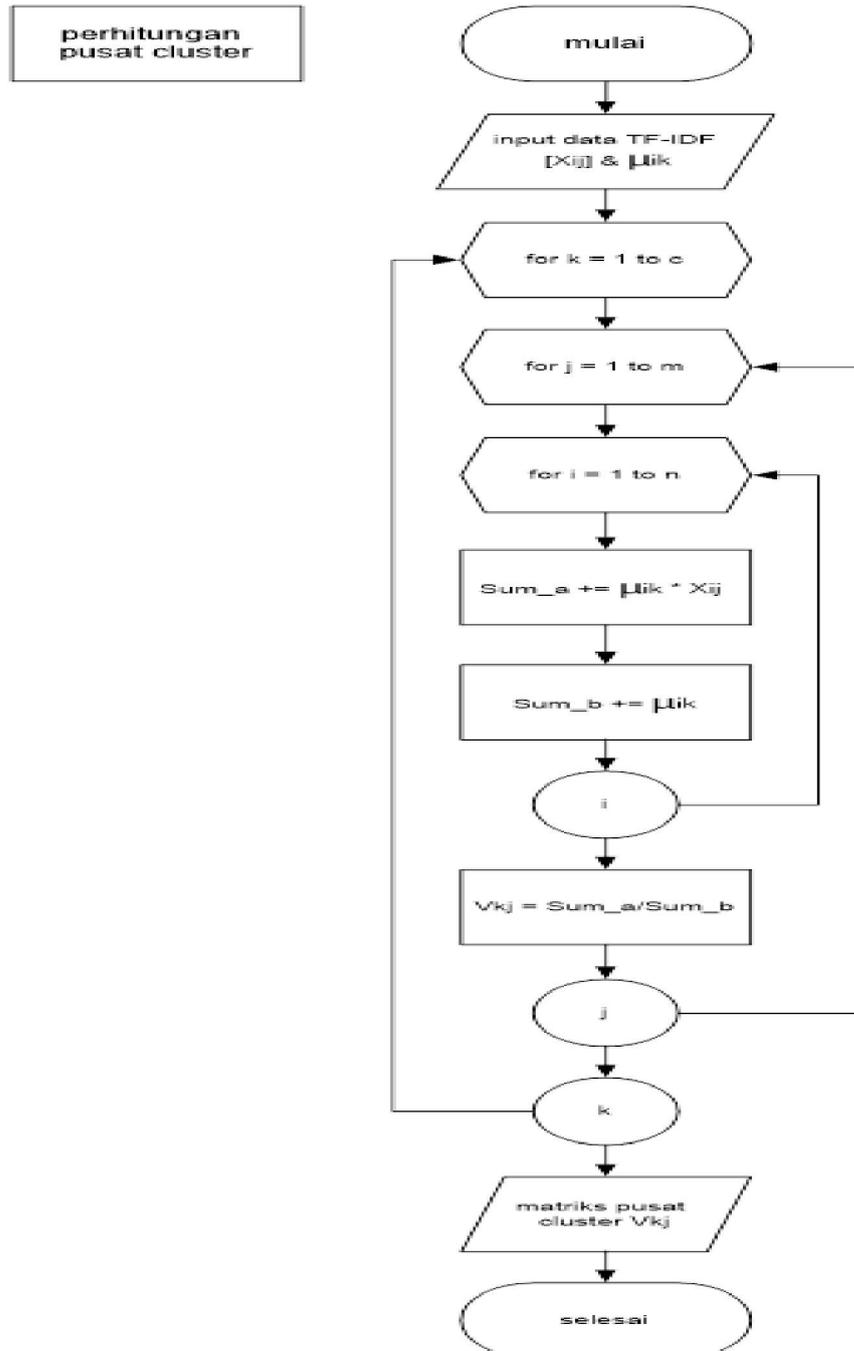
Proses pembentukan matriks *U* awal ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Diagram Alur Pembentukan Matriks Awal *U*

2. Proses Perhitungan Pusat Cluster

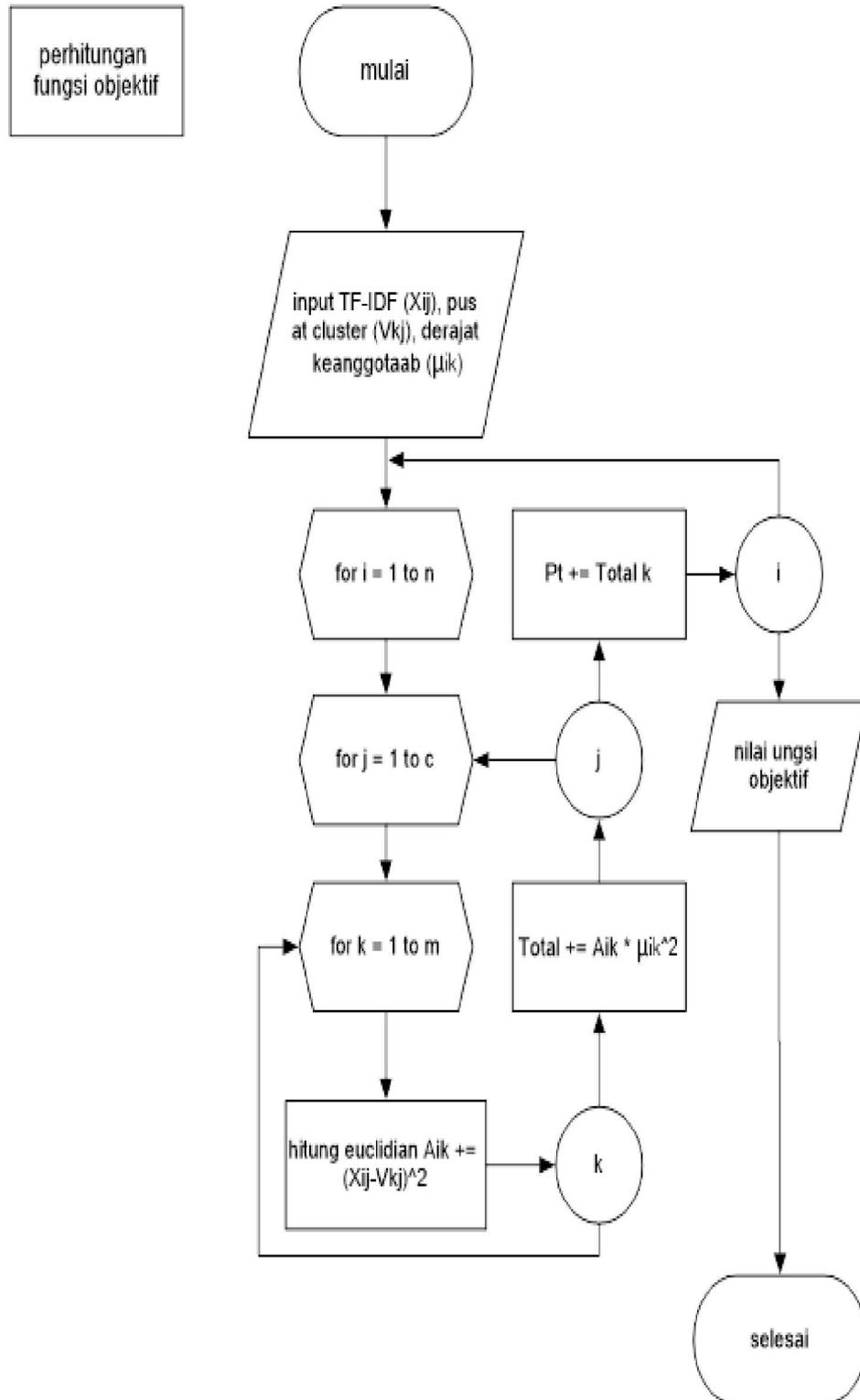
Tahap perhitungan dari pusat *cluster* memerlukan *input* berupa matriks hasil proses TF-IDF X_{ij} dan nilai dari derajat keanggotaan hasil dari proses pembuatan matriks $U \mu_{ik}$. Perhitungan pada pusat *cluster* ini akan menghasilkan suatu matriks pusat *cluster* V_{kj} berdimensi jumlah *cluster* x jumlah fitur ($c \times m$). Proses untuk perhitungan pusat *cluster* ditunjukkan oleh Gambar 4.7.



Gambar 4.7 Diagram Alur Perhitungan Pusat Cluster

3. Proses Perhitungan Fungsi Objektif

Proses untuk perhitungan fungsi objektif ditunjukkan oleh Gambar 4.8.



Gambar 4.8 Diagram Alur Perhitungan Fungsi Objektif

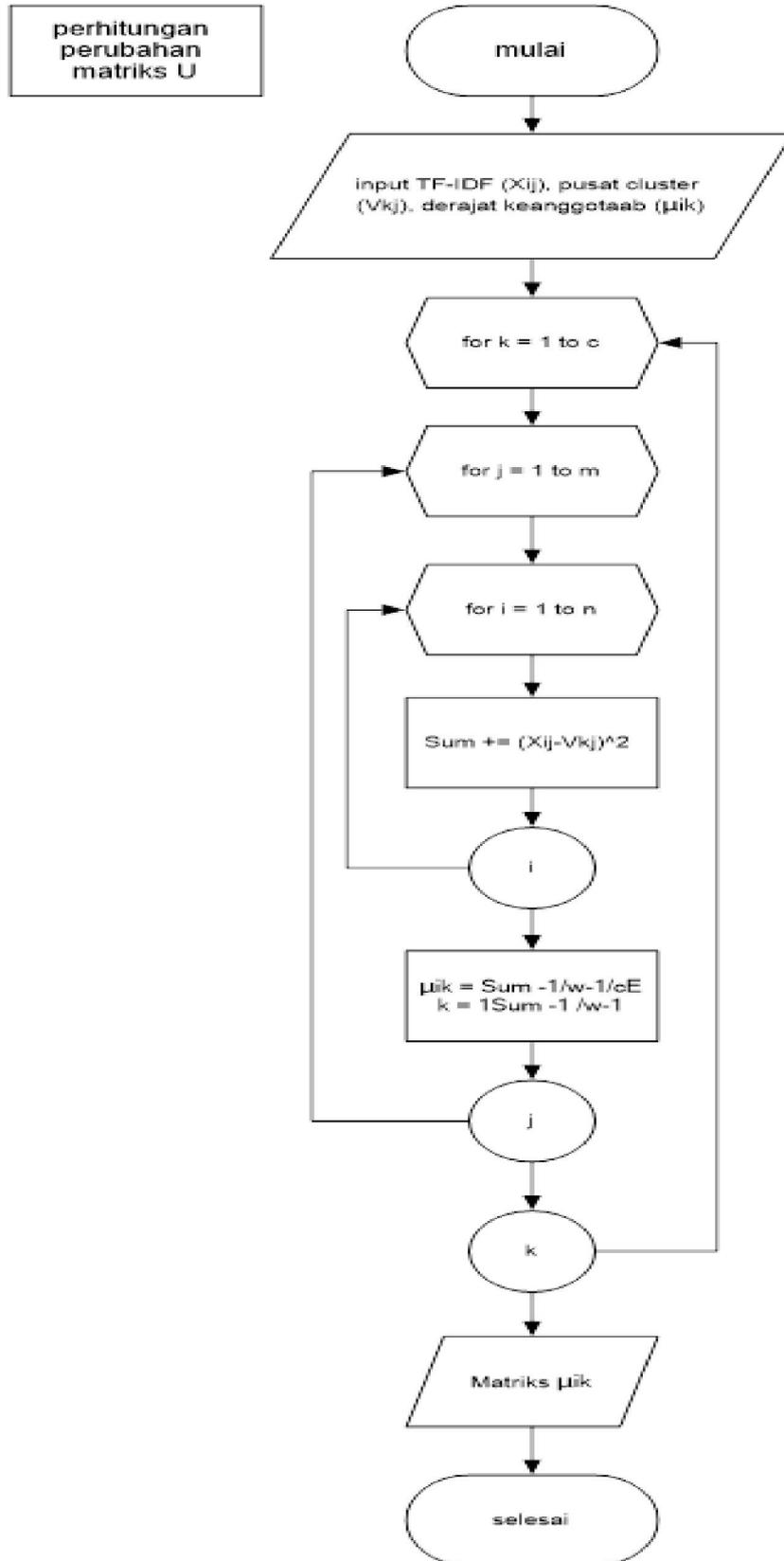
Proses untuk menghitung fungsi objektif akan melibatkan *input* data matriks TF-IDF X_{ij} , matriks pusat *Cluster* V_{kj} serta derajat keanggotaan μ_{ik} . Pada tahap ini dihasilkan *output* sebuah nilai fungsi objektif dari iterasi ke- t P_t . proses untuk menghitung nilai fungsi objektif adalah sebagai berikut.

- Pada tiap iterasi ke- i akan dilakukan iterasi k sebanyak jumlah *cluster* yang akan dibentuk c agar dapat dilakukan perhitungan dari total jarak atribut A_{ik} .
- Proses selanjutnya adalah menghitung total jarak A_{ij} didapat dari hasil penjumlahan antara jarak untuk semua atribut ($A_{ij} = (X_{i1} - V_{k1})^2 + (X_{i2} - V_{k2})^2 + (X_{i3} - V_{k3})^2 + \dots + (X_{ij} - V_{kj})^2$).
- Proses selanjutnya adalah nilai A_{ij} dikali dengan nilai dari derajat keanggotaan μ_{ik} lalu ditotal untuk tiap iterasi i .
- Proses selanjutnya adalah semua nilai total perhitungan akan dijumlah hingga didapat nilai fungsi objektif P_t .
- Selisih Nilai fungsi objektif dengan nilai fungsi objektif pada iterasi sebelumnya P_{t-1} setelah dihitung maka selanjutnya akan dibandingkan dengan nilai dari kesalahan minimum (ξ) yang sudah ditentukan sebelumnya untuk memeriksa kondisi dari berhentinya iterasi pada proses *clustering* data yakni Jika $|P_t - P_{t-1}| < \xi$, sehingga iterasi pada proses *clustering* berhenti.

4. Proses Perubahan Matriks U

Proses perubahan dari matriks U yang menggunakan *input* dari nilai data matriks TF-IDF X_{ij} , matriks pusat *cluster* V_{kj} serta derajat keanggotaan terakhir μ_{ik} . *Output* proses ini merupakan perbaikan nilai dari derajat keanggotaan μ_{ik} yang merepresentasikan hasil dari perubahan pada matriks U . Perubahan matriks U akan berpengaruh terhadap persebaran pada data yang di *clustering* sehingga semakin baik nilai U maka proses *clustering* juga akan semakin baik. Setelah dilakukan proses perubahan matriks U kualitas cluster akan dilihat berdasarkan varian *cluster* apabila iterasi telah berhenti sesuai ketentuan yang ada maka pengecekan varian *cluster* dilakukan.

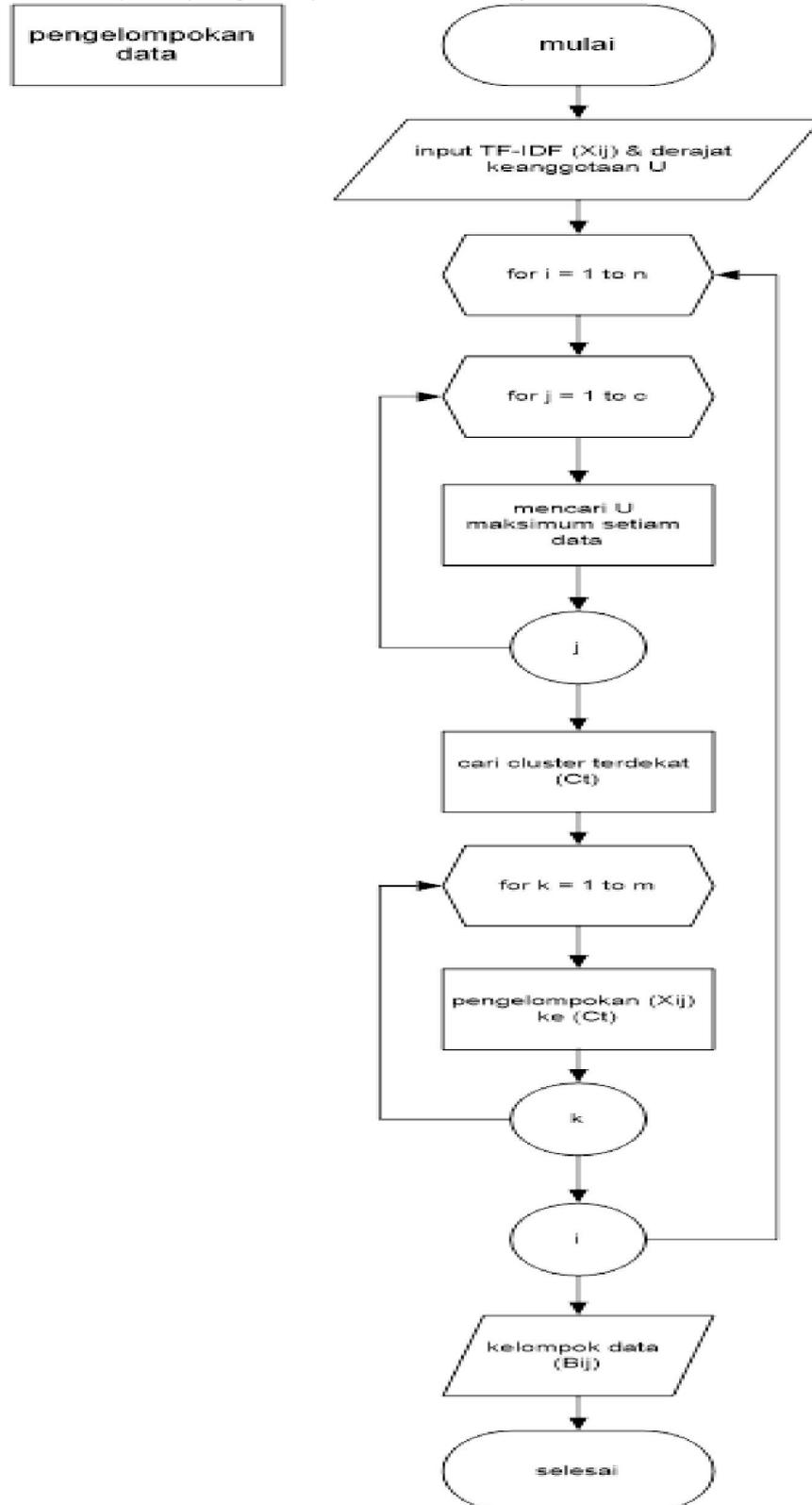
Perubahan pada matriks U bisa membuat nilai dari derajat keanggotaan tiap elemen matriks U μ_{ik} akan menjadi lebih baik. Untuk jumlah μ_{ik} pada setiap iterasi ke- i adalah 1 ($\mu_{i1} + \mu_{i2} + \mu_{i3} + \dots + \mu_{ik} = 1$). Bila jumlah μ_{ik} pada tiap iterasi ke- i tidak sama dengan 1 ($\mu_{i1} + \mu_{i2} + \mu_{i3} + \dots + \mu_{ik} \neq 1$), berarti terjadi suatu kesalahan pada proses *clustering* data. Proses untuk perubahan matriks partisi U ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Diagram Alur Perhitungan Matriks U Baru

1. Proses Pengelompokan Data

Proses pada pengelompokan data ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Diagram Alur Pengelompokan Data

Proses untuk mengelompokkan data akan melibatkan *input* dari nilai data Matriks TF-IDF X_{ij} serta matriks derajat keanggotaan μ_{ik} . Pada proses ini akan dihasilkan *output* kelompok data B_{ij} yang dapat dinyatakan sebagai *cluster* dari setiap data X_{ij} . Tahapan proses untuk mengelompokkan data-data adalah sebagai berikut.

1. Mencari μ_{ik} terbesar pada tiap data, yakni dengan cara membandingkan nilai derajat keanggotaan data μ_{ik} dengan nilai tiap *cluster*. Nilai μ_{ik} yang akan dipilih adalah nilai yang paling dekat dengan nilai 1.
2. Mencari *cluster* yang terpilih kp , yakni dengan cara memilih *cluster* k yang mempunyai nilai μ_{ik} terbesar.
3. Mengelompokkan data ke *cluster*, yakni mengelompokkan data ke *cluster* berdasarkan nilai dari kp .

4.1.1.3 Proses Analisis Varian

Proses untuk perhitungan varian memerlukan data *input* X_{ij} dengan index dari kelompok data B_{ij} . Proses ini akan menghasilkan sebuah nilai *output* batasan varian V yang akan berguna untuk menganalisis jumlah *cluster* terbaik yang terjadi pada proses pelatihan.

Analisis perhitungan varian ini terdiri atas 5 proses perhitungan yaitu:

1. Perhitungan dari rata-rata data tiap *cluster*, rata-rata data tiap *cluster* dihitung hingga didapat nilai rata-rata \bar{y}_c yang berjumlah sebanyak jumlah *cluster* c . Kemudian, dari rata-rata data *cluster* \bar{y}_c , dihitung rata-rata data keseluruhan \bar{y} .
2. Perhitungan varian dari *cluster* V_{c2} dihitung dengan memakai Persamaan 2.20. Hasil dari perhitungan V_{c2} akan dipakai untuk menghitung *variance within Cluster* (V_w) dan *variance between Cluster* (V_b).
3. setelah nilai V_{c2} untuk setiap *cluster* diketahui, selanjutnya perhitungan nilai V_w dapat dilakukan dengan menggunakan Persamaan 2.21 dan nilai V_b dapat dihitung dengan menggunakan Persamaan 2.22.
4. Perhitungan untuk batasan varian V bisa dihitung menggunakan Persamaan 2.23.

Cluster yang baik adalah *cluster* yang memiliki V_w minimum dan V_b maksimum. *Cluster* yang akan dipilih untuk pembangkitan aturan yakni *clustering* yang mempunyai batasan varian V terkecil.

4.1.1.4 Proses Ekstraksi Aturan

Proses untuk melakukan ekstraksi aturan akan melibatkan *input* dari hasil perhitungan pusat *cluster* V_{kj} dan data latih input TF-IDF X_{ij} . *Output* dari proses ini adalah koefisien *output* serta *deviation standard*. Nilai *deviation standard* dan pusat *cluster* akan membentuk suatu fungsi Gauss. Proses pada ekstraksi aturan ini memiliki dua sub-proses yaitu:

1. Proses Perhitungan *Deviation standard*

Deviation standard dapat diperoleh melalui perhitungan atribut ke- j σ_{kj} di tiap data pada *cluster* yang sama. Banyaknya *deviation standard* σ_{kj} sama dengan dimensi pusat *cluster* V_{kj} , yaitu sebanyak jumlah *cluster* x jumlah variabel. Keanggotaan Gauss yang dihasilkan akan mempunyai kombinasi dari nilai pusat *cluster* dengan *deviation standard* yang beda untuk setiap variabel atau fitur pada setiap aturan. Bila diasumsikan atribut ke- m data merupakan nilai *output* data, maka fungsi Gauss yang didapatkan akan sebanyak $m-1$ (jumlah atribut - 1). Maka pasangan pusat *Cluster* V_{kj} dengan *deviation standard* σ_{kj} yang akan dibangkitkan menggunakan fungsi Gauss hanya berupa data *input* saja.

2. Proses Perhitungan Koefisien *Output*

Perhitungan ini dilakukan untuk mencari nilai dari koefisien *output* setiap aturan agar bisa membuat fungsi untuk konsekuen aturan. Perhitungan koefisien *output* dilakukan memakai metode kuadrat terkecil (*least square*) dikarenakan matriks yang diolah tidak berbentuk persegi. Proses perhitungan koefisien *output* melibatkan *input* berupa matriks data latih X_{ij} , matriks pusat cluster V_{kj} , dan matriks derajat keanggotaan μ_{ik} . Proses ini akan menghasilkan nilai koefisien *output* untuk tiap aturan berukuran (variabel+1) x jumlah *cluster*.

Proses perhitunggan koefisien *output* memiliki tiga sub-proses yaitu:

- Normalisasi
Proses ini digunakan untuk menormalisasi data dan kemudian disimpan menjadi suatu matriks.
- Pembentukan matriks U
Proses ini akan membentuk matriks berdimensi jumlah data latih dikali (jumlah fitur x jumlah *cluster*). yang menyimpan data hasil normalisasi
- Perhitungan *Least Square Error*
Proses ini digunakan untuk dapat menghitung nilai dari koefisien *output* yang berguna sebagai penentu nilai dari *output* di proses pengujian.

a) Proses Normalisasi

Proses normalisasi menggunakan tiga *input* yang dilibatkan yaitu matriks data latih X_{ij} , matriks pusat cluster V_{kj} , dan matriks derajat keanggotaan μ_{ik} . *Output* yang akan dihasilkan proses ini adalah matriks d^{kj} yang berisi matriks normalisasi data perkalian data latih dengan pusat *cluster* dan data derajat keanggotaan.

Tahapan-tahapan untuk melakukan normalisasi adalah:

- Data latih X_{ij} dikali derajat keanggotaan μ_{ik} .
- derajat keanggotaan μ_{ik} dijumlahkan untuk semua cluster.
- d^{kj} dibagi dengan hasil penjumlahan μ_{ik} .

b) Proses Pembentukan Matriks U

Proses pembuatan matriks U yang melibatkan input matriks d_{ij}^k ternormalisasi dan akan menghasilkan matriks U . Proses ini dilakukan dengan cara mengambil semua isi data dari d_{ij}^k yang memiliki dimensi tiga ($c \times n \times (m+1)$) diubah ke matriks U yang berdimensi dua ($n \times (c \times (m+1))$). Misalkan cluster yang dibentuk adalah 3, dengan data latih sebanyak 4, dan variabel data latih sebanyak 6. Sehingga dimensi matriks U adalah $4 (3 \times (5+1)) = 4 \times 1$. Matriks U merupakan matriks yang berisi derajat keanggotaan data *output* dan perkalian data latih dengan derajat keanggotaan yang bersesuaian pada tiap *cluster* yang digunakan untuk menghitung nilai koefisien *output* dengan menggunakan metode *Least Squares Error*.

c) Proses Perhitungan *Least Squares Error*

Proses untuk perhitungan *Least Squares Error* melibatkan *input* dari matriks U dan matriks Y . Matriks U didapatkan dari proses pembuatan matriks U dan matriks Y berisi data input latih X_{ij} atribut terakhir X_{im} yang mencerminkan nilai emosi pada data latih. Perhitungan ini akan menghasilkan *output* berupa matriks K yang merupakan koefisien *output* tiap atribut untuk tiap aturan.

Perhitungan *Least Squares Error* akan dijalankan berdasarkan persamaan 2.18 yang melibatkan operasi-operasi matriks yakni perkalian matriks, transpose matriks, dan invers matriks. Operasi matriks tersebut dilakukan dengan menggunakan fungsi yang telah ada di *library* Netbeans, sehingga perhitungan *Least Squares Error* dilakukan dengan menggunakan fungsi *java matrix package* (JAMA).

4.1.2 Proses Pengujian

Proses pengujian dapat dijalankan bila proses pelatihan selesai dijalankan. Proses untuk melakukan pengujian terdiri dari tiga proses, yaitu proses fuzzyfikasi, perhitungan Z tiap aturan, dan defuzzyfikasi. *Input* proses pengujian berupa data pengujian dan aturan hasil proses pelatihan, sedangkan *output* proses akan berupa nilai kedekatan dengan nilai Z serta emosi yang dihasilkan.

Proses fuzzyfikasi berguna untuk mengubah nilai tegas data menjadi suatu nilai *fuzzy*. Perhitungan dari nilai Z digunakan untuk mengetahui nilai dari *output* dari tiap aturan yang terbentuk. Proses defuzzyfikasi digunakan untuk menghitung nilai tegas dari nilai tingkatan emosi.

1. Proses Fuzzyfikasi

fuzzyfikasi merupakan proses mengubah bentuk data dari nilai tegas X_j menjadi *fuzzy* μ_{kj} yang berupa matriks derajat keanggotaan. fuzzyfikasi dilakukan menggunakan fungsi Gauss dengan data yang bersesuaian atau variabel yang bersesuaian. Proses ini akan membentuk *output* berupa *firestrength* α_k . Proses ini

akan melibatkan input data X_j dan menghasilkan *output* matriks derajat keanggotaan μ_{kj} . tahapan proses fuzzyfikasi adalah sebagai berikut.

- Menghitung matriks μ_{kj}
 Derajat keanggotaan μ_{kj} dihitung dengan memakai fungsi Gauss yang sudah dibuat dari proses pelatihan dan pembangkitan aturan.
- Menghitung nilai *firestrength*
Firestrength diperoleh dari perkalian antara derajat keanggotaan pada tiap himpunan *fuzzy* untuk tiap variabel *input* untuk tiap aturan yang terbentuk (α -predikat = $\mu_1 * \mu_2 * \mu_3 * \dots * \mu_j$).

2. Proses Perhitungan Nilai Z

Proses untuk perhitungan nilai Z melibatkan input dari koefisien *output* k_{kj} dan data uji X_j sehingga *output* yang dihasilkan adalah nilai Z untuk tiap aturan yang terbentuk. Proses ini menghitung total penjumlahan dari perkalian k_{kj} dengan input ke- j untuk tiap cluster.

3. Proses Defuzzyfikasi

Proses defuzzyfikasi melibatkan beberapa input yaitu nilai Z dan *firestrength* serta menghasilkan *output* berupa nilai emosi dan kecocokan aturan yang terhadap data uji. defuzzyfikasi yang dilakukan menggunakan metode *weighted average*.

Setelah nilai emosi diketahui dan data uji termasuk kedalam Z tertentu, maka data dapat dikategorikan ke dalam suatu kelas. Pada penelitian ini, digunakan lima kelas emosi yaitu senang, sedih, marah, cinta dan takut. Rentang nilai tingkatan emosi untuk setiap kelas adalah apabila nilai Defuzzyfikasi Z_d dekat dengan Z_1 termasuk senang, Z_d dekat dengan Z_2 maka termasuk sedih, Z_d dekat dengan Z_3 maka termasuk marah, Z_d dekat dengan Z_4 maka termasuk cinta dan Z_d dekat dengan Z_5 maka termasuk kedalam kelas takut.

4.2 Perhitungan Manual

4.2.1 Perhitungan TF-IDF

Pada perhitungan manualisasi ini dilakukan proses perhitungan TF-IDF untuk kategori tiga jenis emosi. Data untuk proses perhitungan TF-IDF bisa dilihat pada Tabel 4.1.

Tabel 4.1 Data Twitter Mentah

data Twitter	
data 2	pergi bersenang-senang dengan teman-teman
data 4	sebahagiannya-bahagiannya kau saat ini ntar sedih-sedih juga
data 8	saat kau pergi dengannya aku pergi saja , bersenang-senang lah dengan teman mu itu
data 9	teman itu ada di saat senang dan sedih tidak saat senang saja

Setelah data dimasukkan maka proses selanjutnya adalah melakukan *stemming* dengan menggunakan metode *Stemming* yang telah dijelaskan pada bab 2 yaitu proses *casefolding* yang dilanjutkan ke *tokenisasi*, *filtering* dan terakhir *Stemming* yaitu mengambil kata dasar. Hasil *Stemming* bisa dilihat pada Tabel 4.2.

Tabel 4.2 Data Twitter Setelah Proses *Stemming*

data 2	pergi	senang	senang	teman	teman		
data 4	bahagia	bahagia	saat	sedih	sedih	juga	
data 8	saat	pergi	pergi	saja	senang	senang	teman
data 9	teman	saat	senang	sedih	saat	senang	saja

Langkah selanjutnya adalah melakukan proses TF-IDF dengan menggunakan metode TF-IDF. Proses awal melakukan DF yaitu menghitung frekuensi kata pada dokumen. Hasil DF dapat dilihat pada Tabel 4.3.

Tabel 4.3 Perhitungan nilai dan Document Frequency (DF)

proses perhitungan frekuensi <i>term</i>					
token/ <i>term</i>	D2	D4	D8	D9	Df
pergi	1	0	2	0	2
senang	2	1	2	1	4
teman	2	0	1	1	3
bahagia	0	2	0	0	1
saat	0	1	0	2	1
saja	0	0	1	1	2
sedih	0	2	0	1	2

Setelah dilakukan proses perhitungan nilai *DF* maka dilanjutkan dengan proses penghitungan nilai *Wtf* dan *Idf*. Nilai *Wtf* didapat dari LOG 10 nilai TF dan nilai *Idf* dari LOG 10 (jumlah *term*/*DF*). Hasil dari DF dapat dilihat pada Tabel 4.4.

Tabel 4.4 Perhitungan Nilai *Wtf* dan *Idf*

<i>Wtf</i>	D2	D4	D8	D9	<i>Idf</i>
pergi	0	0	0,30103	0	0,698970004
senang	0,30103	0	0,30103	0	0,397940009
teman	0,30103	0	0	0	0,522878745
bahagia	0	0,30103	0	0	1
saat	0	0	0	0,30103	1
saja	0	0	0	0	0,698970004
sedih	0	0,30103	0	0	0,698970004

Setelah mendapat nilai *Wtf* dan *Idf* maka proses dilanjutkan dengan menghitung *Wtd*. Nilai *Wtd* didapat dari *Wtf* * *Idf*. Hasil *Wtd* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Perhitungan Nilai Wtd

Wtd	D2	D4	D8	D9
pergi	0	0	0,21041094	0
senang	0,21041094	0	0,21041094	0
teman	0,21041094	0	0	0
bahagia	0	0,21041094	0	0
saat	0	0	0	0,21041094
saja	0	0	0	0
sedih	0	0,21041094	0	0

Proses terakhir dari perhitungan TF idf adalah menghitung normalisasi dari nilai yang dihasilkan Wtd. Normalisasi dilakukan dengan cara menjumlah nilai tiap dokumen dan hasil penjumlahan tersebut menjadi pembagi dari *term* atau kata pada suatu dokumen. Hasil normalisasi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Perhitungan TF-IDF yang Ternormalisasi

Wtd normalisasi	D2	D4	D8	D9
pergi	0	0	0,70710678	0
senang	0,70710678	0	0,70710678	0
teman	0,70710678	0	0	0
bahagia	0	0,70710678	0	0
saat	0	0	0	1
saja	0	0	0	0
sedih	0	0,70710678	0	0

4.2.2 Perhitungan *Fuzzy C-Means*

Tabel *input* untuk proses *fuzzy c-means* yang didapat dari proses TF-IDF. Pada perhitungan ini dilakukan menggunakan tiga *cluster* yaitu senang, sedih dan marah. Tabel *input* dapat dilihat pada Tabel 4.7.

Tabel 4.7 Input Matriks Data Latih (X_{ij}) dari TF-IDF

inputan FCM	fitur/variabel							emosi
	pergi	senang	teman	bahagia	saat	saja	sedih	
OBJEK A1	0	0,70710	0,70710	0	0	0	0	0,3
A2	0	0	0	0,70710	0	0	0,70710	0,6
A3	0,70710	0,70710	0	0	0	0	0	0,3
A4	0	0	0	0	1	0	0	0,9

Proses pertama setelah didapat nilai atau matriks *input* adalah melakukan pembangkitan nilai *random* sebagai keanggotaan *cluster* pada iterasi pertama. Hasil pembangkitan angka *random* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Pembangkitan Angka *Random*

keanggotaan <i>Cluster</i> (angka <i>Random</i>)				
	C1	C2	C3	Jumlah
A1	0,3	0,3	0,4	1
A2	0,3	0,5	0,2	1
A3	0,4	0,2	0,4	1
A4	0,5	0,2	0,3	1

Selanjutnya penetapan aturan *fuzzy c-means* akan didasarkan pada algoritme *fuzzy c-means* yang menentukan jumlah *cluster* yang akan dibuat, maksimal nilai iterasi dan nilai terkecil yang diharapkan. Tabel inisialisasi *fuzzy c-means* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Inisialisasi Awal *Fuzzy C-Means*

Rule FCM	
banyak <i>Cluster</i>	3
maksimal iterasi	100
Nilai pembobot	2
<i>epsilon</i>	0,0001

Setelah dilakukan perhitungan *fuzzy c-means*, berdasarkan algoritme *fuzzy c-means* maka didapatkan hasil pusat *cluster* iterasi pertama. Hasil pusat *cluster* iterasi pertama dapat dilihat pada Tabel 4.10.

Tabel 4.10 Pusat *Cluster* Iterasi Ke-1

	Pusat <i>Cluster</i>						
C1	0,191758	0,299622	0,107864	0,107864	0,423729	0	0,107864
C2	0,067344	0,218866	0,151523	0,420897	0,095238	0	0,420897
C3	0,251416	0,502831	0,251416	0,062854	0,2	0	0,062854

Setelah didapatkan pusat *cluster* maka didapatkan pula hasil fungsi objektif dan selisih fungsi objektif. Selisih fungsi objektif menentukan kapan proses iterasi akan berhenti yaitu apabila nilai selisih fungsi objektif lebih kecil dari *epsilon* atau jumlah iterasi lebih dari nilai maksimal iterasi yang ditentukan. Hasil dari fungsi objektif dan selisih fungsi objektif iterasi pertama dapat dilihat pada Tabel 4.11.

Tabel 4.11 Fungsi Objektif Iterasi Ke-1

Fungsi Objektif	0,882306
Selisih Fungsi Objektif	0,882306

Setelah melalui pembaruan nilai matriks U maka iterasi dinyatakan berhenti pada iterasi ke-6 dikarenakan sudah memenuhi satu syarat berhentinya iterasi yang nilai selisih fungsi objektif sudah lebih kecil dari *error* yang diharapkan. Hasil

pusat *cluster* dan selisih fungsi objektif iterasi ke-6 dapat dilihat pada Tabel 4.12 dan 4.13 .

Tabel 4.12 Pusat *Cluster* Iterasi Ke-6

	Pusat <i>Cluster</i>						
C1	0,007933	0,014757	0,006824	2,01E-07	0,97913	0	2,01E-07
C2	0,00793	0,014751	0,006821	0,692356	1,65E-07	0	0,692356
C3	0,346648	0,707107	0,360458	1,14E-07	1,67E-07	0	1,14E-07

Tabel 4.13 Fungsi Objektif Iterasi Ke-6

Fungsi Objektif	0,39857
Selisih Fungsi Objektif	2,38E05

Derajat keanggotaan akhir dari proses *fuzzy c-means* digunakan sebagai penentuan kecenderungan data terhadap cluster. Tabel derajat keanggotaan akhir atau matriks U dapat dilihat pada Tabel 4.14.

Tabel 4.14 Matriks *U* Sebagai Matriks Derajat Keanggotaan Akhir

Keanggotaan		
0,099193	0,099162	0,801645
0,000538	0,999011	0,000451
0,106948	0,106913	0,786139
0,999131	0,00041	0,000459

Setelah ditemukan pusat *cluster* yang baik dengan menggunakan analisis varian maka ekstraksi aturan dapat dilakukan. Aturan yang dihasilkan dari pusat *cluster* iterasi ke-6 adalah:

ekstraksi aturan

- R1 IF pergi = f11(pergi) AND senang = f12(senang) AND teman = f13(teman) AND bahagia = f14(bahagia) AND saat = f15(saat) AND saja = f16(saja) AND sedih = f17(sedih) THEN Emosi_Senang =
A11x1+A12X2+A13X3+A14X4+A15X5+A16X6+A17X7
- R2 IF pergi = f21(pergi) AND senang = f22(senang) AND teman = f23(teman) AND bahagia = f24(bahagia) AND saat = f25(saat) AND saja = f26(saja) AND sedih = f27(sedih) THEN Emosi_Sedih =
A21x1+A22X2+A23X3+A24X4+A25X5+A26X6+A27X7
- R3 IF pergi = f31(pergi) AND senang = f32(senang) AND teman = f33(teman) AND bahagia = f34(bahagia) AND saat = f35(saat) AND saja = f36(saja) AND sedih = f37(sedih) THEN Emosi_Marah =
A31X1+A32X2+A33X3+A34X4+A35X5+A36X6+A37X7

Setelah didapatkan aturan maka selanjutnya dicari kecenderungan data terhadap *cluster*. Kecenderungan data pada cluster dapat dilihat pada Tabel 4.15.

Tabel 4.15 Kecenderungan Data Pada Cluster

Kecenderungan data terhadap cluster				
Data	Derajat Keanggotaan ke-6			Cluster
A1	0,099	0,099	0,802	3
A2	0,001	0,999	0,000	2
A3	0,107	0,107	0,786	3
A4	0,999	0,000	0,000	1

Setelah kecenderungan data diketahui maka pengelompokan data dapat dilakukan. Selanjutnya adalah mencari rata-rata tiap fitur dari tiap *cluster*. Misal perhitungan rata-rata untuk data *cluster* 3 yang dapat dilihat pada Tabel 4.16.

Tabel 4.16 Perhitungan Rata-rata Untuk Data Cluster 3

data anggota cluster 3							
no	pergi	senang	Teman	bahagia	saat	saja	sedih
1	0	0,707107	0,707107	0	0	0	0
2	0,707107	0,707107	0	0	0	0	0
rata2	0,707107	0,707107	0,707107	0	0	0	0

Setelah proses *clustering* selesai maka dilakukan analisis varian untuk melihat apakah *cluster* sudah baik dan benar bila nilai V_w lebih kecil daripada V_b maka *cluster* dinyatakan baik.

V1	1	Vw	1,25	V	0,144231
V2	1	Vb	8,666667		
V3	0,5				

4.2.3 Perhitungan Koefisien *Output* dan Fungsi Gauss

4.2.3.1 Perhitungan Koefisien *Output*

Perhitungan untuk koefisien output akan dilakukan dengan memakai metode kuadrat terkecil (*least square*). Kudarat terkecil digunakan karena matriks yang digunakan tidak berbentuk persegi. Langkah-langkah untuk melakukan perhitungan koefisien *output* adalah sebagai berikut.

1. Normalisasi

Langkah pertama untuk dapat menghitung koefisien *output* adalah membentuk matriks d_{ij}^k dari perkalian matriks input X_{ij} dengan μ_{ik} dan nilai μ_{ik} untuk tiap *cluster*. Normalisasi akan dilakukan melalui cara membagi d_{ij}^k serta $d_{i(m+1)}^k$ dengan nilai dari jumlah derajat keanggotaan tiap titik data i di *cluster* k . Matriks d_{ij}^k ternormalisasi dapat dilihat pada Tabel 4.17.

Tabel 4.17 Matriks d_{ij}^k Ternormalisasi

0,00000	0,05817	0,05817	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00032	0,00000	0,00000	0,00032
0,06272	0,06272	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,82860	0,00000	0,00000
0,00000	0,05817	0,05817	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,58599	0,00000	0,00000	0,58599
0,06271	0,06271	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00034	0,00000	0,00000
0,00000	0,35680	0,35680	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00020	0,00000	0,00000	0,00020
0,34990	0,34990	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00029	0,00000	0,00000

2. Pemburatan Matriks U

Pembentukan matriks U diperoleh dari matriks d_{ij}^k . Untuk membentuk matriks U dilakukan perubahan dimensi *array* dari matriks d_{ij}^k . Matriks U dapat dilihat pada Tabel 4.19.

Tabel 4.18 Matriks U

0,000000	0,058168	0,058168	0,000000	0,000000
0,000000	0,000000	0,000000	0,000000	0,585989
0,062712	0,062712	0,000000	0,000000	0,000000
0,000000	0,000000	0,000000	0,000201	0,000000

3. Perhitungan *Least Square*

Perhitungan ini dilakukan dengan beberapa langkah. Langkah pertama adalah mencari nilai *transpose* dari matriks U yang dapat dilihat pada Tabel 4.19.

Tabel 4.19 Matriks U Transpose (U^T)

0	0	0,062711876	0
0,058168373	0	0,062711876	0
0,058168373	0	0	0
.....
.....
0	0,585989023	0	0
0	0	0	0,000288945
0	0	0	0
0	0,585989023	0	0

Langkah selanjutnya adalah melakukan perkalian antara matriks U^T dengan matriks U ($U^T * U$). hasil dari perkalian tersebut kemudian di invers sehingga dihasilkan matriks UxT berukuran persegi. Matriks hasil perkalian ($U^T * U$) dapat dilihat pada Tabel 4.20.

Tabel 4.20 Matriks UxT

0,003932	0,003932	0	0	0
0,003932	0,007316	0,003383	0	0
0	0,003383	0,003383	0	0
0	0	0	0	0,485549
.....
0	0	0	7,02892E-05	0	0
0	0	0	0	0,485549	0

Selanjutnya adalah melakukan perkalian antara hasil dari *invers* matriks dengan matriks transpose U ($UxT * U^T$). hasil dari perkalian ($UxT * U^T$) dapat dilihat pada Tabel 4.21.

Tabel 4.21 Matriks ($UxT * U^T$)

0,001632075	0	0,016460636	0
0,002483305	0	0,017974467	0
.....
0	0	0	0
0	0,808727482	0	0,01192606

Langkah terakhir adalah menghitung nilai koefisien *output* (K) yang didapat dari perkalian matriks ($UxT * U^T$) * Y . matriks Y adalah matriks emosi yang di dapat dari index matriks input dimana data latih menunjukkan suatu nilai dengan jarak kedekatan ke $Z1$ maka masuk kategori senang, kedekatan ke $Z2$ maka masuk kategori sedih dan kedekatan ke $Z3$ maka masuk katrgori marah. Hasil nilai K dapat dilihat pada Tabel 4.22.

Tabel 4.22 Matriks Koefisien Ouput (K)

0,005427814
0,006137332
.....
6,72045E-05
0
0,495969944

4.2.3.2 Fungsi Gauss

Untuk membuat fungsi Gauss, maka nilai dari *deviation standard* dan pusat *cluster* harus diketahui. Nilai pusat *cluster* didapat dari perhitungan *fuzzy c-means* dan nilai *deviation standard* didapat melalui persamaan 2.13. Contoh hasil perhitungan nilai *deviation standard* untuk data *cluster* 3 dapat dilihat pada Tabel 4.23.

Tabel 4.23 Deviation standard Cluster 3

Cluster 3		pusat cluster	<i>Deviation standard</i>
pergi	0,353553391	0,346648457	0,5
senang	0,707106781	0,707106549	0
teman	0,353553391	0,360458092	0,5
bahagia	0	1,14259E-07	0
saat	0	1,67155E-07	0
saja	0	0	0
sedih	0	1,14259E-07	0

Dengan didapatkannya nilai *deviation standard* dan nilai pusat *cluster* yang telah diketahui sebelumnya maka dapat dibentuk fungsi Gauss berdasarkan persamaan 2.10. hasil fungsi Gauss adalah sebagai berikut.

1. $f_{11}(\text{pergi}) = \mu_{\text{Var-1;Himp-1}}[X_i] = e^{(X_{i1}-0,007932746)^2/2(0)^2}$
2. $f_{12}(\text{senang}) = \mu_{\text{Var-2;Himp-1}}[X_i] = e^{(X_{i1}-0,0147567)^2/2(0)^2}$
3.
4.
36. $f_{36}(\text{saja}) = \mu_{\text{Var-6;Himp-1}}[X_i] = e^{(X_{i1}-0)^2/2(0)^2}$
37. $f_{37}(\text{sedih}) = \mu_{\text{Var-7;Himp-1}}[X_i] = e^{(X_{i1}-1,14259E-07)^2/2(0)^2}$

4.3 Pengujian Manual

Pada proses pengujian manual dilakukan menggunakan Microsoft Excel, pada proses pengujian manual terdapat langkah-langkah yang terdiri atas:

1. Fuzzyfikasi

Proses fuzzyfikasi dilakukan menggunakan derajat keanggotaan data uji dengan menggunakan hasil dari fungsi keanggotaan Gauss yang bersesuaian. Pada pengujian ini data uji yang akan di ujiakan adalah “pergi saja dengan teman agar sedih”. Data uji menuju proses TF-IDF yang dimasukkan sebagai *query*. Data uji hasil TF-IDF adalah sebagai berikut.

- Pergi : 0,707106781
- Senang : 0
- Teman : 0,707106781
- Bahagia : 0
- Saat : 0
- Saja : 0,707106781
- Sedih : 0,707106781

Dari data tersebut maka dapat dihitung derajat keanggotaan berdasarkan fungsi Gauss dengan atribut yang bersesuaian yang dapat dilihat pada Tabel 4.24.

Tabel 4.24 Derajat Keanggotaan Dari Fungsi Gauss

	Atribut	1	2	3	4	5	6	7
Aturan	1	1,63	1,00	1,63	1	2,60	1,64	1,64
	2	1,63	1,007	1,63	1,61	1	1,64	1,00
	3	1,29	1,64	1,27	1	1	1,64	1,64

2. Perhitungan *Fire-Strength*

Setelah derajat keanggotaan pada data uji didapat maka dicari nilai *fire-strength* untuk setiap aturan dengan cara mengkalikan setiap derajat keanggotaan yang telah didapat tiap *cluster*. Perhitungan *fire-strength* adalah sebagai berikut.

$$\alpha_1 = 18,88092598 \quad \alpha_2 = 7,092504262 \quad \alpha_3 = 7,390454233$$

3. Perhitungan Z Setiap Aturan

Hasil perhitungan vektor Z didapat berdasarkan persamaan 2.17 adalah sebagai berikut.

$$Z_1 = 0,004372787 \quad Z_2 = 0,092359067 \quad Z_3 = 0,373081331$$

4. Defuzzyfikasi

Defuzzyfikasi dilakukan dengan menggunakan metode *weighted average* sebagai berikut.

$$Z = (\alpha_1 Z_1 + \alpha_2 Z_2 + \alpha_3 Z_3) / \alpha_1 + \alpha_2 + \alpha_3 = 0,104749788$$

Dengan hasil tersebut maka data uji lebih dekat ke aturan 2 yaitu 0,0923 dan ber emosi sedih.

4.4 Perancangan Antarmuka

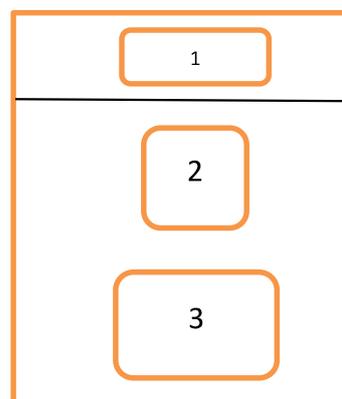
Antarmuka merupakan alat supaya pengguna bisa berkomunikasi dengan sistem. Pada sistem ini terdapat dua antarmuka utama yaitu antar muka untuk pelatihan data dan antarmuka untuk pengujian data.

4.4.1 Antarmuka Pelatihan Data

Antarmuka ini adalah halaman utama yang akan muncul saat program dijalankan. Halaman ini berisi mulai dari *upload* data latih samapai menampilkan hasil pre-pemrosesan, TF-IDF, derajat keanggotaan FCM, pusat *Cluster*, dan hasil dari ekstraksi aturan.

Keterangan:

1. *Header*
2. Menu *input* data latih dan toml proses
3. Hasil *output* dan menu menuju pengujian data



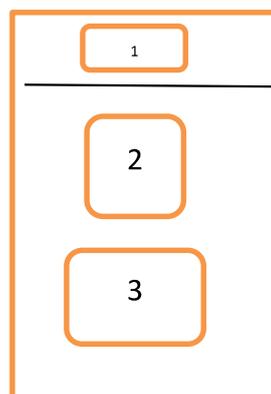
Gambar 4.11 Antarmuka pelatihan data

4.4.2 Antarmuka Pengujian Data

Antarmuka ini adalah antar muka untuk melakukan pengujian data setelah dilakukan proses pelatihan data terlebih dahulu. Halaman ini akan muncul apabila pengguna menekan tombol pengujian data pada halaman awal setelah melakukan pelatihan terlebih dahulu.

Keterangan:

1. *Header*
2. Menu *input* data uji dan tombol uji
3. Hasil *output* pengujian dan akurasi pengujian



Gambar 4.12 Antarmuka pengujian Data

4.5 Penarikan Simpulan

Penarikan simpulan dapat dilakukan apabila perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Simpulan ini diambil dari hasil pengujian yang dilakukan sistem dan analisis dari sistem yang telah dibuat. Selanjutnya akan dilakukan penulisan saran. Saran ditulis sebagai pertimbangan untuk penelitian dengan teknik yang sama yang akan dilakukan untuk pengembangan yang lebih baik.