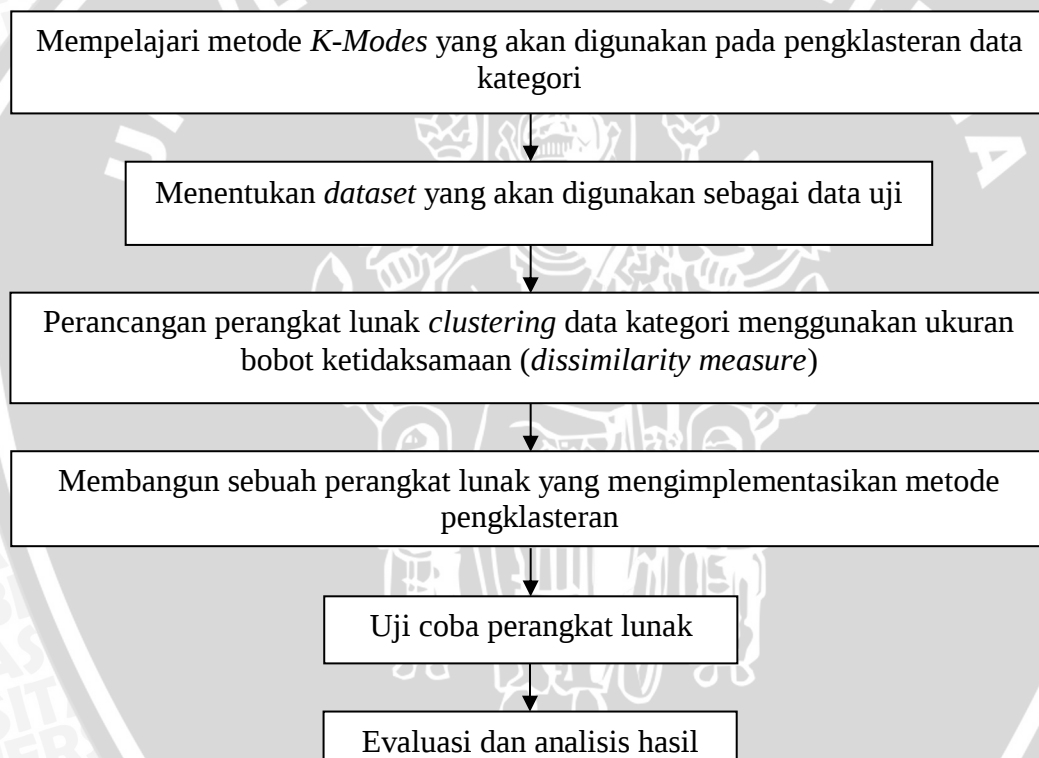


BAB III

METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam menyusun penelitian. Pada bab metodologi dan perancangan ini berisi penjelasan mengenai metode yang digunakan dan langkah – langkah yang dilakukan dalam pembuatan perangkat lunak implementasi *K-Modes* pada *clustering* data kategori menggunakan *new dissimilarity measure*. Tahap – tahap penelitian yang dilakukan dapat diilustrasikan pada gambar 3.1.



Gambar 3.1 Tahapan-tahapan Penelitian

Berdasarkan gambar 3.1 tahap-tahap yang dilakukan pada penelitian ini adalah sebagai berikut.

1. Mempelajari metode yang akan digunakan yaitu *K-Modes* pada *clustering* data kategori menggunakan *new dissimilarity measure* dari jurnal-jurnal terkait yang sudah ada.

2. Menentukan *dataset* yang akan digunakan sebagai data uji. Data uji yang digunakan pada penelitian adalah data uji yang memiliki tipe data *string*.
3. Melakukan analisis dan perancangan perangkat lunak *K-Modes* pada *clustering* data kategori menggunakan *new dissimilarity measure*.
4. Membuat perangkat lunak berdasarkan analisis dan perancangan yang sudah dilakukan.
5. Melakukan uji coba terhadap perangkat lunak.
6. Melakukan evaluasi atau analisis hasil yang diperoleh dari uji coba perangkat lunak.

3.1 Deskripsi Data

Dataset yang digunakan pada penelitian ini adalah *Lenses*. Semua *dataset* yang digunakan diambil dari *UCI Machine Learning Repository* dan semua *dataset* adalah data berkategori. Detail deskripsi dari *dataset* dapat dilihat pada tabel 3.1.

Tabel 3.1 Deskripsi *Dataset*

Dataset	Jumlah Data (object)	Jumlah Atribut	Jumlah Kategori (class)
<i>Balance-Scale</i>	625	4	3
<i>Car Evaluation</i>	1728	6	4
<i>Lenses</i>	24	4	3
<i>Tic Tac Toc</i>	958	9	2

3.2 Analisis Perangkat Lunak

3.2.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang sedang dibuat merupakan perangkat lunak yang dapat mengkluster data berkategori menggunakan *K-Modes*. Ukuran ukuran jarak yang digunakan adalah ukuran *new dissimilarity measure*. Hasil dari perangkat lunak ini nantinya dibandingkan dengan *K-Modes* yang asli yaitu menggunakan ukuran jarak *dissimilarity (simple mismatching)*. Inputan dari perangkat lunak adalah *file*

text dataset dan *output* dari perangkat lunak berupa nilai tingkat kemurnian *cluster* (*purity measure*) dan *F-Measure*.

3.2.2 Batasan Perangkat Lunak

Batasan perangkat lunak yang dikembangkan pada sistem ini adalah.

1. Inputan dari perangkat lunak adalah dataset berupa *file text*.
2. Output dari perangkat lunak berupa nilai tingkat kemurnian *cluster* (*purity measure*) dan *F-Measure*.
3. Ada dua pilihan ukuran jarak yang digunakan oleh algoritma *K-Modes* yaitu *dissimilarity* (*simple mismatching*) dan *new dissimilarity measure*.
4. Perangkat lunak dibuat menggunakan bahasa pemrograman Visual C# 2010.

3.3 Perancangan Perangkat Lunak

3.3.1 Perancangan proses

Sistem yang dibuat terdapat empat buah proses utama, yaitu :

1. proses utama system.
2. proses pengklasteran dengan *K-Modes New Dissimilarity Measure*.
3. proses pengklasteran dengan *K-Modes Konvensional*.
4. evaluasi dengan *Purity-Measure*.
5. evaluasi dengan *F-Measure*.

1. Proses Utama sistem

Proses *request data* terjadi ketika *user* melakukan pemilihan terhadap data uji yang selanjutnya akan digunakan untuk penelitian. Pada Gambar 3.2 dan 3.3 dapat diamati setelah sistem memilih *dataset*, kemudian dilakukan proses *clustering K-Modes* (*K-Modes New Dissimilarity Measure* dan *K-Modes Konvensional*), kemudian evaluasi *clustering* data (evaluasi dilakukan menggunakan *Purity Measure* dan *F-Measure*), dan sistem selesai berjalan. Gambar 3.2 dan 3.3.

2. Proses Metode Pengklasteran *K-Modes New Dissimilarity Measure*

Proses yang terjadi pada pengklasteran dengan *K-Modes New Dissimilarity Measure* adalah melakukan inisialisasi *cluster* 'k' awal sebanyak *class*. Pada *new dissimilarity measure* terdapat bobot yang didapatkan dari perkalian perbandingan nilai atribut di *cluster* dengan perbandingan nilai atribut di *dataset*, yang menggunakan persamaan ukuran jarak *new dissimilarity measure* pada persamaan 2.12. Setelah jarak telah terhitung semua, kelompokkan setiap *record* berdasarkan *cluster* terdekat dan *update cluster* baru. Cek apakah *cluster* baru sama dengan *cluster* lama, jika tidak, ulangi langkah dari mencari jarak hingga perbandingan klaster, jika sama, maka klaster akan dimasukkan ke dalam database dan proses berhenti. *Flowchart* pengklasteran *K-Modes New Dissimilarity Measure* ditunjukkan oleh Gambar 3.6.

3. Proses Metode Pengklasteran *K-Modes Konvensional (Simple Mismatching)*

Proses yang terjadi pada pengklasteran dengan *K-Modes Konvensional* adalah melakukan inisialisasi *cluster* 'k' awal sebanyak *class*. Selanjutnya hitung jarak *Simple Mismatching* masing-masing atribut antar *record*. Setelah jarak telah terhitung semua, kelompokkan setiap *record* berdasarkan *cluster* terdekat dan *update cluster* baru. Cek apakah *cluster* baru sama dengan *cluster* lama, jika tidak, ulangi langkah dari mencari jarak hingga perbandingan klaster, jika sama, maka klaster akan dimasukkan ke dalam database dan proses berhenti. *Flowchart* pengklasteran *K-Modes Konvensional* ditunjukkan oleh Gambar 3.7.

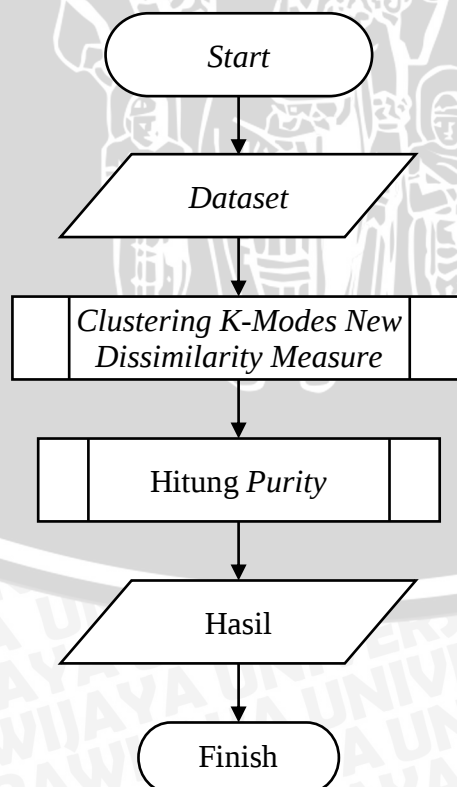
4. Proses Metode Evaluasi *Purity Measure*

Menghitung *purity* digunakan sebagai ukuran akurasi algoritma *clustering*. Tiap data/objek yang ada di *cluster* akan dibandingkan dengan *cluster* yang asli apakah data masuk dalam *cluster* yang sesuai. Kemudian dihitung jumlah objek yang masuk dalam *cluster* yang sesuai menggunakan persamaan $r = \frac{\sum_{i=1}^k a_i}{n}$ dimana a_i adalah objek yang masuk dalam *cluster* C_i dan sesuai dengan hasil proses *clustering*. *Flowchart* hitung *purity* dapat dilihat pada gambar 3.8.

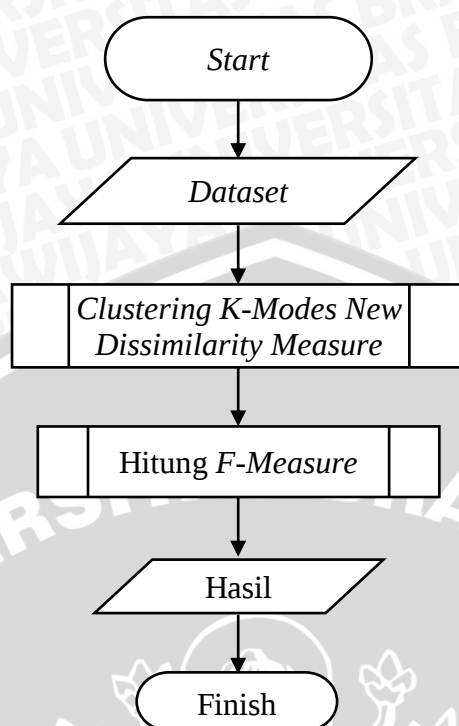
5. Proses Metode Evaluasi *F-Measure*

Sistem mengambil sederet *class* di *dataset* yang sebenarnya untuk dijadikan bahan pengujian. Sistem akan menilai banyaknya anggota *class* dalam sebuah *cluster* untuk dijadikan *true positive* (TP). Selanjutnya, sistem akan mencari anggota yang sebenarnya masuk *cluster* tertentu, tetapi dikelompokkan ke dalam *cluster* yang berbeda untuk dijadikan *false negative* (FN). Kemudian, sistem akan mencari anggota yang tidak dominan pada *cluster* tertentu untuk dijadikan *false positive* (FP). Setelah mendapatkan *true positive* (TP), *false negative* (FN), dan *false positive* (FP), perhitungan *precision*, *recall*, dan *f-measure* akan dilakukan oleh sistem per *cluster*. Setelah semua *cluster* telah dihitung nilainya, maka akan dilakukan rata-rata *f-measure* dari semua *cluster*. Flowchart metode evaluasi *f-measure* akan ditunjukkan oleh Gambar 3.9.

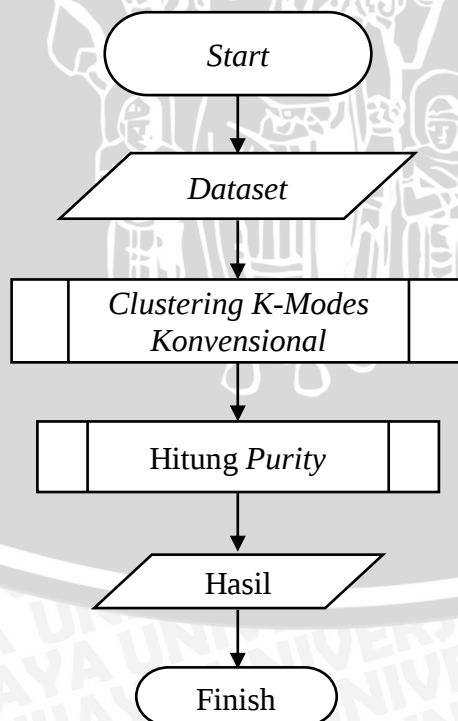
Desain Sistem yang dibuat menggunakan 2 evaluasi, yaitu menggunakan *Purity Measure* dan *F-Measure*.



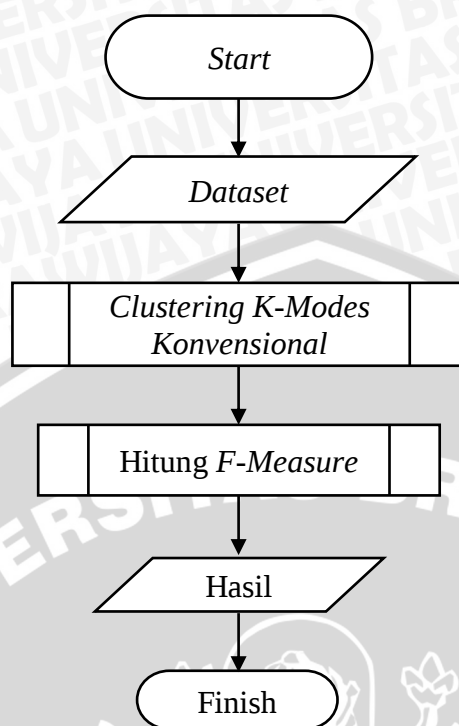
Gambar 3.2 Flowchart Proses Perangkat Lunak K-Modes New Dissimilarity Measure menggunakan evaluasi *Purity Measure*



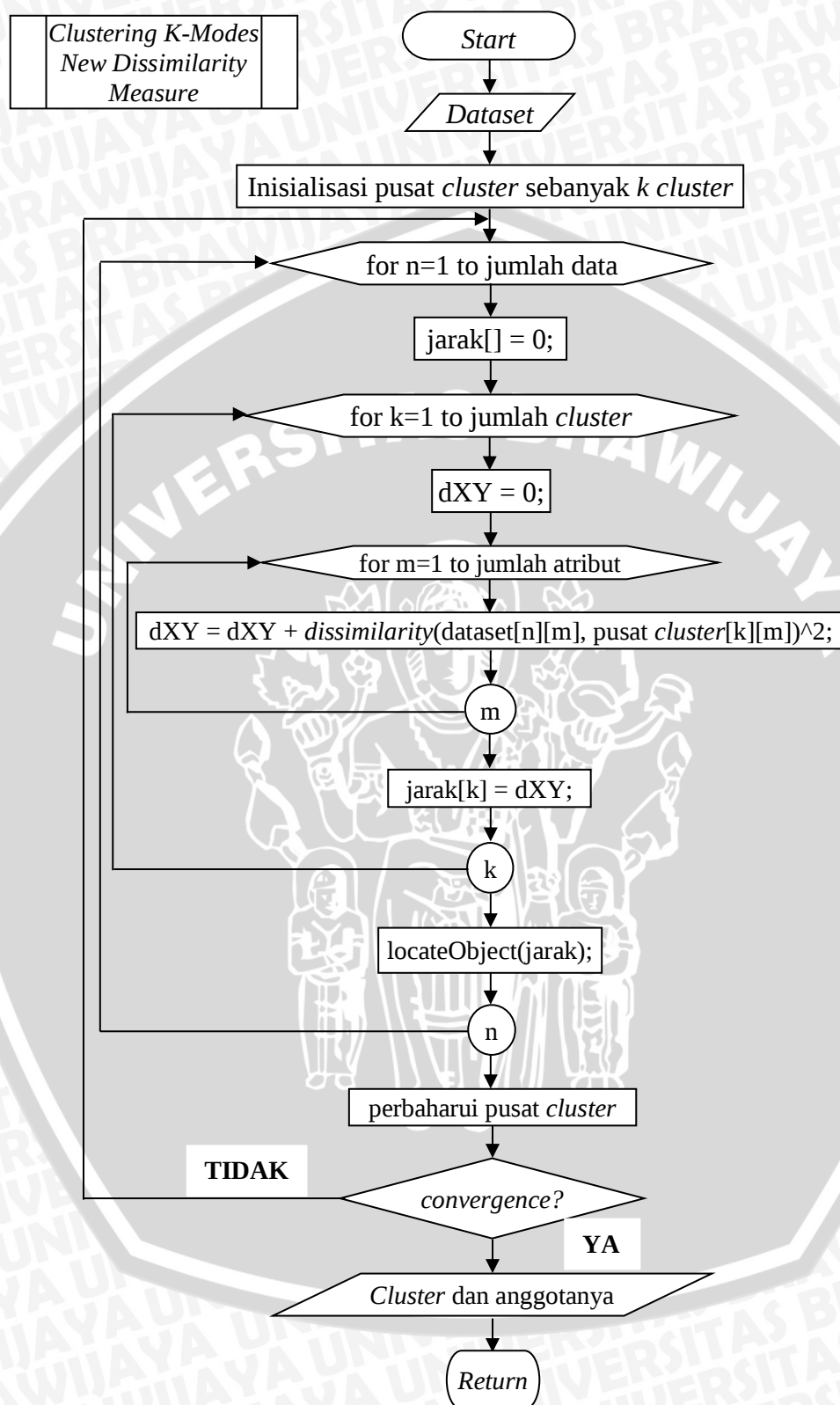
Gambar 3.3 Flowchart Proses Perangkat Lunak K-Modes New Dissimilarity Measure menggunakan evaluasi *F-Measure*



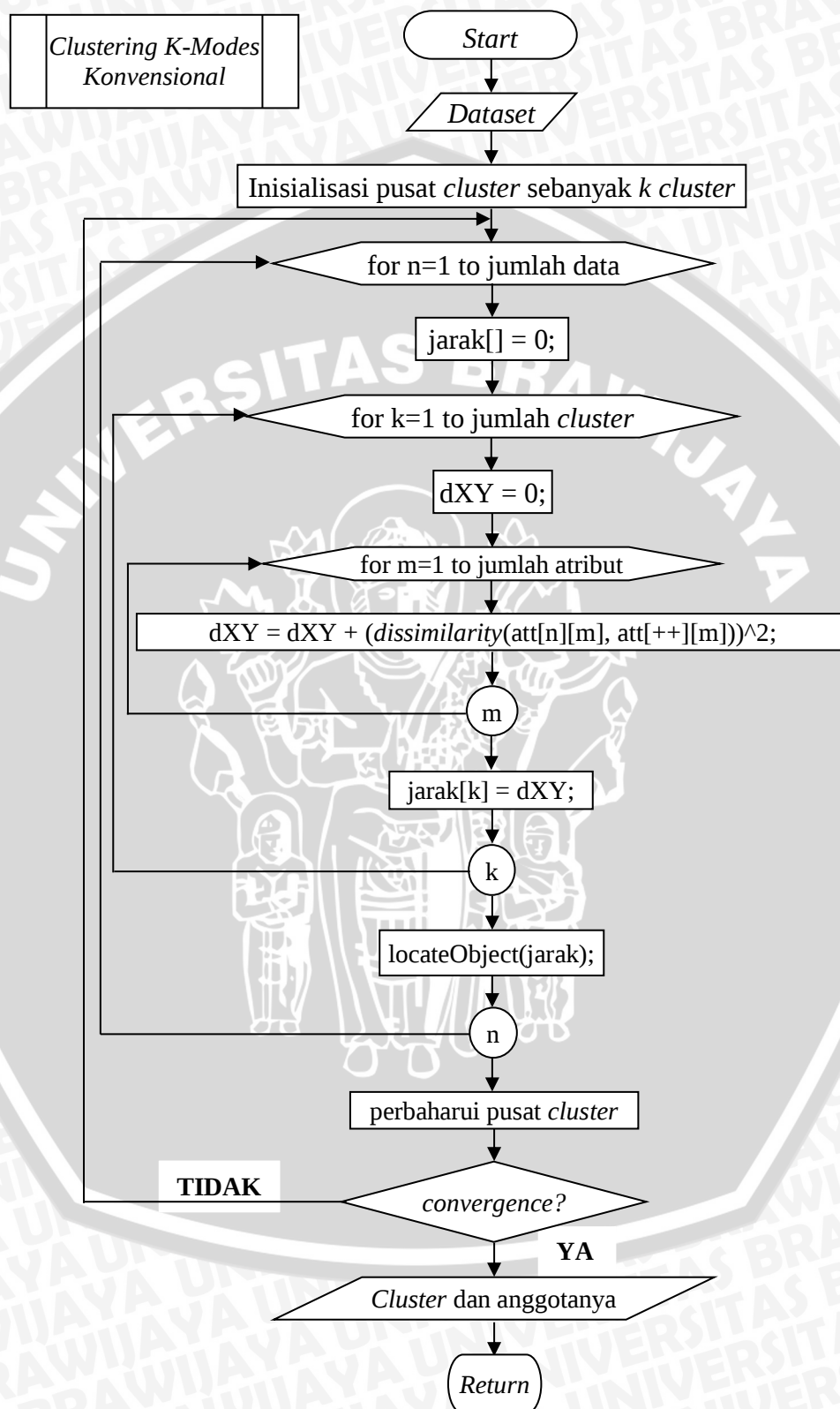
Gambar 3.4 Flowchart Proses Perangkat Lunak K-Modes Konvensional menggunakan evaluasi *Purity Measure*



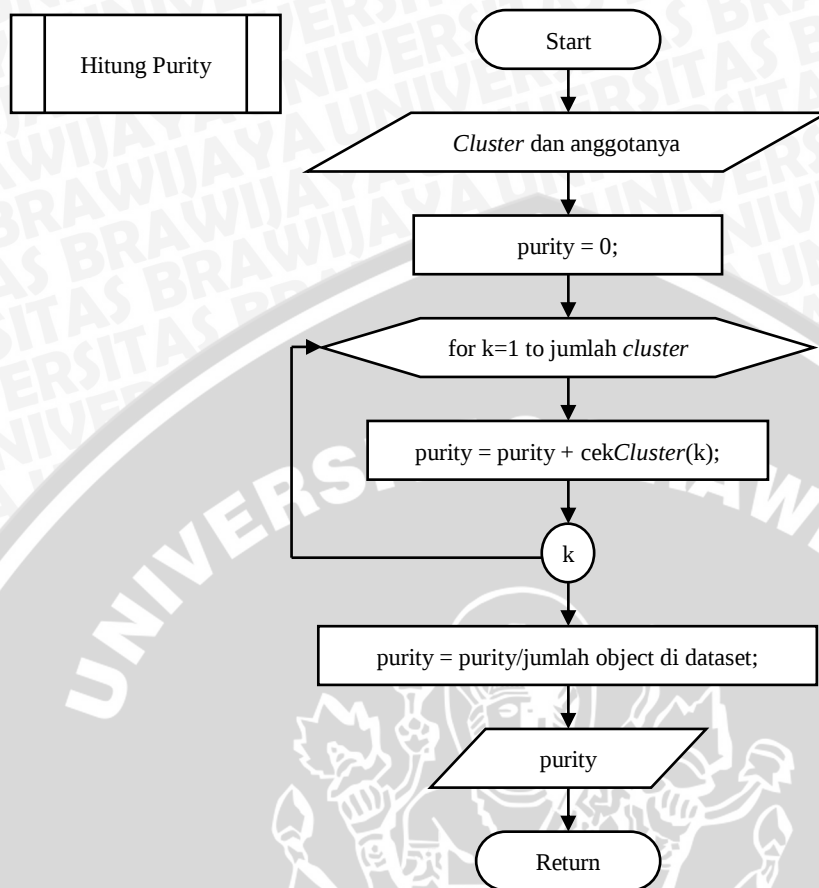
Gambar 3.5 Flowchart Proses Perangkat Lunak K-Modes Konvensional menggunakan evaluasi *F-Measure*



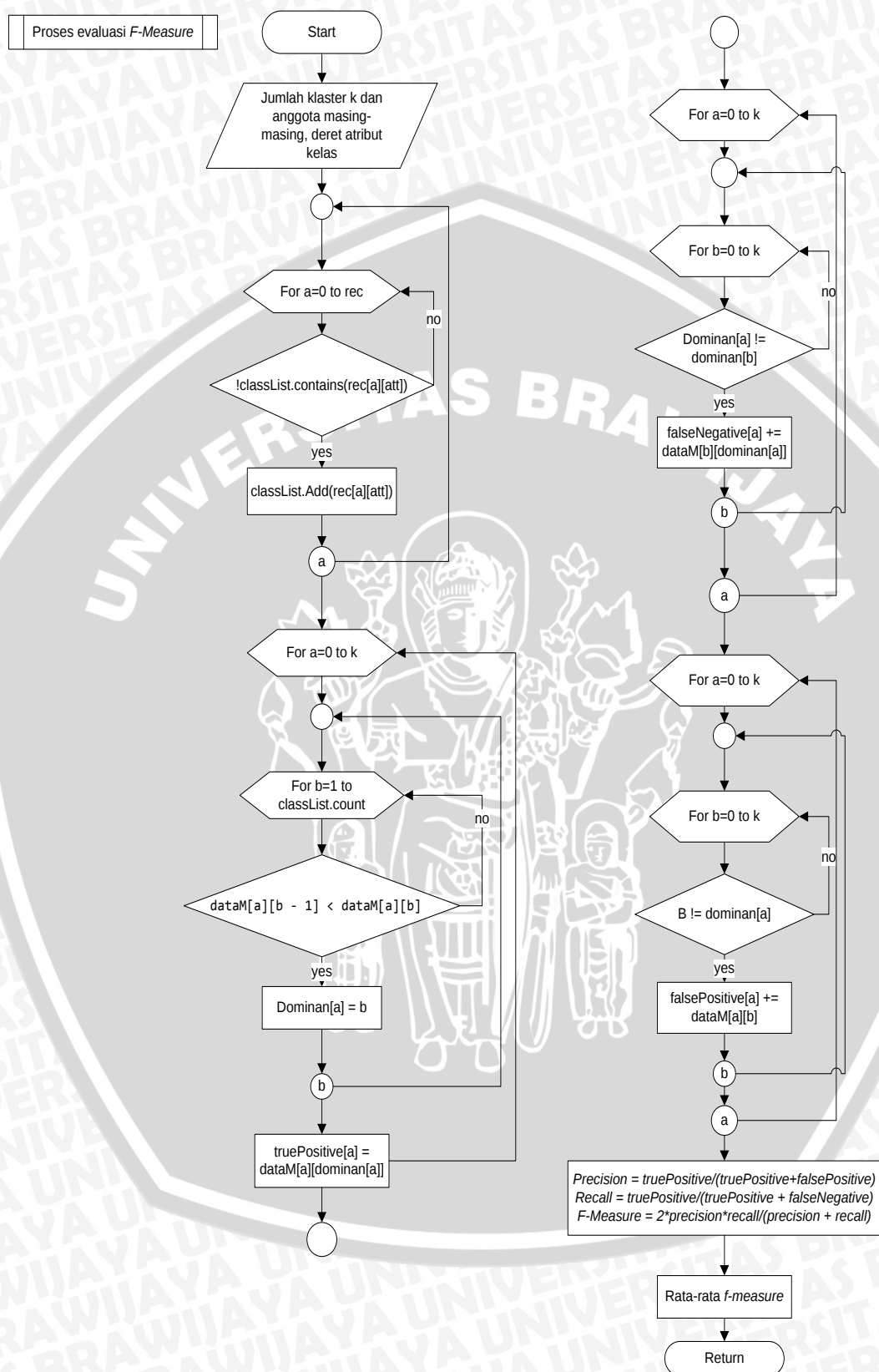
Gambar 3.6 Flowchart Algoritma Clustering K-Modes New Dissimilarity Measure



Gambar 3.7 Flowchart Algoritma Clustering K-Modes Konvensional



Gambar 3.8 Flowchart Hitung Purity Measure

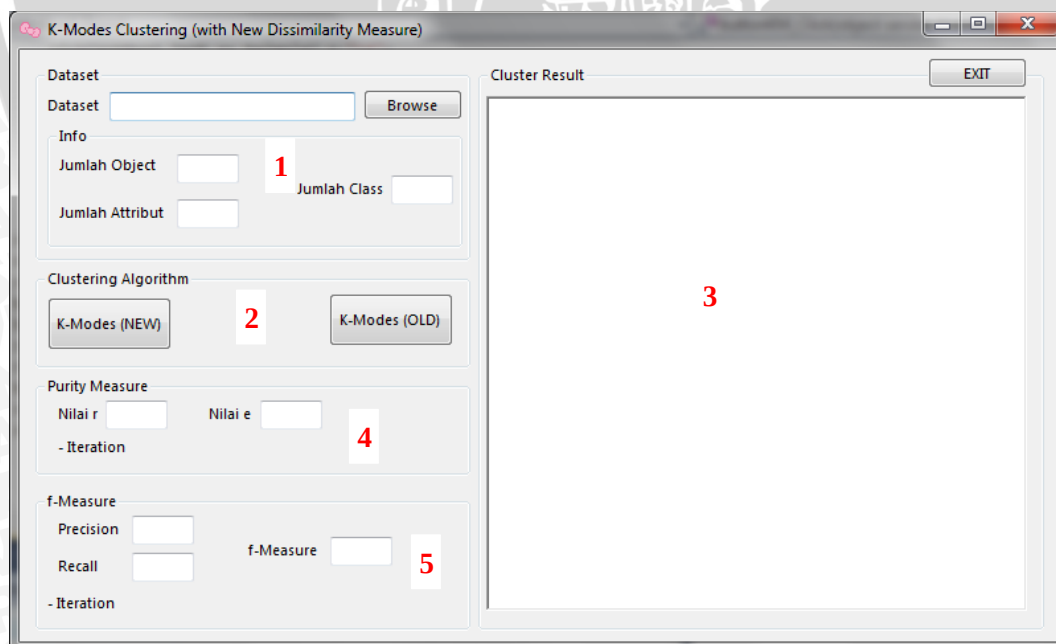


Gambar 3.9 Flowchart Evaluasi F-Measure

3.4 Perancangan Antarmuka

Rancangan antarmuka yang akan digunakan oleh sistem ini terdiri dari satu *form*. Pada satu *form* tersebut terdiri dari 5 bagian/*group* yaitu:

1. Pada bagian pertama berisi *input file text dataset* dan informasi yang didapat dari *file inputan* berupa jumlah *object*, jumlah atribut, dan jumlah *class*.
2. Pada bagian kedua berisi *radio button* untuk memilih algoritma *clustering* yang digunakan yaitu *K-Modes(NEW)* atau *K-Modes(OLD)*.
3. Pada bagian ketiga *cluster result* berisikan hasil dari *clustering* yang menginformasikan pada *cluster* hasil terdiri dari *class* apa saja dan nilai *precision*, *recall*, dan *F-Measure*.
4. Pada bagian keempat *Purity Measure* berisikan hasil perhitungan *purity measure* atau nilai kemurnian *clustering* dan nilai *error*.
5. Pada bagian kelima *F-Measure* berisikan hasil perhitungan *F-Measure* atau pengukuran validitas pada hasil *clustering* yang menghasilkan *precision* (tingkat ketepatan hasil pengelompokan terhadap suatu *event*.), *recall* (ukuran dari jumlah *record* benar yang berhasil dikelompokkan oleh sistem dari keseluruhan *record* yang seharusnya benar), dan *F-Measure*.



Gambar 3.10 Rancangan Antarmuka

3.5 Perancangan Uji Coba

Pada penelitian ini akan dilakukan uji coba tingkat kemurnian (*purity*) algoritma *K-Modes* konvensional dan *K-Modes* dengan ukuran jarak *new dissimilarity measure*. Hasil dari uji coba tersebut kemudian dibandingkan manakah yang menghasilkan *cluster* lebih baik antara algoritma *K-Modes* yang asli dengan *K-Modes* dengan ukuran jarak *new dissimilarity measure*. Pengujian dilakukan menggunakan beberapa dataset dan hasilnya akan dirata-rata. Tabel pengujian dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Uji Coba

Dataset	Attribut	Class	Purity (%)	
			New K-Modes	Old K-Modes
Balance-Scale				
Car Evaluation				
Lenses				
Tic Tac Toc				
Rata-rata				

3.6 Hitung Manual

Tersedia *dataset lenses* dengan atribut sebanyak 4 dan 3 *class*. Inputkan *k* sebagai banyak *cluster*, dalam contoh ini jumlah *k* untuk pengklasteran adalah *k* = 3, *dataset Lenses* pada Tabel 3.3.

Tabel 3.3 Tabel Dataset Lenses

Record	Attribut 1	Attribut 2	Attribut 3	Attribut 4	Class
1	muda	rabun jauh	tidak ada	kurang	3
2	muda	rabun jauh	tidak ada	normal	2
3	muda	rabun jauh	ya	kurang	3
4	muda	rabun jauh	ya	normal	1

5	muda	rabun dekat	tidak ada	kurang	3
6	muda	rabun dekat	tidak ada	normal	2
7	muda	rabun dekat	ya	kurang	3
8	muda	rabun dekat	ya	normal	1
9	pra presbyopic	rabun jauh	tidak ada	kurang	3
10	pra presbyopic	rabun jauh	tidak ada	normal	2
11	pra presbyopic	rabun jauh	ya	kurang	3
12	pra presbyopic	rabun jauh	ya	normal	1
13	pra presbyopic	rabun dekat	tidak ada	kurang	3
14	pra presbyopic	rabun dekat	tidak ada	normal	2
15	pra presbyopic	rabun dekat	ya	kurang	3
16	pra presbyopic	rabun dekat	ya	normal	3
17	presbyopic	rabun jauh	tidak ada	kurang	3
18	presbyopic	rabun jauh	tidak ada	normal	3
19	presbyopic	rabun jauh	ya	kurang	3
20	presbyopic	rabun jauh	ya	normal	1
21	presbyopic	rabun dekat	tidak ada	kurang	3
22	presbyopic	rabun dekat	tidak ada	normal	2
23	presbyopic	rabun dekat	ya	kurang	3
24	presbyopic	rabun dekat	tidak ada	normal	3

Kemudian pilih *cluster* secara acak untuk di uji, misal :

Tabel 3.4 Tabel *Dataset Lenses* yang telah di pilih *random*

<i>Record</i>	<i>Attribut 1</i>	<i>Attribut 2</i>	<i>Attribut 3</i>	<i>Attribut 4</i>	<i>Class</i>
1	muda	rabun jauh	tidak ada	kurang	3
2	muda	rabun jauh	tidak ada	normal	2
3	muda	rabun jauh	ya	kurang	3
4	muda	rabun jauh	ya	normal	1
6	muda	rabun dekat	tidak ada	normal	2
8	muda	rabun dekat	ya	normal	1

Keterangan:

➤ *Class*

- 1 : Pasien yang cocok dengan kontak lens keras (*hard contact lenses*)
- 2 : Pasien yang cocok dengan kontak lens lunak (*soft contact lenses*)
- 3 : Pasien yang tidak perlu memakai kontak lens (*contact lenses*)

Keterangan *record* yang telah dipilih:

a. *Record-1:*

(1 = muda, 1 = rabun jauh (*myope*), 1 = tidak ada, dan 1 = kurang) *Class - 3* = tidak perlu memakai kontak lens (*contact lenses*)

b. *Record-2:*

(1 = muda, 1 = rabun jauh (*myope*), 1 = tidak ada, dan 2 = normal)
Class - 2 = *soft contact lenses*

c. *Record-3:*

(1 = muda, 1 = rabun jauh (*myope*), 2 = ya, dan 1 = kurang)
Class - 3 = tidak perlu memakai kontak lens (*contact lenses*)

d. *Record-4:*

(1 = muda, 1 = rabun jauh (*myope*), 2 = ya, dan 2 = normal)
Class - 1 = *hard contact lenses*

e. *Record-5:*

(1 = muda, 2 = rabun dekat (*hypermetrope*), 1 = tidak ada, dan 2 = normal)

Class - 2 = soft contact lenses

f. *Record-6:*

(1 = muda, 2 = rabun dekat (*hypermetrope*), 2 = ya, dan 2 = normal)

Class - 1 = hard contact lenses

3.6.1 Hitung *K-Modes New Dissimilarity Measure*

Pada algoritma *K-Modes* langkah pertama inisialisasi pusat *cluster* sebanyak *k cluster* dalam contoh ini *k* bernilai 3 sesuai dengan banyak *class*. Pusat *cluster* bisa dipilih dari *record* yang sudah ada misalkan pusat *cluster* pertama ada di *record* nomor 1, pusat *cluster* kedua ada di *record* nomor 2, dan pusat *cluster* ketiga ada di *record* nomor 4.

Tabel 3.5 Tabel Pusat *Cluster*

	Attribut 1	Attribut 2	Attribut 3	Attribut 4
Pusat <i>Cluster</i> 1	muda	rabun jauh	tidak ada	kurang
Pusat <i>Cluster</i> 2	muda	rabun jauh	tidak ada	normal
Pusat <i>Cluster</i> 3	muda	rabun jauh	ya	normal

Setelah ditentukan pusat *cluster* kemudian setiap *record* dialokasikan ke pusat *cluster* terdekat dengan menghitung jarak setiap *record* dengan pusat *cluster* menggunakan persamaan 2.1.

Perhitungan Jarak *Record* dan Pusat *Cluster*

Menggunakan persamaan 2.1, persamaan 2.3, dan persamaan 2.4

1. Perhitungan jarak *record* 1 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned}
 d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\
 &= 0 + (1-4/6) + (1-3/6) + (1-2/6) \\
 &= 1.5
 \end{aligned}$$

pusat *cluster* 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1$$

$$= 0 + (1-4/6) + (1-3/6) + 1$$

$$= 1,833$$

pusat *cluster* 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + 1$$

$$= 0 + (1-4/6) + 1 + 1$$

$$= 2,333$$

2. Perhitungan jarak *record* 2 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1$$

$$= 0 + (1-4/6) + (1-3/6) + 1$$

$$= 1,833$$

pusat *cluster* 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij})$$

$$= 0 + (1-4/6) + (1-3/6) + (1-4/6)$$

$$= 1,166$$

pusat *cluster* 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij})$$

$$= 0 + (1-4/6) + 1 + (1-4/6)$$

$$= 1,666$$

3. Perhitungan jarak *record* 3 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij})$$

$$= 0 + (1-4/6) + 1 + (1-2/6)$$

$$= 2$$

pusat *cluster* 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + 1$$

$$= 0 + (1-4/6) + 1 + 1$$

$$= 2,333$$

pusat *cluster* 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1$$

$$= 0 + (1-4/6) + (1-3/6) + 1 = \mathbf{1.833}$$

4. Perhitungan jarak *record* 4 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + 1 \\ &= 0 + (1-4/6) + 1 + 1 \\ &= \mathbf{2.333} \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij}) \\ &= 0 + (1-4/6) + 1 + (1-4/6) \\ &= \mathbf{1.666} \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + (1-4/6) + (1-3/6) + (1-4/6) \\ &= \mathbf{1.166} \end{aligned}$$

5. Perhitungan jarak *record* 6 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + 1 \\ &= 0 + 1 + (1-3/6) + 1 \\ &= \mathbf{2.5} \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + 1 + (1-3/6) + (1-4/6) \\ &= \mathbf{2.166} \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + (1-w_{ij}) \\ &= 0 + 1 + 1 + (1-4/6) = \mathbf{2.333} \end{aligned}$$

6. Perhitungan jarak *record* 8 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$d(x,y) = (1-w_{ij}) + 1 + 1 + 1$$

$$= 0 + 1 + 1 + 1$$

$$= 3$$

pusat *cluster* 2 ->

$$d(x,y) = (1-w_{ij}) + 1 + 1 + (1-w_{ij})$$

$$= 0 + 1 + 1 + (1-4/6)$$

$$= 2.333$$

pusat *cluster* 3 ->

$$d(x,y) = (1-w_{ij}) + 1 + (1-w_{ij}) + (1-w_{ij})$$

$$= 0 + 1 + (1-3/6) + (1-4/6)$$

$$= 1.833$$

Tabel 3.6. Jarak dan *Cluster* (Iterasi-1)

<i>Record</i>	Jarak dengan Pusat <i>Cluster</i> 1	Jarak dengan Pusat <i>Cluster</i> 2	Jarak dengan Pusat <i>Cluster</i> 3	<i>Cluster</i>
1	1.5	1.833	2.333	1
2	1,833	1,166	1,666	2
3	2	2.333	1.833	3
4	2.333	1.666	1.166	3
6	2.5	2.166	2.333	2
8	3	2.333	1.833	3

Hasil :

Pusat *Cluster* 1 = {1}

Pusat *Cluster* 1 = {2, 6}

Pusat *Cluster* 1 = {3, 4, 8}

Dari hasil perhitungan jarak minimal *record* 1 dengan pusat *cluster* 1 jadi *record* 1 masuk dalam *cluster* 1. Untuk semua *record* dapat dilihat pada tabel 3.7.

Tabel 3.7 Hasil *Cluster* Semua *Record*

<i>Record</i>	<i>Attribut 1</i>	<i>Attribut 2</i>	<i>Attribut 3</i>	<i>Attribut 4</i>	<i>Class</i>
1	muda	rabun jauh	tidak ada	kurang	1
2	muda	rabun jauh	tidak ada	normal	2
3	muda	rabun jauh	ya	kurang	3
4	muda	rabun jauh	ya	normal	3
6	muda	rabun dekat	tidak ada	normal	2
8	muda	rabun dekat	ya	normal	3

Setelah semua *record* masuk dalam *cluster* terdekat, selanjutnya adalah memperbaharui semua pusat *cluster* dengan metode frekuensi. Misalkan untuk pusat *cluster* 1 atribut 1, nilai yang sering muncul adalah 1, atribut 2 nilai 1, atribut 3 nilai 1 dan 2 frekuensinya sama jadi nilainya tetap, atribut 4 nilai 1. Jadi pusat *cluster* 1 yang baru adalah 1, 1, 1, 1. Tabel 3.7 menunjukkan pusat *cluster* yang telah diperbaharui.

Tabel 3.8 Tabel Pusat *Cluster* yang Baru

	<i>Attribut 1</i>	<i>Attribut 2</i>	<i>Attribut 3</i>	<i>Attribut 4</i>
Pusat <i>Cluster</i> 1	muda	rabun jauh	tidak ada	kurang
Pusat <i>Cluster</i> 2	muda	rabun jauh	tidak ada	normal
Pusat <i>Cluster</i> 3	muda	rabun jauh	ya	kurang

Setelah ditentukan pusat *cluster* kemudian setiap *record* dialokasikan ke pusat *cluster* terdekat dengan menghitung jarak setiap *record* dengan pusat *cluster* menggunakan persamaan 2.1.

Perhitungan Jarak *Record* dan Pusat *Cluster*

Menggunakan persamaan 2.1, persamaan 2.3, dan persamaan 2.4

7. Perhitungan jarak *record* 1 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij})$$

$$= 0 + (1-4/6) + (1-3/6) + (1-2/6)$$

$$= 1.5$$

pusat cluster 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1$$

$$= 0 + (1-4/6) + (1-3/6) + 1$$

$$= 1.833$$

pusat cluster 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij})$$

$$= 0 + (1-4/6) + 1 + (1-2/6)$$

$$= 2$$

8. Perhitungan jarak *record* 2 dengan semua pusat *cluster*.

pusat cluster 1 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1$$

$$= 0 + (1-4/6) + (1-3/6) + 1$$

$$= 1.833$$

pusat cluster 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij})$$

$$= 0 + (1-4/6) + (1-3/6) + (1-4/6)$$

$$= 1.166$$

pusat cluster 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + 1$$

$$= 0 + (1-4/6) + 1 + 1$$

$$= 2.333$$

9. Perhitungan jarak *record* 3 dengan semua pusat *cluster*.

pusat cluster 1 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij})$$

$$= 0 + (1-4/6) + 1 + (1-2/6) = 2$$

pusat cluster 2 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + 1$$

$$= 0 + (1-4/6) + 1 + 1 = 2.333$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + (1-4/6) + (1-3/6) + (1-2/6) = \mathbf{1.5} \end{aligned}$$

10. Perhitungan jarak *record* 4 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + 1 \\ &= 0 + (1-4/6) + 1 + 1 \\ &= \mathbf{2.333} \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij}) \\ &= 0 + (1-4/6) + 1 + (1-4/6) \\ &= \mathbf{1.666} \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1 \\ &= 0 + (1-4/6) + (1-3/6) + 1 \\ &= \mathbf{1.833} \end{aligned}$$

11. Perhitungan jarak *record* 6 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + 1 \\ &= 0 + 1 + (1-3/6) + 1 \\ &= \mathbf{2.5} \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + 1 + (1-3/6) + (1-4/6) \\ &= \mathbf{2.166} \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + 1 \\ &= 0 + 1 + 1 + 1 = \mathbf{3} \end{aligned}$$

12. Perhitungan jarak *record* 8 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + 1 \\ &= 0 + 1 + 1 + 1 \\ &= 3 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + (1-w_{ij}) \\ &= 0 + 1 + 1 + (1-4/6) \\ &= 2.333 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + 1 \\ &= 0 + 1 + (1-3/6) + 1 \\ &= 2,5 \end{aligned}$$

Tabel 3.9 Jarak dan *Cluster* (Iterasi-2)

<i>Record</i>	Jarak dengan Pusat <i>Cluster</i> 1	Jarak dengan Pusat <i>Cluster</i> 2	Jarak dengan Pusat <i>Cluster</i> 3	<i>Cluster</i>
1	1.5	1.833	2	1
2	1,833	0.833	2,333	2
3	2	2.333	1.5	3
4	2.333	1.666	1.833	2
6	2.5	2.166	3	2
8	3	2.333	2,5	2

Hasil :

Pusat *Cluster* 1 = {1}

Pusat *Cluster* 1 = {2, 4, 6, 8}

Pusat *Cluster* 1 = {3}

Dari hasil perhitungan jarak minimal *record* 1 dengan pusat *cluster* 1 jadi *record* 1 masuk dalam *cluster* 1. Untuk semua *record* dapat dilihat pada tabel 3.10.

Tabel 3.10 Hasil *Cluster* Semua *Record*

<i>Record</i>	Attribut 1	Attribut 2	Attribut 3	Attribut 4	<i>Class</i>
1	muda	rabun jauh	tidak ada	kurang	1
2	muda	rabun jauh	tidak ada	normal	2
3	muda	rabun jauh	ya	kurang	3
4	muda	rabun jauh	ya	normal	2
6	muda	rabun dekat	tidak ada	normal	2
8	muda	rabun dekat	ya	normal	2

Tabel 3.11 Tabel Pusat *Cluster* yang Baru

	Attribut 1	Attribut 2	Attribut 3	Attribut 4
Pusat <i>Cluster</i> 1	muda	rabun jauh	tidak ada	kurang
Pusat <i>Cluster</i> 2	muda	rabun jauh	ya	normal
Pusat <i>Cluster</i> 3	muda	rabun jauh	ya	kurang

Perhitungan Jarak *Record* dan Pusat *Cluster*

Menggunakan persamaan 2.1, persamaan 2.3, dan persamaan 2.4

13. Perhitungan jarak *record* 1 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned}
 d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\
 &= 0 + (1-4/6) + (1-3/6) + (1-2/6) \\
 &= 1.5
 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned}
 d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + 1 \\
 &= 0 + (1-4/6) + 1 + 1 \\
 &= 2,333
 \end{aligned}$$

pusat *cluster* 3 ->

$$d(x,y) = (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij})$$

$$= 0 + (1-4/6) + 1 + (1-2/6) = 2$$

14. Perhitungan jarak *record* 2 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1 \\ &= 0 + (1-4/6) + (1-3/6) + 1 = 1,833 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij}) \\ &= 0 + (1-4/6) + 1 + (1-4/6) = 1,666 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + 1 \\ &= 0 + (1-4/6) + 1 + 1 = 2,333 \end{aligned}$$

15. Perhitungan jarak *record* 3 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + (1-w_{ij}) \\ &= 0 + (1-4/6) + 1 + (1-2/6) = 2 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1 \\ &= 0 + (1-4/6) + (1-3/6) + 1 = 1,833 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + (1-4/6) + (1-3/6) + (1-2/6) = 1.5 \end{aligned}$$

16. Perhitungan jarak *record* 4 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + 1 + 1 \\ &= 0 + (1-4/6) + 1 + 1 = 2.333 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + (1-4/6) + (1-3/6) + (1-4/6) \\ &= 1.166 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + (1-w_{ij}) + (1-w_{ij}) + 1 \\ &= 0 + (1-4/6) + (1-3/6) + 1 \\ &= 1.833 \end{aligned}$$

17. Perhitungan jarak *record* 6 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + 1 \\ &= 0 + 1 + (1-3/6) + 1 \\ &= 2.5 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + (1-w_{ij}) \\ &= 0 + 1 + 1 + (1-4/6) \\ &= 2.33 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + 1 \\ &= 0 + 1 + 1 + 1 \\ &= 3 \end{aligned}$$

18. Perhitungan jarak *record* 8 dengan semua pusat *cluster*.

pusat *cluster* 1 ->

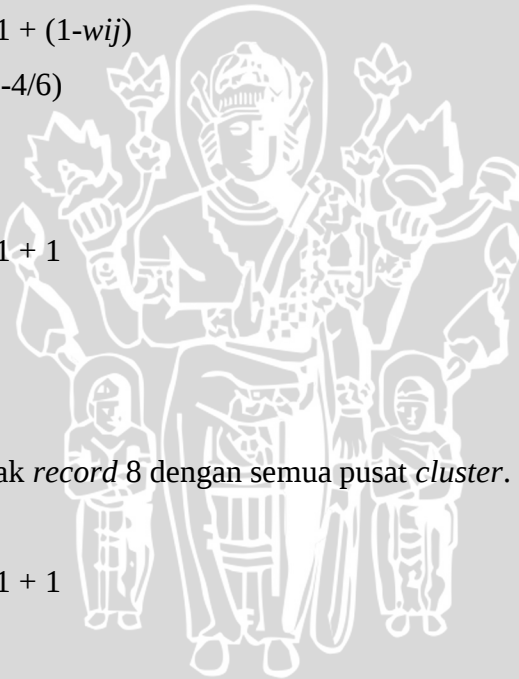
$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + 1 + 1 \\ &= 0 + 1 + 1 + 1 \\ &= 3 \end{aligned}$$

pusat *cluster* 2 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + (1-w_{ij}) \\ &= 0 + 1 + (1-3/6) + (1-4/6) \\ &= 1.833 \end{aligned}$$

pusat *cluster* 3 ->

$$\begin{aligned} d(x,y) &= (1-w_{ij}) + 1 + (1-w_{ij}) + 1 \\ &= 0 + 1 + (1-3/6) + 1 \\ &= 2.5 \end{aligned}$$



Tabel 3.12 Jarak dan *Cluster* (Iterasi-3)

<i>Record</i>	Jarak dengan Pusat <i>Cluster</i> 1	Jarak dengan Pusat <i>Cluster</i> 2	Jarak dengan Pusat <i>Cluster</i> 3	<i>Cluster</i>
1	1.5	2.333	2	1
2	1.833	1.666	2.333	2
3	2	1.833	1.5	3
4	2.333	1.166	1.833	2
6	2.5	2.233	3	2
8	3	1.833	2.5	2

Hasil :

Pusat *Cluster* 1 = {1}

Pusat *Cluster* 1 = {2, 4, 6, 8}

Pusat *Cluster* 1 = {3}

Dari hasil perhitungan jarak minimal *record* 1 dengan pusat *cluster* 1 jadi *record* 1 masuk dalam *cluster* 1. Untuk semua *record* dapat dilihat pada tabel 3.13.

Tabel 3.13 Hasil *Cluster* Semua *Record*

<i>Record</i>	Attribut 1	Attribut 2	Attribut 3	Attribut 4	<i>Class</i>
1	muda	rabun jauh	tidak ada	kurang	1
2	muda	rabun jauh	tidak ada	normal	2
3	muda	rabun jauh	ya	kurang	3
4	muda	rabun jauh	ya	normal	2
6	muda	rabun dekat	tidak ada	normal	2
8	muda	rabun dekat	ya	normal	2

Tabel 3.14 Tabel Pusat *Cluster* yang Baru

	Attribut 1	Attribut 2	Attribut 3	Attribut 4
Pusat <i>Cluster</i> 1	muda	rabun jauh	tidak ada	kurang

Pusat <i>Cluster</i> 2	muda	rabun jauh	ya	normal
Pusat <i>Cluster</i> 3	muda	rabun jauh	ya	kurang

Tahap selanjutnya adalah kembali ke langkah sebelumnya yaitu mengalokasikan semua *record* ke *cluster* terdekat dan seterusnya sampai didapatkan pusat *cluster* yang tidak berubah (*convergence*).

3.6.2 Hitung Purity

Hitung *purity* digunakan untuk menghitung akurasi *cluster*. *Cluster* 100% murni (*pure*) apabila dalam satu *cluster* semua *record* memiliki kategori *class* yang sama. Misalkan ada *data set lense* dan hasil *clustering* seperti pada tabel 3.15 dan 3.16.

Tabel 3.15 Tabel *Data set Lenses*

Attribut 1	Attribut 2	Attribut 3	Attribut 4	Class
muda	rabun jauh	tidak ada	kurang	3
muda	rabun jauh	tidak ada	normal	2
muda	rabun jauh	ya	kurang	3
muda	rabun jauh	ya	normal	1
muda	rabun dekat	tidak ada	normal	2
muda	rabun dekat	ya	normal	1

Tabel 3.16 Hasil *Cluster* Semua *Record*

Attribut 1	Attribut 2	Attribut 3	Attribut 4	Masuk Ke Dalam <i>Cluster</i>
muda	rabun jauh	tidak ada	kurang	1
muda	rabun jauh	tidak ada	normal	2
muda	rabun jauh	ya	kurang	1
muda	rabun jauh	ya	normal	3
muda	rabun dekat	tidak ada	normal	2
muda	rabun dekat	ya	normal	2

Dilihat dari kedua tabel tersebut *record* dengan label *class* 3 yaitu *record* 1 dan 3 masuk dalam *cluster* 1 dan tidak ada *record* dengan label *class* 3 yang berada pada *cluster* selain 1. Sedangkan *record* dengan label *class* 1 masuk dalam *cluster* 2 dan 3. Tabel 3.17 menunjukkan rangkuman hasil *clustering*.

Tabel 3.17 Hasil *Clustering*

<i>Class</i>	<i>Cluster 1</i>	<i>Cluster 2</i>	<i>Cluster 3</i>
1	0	1	1
2	0	2	0
3	2	0	0

Dilihat dari tabel 3.16 pada *cluster* 1 terdapat 2 objek pada *class*3, *cluster* 2 terdapat 1 objek *class*1 dan 2 objek *class*2, dan pada *cluster* 3 terdapat 1 objek di *class* 1. Maka objek pada *cluster* dapat dihitung nilai *purity* hasil *clustering* menggunakan persamaan 2.5 yaitu

Nilai Kemurnian (*purity*) :

$$r = \frac{2+2+1}{6} * 100\%$$

$$r = 83,333\%$$

Error cluster :

$$e = 1 - r$$

$$e = 1 - 0,8333$$

$$= 0,1777$$

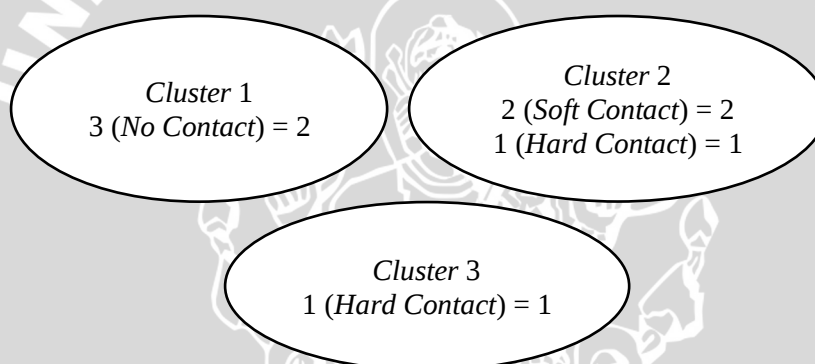
3.6.3 Contoh Perhitungan Metode Evaluasi *f-measure*

Evaluasi *f-measure* dilakukan dengan melihat *cluster* dan anggotanya, kemudian dihitung berdasarkan atribut *class* dalam tiap anggota *cluster*. Keterangan *cluster* dan kelas per *record* akan ditunjukkan Tabel 3.18.

Tabel 3.18 Tabel *cluster* dan *class* per record

Nomor Record	Cluster	Class
1	1	3
2	2	2
3	1	3
4	2	1
6	2	2
8	3	1

Penjelasan lain akan ditunjukkan oleh Gambar 3.11



Gambar 3.11 Gambar *cluster* dan anggota *cluster*

Setelah anggota dikelompokkan kedalam *cluster* masing-masing, hitung *true positive*, *false positive*, dan *false negative*. *True positive* didapat dengan cara mencari anggota kelas terbanyak dari suatu *cluster*. Untuk *cluster 1*, *true positive* (TP) bernilai 3 (*No Contact*), sedangkan untuk *cluster 2*, *true positive* (TP) bernilai 2 (*Soft Contact*), dan untuk *cluster 3*, *true positive* (TP) bernilai 1 (*Hard Contact*). *False positive* (FP) didapat dengan melihat anggota *class* dalam *cluster* selain yang terbanyak. Pada *cluster 2*, *false positive* (FP) bernilai 1 (*Hard Contact*). Untuk nilai *false negative* (FN), lihat *cluster* lain yang nilai terbanyak tidak sama dengan *cluster* yang dimaksud. Untuk nilai *false negative* (FN), *cluster 1* dan *cluster 2* bernilai 0, sedangkan *cluster 3* bernilai 1.

Precision, recall, dan F-Measure dapat dihitung jika true positive, false positive, dan false negative sudah ditemukan.

➤ **Precision:**

- Cluster 1:

$$precision = \frac{TP}{TP + FP} = \frac{2}{2} = 1$$

- Cluster 2:

$$precision = \frac{TP}{TP + FP} = \frac{2}{2} = 1$$

- Cluster 3:

$$precision = \frac{TP}{TP + FP} = \frac{1}{1} = 1$$

➤ **Recall:**

- Cluster 1:

$$recall = \frac{TP}{TP + FN} = \frac{2}{2 + 0} = 1$$

- Cluster 2:

$$recall = \frac{TP}{TP + FN} = \frac{2}{2 + 1} = \frac{2}{3} = 0.6666$$

- Cluster 3:

$$recall = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = 1$$

➤ **F-Measure:**

- Cluster 1:

$$F = \frac{2 \times recall \times precision}{recall + precision} = \frac{2 \times 1 \times 1}{1 + 1} = \frac{2}{2} = 1$$

- Cluster 2:

$$F = \frac{2 \times recall \times precision}{recall + precision} = \frac{2 \times 0.666 \times 1}{0.666 + 1} = \frac{1.332}{1.666} = 0.799$$

- Cluster 3:

$$F = \frac{2 \times recall \times precision}{recall + precision} = \frac{2 \times 1 \times 1}{1 + 1} = \frac{2}{2} = 1$$

Nilai *precision*, *recall*, dan *F-Measure* akan ditunjukkan Tabel 3.19.

Tabel 3.19 *Precision, Recall, dan F-measure*

Claster	Precision	Recall	F-measure
1	1	1	1
2	1	0.666	0.799
3	1	1	1
Jumlah			2.799
Rata-Rata F-Measure			0.933

Pada tabel 3.19, yang akan dijadikan nilai akhir pengujian adalah rata-rata *F-Measure*. Dari hasil di atas dapat di peroleh hasil *F-Measure* adalah **0.933**.

