Indonesian Journal of Data and Science (IJODAS) ISSN: 2715-9930
Vol 3, No 1, March 2022, pp. 35-46

Analisis Performa Metode *Cluster* K-*Means* pada *Dataset Ocular Disease Recognition*

Mulyanul Ilmi Mashur[a,1], Yulita Salim[a,2]

[a] *Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM.05, Makassar dan 90231, Indonesia*
[1] *13020170174@umi.ac.id;* [2] *yulita.salim@umi.ac.id;*

INFORMASI ARTIKEL

Diterima : 19 – 01 – 2022
Direvisi : 22 – 02 – 2022
Diterbitkan : 31 – 03 – 2022

Kata Kunci:
K-*means*
Clustering
Moment invariant
Rand Index
Mutual Information based Scores

ABSTRAK

Penelitian ini bertujuan untuk melakukan perbandingan menggunakan teknik *cluster* yang dapat mengolah data dalam jumlah besar untuk menemukan *cluster* baru pada *dataset Ocular Disease Recognition*. Pengolahan data tersebut digunakan untuk mengelompokkan penyakit pasien melalui fundus mata. Teknik Pengelompokkan menggunakan metode K-*Means* di mana metode ini efisien dan efektif dalam mengolah data dengan jumlah banyak. Pengukuran performa yang digunakan yaitu dengan menggunakan *rand index* dan *mutual information based scores*. Inputan yang digunakan yaitu 7 atribut dari hasil ekstraksi fitur moment invariant dataset citra fundus pasien. Data tersebut merupakan data testing yang digunakan untuk menguji performa pada metode K-*Means*. Berdasarkan hasil pengujian performa pada metode *cluster* k-*means*, untuk pengukuran rand *index* di dapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik, kemudian untuk *mutual information based scores* didapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik. Dari hasil perbandingan k=8 dan k=9 dengan *dataset* versi pertama dengan *dataset* versi kedua.

(cc) BY-NC

I. Pendahuluan

Clustering /*cluster* adalah salah satu metode yang digunakan dalam data mining yang cara kerjanya mencari dan mengelompokkan data yang mempunyai kemiripan karakteristik antara data satu dengan data yang lainnya. Metode *Clustering* yang mempunyai sifat efisien dan cepat yang dapat digunakan salah satunya adalah Metode K-*Means*.[1], [2]

Metode K-*Means* pertama kali diperkenalkan oleh MacQueen JB pada tahun 1976. Metode ini bertujuan untuk membuat *cluster* objek berdasarkan atribut menjadi k partisi. Cara kerja metode ini adalah mula-mula ditentukan *cluster* yang akan dibentuk, pada elemen pertama dalam tiap *cluster* dapat dipilih untuk dijadikan sebagai titik tengah (*centroid*), selanjutnya akan dilakukan pengulangan langkah-langkah hingga tidak ada objek yang dapat dipindahkan lagi.[3], [4]. Metode K-*Means* ini memiliki ketelitian yang cukup tinggi terhadap ukuran objek, sehingga metode ini relatif lebih terukur dan efisien untuk pengolahan objek dalam jumlah besar. Selain itu metode ini tidak terpengaruh oleh urutan objek.[5], [6]

Metode K-*Means* sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan *outlier* dengan sangat cepat.[7] Salah satu tahapan penting dalam menerapkan metode K-*Means* adalah menentukan *centroid*, banyaknya *cluster* dan jarak *centroid*. Dengan membentuk beberapa *cluster* menggunakan K-*Means* dapat juga mengetahui jarak antara *cluster* pusat (*centroid*) pada data yang akan dianalisa. Hasil ini menjadi dasar untuk mengklasifikasi data baru yang kemudian muncul sehingga diketahui kelompoknya.[8]

Kesehatan merupakan hal yang berharga bagi manusia karena siapa saja dapat mengalami gangguan kesehatan, begitu pula pada manusia yang sangat rentang terhadap berbagai macam penyakit tetapi penyebabnya tidak kita sadari.[5] Deteksi mata sejak dini adalah cara yang ekonomis dan efektif untuk mencegah kebutaan yang disebabkan oleh diabetes, glaukoma, katarak, *age-related macular degeneration* (AMD), hipertensi, *myopia patalogis* dan penyakit lainnya. Menurut Organisasi Kesehatan Dunia (WHO) saat ini, setidaknya 2,2 miliar memiliki gangguan penglihatan yang sebenarnya bisa dicegah.[9] Cara mendeteksi

penyakit tersebut dapat dilakukan melalui funduskopi. Funduskopi adalah serangkaian tes yang dilakukan oleh dokter mata untuk memeriksa bagian belakang dan dalam mata (*fundus*).[10]

Beberapa penelitian terutama *Clustering* K-*Means* sering digunakan untuk pengelompokan penyakit diantaranya penelitian yang dilakukan oleh Bastian et al., dengan menerapkan algoritma K-*Means Clustering Analysis* pada penyakit menular manusia. Penelitian tersebut memperoleh *cluster* terbanyak yaitu penyakit diare.[5] Penelitian lain dilakukan oleh Silitonga & Morina dalam penelitiannya Klusterisasi Pola Penyebaran Penyakit Pasien Berdasarkan Usia Pasien dengan menggunakan K-*Means Clustering* dari sejumlah pasien yang ada, presentasi usia pasien paling tinggi adalah pasien dengan usia tua kemudian pasien dengan usia parobaya.[11]

Dataset yang akan digunakan sebagai objek penelitian ini merupakan *dataset multiclass* dimana datanya berupa citra *fundus*. *Dataset multiclass* merupakan dataset unik yang memiliki lebih dari dua label[12]–[16].[17] *Dataset* ini memiliki 8 label, yaitu normal, diabetes, glaukoma, katarak, AMD, hipertensi, *myopia patalogis* dan penyakit lain. Keseluruhan *dataset* ini diambil dari *repository Kaggle*. Saat ini belum ada penelitian mengenai performa metode *Cluster* K-*Means* pada dataset *Ocular Disease Recognition*.

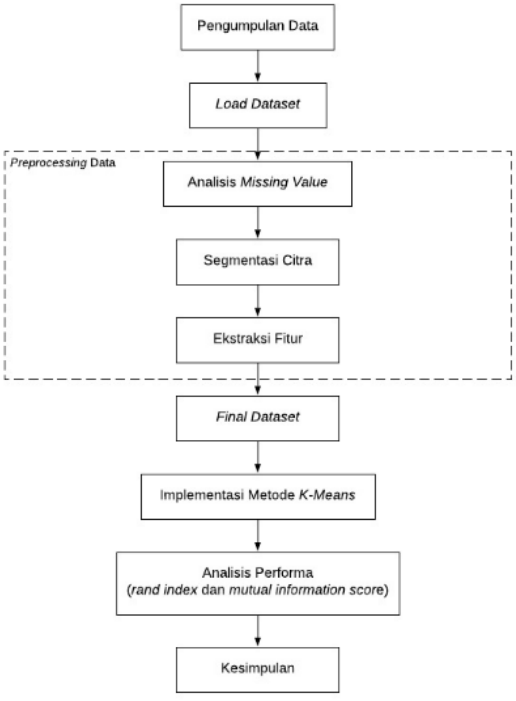
Menurut Aisyah, model segmentasi citra *canny* dan ekstraksi fitur *moment invariant* merupakan pasangan metode yang cukup baik pada tahapan *preprocessing* citra sebagai tahap sebelum proses *clustering*. Berdasarkan uraian di atas, maka penulis mencoba menganalisis performa *rand index* dan *mutual information score* metode K-*Means* pada *dataset Ocular Disease Recognition*.[18]

Rand index adalah ukuran kesamaan antara dua kelompok data.[19] *Rand index* terletak antara nilai 0 dan nilai 1.[20] Nilai 0 menunjukkan bahwa kedua *cluster* data tidak setuju pada setiap pasangan titik dan nilai 1 menunjukkan bahwa *cluster* data sama persis. [19]

Berdasarkan analisis dari beberapa penelitian sebelumnya maka peneliti melakukan penelitian dengan tujuan penelitian ini adalah untuk mengetahui bagaimana hasil evaluasi performa *rand index* dan *mutual information score*.

II. Metode

Pada penelitian ini menggunakan metode seperti yang tertuang pada Gambar 1. Berikut ini:



Gambar 1. Alur Perancangan Proses

1) Pengumpulan Data

Dalam tahapan pengumpulan data untuk penelitian ini digunakan metode pengumpulan studi pustaka dan *dataset* dari *repository Kaggle*. Data tersebut diolah oleh Shanggong Medical Technology Co., Ltd. dari berbagai rumah sakit dan pusat medis di China. Data tersebut di unggah pada tanggal 20 April 2020 dan di perbaharui pada tanggal 24 September 2020.

2) Load Dataset

Dataset yang digunakan pada penelitian ini akan menggunakan dua versi, dimana versi pertama berjumlah 7.821 data, sedangkan versi kedua berjumlah 6.977 data.

3) Preprocessing Data

Tahap ini dilakukan untuk menghilangkan permasalahan-permasalahan yang dapat mengganggu hasil dari proses data. *Preprocessing* data terdiri dari 4 tahapan, yaitu:

a) Analisis Missing Value

Pada tahap ini dilakukan pengecekan hilangnya suatu informasi dari data karena alasan tertentu.

b) Segmentasi Citra

Pada tahap ini menggunakan metode canny sebagai segmentasi citra untuk mendeteksi tepi pada saat proses pengolahan citra. Dalam proses segmentasi citra menggunakan library cv2.

c) Ekstraksi Fitur

Pada tahap ini menggunakan moment invariant sebagai ekstraksi fitur untuk mengkonversi data citra menjadi data numerik, dimana hasilnya berupa 8 fitur yang diberi label H1 sampai H7 dan target. 8 fitur tersebut akan digunakan untuk tahap clustering. Hasil konversi data tersebut kemudian diekspor dalam format .csv (Comma Separated Values). Dalam proses ekstraksi fitur menggunakan library cv2.

4) Implementasi Metode

Pada tahap ini, nilai k yang digunakan yaitu k=8 dan k=9, menggunakan rumus dari metode k-means.

5) Perhitungan Performa

Pada tahap ini, melakukan perhitungan performa rand index dan mutual information score[21]. Performa akan mengitung persamaan antara dua cluster dengan mempertimbangkan semua pasangan sampel dan menghitung pasangan yang ditugaskan dalam cluster yang sama atau berbeda dalam cluster yang diprediksi dan sebenarnya.

6) Pengambilan Kebutusan

Kesimpulan diambil berdasarkan hasil evaluasi performa metode K-*Means* pada *dataset Ocular Disease Recognition*.

III. Hasil dan Pembahasan

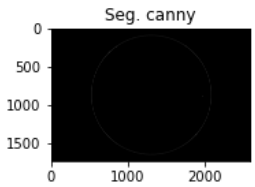
A. Implementasi

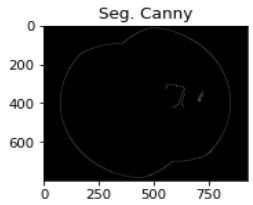
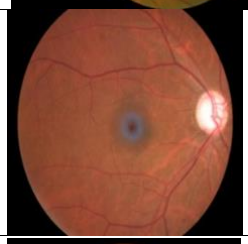
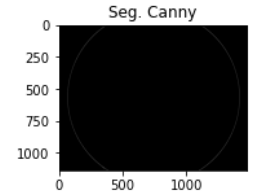
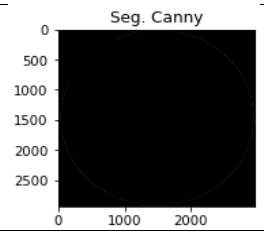
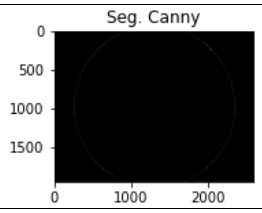
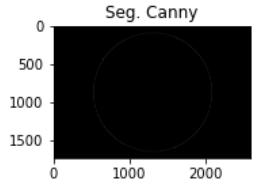
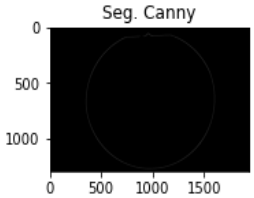
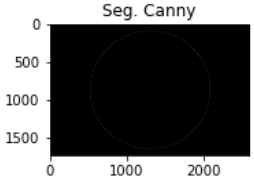
Pada tahap implemetasi menggambarkan beberapa tahap yang terdiri dari implementasi kebutuhan perangkat lunak, perangkat keras, implementasi analisis *missing value*, implementasi segmentasi citra *canny*, implementasi ekstraksi fitur *moment invariant*, implementasi K-*Means Clustering*, implementasi *rand index*, dan implementasi *mutual information scores*.

1) Implementasi Segmentasi Citra Canny

Pada tahap ini dilakukan deteksi tepi pada citra *fundus* dengan menggunakan metode *canny*. Deteksi tepi ini mengoptimalkan pendeteksian tepi pada citra yang ber*noise*. Hasil implementasi segmentasi citra *canny* ditunjukkan pada Tabel 1.

Tabel 1. Prbandingan Nilai *Threshold* Citra Penyakit Malaria

Id	Citra Asli	Hasil Segmentasi Citra Canny
Normal (1)		

Id	Citra Asli	Hasil Segmentasi Citra Canny
Diabetes (2)		
Glaukoma (3)		
Katarak (4)		
AMD (5)		
Hipertensi (6)		
Miopia (7)		
Penyakit Lain (8)		

.

2) Implementasi Ekstraksi Fitur Moment Variant

Implementasi Ekstraksi Fitur pada tahap ini yang dimana terjadinya proses perubahan data citra lalu dikonversi menjadi data numerik, kemudian menghasilkan 7 nilai *array* yang berlabel H1-H7, serta label *target* merupakan *class* dari *dataset multiclass* penyakit. Setelah melakukan konversi maka data numerik tersebut akan disimpan dalam format .csv (*Comma Separated Values*). Hasil dari implementasi ekstraksi fitur *moment invariant* pada dataset versi pertama ditunjukkan pada Tabel 2. dan hasil implementasi ektraksi fitur pada *dataset* versi kedua ditunjukkan pada Table 3.

Tabel 2. Hasil Implementasi Ektraksi Fitur Versi Pertama

Id	H1	H2	H3	H4	H5	H6	H7	Target
0	0.108699196	7.27E-06	4.16E-07	0.000137695	4.75E-10	-3.49E-07	-9.28E-10	1
1	0.219129163	0.000136238	5.82E-06	0.000755555	-4.08E-08	8.71E-06	2.91E-08	1
2	0.454417233	0.000506943	6.49E-05	0.000414883	2.69E-08	9.34E-06	6.25E-08	1
3	0.152998238	0.00017522	2.45E-06	0.00048035	1.63E-08	6.36E-06	-2.58E-09	1
...	...	...	...	...	...	...	...	...
7819	0.281375358	0.000348685	1.11E-06	0.000139186	-5.17E-10	1.83E-06	-1.65E-09	8
7820	0.232927806	0.00428628	0.000172629	0.000398216	9.57E-08	2.09E-05	-4.17E-08	8

Tabel 3. Hasil Implementasi Ekstraksi Fitur Versi Kedua

Id	H1	H2	H3	H4	H5	H6	H7	Target
0	0.108699196	7.27E-06	4.16E-07	0.000137695	4.75E-10	-3.49E-07	-9.28E-10	1
1	0.219129163	0.000136238	5.82E-06	0.000755555	-4.08E-08	8.71E-06	2.91E-08	1
2	0.454417233	0.000506943	6.49E-05	0.000414883	2.69E-08	9.34E-06	6.25E-08	1
3	0.152998238	0.00017522	2.45E-06	0.00048035	1.63E-08	6.36E-06	-2.58E-09	1
...	...	...	...	...	...	...	...	...
6975	0.232927806	0.00428628	0.000172629	0.000398216	9.57E-08	2.09E-05	-4.17E-08	8
6976	0.987623117	0.224968947	0.252268692	0.054783659	0.004161237	0.013587075	-0.004915484	8

3) Implemetasi Metode K-Means

Implementasi metode k-*means* pada tahap ini merupakan contoh proses perhitungan manual mulai dari penetapan data *testing* hingga implementasi metode k-*means*, dimana penulis menggunakan *dataset* versi pertama sebagai contoh. Pada Tabel 4. merupakan 20 *sample* yang akan digunakan untuk perhitungan manual.

Tabel 4. *Sample Dataset* Versi Pertama

H1	H2	H3	H4	H5	H6	H7	Target
0.218237	5.91E-06	6.41E-06	0.000327	1.30E-08	5.11E-07	7.29E-09	1
0.656048	0.105203	0.000529	0.000219	-3.16E-08	-6.83E-05	6.76E-08	1
0.21237	4.00E-06	3.39E-06	0.000242	-1.07E-09	1.73E-07	6.87E-09	1
0.1804	1.11E-05	2.13E-06	0.000488	9.01E-09	1.48E-06	1.29E-08	2
0.23943	0.000176	3.78E-07	0.000258	2.24E-09	9.56E-07	1.21E-09	2
0.3404	0.000485	0.000103	0.000623	-1.10E-07	5.74E-06	1.13E-07	2
0.17961	0.000391	8.37E-06	0.000463	-2.87E-08	8.25E-06	-3.05E-09	3
0.5487	0.000765	1.06E-05	1.44E-05	-1.72E-10	2.80E-07	4.70E-11	3
0.4018	0.000226	3.03E-05	0.000252	-1.74E-08	3.78E-06	1.36E-08	3
0.04372	7.38E-05	6.62E-07	5.98E-07	3.43E-13	-7.15E-10	1.57E-13	4
0.5783	0.001134	0.000477	0.000191	-5.73E-08	-6.42E-06	6.60E-09	4
0.3943	0.000268	1.45E-05	0.001177	-7.64E-08	1.91E-05	1.33E-07	4
0.2806	2.59E-05	9.60E-07	0.000538	-8.63E-10	2.73E-06	-1.22E-08	5
0.40354	7.57E-06	0.000173	0.000357	-8.88E-08	-9.06E-07	-3.09E-09	5
0.12663	0.000195	8.50E-06	0.000334	8.06E-09	4.02E-06	-1.58E-08	6
0.070331	1.35E-05	1.15E-06	4.77E-05	2.26E-10	1.74E-07	2.70E-10	6

0.39827	4.72E-06	7.65E-06	9.20E-06	-5.26E-11	1.84E-08	-5.65E-11	7
0.21888	3.89E-05	4.04E-05	4.96E-05	-1.09E-09	2.37E-08	-1.93E-09	7
0.16846	7.68E-06	5.39E-06	1.71E-05	9.78E-11	3.06E-08	1.33E-10	8
0.31049	1.93E-05	1.84E-05	0.000395	3.22E-08	1.69E-06	9.96E-09	8

4) Perhitungan jarak cluster

Untuk menghitung jarak antara data dengan pusat *cluster* digunakan persamaan 1. Maka di dapatkan nilai matrik sebagai berikut:

Jarak data ke-1 ke pusat cluster

$$C1 = \sqrt{\begin{matrix}(0.218237-0.218237)^2+(5.91\text{E}-06-5.91\text{E}-06)^2\\+(6.41\text{E}-06-6.41\text{E}-06)^2+(0.000327-0.000327)^2\\+(1.30\text{E}-08-1.30\text{E}-08)^2+(5.11\text{E}-07-5.11\text{E}-07)^2\\+(7.29\text{E}-09-7.29\text{E}-09)^2\end{matrix}} = 0$$

$$C2 = \sqrt{\begin{matrix}(0.1804-0.218237)^2+(1.11\text{E}-05-5.91\text{E}-06)^2\\+(2.13\text{E}-06-6.41\text{E}-06)^2+(0.000488-0.000327)^2\\+(9.01\text{E}-09-1.30\text{E}-08)^2+(1.48\text{E}-06-5.11\text{E}-07)^2\\+(1.29\text{E}-08-7.29\text{E}-09)^2\end{matrix}} = 0.03784$$

$$C3 = \sqrt{\begin{matrix}(0.5487-0.218237)^2+(0.000765-5.91\text{E}-06)^2\\+(1.06\text{E}-05-6.41\text{E}-06)^2+(1.44\text{E}-05-0.000327)^2\\+(-1.72\text{E}-10-1.30\text{E}-08)^2+(2.80\text{E}-07-5.11\text{E}-07)^2\\+(4.70\text{E}-11-7.29\text{E}-09)^2\end{matrix}} = 0.33046$$

$$C4 = \sqrt{\begin{matrix}(0.5783-0.218237)^2+(0.001134-5.91\text{E}-06)^2\\+(0.000477-6.41\text{E}-06)^2+(0.000191-0.000327)^2\\+(-5.73\text{E}-08-1.30\text{E}-08)^2+(-6.42\text{E}-06-5.11\text{E}-07)^2\\+(6.60\text{E}-09-7.29\text{E}-09)^2\end{matrix}} = 0.36007$$

$$C5 = \sqrt{\begin{matrix}(0.39827-0.218237)^2+(4.72\text{E}-06-5.91\text{E}-06)^2\\+(7.65\text{E}-06-6.41\text{E}-06)^2+(9.20\text{E}-06-0.000327)^2\\+(-5.26\text{E}-11-1.30\text{E}-08)^2+(1.84\text{E}-08-5.11\text{E}-07)^2\\+(-5.65\text{E}-11-7.29\text{E}-09)^2\end{matrix}} = 0.18003$$

$$C6 = \sqrt{\begin{matrix}(0.31049-0.218237)^2+(1.93\text{E}-05-5.91\text{E}-06)^2\\+(1.84\text{E}-05-6.41\text{E}-06)^2+(0.000395-0.000327)^2\\+(3.22\text{E}-08-1.30\text{E}-08)^2+(1.69\text{E}-06-5.11\text{E}-07)^2\\+(9.96\text{E}-09-7.29\text{E}-09)^2\end{matrix}} = 0.09225$$

$$C7 = \sqrt{\begin{matrix}(0.12663-0.218237)^2+(0.000195-5.91\text{E}-06)^2\\+(8.50\text{E}-06-6.41\text{E}-06)^2+(0.000334-0.000327)^2\\+(8.06E-09-1.30E-08)^2+(4.02\text{E}-06-5.11\text{E}-07)^2\\+(-1.58\text{E}-08-7.29\text{E}-09)^2\end{matrix}} = 0.09161$$

$$C8 = \sqrt{\begin{matrix}(0.2806-0.218237)^2+(2.59\text{E}-05-5.91\text{E}-06)^2\\+(9.60\text{E}-07-6.41\text{E}-06)^2+(0.000538-0.000327)^2\\+(-8.63\text{E}-10-1.30\text{E}-08)^2+(2.73\text{E}-06-5.11\text{E}-07)^2\\+(-1.22\text{E}-08-7.29\text{E}-09)^2\end{matrix}} = 0.06236$$

Proses selanjutnya dilakukan seperti pada langkah diatas, dan dilanjutkan menghitung sampai data ke-2 .....n terhadap pusat *cluster* awal hingga didapatkan matrik jarak.

5) Pengelompokkan Data

Jarak hasil perhitungan pada *point* ke dua akan dilakukan suatu perbandingan dan jarak yang terdekat dipilih antara pusat *cluster* dengan data, jarak tersebut akan menunjukkan bahwa data tersebut memiliki jarak paling dekat berada dalam satu kelompok dengan pusat *cluster*, pembagian data dapat dilihat pada Tabel 5. di bawah ini. Nilai 1 berarti data tersebut berada dalam kelompok.

Tabel 5. Pengelompokkan data

C1	C2	C3	C4	C5	C6	C7	C8
1							
			1				
1							
	1						
1							
					1		
	1						
		1					
				1			
						1	
			1				
				1			
							1
				1			
						1	
						1	
				1			
1							
	1						
					1		

Berdasarkan matrik yang didapatkan pada tabel diatas maka didapatkan pengelompokkan sebagai berikut:

C1: 4 data (0,2,4,17)

C2: 3 data (3,6,18)

C3: 1 data (7)

C4: 2 data (1,10)

C5: 4 data (8,11,13,16)

C6: 2 data (5,19)

C7: 3 data (9,14,15)

C8: 1 data (12)

6) Penentuan pusat cluster baru

Setelah didapatkan member dari setiap *cluster* kemudian *cluster* bari dihitung berdasarkan data member tiap-tiap *cluster* yang sudah didapatkan menggunakan persamaan 3 yang sesuai dengan pusat member *cluster* sebagai berikut:

Proses selanjutnya dilakukan seperti pada langkah diatas hingga. Setelah semua nilai *centroid* baru diperoleh seperti yang terlihat pada Tabel 6.

Tabel 6. Pusat *cluster* baru

Centroid - 1	0.222229	5.61E-05	1.26E-05	0.000219	3.27E-09	4.16E-07	3.36E-09
Centroid - 2	0.176157	0.000136	5.3E-06	0.000323	-6.5E-09	3.25E-06	3.33E-09
Centroid - 3	0.5487	0.000765	1.06E-05	1.44E-05	-1.7E-10	2.8E-07	4.7E-11
Centorid - 4	0.617174	0.053169	0.000503	0.000205	-4.4E-08	-3.7E-05	3.71E-08
Centorid - 5	0.399478	0.000127	5.65E-05	0.000449	-4.6E-08	5.5E-06	3.59E-08
Centroid - 6	0.26143	1.17E-05	1.09E-05	0.000319	1.56E-08	9.32E-07	8.42E-09
Centroid - 7	0.080227	9.4E-05	3.44E-06	0.000127	2.76E-09	1.4E-06	-5.2E-09
Centroid - 8	0.2806	2.59E-05	9.6E-07	0.000538	-8.6E-10	2.73E-06	-1.2E-08

Langkah selanjutnya hitung nilai *Euclidean* dari semua data ke titik *cluster* baru seperti yang dilakukan pada *point* kedua. Pada Tabel 7, Tabel 8, Tabel 9 dan Tabel 10. merupakan hasil pengelompokan iterasi ke-2 sampai iterasi ke-5.

Tabel 7. Hasil pengelompokkan iterasi ke-2

C1	C2	C3	C4	C5	C6	C7	C8
1							
			1				
1							
	1						
1							
				1			
	1						
		1					
				1			
						1	
		1					
				1			
							1
				1			
						1	
						1	
				1			
1							
	1						
						1	

Tabel 8. Hasil pengelompokkan iterasi ke-3

C1	C2	C3	C4	C5	C6	C7	C8
1							
			1				
1							
	1						
1							
				1			
	1						
		1					
				1			
					1		
		1					
				1			
							1
				1			
						1	
						1	
				1			
1							
	1						
							1

Tabel 9. Hasil pengelompokkn iterasi ke-4

C1	C2	C3	C4	C5	C6	C7	C8
1							
			1				
1							
	1						
1							
							1
	1						
		1					
				1			
					1		
		1					
				1			
							1
				1			
						1	
					1		

				1			
1							
	1						
							1

Tabel 10. Hasil pengelompokkn iterasi ke-5

C1	C2	C3	C4	C5	C6	C7	C8
1							
			1				
1							
	1						
1							
							1
	1						
		1					
				1			
					1		
		1					
				1			
							1
				1			
						1	
					1		
				1			
1							
	1						
							1

Pada pengujian iterasi ke-4 dan iterasi ke-5 tidak mengalami perubahan maka proses iterasi selesai dan diperoleh 8 *cluster* dengan 5 iterasi. Iterasi selanjutnya dilakukan dengan cara yang sama hingga tidak ada perubahan data dalam suatu *cluster*.

B. *Pembahasan*

Dataset yang digunakan sebagai objek penelitian ini merupakan *dataset multiclass* penyakit, dimana datanya berupa citra *fundus*. Data tersebut memiliki 8 label, yaitu (1) Normal, (2) *Diabetes*, (3) Glaukoma, (4) Katarak, (5) AMD, (6) Hipertensi, (7) *Miopia*, (8) Penyakit lainnya. *Dataset* ini berasal dari *repository Kaggle*. Data tersebut diolah oleh Shanggong Medical Technology Co., Ltd. dari berbagai rumah sakit dan pusat medis di China. Data tersebut di unggah pada tanggal 20 April 2020 dan di perbaharui pada tanggal 24 September 2020.

Pada *dataset* ini, satu data ada yang memiliki 1 label, 2 label, bahkan lebih dari 2 label. *Dataset* yang digunakan dibagi menjadi 2 versi, dimana versi pertama merupakan *dataset* yang datanya memiliki 2 label atau dipisahkan ke folder masing-masing berdasarkan labelnya sehingga total data pada *dataset* bversi pertama menjadi 7821 data. Versi kedua merupakan *dataset* yang datanya dipisahkan ke fordel masing-masing berdasarkan salah satu labelnya saja, sehingga total data pada *dataset* versi kedua menjadi 6.977 data.

Pembahasan *source code* program pada tahap ini yang terdiri dari *source code* deklarasi *library* yang digunakan, *load image dataset*, visualisasi data, implementasi segmentasi citra menggunakan metode *canny*, ekstraksi fitur menggunakan *moment invariant*, ekspor data kedalam format .csv(*Comma Separeted Values*), *load* csv *dataset*, split atribut dan target, implementasi metode K-*Means Clustering*, dan perhitungan performa berupa *rand index* dan mutual *information score*. *Source Code* program tersebut ditunjukkan pada Gambar 2 sampai Gambar 11.

```
import pandas as pd
import numpy as np
import math
import os

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

import sklearn.metrics as metrics

from skimage import io
from skimage import feature

import matplotlib.pyplot as plt

import cv2
```

Gambar 2. Deklarasi *Library*

Pada Gambar 2 merupakan deklarasi *library* untuk mempermudah proses perhitungan yang akan berfungsi pada tahap selanjutnya.

```
normal_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/1. Normal'
diabetes_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/2. Diabetes'
glaukoma_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/3. Glaucoma'
katarak_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/4. Cataract'
amd_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/5. Age Related Macular Degeneration'
hipertensi_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/6. Hypertension'
miopia_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/7. Pathological Myopia'
lain_path = '../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/8. Other Diseases'

normal = os.listdir(normal_path)
diabetes = os.listdir(diabetes_path)
glaukoma = os.listdir(glaukoma_path)
katarak = os.listdir(katarak_path)
amd = os.listdir(amd_path)
hipertensi = os.listdir(hipertensi_path)
miopia = os.listdir(miopia_path)
lain = os.listdir(lain_path)

print('Done')
```

Gambar 3. *Load Image Dataset*

Pada Gambar 3. merupakan proses *load* data citra yang akan diolah ke dalam variabel.

```
plt.figure(figsize = (8,8))
for i in range(3):
    plt.subplot(1, 3, i+1)
    img = cv2.imread(normal_path + "/" + normal[i])
    plt.imshow(img)
    plt.title('Normal actual')
    plt.tight_layout()
    print(normal[i])

plt.figure(figsize = (8,8))
for i in range(3):
    plt.subplot(1, 3, i+1)
    img = cv2.imread(normal_path + "/" + normal[i])
    edges = cv2.Canny(img,25,255,L2gradient=False)
    plt.imshow(edges,cmap='gray')
    plt.title('Seg. canny')
    plt.tight_layout()
```

Gambar 4. Visuaisasi Data

Pada Gambar 4 merupakan proses visualisasi, yang dimana untuk mengatur ukuran skala tampilan citra dan melakukan proses segmentasi *canny*.

```
x= 0
x = np.array([['h1','h2','h3','h4','h5','h6','h7','target']])

for i in range(len(normal)):
    img = cv2.imread('../input/dataset-multiclass-penyakit/Dataset Multiclass Penyakit/1. Normal' + "/" + normal[i]
    edges = cv2.Canny(img,25,100)
    a = cv2.HuMoments(cv2.moments(edges)).flatten()
    a = np.append(a, 1)
    x = np.vstack((x,a))
```

Gambar 5. *Canny* dan *Moment Invariant*

Pada Gambar 5. merupakan proses untuk membuat *array* H1-H7 dan target, sebagai *header* pada file .csv dan mengkonversi data citra menjadi data numerik menggunakan *moment invariant.*

```
np.savetxt("/kaggle/working/data.csv", x, fmt='%s',delimiter=',' )
print('Done')
```

Gambar 6. *Export to* CSV

Pada Gambar 6. merupakan proses mengubah data variabel x kedalam bentuk .csv (*Comma Separated Values*).

```
dataset = pd.read_csv('/kaggle/working/data.csv')
print (len(dataset))
print (dataset)
```

Gambar 7. *Load* CSV *Dataset*

Pada Gambar 7. merupakan proses *load file* .csv kedalam variabel *dataset* dan mencetak informasi jumlah data yang ada dalam variabel.

```
x = dataset.iloc[:,0:7]
y = dataset.iloc[:,7]
```

Gambar 8. Split atribut dan Target

Pada Gambar 8. merupakan proses split atribut dan target, dimana variabel x merupakan atribut dan variabel y merupakan target.

```
sc_x = StandardScaler()
x = sc_x.fit_transform(x)
```

Gambar 9. *Scaling* Data

Pada Gambar 9. merupakan proses *scaling* data untuk membuat data yang kita miliki berada dintara 0-1.

```
kmeans = KMeans(n_clusters=8, random_state=0).fit(x)
y_pred=kmeans.labels_
```

Gambar 10. Implementasi Metode K-*Means*

Pada Gambar 10 merupakan proses menentukan dan mengkonfigurasi fungsi k-*means*.

```
print(type(y))
print(type(y_pred))
Tlabel = list(y)
Plabel = list(y_pred)
```

```
metrics.adjusted_rand_score(Tlabel,Plabel)
```

```
metrics.adjusted_mutual_info_score(Tlabel,Plabel)
```

Gambar 11. Performa RI dan MI

Pada Gambar 11 merupakan proses performa *rand index* dan *mutual information score*

C. Kesimpulan pengujian

Pengujian performa *Cluster* K-*means* dilakukan pada *dataset Ocular Disease Recognition*. *Dataset* dibagi 2 versi, pada *dataset* versi pertama berjumlah 7.821 data dan *dataset* versi kedua berjumlah 6.977 data. Berdasarkan hasil pengujian performa pada metode *cluster* k-*means*, untuk pengukuran *rand index* di dapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik, kemudian untuk *mutual information based scores* didapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik. Dari hasil perbandingan k=8 dan k=9 dengan *dataset* versi pertama dengan *dataset* versi kedua

IV. Kesimpulan

Berdasarkan hasil penelitian ini maka dapat disimpulkan bahwa algoritma k-*means* dapat digunakan untuk mengelompokkan data penyakit *multiclass* dan multilabel dan metode *cluster* k-*means*, untuk pengukuran *rand index* di dapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik, kemudian untuk *mutual information based scores* didapatkan hasil nilai 1.0 dengan k=8 untuk *cluster* yang identik. Dari hasil perbandingan k=8 dan k=9 dengan *dataset* versi pertama dengan *dataset* versi kedua.

Daftar Pustaka

[1] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.

[2] D. Sartika and J. Jumadi, "Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Clustering Penilaian Kinerja Dosen Menggunakan Algoritma K-Means (Studi Kasus: Universitas Dehasen Bengkulu)," pp. 703–709, 2019.

[3] A. K. Wardhani, "K-Means Algorithm Implementation for Clustering of Patients Disease in Kajen Clinic of Pekalongan," *J. Transform.*, vol. 14, no. 1, p. 30, 2016, doi: 10.26623/transformatika.v14i1.387.

[4] A. Ali, "Klasterisasi Data Rekam Medis Pasien Menggunakan Metode K-Means Clustering di Rumah Sakit Anwar Medika Balong Bendo Sidoarjo," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 1, pp. 186–195, 2019, doi: 10.30812/matrik.v19i1.529.

[5] A. Bastian, H. Sujadi, and G. Febrianto, "Penerapan Algoritma K-Means Clustering Analysis Pada Penyakit Menular Manusia (Studi Kasus Kabupaten Majalengka)," no. 1, pp. 26–32.

[6] A. Asroni and R. Adrian, "Penerapan Metode K-Means Untuk Clustering Mahasiswa Berdasarkan Nilai Akademik Dengan Weka Interface Studi Kasus Pada Jurusan Teknik Informatika UMM Magelang," *Semesta Tek.*, vol. 18, no. 1, pp. 76–82, 2016, doi: 10.18196/st.v18i1.708.

[7] W. Safira Azis and dan Dedy Atmajaya, "Pengelompokan Minat Baca Mahasiswa Menggunakan Metode K-Means," *Ilk. J. Ilm.*, vol. 8, no. 2, pp. 89–94, 2016.

[8] B. S. Prayoga and N. N. Fatriani, "Penerapan Metode K-Means Cluster Analysis Untuk," pp. 73–78, 2014.

[9] WHO, "World report on vision," 2019.

[10] F. W. Aprilia and D. Kuswanto, "Desain Alat Periksa Mata Fundus Portable Berbasis Rapid Prototyping untuk Mendukung Diagnosa Secara Telemedicine di Indonesia," *J. Sains dan Seni ITS*, vol. 7, no. 1, 2018, doi: 10.12962/j23373520.v7i1.29933.

[11] P. M. Silitonga Irene Sri, "Klusterisasi Pola Penyebaran Penyakit Pasien Berdasarkan Usia Pasien Dengan Menggunakan K-Means Clustering," *J. TIMES*, vol. VI, no. Vol 6, No 2 (2017), pp. 22–25, 2017.

[12] H. Azis, F. T. Admojo, and E. Susanti, "Analisis Perbandingan Performa Metode Klasifikasi pada Dataset Multiclass Citra Busur Panah," *Techno.Com*, vol. 19, no. 3, 2020.

[13] M. M. Baharuddin, T. Hasanuddin, and H. Azis, "Analisis Performa Metode K-Nearest Neighbor untuk Identifikasi Jenis Kaca," *Ilk. J. Ilm.*, vol. 11, no. 28, pp. 269–274, 2019.

[14] A. Fitria and H. Azis, "Analisis Kinerja Sistem Klasifikasi Skripsi menggunakan Metode Naïve Bayes Classifier," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 102–106, 2018.

[15] N. Fadhillah, H. Azis, and D. Lantara, "Validasi Pencarian Kata Kunci Menggunakan Algoritma Levenshtein Distance Berdasarkan Metode Approximate String Matching," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 3–7, 2018.

[16] A. A. Karim, H. Azis, and Y. Salim, "Kinerja Metode C4.5 dalam Penyaluran Bantuan Dana Bencana 1," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 84–87, 2018.

[17] H. Azis, F. Tangguh Admojo, and E. Susanti, "Analisis Perbandingan Performa Metode Klasifikasi pada Dataset Multiclass Citra Busur Panah," *Techno.Com*, vol. 19, no. 3, pp. 286–294, 2020.

[18] Aisyah, "Analisis Penerapan Metode K-Nearest Neighbor (K- NN) Pada Dataset Citra Penyakit Malaria," Universitas Muslim Indonesia, 2020.

[19] B. Sathya and R. Manavalan, "Image Segmentation by Clustering Methods: Performance Analysis," *Int. J. Comput. Appl.*, vol. 29, no. 11, pp. 27–32, 2011, doi: 10.5120/3688-5127.

[20] K. Y. Yeung and W. Ruzzo, "Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper 'An empirical study on Principal Component Analysis for clustering gene expression data' (to appear in Bioinformatics)," *Science (80-.).*, vol. 17, 2001.

[21] Wahyu ngestisari, "The Perbandingan Metode ARIMA dan Jaringan Syaraf Tiruan untuk Peramalan Harga Beras," *Indones. J. Data Sci.*, vol. 1, no. 3, pp. 96–107, 2020, doi: 10.33096/ijodas.v1i3.18.