

Implementasi Metode K-Means untuk Klustering pada Hasil Tanaman Padi di Aceh

Abdul Sidik*¹, Haekal Ilmandry², Hendrik Asta Manggala³, Reza Aqib Setyoanggoro⁴

¹⁻⁴Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Indonesia
abdulsidik134@gmail.com¹, haekalilmandry10@gmail.com², astahendrik@gmail.com³,
rezaaqib31@gmail.com⁴

Alamat Kampus: Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten 15310

Korespondensi penulis : abdulsidik134@gmail.com*

Abstract. Rice growth is a crucial indicator in determining the success of the agricultural sector, especially in agrarian regions like Aceh. This study aims to cluster rice growth patterns using the K-Means method, a widely recognized and efficient clustering technique. The analyzed data include land area, harvest yield, rice varieties, and environmental factors such as rainfall, temperature, and soil fertility. By applying the K-Means method, the data were grouped into several clusters representing the levels of rice growth across different regions. The results revealed significant differences in rice productivity among clusters, influenced by environmental factors and agricultural management. These findings provide strategic insights for decision-making, particularly in resource optimization and policy planning, to sustainably enhance rice growth outcomes in Aceh.

Keywords: K-Means, rice growth, clustering, Aceh, agricultural productivity.

Abstrak. Pertumbuhan padi merupakan salah satu indikator penting dalam menentukan keberhasilan sektor pertanian, terutama di wilayah agraris seperti Aceh. Penelitian ini bertujuan untuk mengelompokkan pola pertumbuhan padi menggunakan metode K-Means, yang dikenal sebagai teknik klusterisasi yang efisien. Data yang dianalisis mencakup luas lahan tanam, hasil panen, varietas padi, serta kondisi lingkungan seperti curah hujan, suhu, dan kesuburan tanah. Melalui implementasi metode K-Means, data dibagi menjadi beberapa kluster yang menggambarkan tingkat pertumbuhan padi di berbagai wilayah. Hasil penelitian menunjukkan adanya perbedaan signifikan dalam produktivitas padi antar kluster, yang dipengaruhi oleh faktor lingkungan dan manajemen agrikultur. Temuan ini memberikan wawasan strategis bagi pengambilan keputusan, khususnya dalam optimasi sumber daya dan perencanaan kebijakan untuk meningkatkan hasil pertumbuhan padi di Aceh secara berkelanjutan.

Kata kunci: K-Means, pertumbuhan padi, klusterisasi, Aceh, produktivitas pertanian.

1. LATAR BELAKANG

Produksi padi merupakan aspek penting dalam ketahanan pangan nasional. Aceh, sebagai salah satu wilayah penghasil padi utama di Indonesia, memiliki karakteristik produksi yang bervariasi di setiap wilayahnya. Perbedaan dalam luas lahan, tingkat produktivitas, dan volume produksi antar wilayah membutuhkan pendekatan berbasis data untuk mengelompokkan wilayah dengan karakteristik yang serupa.

2. KAJIAN TEORITIS

Metode K-Means adalah salah satu algoritma *clustering* non-hierarki yang digunakan untuk mengelompokkan data berdasarkan kesamaan atau kedekatan antar data. Algoritma ini membagi data ke dalam kkk kelompok (cluster) dengan cara meminimalkan jumlah kuadrat jarak antara titik-titik dalam cluster dengan centroid (pusat cluster).

3. METODE PENELITIAN

1. Pengelompokan Data (Clustering)

Clustering adalah metode pengelompokan data berdasarkan kesamaan karakteristik tertentu. Salah satu metode yang paling banyak digunakan adalah K-Means clustering.

2. Data dan Sumber Data

Data produksi padi diperoleh dari dataset yang tersedia di platform Kaggle. Variabel yang digunakan adalah: pada Tabel 1 dan Gambar 1.

Tabel 1. Variabel Data

Nama	Satuan
Luas lahan	(ha)
Produktivitas	(ton/ha)
Produksi	(ton)

Tabel 2. Variabel Data Produksi data pertahun 1993 - 2020

Provinsi	Tahun	Produk	Luas Panen	Curah hujan	Kelembapan	Suhu rata-rata
Aceh	1993	1329536	323589	1627	82	26.06
Aceh	1994	1299699	329041	1521	82.12	26.92
Aceh	1995	1382905	339253	1476	82.72	26.27
Aceh	1996	1419128	348223	1557	83	26.08
Aceh	1997	1368074	337561	1339	82.46	26.31
Aceh	1998	1404580	365892	1465	82.6	26.84
Aceh	1999	1478712	359817	1778	82.79	26.14
Aceh	2000	1486909	336765	1974.7	90.6	27.1
Aceh	2001	1547499	295212	1688.7	69.48	28.9
Aceh	2002	1314165	315131	1296.8	68.75	29.2
Aceh	2003	1246614	367636	1507.2	70.66	29.4
Aceh	2004	1350748	370966	1097	80.84	29.4
Aceh	2005	1411650	337893	710.5	79.5	26.8
Aceh	2006	1552078	320789	506.5	80.8	26.73
Aceh	2007	1556858	360717	1414	81.5	26.38
Aceh	2008	1402287	329109	1270.4	78.5	27
Aceh	2009	1533369	359375	1577	78.7	26.9
Aceh	2010	1788738	352281	1986	81.4	27.1
Aceh	2011	1772962	380686	1268	79.4	27.1
Aceh	2012	1582393	387803	1098	79.6	26.9
Aceh	2013	2331046	419183	1623.6	80.7	27
Aceh	2014	1820062	376137	2264.4	78.3	27.1
Aceh	2015	1956940	461060	1575	80	27.1
Aceh	2016	2180754	293067	1096	83.32	27.12
Aceh	2017	2478922	294483	1905.9	85.57	26.51
Aceh	2018	1751996.9	329515.78	1427.8	83.98	26.48
Aceh	2019	1714437.6	310012.46	1931.4	83.9	26.65
Aceh	2020	1861567.1	317869.41	1619.2	80.82	25.41

4. HASIL DAN PEMBAHASAN

Metode penelitian yang digunakan dalam pengelompokan hasil produksi padi ini adalah analisis klastering dengan pendekatan K-Means Clustering. Penelitian ini bertujuan untuk mengidentifikasi pola produksi padi di suatu daerah berdasarkan indikator-indikator kunci seperti produksi padi, luas panen, curah hujan, kelembapan udara, dan suhu rata-rata. Data produksi padi diperoleh dari dataset yang tersedia di platform Kaggle. Variabel utama yang digunakan dalam penelitian ini meliputi luas lahan (ha), produktivitas (ton/ha), dan volume

produksi (ton). Tahap pertama dalam metode ini adalah pengumpulan data, di mana data dari sumber tersebut dikumpulkan dan disusun dalam bentuk tabel. Sumber data ini menyediakan gambaran historis produksi padi dari tahun ke tahun, sehingga memudahkan analisis untuk menemukan pola dan tren.

Tahap berikutnya adalah preprocessing data yang terdiri dari beberapa langkah penting. Langkah pertama dalam preprocessing adalah pembersihan data, yaitu menghapus missing values atau nilai yang hilang serta mengidentifikasi dan menghapus outliers atau data yang tidak wajar agar tidak memengaruhi hasil analisis klaster. Data yang bersih akan memberikan hasil klustering yang lebih akurat dan dapat diandalkan. Setelah pembersihan data, dilakukan normalisasi data untuk memastikan semua variabel memiliki skala yang setara. Normalisasi ini penting karena variabel seperti luas lahan, produktivitas, dan volume produksi memiliki satuan yang berbeda, dan skala yang tidak setara dapat memengaruhi proses pembentukan klaster. Dengan normalisasi, semua variabel akan berada dalam rentang nilai yang seragam, sehingga meminimalkan bias dalam proses klustering.

Selanjutnya, proses K-Means Clustering dilakukan untuk mengelompokkan data ke dalam beberapa klaster berdasarkan kemiripan karakteristik. Variabel utama seperti luas lahan, produktivitas, dan volume produksi digunakan sebagai acuan untuk membentuk klaster. Sebelum menjalankan algoritma K-Means, jumlah klaster (k) ditentukan terlebih dahulu, misalnya dengan menggunakan Elbow Method untuk menemukan jumlah klaster yang optimal. Algoritma K-Means akan mengelompokkan data ke dalam klaster-klaster yang memiliki centroid atau pusat klaster dengan jarak terdekat. Misalnya, data dapat dibagi ke dalam tiga klaster utama, yakni klaster dengan hasil produksi padi "tinggi," "sedang," dan "rendah."

Hasil klustering akan divisualisasikan menggunakan scatter plot, diagram batang, atau bentuk visualisasi data lainnya untuk memberikan gambaran yang jelas terkait distribusi data dalam setiap klaster. Visualisasi ini membantu dalam menginterpretasi pola yang terbentuk, seperti kelompok dengan luas lahan yang kecil tetapi produktivitas tinggi, atau kelompok dengan luas lahan besar tetapi produktivitas rendah. Dari analisis ini, kita dapat mengidentifikasi klaster yang memiliki kinerja optimal dan klaster yang memerlukan perhatian lebih lanjut.

Tahap terakhir dalam penelitian ini adalah interpretasi hasil dan rekomendasi solusi. Setiap klaster dianalisis untuk mengenali karakteristik dan faktor-faktor yang memengaruhi kinerja produksinya. Klaster dengan hasil produksi rendah, misalnya, mungkin disebabkan oleh luas lahan yang sempit, produktivitas yang rendah, atau faktor lingkungan seperti curah hujan yang tidak memadai. Rekomendasi solusi untuk klaster ini dapat berupa penyediaan

teknologi pertanian yang lebih modern, penyuluhan terkait praktik pertanian yang baik, serta distribusi pupuk dan bibit unggul. Di sisi lain, kluster dengan hasil produksi tinggi dapat dijadikan sebagai benchmark atau contoh praktik terbaik yang bisa direplikasi di kluster lainnya.

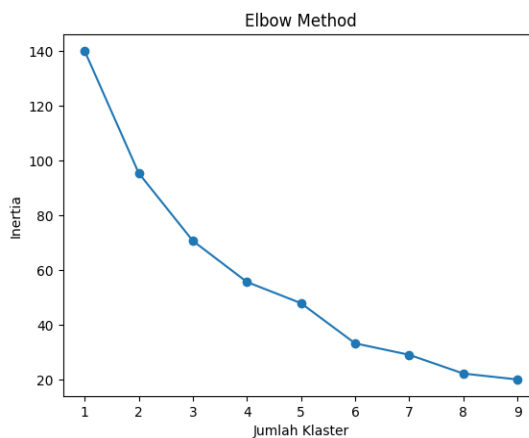
Metode Yang Digunakan

Menentukan data dengan algoritma K-Means dengan langkah berikut

1. Menentukan jumlah kluster optimal menggunakan metode Elbow dan Silhouette.
2. Menginisialisasi centroid awal secara acak.
3. Menghitung jarak Euclidean antara data dan centroid.
4. Mengelompokkan data ke dalam kluster berdasarkan jarak terdekat.
5. Memperbarui posisi centroid dan mengulangi langkah 3-4 hingga kluster stabil.

Menentukan jumlah kluster optimal menggunakan metode Elbow dan Silhouette

Metode Elbow dan Silhouette Score digunakan untuk menentukan jumlah kluster optimal. Penentuan jumlah kluster optimal dilakukan menggunakan Elbow Method dan Silhouette Score. Grafik Elbow menunjukkan titik optimal pada $K = 3$, di mana penurunan nilai inerti mulai melambat.



Gambar 1. Grafik Elbow Method

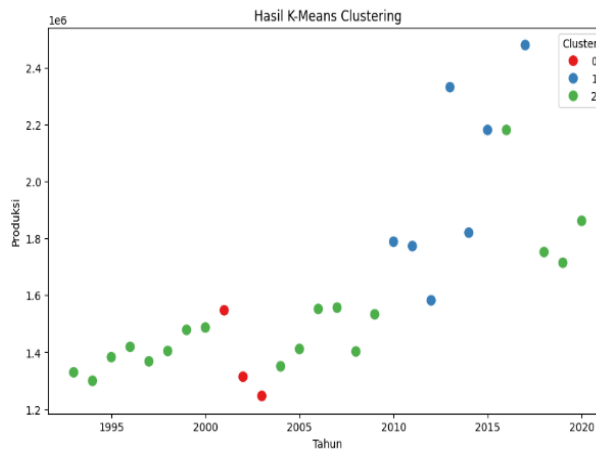
a) Hasil klasterisasi

Pengelompokan wilayah produksi padi menghasilkan tiga kluster dengan karakteristik sebagai berikut:

1. Kluster 1: Wilayah dengan produktivitas tinggi, luas lahan kecil, tetapi volume produksi besar.
2. Kluster 2: Wilayah dengan luas lahan besar tetapi produktivitas rendah.
3. Kluster 3: Wilayah dengan luas lahan sedang dan produktivitas sedang.

b) Visualisasi Hasil Klasterisasi

1. Berikut ini penjelasan untuk Sub-sub-sub judul. Sumbu X yaitu tahun produksi.
2. Sumbu Y yaitu volume produksi.



Gambar 2. Grafik Hasil K-Means Klaster

c) Analisis Tren Produksi

Metode Elbow dan Silhouette Score digunakan untuk menentukan jumlah klaster optimal. Penentuan jumlah klaster optimal dilakukan menggunakan Elbow Method dan Silhouette Score. Grafik Elbow menunjukkan titik optimal pada $K = 3$, di mana penurunan nilai inertia mulai melambat.

Menginisialisasi Centroid Awal

a) Menentukan Jumlah Klaster

1. Sebelumnya menentukan jumlah klaster k berdasarkan metode seperti Elbow Method atau aturan lainnya.
2. Pilih k titik dari dataset secara acak.
3. Setiap titik yang dipilih menjadi centroid awal untuk masing-masing klaster.
4. Centroid awal akan menjadi centroid pusat dari klaster-klaster awal.
5. Titik-titik data akan dikelompokkan ke centroid terdekat berdasarkan jarak Euclidean.

b) Pemilihan Centroid

1. K-Means : Centroid awal dipilih secara lebih strategis, bukan murni acak.
2. Multiple Runs : Menjalankan K-Means beberapa kali dengan inisialisasi berbeda dan memilih hasil terbaik.

Jika dengan google collab

```
import pandas as pd
import numpy as np

//Membaca data dari file CSV
file_path = 'data_aceh.csv' //Ganti dengan path file CSV Anda
df = pd.read_csv(file_path)

//Menentukan fitur yang akan digunakan untuk clustering
//Misalnya kita gunakan kolom 'Produksi' dan 'Luas Panen'
data = df[['Produksi', 'Luas Panen']] //Perhatikan nama kolom sesuai file CSV

//Menentukan jumlah kluster (k)
k = 3 //Jumlah kluster optimal dari metode Elbow

//Menginisialisasi centroid secara acak
np.random.seed(42) //Untuk hasil yang konsisten
centroid_indices = np.random.choice(data.index, size=k, replace=False)
centroids = data.loc[centroid_indices]

//Menampilkan centroid awal
print("Centroid Awal:")
print(centroids)
```

Output:

Centroid Awal:

```
   Produksi  Luas Panen
9  1314165.0  315131.00
25 1751996.9  329515.78
8  1547499.0  295212.00
```

Tabel 3. Data Cnrtroid Awal

Index	Produksi	Luas Panen
9	1314165.0	315131.00
25	1751996.9	329515.78
8	1547499.0	295212.00

Menghitung jarak Euclidean antara data dan centroid

Berikut adalah implementasi kode Python untuk menghitung jarak Euclidean antara data dan centroid:

```
import numpy as np

//Membaca data dari file CSV
file_path = 'data_aceh.csv' //Ganti dengan path file Anda
df = pd.read_csv(file_path)

//Menentukan fitur yang akan digunakan untuk clustering
data = df[['Produksi', 'Luas Panen']]

//Centroid awal dari hasil sebelumnya
centroids = np.array([
    [1314165.0, 315131.00], //Centroid 1
    [1751996.9, 329515.78], //Centroid 2
    [1547499.0, 295212.00] //Centroid 3
])

//Fungsi untuk menghitung jarak Euclidean
def euclidean_distance(point, centroid):
    return np.sqrt(np.sum((point - centroid) ** 2))

//Menghitung jarak Euclidean dari setiap titik data ke centroid
distances = []
for i, row in data.iterrows():
    point = row.values
    dist_to_centroids = [euclidean_distance(point, centroid) for centroid in centroids]
    distances.append(dist_to_centroids)

//Menyimpan hasil jarak dalam DataFrame
distance_df = pd.DataFrame(distances, columns=['Centroid_1', 'Centroid_2', 'Centroid_3'])
data_with_distance = pd.concat([data, distance_df], axis=1)
```

```
//Menampilkan jarak Euclidean  
print("Jarak Euclidean antara setiap titik data dan centroid:")  
print(data_with_distance)
```

Output:

Jarak Euclidean antara setiap titik data dan centroid:

	Produksi	Luas Panen	Centroid_1	Centroid_2	Centroid_3
0	1329536.0	323589.00	1.754438e+04	422502.471886	219802.464722
1	1299699.0	329041.00	2.006871e+04	452298.149190	250098.463092
2	1382905.0	339253.00	7.284956e+04	369220.319185	170384.255484
3	1419128.0	348223.00	1.100559e+05	333394.158118	138885.851554
4	1368074.0	337561.00	5.838908e+04	384007.185752	184355.006512
5	1404580.0	358592.00	1.003181e+05	348631.508867	156342.140707
6	1478712.0	359817.00	1.705068e+05	274959.634313	94368.731018
7	1486909.0	396765.00	1.910618e+05	273485.012966	118254.640116
8	1547499.0	295212.00	2.341827e+05	207355.107067	0.000000
9	1314165.0	315131.00	0.000000e+00	438068.139167	234182.668268
10	1246614.0	367636.00	8.555649e+04	506818.534374	309478.624465
11	1350748.0	370896.00	6.669371e+04	403376.997805	210805.654234
12	1411563.0	337893.00	1.000224e+05	340536.955534	142478.994441
13	1552078.0	320789.00	2.379803e+05	200109.278312	25983.651976
14	1556858.0	360717.00	2.469372e+05	197617.576199	66170.204065
15	1402827.0	329109.00	8.975709e+04	349170.136948	148590.020503
16	1533369.0	359375.00	2.236245e+05	220657.498575	65700.437358
17	1788325.0	352281.00	4.756131e+05	42871.740007	247495.521246
18	1772962.0	380686.00	4.634567e+05	55298.524690	241121.067194
19	1582393.0	387803.00	2.778983e+05	179340.131902	98947.887885
20	2331046.0	419183.00	1.022191e+06	585950.570060	793293.583770
21	1820062.0	376137.00	5.095621e+05	82500.884797	284322.782404
22	2180734.0	461060.00	8.787702e+05	448463.357179	654593.098290
23	2180754.0	293067.00	8.668698e+05	430303.572335	633258.632827
24	2478922.0	294483.00	1.164940e+06	727768.779685	931423.285284
25	1751996.9	329515.78	4.380681e+05	0.000000	207355.107067

26	1714437.6	310102.46	4.003042e+05	42279.758868	167601.378183
27	1861567.1	317869.41	5.474089e+05	110187.416079	314884.311558

Tabel 4. Data Jarak Euclidean Antara Data dan Centroid

	A	B	C	D	E
1	Produksi	Luas Panen	Centroid_1	Centroid_2	Centroid_3
2	1329536	323589	17544,38	422502,47	219802,46
3	1299699	329041	20068,71	452298,15	250098,46
4	1382905	339253	72849,56	369220,32	170384,26
5	1419128	348223	110055,9	333394,16	138885,85
6	1368074	337561	58389,08	384007,19	184355,01
7	1404580	358592	100318,1	348631,51	156342,14
8	1478712	359817	170506,8	274959,63	94368,73
9	1486909	396765	191061,8	273485,01	118254,64
10	1547499	295212	234182,7	207355,11	0
11	1314165	315131	0	438068,14	234182,67
12	1246614	367636	85556,49	506818,53	309478,62
13	1350748	370896	66693,71	403377	210805,65
14	1411563	337893	100022,39	340536,96	142478,99
15	1552078	320789	237980,3	200109,28	25983,65
16	1556858	360717	246937,18	197617,58	66170,2
17	1402827	329109	89757,09	349170,14	148590,02
18	1533369	359375	223624,53	220657,5	65700,44
19	1788325	352281	475613,13	42871,74	247495,52
20	1772962	380686	463456,7	55298,52	241121,07
21	1582393	387803	277898,3	179340,13	98947,89
22	2331046	419183	1022191	585950,57	793293,58
23	1820062	376137	509562,1	82500,88	284322,78
24	2180734	461060	878770,18	448463,36	654593,1
25	2180754	293067	866869,76	430303,57	633258,63
26	2478922	294483	1164940	727768,78	931423,29
27	1751996,9	329515,78	438068,14	0	207355,11
28	1714437,6	310102,46	400304,2	42279,76	167601,38
29	1861567,1	317869,41	547408,87	110187,42	314884,31

Mengelompokkan data ke dalam kluster berdasarkan jarak terdekat

Untuk mengelompokkan data ke dalam kluster berdasarkan jarak terdekat ke centroid, kita dapat menentukan centroid dengan nilai jarak terkecil untuk setiap baris data. Langkah ini melibatkan yaitu, mencari centroid dengan jarak minimum pada setiap baris dan menentukan kluster berdasarkan indeks centroid tersebut.

```
//Menentukan kolom jarak ke centroid
distance_columns = ['Centroid_1', 'Centroid_2', 'Centroid_3']

//Menentukan kluster berdasarkan jarak terdekat (centroid dengan jarak minimum)
data_with_distance['Cluster'] = data_with_distance[distance_columns].idxmin(axis=1)

//Mengganti nama kluster agar lebih mudah dibaca (opsional)
data_with_distance['Cluster'] = data_with_distance['Cluster'].replace({
    'Centroid_1': 'Cluster 1',
    'Centroid_2': 'Cluster 2',
    'Centroid_3': 'Cluster 3'
})
```

```
//Menampilkan data yang sudah dikelompokkan  
print("Data dengan klaster berdasarkan jarak terdekat:")  
print(data_with_distance[['Produksi', 'Luas Panen', 'Cluster']])
```

Output:

Data dengan klaster berdasarkan jarak terdekat:

	Produksi	Luas Panen	Cluster
0	1329536.0	323589.00	Cluster 1
1	1299699.0	329041.00	Cluster 1
2	1382905.0	339253.00	Cluster 1
3	1419128.0	348223.00	Cluster 1
4	1368074.0	337561.00	Cluster 1
5	1404580.0	358592.00	Cluster 1
6	1478712.0	359817.00	Cluster 3
7	1486909.0	396765.00	Cluster 3
8	1547499.0	295212.00	Cluster 3
9	1314165.0	315131.00	Cluster 1
10	1246614.0	367636.00	Cluster 1
11	1350748.0	370896.00	Cluster 1
12	1411563.0	337893.00	Cluster 1
13	1552078.0	320789.00	Cluster 3
14	1556858.0	360717.00	Cluster 3
15	1402827.0	329109.00	Cluster 1
16	1533369.0	359375.00	Cluster 3
17	1788325.0	352281.00	Cluster 2
18	1772962.0	380686.00	Cluster 2
19	1582393.0	387803.00	Cluster 3
20	2331046.0	419183.00	Cluster 2
21	1820062.0	376137.00	Cluster 2
22	2180734.0	461060.00	Cluster 2
23	2180754.0	293067.00	Cluster 2
24	2478922.0	294483.00	Cluster 2
25	1751996.9	329515.78	Cluster 2

26	1714437.6	310102.46	Cluster 2
27	1861567.1	317869.41	Cluster 2

Tabel 5. Kelompok Data Sesuai Jarak Terdekat

1	Produksi	Luas Panen	Cluster
2	1329536	323589	Cluster 1
3	1299699	329041	Cluster 1
4	1382905	339253	Cluster 1
5	1419128	348223	Cluster 1
6	1368074	337561	Cluster 1
7	1404580	358592	Cluster 1
8	1478712	359817	Cluster 3
9	1486909	396765	Cluster 3
10	1547499	295212	Cluster 3
11	1314165	315131	Cluster 1
12	1246614	367636	Cluster 1
13	1350748	370896	Cluster 1
14	1411563	337893	Cluster 1
15	1552078	320789	Cluster 3
16	1556858	360717	Cluster 3
17	1402827	329109	Cluster 1
18	1533369	359375	Cluster 3
19	1788325	352281	Cluster 2
20	1772962	380686	Cluster 2
21	1582393	387803	Cluster 3
22	2331046	419183	Cluster 2
23	1820062	376137	Cluster 2
24	2180734	461060	Cluster 2
25	2180754	293067	Cluster 2
26	2478922	294483	Cluster 2
27	1751996,9	329515,78	Cluster 2
28	1714437,6	310102,46	Cluster 2
29	1861567,1	317869,41	Cluster 2

Memperbarui posisi centroid dan mengulangi langkah 3-4 hingga kluster stabil

Menghitung centroid baru: Rata-rata koordinat (Produksi, Luas Panen) dari semua titik data dalam setiap kluster, Mengulangi perhitungan jarak Euclidean: menghitung jarak dari setiap titik ke centroid baru, mengelompokkan ulang: Menentukan kluster baru berdasarkan jarak minimum dan memeriksa stabilitas: Jika kluster tidak berubah, iterasi dihentikan.

```
//Inisialisasi variabel
max_iter = 100 //Maksimal jumlah iterasi
tolerance = 1e-4 //Toleransi perbedaan untuk menghentikan iterasi

//Proses iterasi untuk memperbarui centroid dan mengelompokkan data
for i in range(max_iter):
    print(f"Iterasi ke- {i+1}")

    //Menghitung centroid baru berdasarkan rata-rata titik data di setiap kluster
    centroids = data_with_distance.groupby('Cluster')[['Produksi', 'Luas Panen']].mean()
    print("Centroid Baru:")
    print(centroids)
```

```
//Menentukan jarak Euclidean ke centroid baru
for idx, centroid in centroids.iterrows():
    col_name = f"Centroid_{idx[-1]}" //Cluster 1 -> Centroid_1
    data_with_distance[col_name] = np.sqrt(
        (data_with_distance['Produksi'] - centroid['Produksi'])**2 +
        (data_with_distance['Luas Panen'] - centroid['Luas Panen'])**2
    )

//Menentukan klaster baru berdasarkan jarak minimum
previous_clusters = data_with_distance['Cluster'].copy()
distance_columns = [f"Centroid_{i+1}" for i in range(len(centroids))]
data_with_distance['Cluster'] = data_with_distance[distance_columns].idxmin(axis=1)

//Mengganti nama klaster agar lebih mudah dibaca
data_with_distance['Cluster'] = data_with_distance['Cluster'].replace({
    'Centroid_1': 'Cluster 1',
    'Centroid_2': 'Cluster 2',
    'Centroid_3': 'Cluster 3'
})

//Mengecek stabilitas klaster
if previous_clusters.equals(data_with_distance['Cluster']):
    print("Klaster telah stabil. Iterasi dihentikan.")
    break
else:
    print("Mencapai iterasi maksimum.")

//Menampilkan hasil akhir
print("Data dengan klaster akhir:")
print(data_with_distance[['Produksi', 'Luas Panen', 'Cluster']])
```

Output:

```
Iterasi ke-1
Centroid Baru:
```

	Produksi	Luas Panen
Cluster		
Cluster 1	1.357258e+06	341538.545455
Cluster 2	1.988081e+06	353438.465000
Cluster 3	1.533974e+06	354354.000000
Iterasi ke-2		
Centroid Baru:		
	Produksi	Luas Panen
Cluster		
Cluster 1	1.357258e+06	341538.545455
Cluster 2	2.051797e+06	361845.801250
Cluster 3	1.578250e+06	346677.360000
Iterasi ke-3		
Centroid Baru:		
	Produksi	Luas Panen
Cluster		
Cluster 1	1.357258e+06	341538.545455
Cluster 2	2.142181e+06	360299.901667
Cluster 3	1.615049e+06	350278.476364
Iterasi ke-4		
Centroid Baru:		
	Produksi	Luas Panen
Cluster		
Cluster 1	1367379.25	343061.750000
Cluster 2	2292864.00	366948.250000
Cluster 3	1664038.05	348937.720833
Iterasi ke-5		
Centroid Baru:		
	Produksi	Luas Panen
Cluster		
Cluster 1	1.376574e+06	347192.769231
Cluster 2	2.292864e+06	366948.250000
Cluster 3	1.680141e+06	344589.786364

Klaster telah stabil. Iterasi dihentikan.

Data dengan klaster akhir:

	Produksi	Luas Panen	Cluster
0	1329536.0	323589.00	Cluster 1
1	1299699.0	329041.00	Cluster 1
2	1382905.0	339253.00	Cluster 1
3	1419128.0	348223.00	Cluster 1
4	1368074.0	337561.00	Cluster 1
5	1404580.0	358592.00	Cluster 1
6	1478712.0	359817.00	Cluster 1
7	1486909.0	396765.00	Cluster 1
8	1547499.0	295212.00	Cluster 3
9	1314165.0	315131.00	Cluster 1
10	1246614.0	367636.00	Cluster 1
11	1350748.0	370896.00	Cluster 1
12	1411563.0	337893.00	Cluster 1
13	1552078.0	320789.00	Cluster 3
14	1556858.0	360717.00	Cluster 3
15	1402827.0	329109.00	Cluster 1
16	1533369.0	359375.00	Cluster 3
17	1788325.0	352281.00	Cluster 3
18	1772962.0	380686.00	Cluster 3
19	1582393.0	387803.00	Cluster 3
20	2331046.0	419183.00	Cluster 2
21	1820062.0	376137.00	Cluster 3
22	2180734.0	461060.00	Cluster 2
23	2180754.0	293067.00	Cluster 2
24	2478922.0	294483.00	Cluster 2
25	1751996.9	329515.78	Cluster 3
26	1714437.6	310102.46	Cluster 3
27	1861567.1	317869.41	Cluster 3

Output table

Tabel 6. Klaster Data Yang Telah StabilS

Iterasi ke-1			Iterasi ke-2		
Centroid Baru:			Centroid Baru:		
Cluster	Produksi	Luas Panen	Cluster	Produksi	Luas Panen
Cluster 1	1.357258e+06	341538.545455	Cluster 1	1.357258e+06	341538.545455
Cluster 2	1.988081e+06	353438.465000	Cluster 2	2.051797e+06	361845.801250
Cluster 3	1.533974e+06	354354.000000	Cluster 3	1.578250e+06	346677.360000
Iterasi ke-3			Iterasi ke-4		
Centroid Baru:			Centroid Baru:		
Cluster	Produksi	Luas Panen	Cluster	Produksi	Luas Panen
Cluster 1	1.357258e+06	341538.545455	Cluster 1	1367379.25	343061.750000
Cluster 2	2.142181e+06	360299.901667	Cluster 2	2292864.00	366948.250000
Cluster 3	1.615049e+06	350278.476364	Cluster 3	1664038.05	348937.720833
Iterasi ke-5			Centroid Baru:		
Cluster	Produksi	Luas Panen			
Cluster 1	1.376574e347192.769231				
Cluster 2	2.292864e366948.250000				
Cluster 3	1.680141e344589.786364				
Klaster telah stabil. Iterasi dihentikan.					

Analisa Kebutuhan

Dalam penelitian ini, beberapa kebutuhan penting telah diidentifikasi untuk memastikan implementasi metode K-Means dapat menghasilkan analisis klaster yang akurat dan mendukung tujuan penelitian. Pertama, diperlukan dataset yang berkualitas tinggi. Dataset tersebut harus mencakup data produksi padi di Aceh dengan variabel utama seperti luas lahan, produktivitas dalam satuan ton per hektar (ton/ha), dan volume produksi dalam ton. Selain itu, data harus mencakup periode waktu yang memadai untuk analisis tren historis, sehingga dapat menggambarkan perkembangan produksi secara menyeluruh.

Tahapan preprocessing data menjadi kebutuhan kedua yang sangat penting. Data yang diperoleh harus melalui proses pembersihan untuk menghapus nilai yang hilang (missing values) serta mengidentifikasi dan menghilangkan data pencilan (outliers). Hal ini dilakukan agar hasil analisis lebih representatif dan bebas dari bias. Normalisasi data juga merupakan langkah krusial untuk memastikan semua variabel berada dalam skala yang setara, mengingat variabel-variabel tersebut memiliki satuan yang berbeda.

Ketiga, metode klasterisasi yang digunakan harus tepat dan relevan. Penentuan jumlah klaster yang optimal, misalnya, dilakukan menggunakan metode Elbow atau Silhouette Score. Setelah jumlah klaster ditentukan, algoritma K-Means diterapkan untuk membentuk klaster yang stabil dengan karakteristik yang dapat dianalisis lebih lanjut.

Sumber daya teknologi juga menjadi kebutuhan yang tidak dapat diabaikan. Perangkat keras seperti komputer dengan kemampuan pemrosesan data besar, serta perangkat lunak seperti Python dengan pustaka pendukung seperti Pandas, NumPy, dan Matplotlib, digunakan untuk mendukung proses analisis data dan visualisasi hasil.

Hasil analisis harus divisualisasikan dengan jelas untuk mempermudah interpretasi. Visualisasi ini dapat berupa scatter plot atau diagram batang yang memberikan gambaran distribusi data dalam setiap klaster. Selain itu, analisis karakteristik setiap klaster dilakukan

untuk mengidentifikasi wilayah-wilayah yang membutuhkan perhatian khusus atau dapat dijadikan acuan sebagai wilayah dengan praktik pertanian yang optimal.

Terakhir, hasil penelitian ini harus mampu memberikan rekomendasi kebijakan yang dapat diterapkan secara nyata. Rekomendasi ini mencakup strategi berbasis data untuk meningkatkan produktivitas di wilayah dengan hasil rendah, serta pengembangan dan replikasi praktik terbaik dari wilayah dengan hasil tinggi. Dengan demikian, penelitian ini tidak hanya memberikan wawasan analitis tetapi juga memberikan dampak nyata dalam pengelolaan pertanian.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil menerapkan metode K-Means untuk mengelompokkan wilayah produksi padi di Aceh ke dalam tiga kluster utama berdasarkan variabel luas lahan, produktivitas, dan volume produksi. Proses klusterisasi ini menghasilkan tiga kluster dengan karakteristik yang berbeda. Kluster pertama terdiri dari wilayah-wilayah dengan produktivitas tinggi tetapi luas lahan kecil, yang tetap mampu menghasilkan volume produksi yang besar. Kluster kedua mencakup wilayah dengan luas lahan besar tetapi produktivitasnya relatif rendah, sehingga memerlukan perhatian khusus untuk meningkatkan efisiensinya. Kluster ketiga adalah wilayah dengan luas lahan dan produktivitas sedang, yang cenderung stabil tetapi memiliki potensi untuk dikembangkan lebih lanjut.

Analisis tren produksi menunjukkan adanya peningkatan produksi dari waktu ke waktu. Pergeseran data dari kluster dengan hasil rendah ke kluster dengan hasil sedang dan tinggi menunjukkan adanya perbaikan dalam praktik pertanian dan manajemen lahan di Aceh selama periode pengamatan.

Metode K-Means terbukti efektif dalam mengidentifikasi pola produksi dan memberikan wawasan berbasis data yang dapat digunakan sebagai dasar pengambilan kebijakan. Hasil penelitian ini memberikan peluang untuk meningkatkan produktivitas melalui distribusi sumber daya yang lebih adil, pengembangan teknologi pertanian modern, dan penyuluhan berkelanjutan bagi petani. Dengan langkah-langkah ini, diharapkan hasil produksi padi di Aceh dapat terus meningkat, mendukung ketahanan pangan, dan memberikan kontribusi signifikan bagi perekonomian daerah.

DAFTAR REFERENSI

- Hastuti, R., & Setiawan, D. A. (2021). Pengelompokan data penyakit TBC menggunakan algoritma K-Means. *Jurnal Teknologi Informasi dan Komputer*, 3(4).
- Putra, M. A., & Hidayat, A. F. (2020). Clustering produktivitas pertanian menggunakan metode K-Means. *Jurnal Teknologi Informasi Pertanian*, 2(1).
- Saputra, A. Z., Suarna, N., & Lestari, G. D. (2020). Klasterisasi nilai ujian sekolah menggunakan metode algoritma K-Means. *Jurnal Sistem Informasi dan Komputer*, 6(2).
- Wardhani, A. K. (2019). K-Means algorithm implementation for clustering of patients disease in Kajen Clinic of Pekalongan. *Journal of Health Informatics*, 5(1).