

**Analisis Potensi Tanaman Florikultura di Indonesia untuk Menyongsong
Indonesia Emas 2045 dengan Metode Clustering: K-Nearest Neighbour**



Yogyakarta, 31 Mei 2023

KELOMPOK 2

Anggota Kelompok:

Syakhrul Afandi	21/483096/PA/21060
Silvy Aqila Maharani	21/478973/PA/20777
Sintikke Finnesya Bisay	21/479646/PA/20792
Pramana Cahya Girindra	21/479923/PA/20825
Ahmad Syarofi Hikam	21/480410/PA/20868
Dinda Azhar Maulida	21/481852/PA/20994

**LABORATORIUM KOMPUTASI MATEMATIKA DAN STATISTIKA
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
2023**

ABSTRAK

Produksi tanaman hias merupakan salah satu bidang usaha yang menjanjikan untuk dilakukan di negara yang merupakan negara tropis. Dengan iklim yang sesuai dan kekayaan genetik dari florikultura Indonesia, sektor ini merupakan salah sektor yang tepat dilakukan di negara ini. Untuk memajukan sektor ini diperlukan kerja sama antara pemerintah dan petani tanaman hias. Pemerintah harus aktif memberi bantuan dalam berbagai macam cara. Salah satunya adalah dengan memberikan para petani ilmu tentang cara memproduksi tanaman hias. Para petani pun harus membuat inovasi-inovasi yang dapat meningkatkan daya jual pada tanaman hias. Sehingga dengan adanya ilmu yang memadai dan inovasi yang terus dikembangkan, diharapkan nantinya potensi sektor ini dapat digunakan dengan semaksimal mungkin.

Antusias dari tanaman hias makin hari kian tinggi. Tak hanya dalam negeri, penjualan tanaman hias ini sudah sampai ke manca negara. Banyak negara seperti Jepang, Saudi Arabia, Inggris, Amerika, dan sebagainya, menyukai tanaman hias khas Indonesia sehingga potensi pasar dari produksi tanaman hias sangat besar, bahkan lebih besar dibandingkan dengan kopi dan teh. Maka dari itu, potensi dari sektor ini tidak boleh disia-siakan. Pengembangan industri florikultura memerlukan dukungan inovasi berkelanjutan berupa varietas unggul baru dan teknologi pendukungnya. faktor kunci dalam pengembangan sektor florikultura adalah Ketersediaan inovasi unggul. Inovasi yang ada harus dapat meningkatkan kapasitas produksi dan produktivitas, sehingga terjadi pertumbuhan jumlah produksi dan meningkatnya daya saing. Sehingga nantinya sektor ini dapat menjadi salah satu sektor yang dapat membangun perekonomian bangsa.

Hal ini memotivasi penulis untuk menjadikan produksi tanaman hias di berbagai daerah Indonesia menjadi tema dari makalah ini. Dengan menganalisis data produksi tanaman florikultura tahun 2022 yang bersumber dari Badan Pusat Statistik, penulis berharap dapat mengetahui daerah di negara Indonesia yang memiliki potensi besar dalam membudidayakan tanaman hias di masa depan. Sehingga nantinya dapat dilakukan evaluasi yang mendalam dan potensi yang ada dapat digunakan secara maksimal.

Kata Kunci : tanaman hias, florikultura, data clustering, data mining

BAB I

PENDAHULUAN

A. Latar Belakang

Florikultura hias adalah industri yang berhubungan dengan budidaya, produksi, dan perdagangan tanaman hias, seperti bunga potong, tanaman pot, dan tanaman hias lainnya. Tanaman florikultura hias sering digunakan untuk menghiasi taman, ruangan, acara pernikahan, dan berbagai kesempatan lainnya.

Indonesia sendiri memiliki kekayaan alam yang melimpah, iklim yang sesuai, serta keahlian dalam budidaya tanaman hias. Keanekaragaman hayati yang dimiliki Indonesia memberikan peluang bisnis tanaman hias baik untuk penyediaan kebutuhan dalam negeri maupun dunia yang pasarnya masih terbuka lebar. Ada beberapa tanaman hias yang populer di Indonesia contohnya mawar, anggrek, krisan, melati, bougainvillea, dan masih banyak lagi. Dampak dari produksi tanaman florikultura hias terhadap perekonomian di Indonesia dapat mencakup banyak hal, salah satu contohnya membuka lapangan pekerjaan bagi masyarakat dan kegiatan ekspor yang sangat menguntungkan.

Menurut Menteri Koordinator Bidang Perekonomian, Airlangga Hartanto, pada tahun 2021, *global market value* tanaman hias mencapai nilai US\$22,329 miliar, lebih tinggi dibandingkan kopi dan teh. Namun, Indonesia baru memenuhi pasar dunia sebesar 0,1%. Padahal apabila potensi ini digunakan secara maksimal, tanaman hias bisa menjadi salah satu penyokong perekonomian negara.

Sebagai sebuah potensi yang dapat membantu memulihkan ekonomi nasional, tanaman hias juga memiliki peran untuk membuka lapangan pekerjaan dalam hal budidaya. Setiap tanaman hias memerlukan perawatan yang cukup detail sehingga menghasilkan tanaman hias yang layak untuk diperjualbelikan. Apalagi untuk bersaing di pasar internasional, budidaya tanaman hias perlu dipikirkan secara seksama.

Selain itu, tanaman hias juga dapat menjadi salah satu daya tarik wisatawan. Dengan keunikan dan keindahan alami yang jarang ditemukan di negara lain, tanaman hias juga berpotensi meningkatkan pertumbuhan sektor pariwisata. Festival bunga menjadi salah satu cara yang dapat digunakan sebagai media promosi budaya satu negara guna menarik minat wisatawan untuk datang ke daerah atau negara tersebut. Festival ini merupakan adaptasi dari festival bunga di Pasedena yang bernama *Tournament of Rose* (TOR).

B. Tujuan dan manfaat yang diperoleh dari proses analisis data

1. Tujuan

Setelah mengetahui bagaimana dampak dari produksi tanaman florikultura hias terhadap perekonomian di Indonesia, kami ingin tahu Provinsi manakah yang paling berpotensi membudidayakan tanaman hias serta tanaman yang paling banyak diproduksi.

2. Manfaat

- Mengetahui provinsi manakah yang paling berpotensi membudidayakan tanaman hias
- Mengetahui tanaman manakah yang paling banyak diproduksi pada provinsi di Indonesia
- Dapat memaksimalkan potensi tanaman hias

-

BAB II PEMBAHASAN

A. Metode analisis data

1. Data Mining

Data Mining Data Mining merupakan proses ekstraksi data menjadi informasi yang sebelumnya belum tersampaikan, dengan teknik yang tepat proses data mining akan memberikan hasil yang optimal. Adapun pengelompokan data mining dapat dibagi menjadi empat kelompok, diantaranya yaitu model prediksi (*prediction modelling*), analisis kelompok (*Cluster analysis*), analisis asosiasi (*association analysis*) dan deteksi anomaly (*anomaly detection*). Riset ini salah satunya menggunakan algoritma *Clustering : K-Nearest Neighbour* yang termasuk pada tipe peran clustering pada data mining.

2. *Clustering*

Clustering merupakan salah satu metode Data Mining yang bersifat tanpa arahan (*unsupervised*), maksudnya metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*) serta tidak memerlukan target output.

3. Algoritma *Clustering : K-Nearest Neighbour*

Algoritma *Clustering : K-Nearest Neighbour* adalah algoritma yang mempartisi data ke dalam beberapa cluster sehingga data yang memiliki kemiripan berada pada satu cluster yang sama dan data yang memiliki ketidaksamaan berada pada cluster yang lain. Dalam algoritma ini, setiap data harus termasuk ke cluster tertentu pada suatu tahapan proses, pada tahapan proses berikutnya dapat berpindah ke cluster yang lain.

4. Alat Bantu

Alat bantu yang dipakai untuk keperluan analisis data adalah Phyton, alat ini membantu dalam proses meng-*import* data, *preprocessing* data, dan K-Means *Clustering*. Alasan memilih Phyton untuk analisis ini adalah Python memiliki ekosistem pustaka yang sangat kaya, seperti NumPy, Pandas, dan scikit-learn, yang menyediakan berbagai fungsi dan alat untuk melakukan analisis data dan implementasi algoritma K-means. Dengan menggunakan pustaka-pustaka ini, dapat mengurangi waktu dan usaha dalam mengimplementasikan algoritma K-means dari awal. Meskipun Python adalah bahasa pemrograman interpretatif, pustaka-pustaka yang dioptimalkan seperti NumPy dapat mempercepat komputasi numerik.

B. Analisis Data

1. Pengolahan Data

- Data Sampel

Data sampel yang digunakan yaitu data produksi tanaman florikultura (hias) 2022. Data berisikan tentang jumlah dari dua belas jenis bunga yang ada di 34 provinsi di Indonesia.

- *Preprocessing Data*

Setelah didapatkan data produksi tanaman florikultura (hias) 2022 diperlukan adanya tahap *preprocessing* data sebelum dilakukan tahap pengimplementasian pada sistem analisis. Adapun tahapan *preprocessing* data yang dilakukan sebagai berikut ini:

- a. Seleksi Atribut

Dari beragamnya atribut pada data awal, setelah dilakukan penyeleksian atribut terdapat atribut yang tidak dipakai karena tidak berisikan data sama sekali, atribut tersebut yaitu Anyelir (Tangkai), Gladiol (Tangkai), dan Melati (Pohon).

- b. Pembersihan Data (*Data Cleaning*)

Setelah 3 (Tiga) atribut diseleksi, maka dilakukan juga pembersihan data yang tidak sesuai sebelum proses pengolahan data lebih lanjut. Seperti mengubah jenis *value* yang semula berjenis object atau string menjadi float, dan mengisi *missing value* dengan mean.

- c. Cek Distribusi Data

Dataset dari hasil proses seleksi atribut dan pembersihan data sebelumnya, selanjutnya dilakukan pengecekan data, apakah data sudah berdistribusi normal atau tidak. Berdasarkan *output* yang diperoleh terlihat bahwa data tidak berdistribusi normal sehingga perlu dilakukan proses normalisasi sebelum dilakukan proses *clustering*.

2. *K-Means Clustering*

- Pertama dilakukan upload dataset pada Aplikasi Phyton seperti pada gambar dibawah ini.

```
import pandas as pd
data = pd.read_csv(r"C:\Users\Lenovo\Downloads\atas lg_files\Produksi Tanaman Florikultura (Hias).zip")
data.head()
```

Dan diperoleh output sebagai berikut:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Interpretasi :

Output tersebut merupakan hasil pembacaan file Excel dengan menggunakan library pandas di Python. Data yang ada dalam file Excel dengan nama "Produksi Tanaman Florikultura (Hias).xlsx" telah diubah menjadi objek data frame dalam pandas. Data frame adalah struktur data tabular dua dimensi yang terdiri dari baris dan kolom. Dalam hal ini, variabel 'data' berisi data frame yang berisi informasi yang terdapat dalam file Excel tersebut.

- mengembalikan hasil dari operasi sebelumnya, yaitu nilai-nilai unik dari baris data pada indeks ke-1. Dengan syntax seperti pada gambar dibawah ini.

```
data.loc[1].unique()
```

Dan diperoleh output sebagai berikut:

```
array([nan, '2021'], dtype=object)
```

Interpretasi:

Array yang diberikan memiliki dua elemen. Elemen pertama memiliki nilai NaN, yang merupakan singkatan dari "Not a Number" dan menunjukkan bahwa nilai tersebut tidak dapat diidentifikasi sebagai angka atau data yang valid. Elemen kedua adalah string '2021', yang merupakan representasi tahun 2021 dalam format teks. Dengan demikian, array ini memiliki kombinasi nilai numerik yang tidak valid dan sebuah string yang menyatakan tahun.

- Untuk Merapikan Dataframe dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
import numpy as np
#Slicing isi dataframe yang diperlukan
df = data.iloc[2:,0:]
df = df.drop(np.array(range(36,41)))
#Mengganti nama kolom data frame
df = df.reset_index()
df.drop(columns=["index"], inplace=True)
df = df.rename(columns=dict(zip(df.columns[1:], np.array(data.iloc[0, 1:])))
```

interpretasi:

Output yang dihasilkan adalah data frame yang telah disesuaikan dengan slicing, penghapusan baris, pengaturan ulang indeks, penghapusan kolom "index", dan penggantian nama kolom. Data frame ini merupakan hasil transformasi data yang mungkin digunakan untuk tujuan analisis atau pemrosesan selanjutnya.

- Untuk Melihat Dataframe dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Mengecek dimensi data frame
df.shape
```

Dan diperoleh output sebagai berikut:

```
(34, 16)
```

Interpretasi:

Sintaks "df.shape" digunakan untuk mengakses dimensi atau ukuran data frame "df". Output "(34, 16)" menunjukkan bahwa data frame tersebut memiliki 34 baris dan 16 kolom. Dengan kata lain, data frame "df" terdiri dari 34 entri atau observasi yang terorganisir dalam baris, dan memiliki 16 variabel atau fitur yang diwakili oleh kolom-kolomnya. Informasi ini berguna untuk memahami sejauh mana data frame tersebut meluas dan menyediakan gambaran tentang jumlah entri dan variabel yang ada di dalamnya.

- Untuk Mengganti isi "-" menjadi nan dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Mengganti isi "-" menjadi nan
df.replace("-", np.nan, inplace=True)
```

interpretasi:

Perintah ini bertujuan untuk mengganti semua nilai tanda hubung ("-") dengan nilai NaN dalam data frame. Biasanya, ini dilakukan ketika nilai tanda hubung digunakan untuk mewakili data yang hilang atau tidak tersedia. Dengan menggantikan nilai tersebut dengan NaN, dapat mempermudah analisis dan pemrosesan data selanjutnya, seperti menghapus atau mengisi nilai yang hilang.

- Untuk Mengecek *missing value* dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Mengecek missing value
df.isna().sum()
```

Dan diperoleh output sebagai berikut:

```
Provinsi      0
Anggrek Potong (Tangkai)    3
Anthurium Bunga (Tangkai)   7
Anyelir (Tangkai)          34
Gerbera/Herbras (Tangkai)   15
Gladiol (Tangkai)          34
Heliconia/Pisang-pisangan (Tangkai)  4
Krisan (Tangkai)           8
Mawar (Tangkai)            4
Sedap Malam (Tangkai)      11
Dracaena (Pohon)           11
Melati (Pohon)             34
Palem (Pohon)              3
Anggrek Pot (Pohon)         6
Pakis (Pohon)              6
Melati (Kg)                 6
dtype: int64
```

Interpretasi:

Interpretasi singkatnya adalah bahwa output tersebut memberikan informasi tentang berapa banyak nilai yang hilang dalam setiap kolom data frame `df`. Menggunakan metode `.isna().sum()` sangat berguna untuk mendapatkan gambaran tentang kualitas data, melihat apakah terdapat kekurangan atau nilai yang hilang dalam kolom-kolom tertentu, dan memutuskan langkah-langkah yang diperlukan untuk menangani data yang hilang tersebut, seperti menghapus baris atau mengisi nilai yang hilang. Dari output di atas, dapat dilihat bahwa terdapat beberapa kolom yang tidak memiliki isi sama sekali (berisi missing value semua) sehingga variabel-variabel tersebut dapat dihapus

- Untuk Menghapus kolom kosong dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
df.drop(columns=["Anyelir (Tangkai)", "Gladiol (Tangkai)", "Melati (Pohon)"], inplace=True)
```

Interpretasi:

Output dari perintah `df.drop(columns=["Anyelir (Tangkai)", "Gladiol (Tangkai)", "Melati (Pohon)"], inplace=True)` adalah perubahan pada data frame `df`. Perintah ini menghapus kolom-kolom yang ditentukan dari data frame. Opsi `inplace=True` menunjukkan bahwa perubahan dilakukan langsung pada objek `df`, tanpa menghasilkan objek data frame baru. Perintah ini bertujuan untuk menghapus kolom-kolom "Anyelir (Tangkai)", "Gladiol (Tangkai)", dan "Melati (Pohon)" dari data frame. Dengan menghapus kolom-kolom ini, kita dapat mengurangi dimensi data frame atau menghilangkan informasi yang tidak diperlukan untuk analisis atau pemrosesan data selanjutnya.

- Untuk Mengubah jenis *value* dari *object* atau *string* menjadi *float* dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Mengubah jenis value yang semula berjenis object atau string menjadi float
for i in df.columns[1:]:
    df[i]=df[i].astype(float)
```

interpretasi:

Perintah ini bertujuan untuk mengubah tipe data kolom-kolom dalam data frame menjadi float. Dengan mengubah tipe data menjadi float, kita dapat melakukan operasi numerik dan analisis statistik yang lebih lanjut pada kolom-kolom tersebut. Perubahan tipe data ini berguna jika kolom-kolom tersebut semula memiliki tipe data yang tidak sesuai untuk operasi numerik, seperti string atau objek.

- Untuk Mengisi nilai *missing value* dengan *mean* dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Mengisi missing value dengan mean
for i in df.columns[1:]:
    df[i] = df[i].fillna(df[i].mean())
df.iloc[0:,1:]=df.iloc[0:,1:].apply(lambda x: x.apply('{0:.5f}'.format))
```

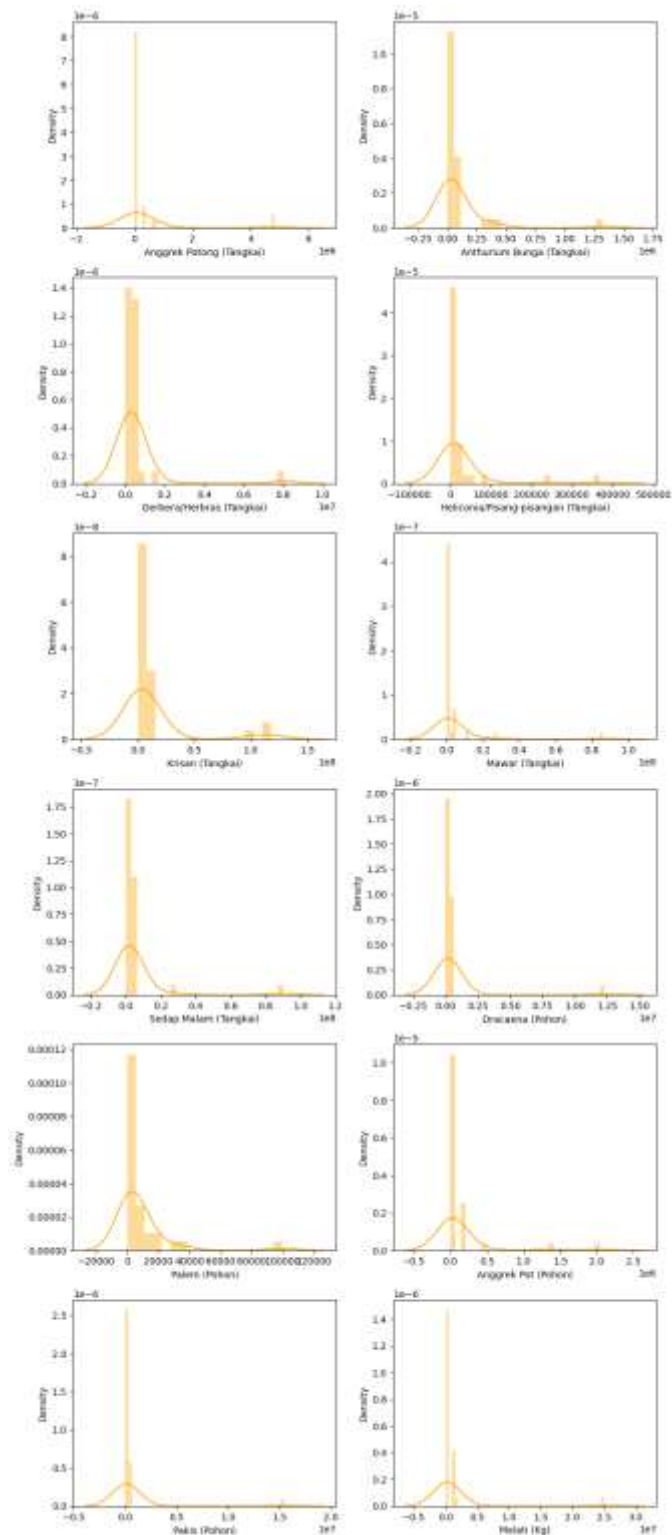
Interpretasi:

Perintah-perintah tersebut bertujuan untuk mengisi nilai yang hilang dengan nilai rata-rata kolom yang sesuai dan mengubah format angka dalam kolom-kolom menjadi format angka dengan 5 angka desimal. Dengan melakukan ini, kita dapat mengatasi nilai yang hilang dengan menggunakan nilai rata-rata, dan memformat angka untuk presentasi atau analisis yang lebih tepat.

- Untuk membuat EDA dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#EDA
plt.figure(figsize=(10, 30))
for i in range(0, len(df.columns[1:])):
    plt.subplot(8, 2,i+1)
    sns.distplot(df[df.columns[1:][i]], color='orange')
plt.tight_layout()
```

Dan diperoleh output sebagai berikut:



Interpretasi:

Kode tersebut menghasilkan plot dengan beberapa subplots yang menunjukkan distribusi data dari setiap kolom dalam data frame. Setiap subplot menampilkan histogram yang menggambarkan frekuensi kemunculan nilai-nilai dalam kolom tersebut. Ukuran keseluruhan plot diatur menjadi 10x30 untuk memastikan subplots

yang cukup besar. Plot ini membantu dalam pemahaman visual tentang distribusi data dalam setiap kolom, sehingga dapat melihat pola, kecenderungan, dan outlier dalam data. Dari grafik distribusi di atas, dapat terlihat bahwa semuanya tidak berdistribusi normal sehingga perlu dilakukan normalisasi

- Untuk melakukan Normalisasi dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df1 = df.copy()
df1[df1.columns[1:]] = scaler.fit_transform(df1[df1.columns[1:]])
```

Interpretasi:

Kode tersebut melakukan normalisasi data pada kolom-kolom (mulai dari kolom kedua) dalam data frame `df1` menggunakan metode skala Min-Max. Normalisasi skala Min-Max mengubah nilai-nilai dalam kolom menjadi rentang antara 0 dan 1, di mana nilai terkecil dalam kolom akan menjadi 0 dan nilai terbesar akan menjadi 1, sedangkan nilai-nilai di antara akan diubah proporsional. Hal ini membantu dalam mengubah data ke dalam rentang yang seragam dan menghilangkan perbedaan skala antar kolom. Dengan melakukan normalisasi ini, data dapat diolah dan dibandingkan dengan lebih baik dalam beberapa algoritma pemodelan dan analisis statistik.

- Untuk melakukan Clustering dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

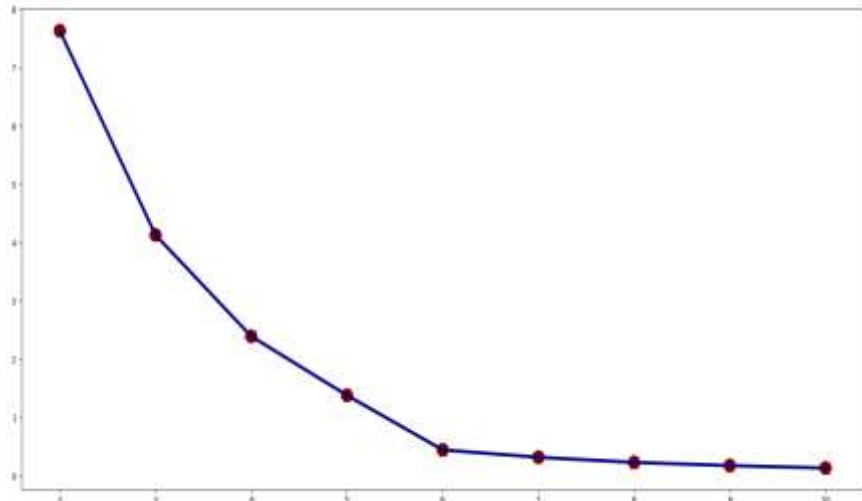
```
from sklearn.cluster import KMeans
inertia = []

for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, random_state=0)
    kmeans.fit(df1.iloc[0:,1:])
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(20, 10))

sns.lineplot(x=range(2, 11), y=inertia, color='#000087', linewidth = 4)
sns.scatterplot(x=range(2, 11), y=inertia, s=300, color='#800000', linestyle='--')
plt.show()
```

Dan diperoleh output sebagai berikut:



Interpretasi:

Output dari kode tersebut adalah sebuah plot garis dan titik yang menampilkan inertia (inersia) dari algoritma K-Means clustering pada data yang telah diubah (`df1`) dengan menggunakan jumlah kluster yang berbeda. Kode tersebut melakukan iterasi melalui rentang dari 2 hingga 10 untuk jumlah kluster dalam algoritma K-Means clustering. Pada setiap iterasi, algoritma K-Means clustering diterapkan pada data frame `df1` dengan jumlah kluster yang berbeda, dan inertia dari setiap iterasi dicatat dalam list `inertia`. Inertia mengukur sejauh mana titik-titik dalam kluster berdekatan satu sama lain, sehingga nilai inertia yang lebih rendah menunjukkan kluster yang lebih padat dan terpisah dengan baik. Setelah selesai iterasi, plot garis dan titik dibuat menggunakan library `matplotlib.pyplot` dan `seaborn`. Garis menunjukkan perubahan inertia seiring dengan jumlah kluster yang berbeda, sedangkan titik menunjukkan inertia pada setiap jumlah kluster. Plot ini membantu dalam menentukan jumlah kluster yang optimal dengan melihat titik-titik di mana perubahan inertia mulai melambat.

- Untuk melakukan lanjutan Clustering diatas dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
kmeans = KMeans(n_clusters=3, random_state=0).fit(df1.iloc[0:,1:])
df['cluster'] = kmeans.labels_
```

Interpretasi:

Output dari kode tersebut adalah penambahan kolom baru 'cluster' ke dalam data frame `df`, yang berisi label kluster dari hasil algoritma K-Means clustering. Algoritma K-Means clustering diterapkan pada data frame `df1` dengan jumlah kluster 3. Kode tersebut melakukan pengelompokan data menggunakan algoritma K-Means clustering dengan 3 kluster. Setelah klusterisasi selesai, label kluster dari setiap data diatribusikan ke kolom 'cluster' dalam data frame `df`. Ini memungkinkan untuk membedakan setiap data berdasarkan kluster yang mereka miliki, sehingga memudahkan

dalam analisis atau visualisasi lebih lanjut terkait pola atau karakteristik yang mungkin ada dalam setiap kluster.

- Untuk melihat anggota Cluster 0 dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Cluster 0
df[df['cluster']==0]
```

Dan diperoleh output sebagai berikut:

Interpretasi:

Output `df[df['cluster']==0]` menghasilkan subset dari data frame `df` yang hanya berisi baris-baris yang memiliki nilai 'cluster' sama dengan 0. Output tersebut adalah subset dari data frame `df` yang hanya berisi baris-baris yang termasuk ke dalam kluster dengan label 0. Dengan menggunakan kondisi `df['cluster']==0`, kita dapat memfilter data frame untuk mendapatkan baris-baris yang terkait dengan kluster tertentu. Hal ini berguna untuk melakukan analisis atau pemrosesan lebih lanjut pada kelompok data yang memiliki karakteristik atau pola yang serupa dalam kluster yang ditentukan.

- Untuk melihat anggota Cluster 1 dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
#Cluster 1
df[df['cluster']==1]
```

Dan diperoleh output sebagai berikut:

Interpretasi:

Output `df[df['cluster']==1]` menghasilkan subset dari data frame `df` yang hanya berisi baris-baris yang memiliki nilai 'cluster' sama dengan 1. Output tersebut adalah subset dari data frame `df` yang hanya berisi baris-baris yang termasuk ke dalam kluster dengan label 1. Dengan menggunakan kondisi `df['cluster']==1`, kita dapat memfilter data frame untuk mendapatkan baris-baris yang terkait dengan kluster tertentu. Hal ini berguna untuk melakukan analisis atau pemrosesan lebih lanjut pada kelompok data yang memiliki

karakteristik atau pola yang serupa dalam klaster yang ditentukan. Dalam konteks ini, subset ini berisi data yang termasuk ke dalam klaster dengan label 1.

- Untuk melihat anggota Cluster 2 dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
df[df['cluster']==2]
```

Dan diperoleh output sebagai berikut:

Particulars	2019-2020	2018-2019	2017-2018	2016-2017	2015-2016	2014-2015	2013-2014	2012-2013	2011-2012	2010-2011	2009-2010	2008-2009	2007-2008	2006-2007	2005-2006	2004-2005	2003-2004	2002-2003	2001-2002	2000-2001	1999-2000	1998-1999	1997-1998	1996-1997	1995-1996	1994-1995	1993-1994	1992-1993	1991-1992	1990-1991	1989-1990	1988-1989	1987-1988	1986-1987	1985-1986	1984-1985	1983-1984	1982-1983	1981-1982	1980-1981	1979-1980	1978-1979	1977-1978	1976-1977	1975-1976	1974-1975	1973-1974	1972-1973	1971-1972	1970-1971	1969-1970	1968-1969	1967-1968	1966-1967	1965-1966	1964-1965	1963-1964	1962-1963	1961-1962	1960-1961	1959-1960	1958-1959	1957-1958	1956-1957	1955-1956	1954-1955	1953-1954	1952-1953	1951-1952	1950-1951	1949-1950	1948-1949	1947-1948	1946-1947	1945-1946	1944-1945	1943-1944	1942-1943	1941-1942	1940-1941	1939-1940	1938-1939	1937-1938	1936-1937	1935-1936	1934-1935	1933-1934	1932-1933	1931-1932	1930-1931	1929-1930	1928-1929	1927-1928	1926-1927	1925-1926	1924-1925	1923-1924	1922-1923	1921-1922	1920-1921	1919-1920	1918-1919	1917-1918	1916-1917	1915-1916	1914-1915	1913-1914	1912-1913	1911-1912	1910-1911	1909-1910	1908-1909	1907-1908	1906-1907	1905-1906	1904-1905	1903-1904	1902-1903	1901-1902	1900-1901	1899-1900	1898-1899	1897-1898	1896-1897	1895-1896	1894-1895	1893-1894	1892-1893	1891-1892	1890-1891	1889-1890	1888-1889	1887-1888	1886-1887	1885-1886	1884-1885	1883-1884	1882-1883	1881-1882	1880-1881	1879-1880	1878-1879	1877-1878	1876-1877	1875-1876	1874-1875	1873-1874	1872-1873	1871-1872	1870-1871	1869-1870	1868-1869	1867-1868	1866-1867	1865-1866	1864-1865	1863-1864	1862-1863	1861-1862	1860-1861	1859-1860	1858-1859	1857-1858	1856-1857	1855-1856	1854-1855	1853-1854	1852-1853	1851-1852	1850-1851	1849-1850	1848-1849	1847-1848	1846-1847	1845-1846	1844-1845	1843-1844	1842-1843	1841-1842	1840-1841	1839-1840	1838-1839	1837-1838	1836-1837	1835-1836	1834-1835	1833-1834	1832-1833	1831-1832	1830-1831	1829-1830	1828-1829	1827-1828	1826-1827	1825-1826	1824-1825	1823-1824	1822-1823	1821-1822	1820-1821	1819-1820	1818-1819	1817-1818	1816-1817	1815-1816	1814-1815	1813-1814	1812-1813	1811-1812	1810-1811	1809-1810	1808-1809	1807-1808	1806-1807	1805-1806	1804-1805	1803-1804	1802-1803	1801-1802	1800-1801	1799-1800	1798-1799	1797-1798	1796-1797	1795-1796	1794-1795	1793-1794	1792-1793	1791-1792	1790-1791	1789-1790	1788-1789	1787-1788	1786-1787	1785-1786	1784-1785	1783-1784	1782-1783	1781-1782	1780-1781	1779-1780	1778-1779	1777-1778	1776-1777	1775-1776	1774-1775	1773-1774	1772-1773	1771-1772	1770-1771	1769-1770	1768-1769	1767-1768	1766-1767	1765-1766	1764-1765	1763-1764	1762-1763	1761-1762	1760-1761	1759-1760	1758-1759	1757-1758	1756-1757	1755-1756	1754-1755	1753-1754	1752-1753	1751-1752	1750-1751	1749-1750	1748-1749	1747-1748	1746-1747	1745-1746	1744-1745	1743-1744	1742-1743	1741-1742	1740-1741	1739-1740	1738-1739	1737-1738	1736-1737	1735-1736	1734-1735	1733-1734	1732-1733	1731-1732	1730-1731	1729-1730	
-------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	--

Interpretasi:

Output ``df[df['cluster']==2]`` menghasilkan subset dari data frame ``df`` yang hanya berisi baris-baris yang memiliki nilai 'cluster' sama dengan 2. Output tersebut adalah subset dari data frame ``df`` yang hanya berisi baris-baris yang termasuk ke dalam kluster dengan label 2. Dengan menggunakan kondisi ``df['cluster']==2``, kita dapat memfilter data frame untuk mendapatkan baris-baris yang terkait dengan kluster tertentu. Hal ini berguna untuk melakukan analisis atau pemrosesan lebih lanjut pada kelompok data yang memiliki karakteristik atau pola yang serupa dalam kluster yang ditentukan. Dalam konteks ini, subset ini berisi data yang termasuk ke dalam kluster dengan label 2.

- Untuk memilih kolom untuk dikelompokkan dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
df[df.columns[1:].tolist()].groupby('cluster').mean()
```

Dan diperoleh output sebagai berikut:

	Aggregat Refugee (Tongkat)	Author's Refugee (Tongkat)	Barbara/Warner (Tongkat)	Helmut/Fleming (Tongkat)	Bruce (Tongkat)	Rae (Tongkat)	Sadie Helen (Tongkat)	Bracewell (Pauze)	Helen (Pauze)	Aggregat Pet (Pauze)	Pauze (Pauze)
cluster											
8	2.21208e+05	4.45093e+04	2.35806e+05	1.05527e+05	3.75131e+05	7.93472e+05	1.67155e+06	1.02225e+06	405142e+07	3.32273e+06	1.15988e+07
9	3.48205e+05	5.53741e+05	8.84627e+05	4.06139e+05	1.96884e+06	6.00675e+05	9.02517e+05	4.46249e+06	6.93830e+06	8.03263e+06	7.78139e+06
3	4.82673e+05	1.21023e+06	5.94027e+05	2.69114e+05	1.02374e+06	1.15927e+05	3.46822e+05	1.27788e+07	1021200000	2.03225e+06	1.30325e+06

di sini kita akan memilih semua kolom kecuali kolom pertama (kolom 'cluster'). Kemudian, DataFrame yang dipilih akan dikelompokkan berdasarkan nilai pada kolom 'cluster' menggunakan metode ``groupby('cluster')``. Dan kita akan menghitung nilai rata-ratanya.

```
cluster
1      2.128779e+07
2      1.205789e+07
0      6.325588e+05
dtype: float64
```

Output di atas, merupakan rata-rata dari rata-rata masing-masing fitur yang dimiliki oleh masing-masing klaster. Dari output tersebut,

dapat disimpulkan bahwa kluster provinsi yang paling berpotensi yakni kluster 1 yang terdiri dari Jawa Tengah dan Jawa Timur.

- Untuk melakukan normalisasi dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df2 = data_clean.transpose()
df2.set_axis(df2.iloc[0], axis='columns', inplace=True)
df2 = df2[1:] # Remove the first row
df2.reset_index(inplace=True)
df2.rename(columns={'index': 'Bunga'}, inplace=True)
df2_clean = df2.copy()
df2[df2.columns[1:]] = scaler.fit_transform(df2[df2.columns[1:]])
```

- Untuk melihat visualisasi Data Cluster dilakukan analisis dengan syntax seperti pada gambar dibawah ini.

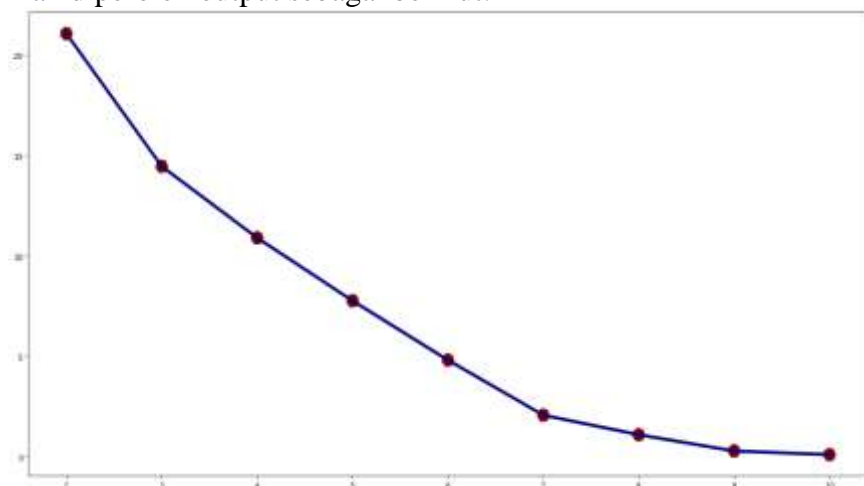
```
from sklearn.cluster import KMeans
inertia = []

for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, random_state=0)
    kmeans.fit(df2.iloc[0:,1:])
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(20, 10))

sns.lineplot(x=range(2, 11), y=inertia, color='#000087', linewidth = 4)
sns.scatterplot(x=range(2, 11), y=inertia, s=300, color='#800000', linestyle='--')
plt.show()
```

Dan diperoleh output sebagai berikut:



Output yang dihasilkan adalah grafik garis dan titik-titik yang menunjukkan perubahan inertia terhadap jumlah kluster dari 2 hingga 10.


```
kmeans = KMeans(n_clusters=7, random_state=0).fit(df2.iloc[0:,1:])
df2_clean['cluster'] = kmeans.labels_
```

Output yang dihasilkan adalah dataset `df2_clean` yang telah ditambahkan kolom 'cluster' yang berisi label klaster dari algoritma K-Means. Kolom 'cluster' menunjukkan kelompok atau klaster yang diberikan oleh algoritma K-Means untuk setiap baris dalam dataset. Label klaster tersebut dapat digunakan untuk analisis atau visualisasi lebih lanjut terhadap data.

```
for i in range(0,7):
    print("Cluster", i, "\n", df2_clean[df2_clean['cluster']==i]["Bunga"], "\n")
```

```
Cluster 0
5    Mawar (Tangkai)
Name: Bunga, dtype: object

Cluster 1
1    Anthurium Bunga (Tangkai)
3    Heliconia/Pisang-pisangan (Tangkai)
8    Palem (Pohon)
9    Anggrek Pot (Pohon)
10   Pakis (Pohon)
11   Melati (Kg)
Name: Bunga, dtype: object

Cluster 2
4    Krisan (Tangkai)
Name: Bunga, dtype: object

Cluster 3
6    Sedap Malam (Tangkai)
Name: Bunga, dtype: object

Cluster 4
2    Gerbera/Herbras (Tangkai)
Name: Bunga, dtype: object

Cluster 5
7    Dracaena (Pohon)
Name: Bunga, dtype: object

Cluster 6
0    Anggrek Potong (Tangkai)
Name: Bunga, dtype: object
```

Output yang dihasilkan adalah pemisahan baris-baris dataset berdasarkan label klaster. Setiap kelompok klaster akan ditampilkan dengan memisahkan baris-baris yang termasuk dalam klaster tersebut dan menampilkan kolom 'Bunga' dari baris-baris tersebut.

```
df2_clean[df2_clean.columns[1:].tolist()].groupby('cluster').mean()
```

Provinsi	ACEH	SUMATERA UTARA	SUMATERA BARAT	RIAU	JAMBI	SUMATERA SELATAN	BENG
cluster							
0	5.195000e+03	3.001666e+06	124563.000000	21977.000000	278.000000	2473.000000	1807.00
1	3.308500e+03	2.973667e+03	23739.166667	2297.166667	161.666667	253.000000	140.83
2	3.300000e+01	3.793943e+06	712595.000000	1270.000000	600.000000	171.000000	85.00
3	5.340527e+06	1.482936e+06	5535.000000	8775.000000	50.000000	491.000000	18.00
4	5.565225e+05	7.154950e+05	3503.000000	465.000000	556522.526316	22.000000	517.00
5	1.450000e+02	2.402000e+03	2834.000000	357.000000	539593.695652	539593.695652	10.00
6	6.200000e+01	3.663700e+04	7880.000000	9104.000000	300.000000	3866.000000	946.00

7 rows x 34 columns

Output yang dihasilkan adalah nilai rata-rata dari setiap kolom (kecuali kolom 'Bunga') dalam dataset yang dikelompokkan berdasarkan label klaster. Output ini memberikan informasi tentang rata-rata nilai dari setiap kolom dalam dataset untuk setiap klaster yang terbentuk.

```
cluster
2    1.323196e+07
3    5.340527e+06
0    4.321919e+06
4    5.565225e+05
5    5.395937e+05
6    3.661811e+05
1    3.077713e+05
dtype: float64
```

Output di atas, merupakan rata-rata dari rata-rata masing-masing fitur yang dimiliki oleh masing-masing klaster. Dari output tersebut, dapat disimpulkan bahwa klaster bunga yang paling berpotensi yakni klaster 2 yang terdiri dari Krisan (tangkai).

BAB III PENUTUP

A. Kesimpulan

Jenis pengelompokan yang digunakan penulis pada analisis data ini adalah clustering data mining yang bersifat tanpa arahan. Penulis melakukan analisis dengan menggunakan K-Means Clustering dimana algoritma yang mempartisi data ke dalam beberapa cluster sehingga data yang memiliki kemiripan berada pada satu cluster yang sama dan data yang memiliki ketidaksamaan berada pada cluster yang lain. Penulis menggunakan Alat bantu yang dipakai untuk keperluan analisis data yaitu Phyton, yang membantu dalam proses meng-*import* data, *preprocessing* data, dan K-Means *Clustering*. Data yang gunakan merupakan data tanaman florikultura tahun 2022 yang bersumber dari *website* Badan Pusat Statistika. Pertama dilakukan tahap *preprocessing* data yang berisi seleksi atribut, pembersihan data, lalu pengecekan distribusi data. Adapun pun atribut yang diseleksi yaitu Anyelir (Tangkai), Gladiol (Tangkai), dan Melati (Pohon). Lalu pembersihan data bertujuan untuk mengubah jenis value menjadi float dan mengisi missing value dengan mean. Lalu dilakukan pengecekan data dengan memeriksa apakah data yang ada berdistibu normal atau tidak. Berdasarkan analisis yang dilakukan diperoleh bahwa data tidak berdistribusi normal sehingga perlu dilakukan proses normalisasi. Lalu dilakukanlah analisis yang bertujuan untuk membagi data menjadi beberapa clustering.

Pada analisis yang dilakukan, diperoleh bahwa data yang dapat dibagi menjadi beberapa cluster. Pembagian data dilakukan berdasarkan provinsi dan jenis dari tanamas hias. Pembagian data berdasarkan provinsi menghasilkan 3 cluster. Sedangkan pembagian data berdasarkan jenis bunga, menghasilkan 7 cluster. Pembagian data berdasarkan provinsi berisikan cluster 0 yang beranggotakan 30 provinsi di Indonesia. 30 provinsi tersebut merupakan Aceh, Sumatra Utara, Sumatra Barat, Riau, Jambi, Sumatra Selatan, Bengkulu, Lampung, Kepulauan Bangka Belitung, Kepulauan Riau, DKI Jakarta, DI Yogyakarta, Banten, Bali, Nusa Tenggara Barat, Nusa Tenggara Timur, Kalimantan Barat, Kalimantan Tengah, , Kalimantan Selatan, Kalimantan Timur, Kalimantan Utara, Sulawesi Utara, SulawesiTengah, Sulawesi Selatan, Sulawesi Tenggara, Gorontalo, Sulawesi Barat, Maluku, Maluku Utara, dan Papua Barat. Adapun Klaster 1 beranggotakan 2 provinsi di Indonesia yaitu Jawa Tengah dan Jawa Timur. Dan terakhir klaster 2 yang hanya memiliki 1 anggota yaitu Jawa Barat.

Pembagian data berdasarkan jenis tanaman berisikan cluster 0 yang memiliki hanya 1 anggota yaitu Mawar (Tangkai). Sedangkan cluster 1 memiliki 6 anggota. 6 anggota tersebut adalah Anthurium Bunga (Tangkai), Heliconia/Pisang-pisangan (Tangkai), Palem (Pohon), Anggrek Pot (Pohon), Pakis (pohon), Melati (Kg). Cluster 2 pun hanya berisi 1 anggota yaitu Krisan (Tangkai). Begitupun dengan

cluster 3 yang beranggotakan Sedap Malam (Tangkai). Dan cluster 4, 5, 6 pun hanya memiliki 1 anggota yaitu Gerbera/Hebras (Tangkai), Dracaena (Pohon), dan Anggrek Potong (Tangkai) secara berturut-turut.

Dari analisis yang dilakukan, data berdasarkan provinsi, menghasilkan 3 klaster data, dimana cluster 1 mempunyai rata-rata paling tinggi. hal tersebut menandakan bahwa anggota dari cluster 1 yaitu provinsi Jawa timur dan Jawa Tengah merupakan daerah yang paling berpotensi dalam pembudidayaan tanaman hias. Lalu berdasarkan jenis tanaman hias, data dapat dibagi menjadi 7 cluster, dimana cluster 2 mempunyai rata-rata tertinggi. Hal ini menandakan bahwa anggota dari cluster 2 yaitu tangkai krisan merupakan jenis tanaman hias yang memiliki potensi paling tinggi dalam pembudidayaan tanaman hias. Pembagian data ini bertujuan untuk memudahkan dalam analisis atau visualisasi lebih lanjut terkait pola atau karakteristik yang mungkin ada dalam setiap klaster. Sehingga diperoleh hasil bahwa provinsi Jawa Barat dan tangkai krisan memiliki potensi tertinggi dalam pembudidayaan tanaman hias. Setelah diperoleh pembagian cluster dapat disimpulkan bahwa setiap klaster memiliki karakteristik masing-masing.

B. Saran

Analisis yang dilakukan tentu jauh dari kata sempurna. Penulis berharap bahwa dengan diketahuinya provinsi mana yang paling berpotensi membudidayakan tanaman hias maupun Provinsi yang memiliki peluang kecil dalam membudidayakan tanaman hias, pemerintah dapat mengevaluasi penyebab dari kecilnya peluang-peluang pembudidayaan tanaman hias di daerah tersebut. Sehingga potensi akan tanaman hias seluruh daerah di Indonesia dapat digunakan semaksimal mungkin.

C. Link video

<https://youtu.be/a-Pnelbeiqw>

DAFTAR PUSTAKA

- Oktaviano, Dino. 2021. Mengembangkan Potensi Tanaman Hias di Indonesia. Kompas Images. <https://foto.kompas.com/photo/read/2021/12/17/163973451033a/Mengembangkan-Potensi-Tanaman-Hias-di-Indonesia>. Diakses 6 Juni 2023.
- Limansetyo, Haryo. 2021. Airlangga Hartanto: Tingkatkan Ekspor Florikultura, Penuhi Ceruk Pasar Dunia. Kementerian Koordinator Bidang Perekonomian RI. <https://ekon.go.id/publikasi/detail/2978/airlangga-hartarto-tingkatkan-ekspor-florikultura-penuhi-ceruk-pasar-dunia>. Diakses 6 Juni 2023.
2023. Cara Budidaya Tanaman Hias Yang Baik dan Benar. Bibit Bunga. <https://bibitbunga.com/cara-budidaya-tanaman-hias-yang-baik-dan-benar/>. Diakses 6 Juni 2023.
- Hidranto, Firman. 2021. Genjot Ekspor Tanaman Bunga Indonesia. Portal Informasi Indonesia. <https://indonesia.go.id/kategori/editorial/2818/genjot-ekspor-tanaman-bunga-indonesia>. Diakses 6 Juni 2023.
- Humas. 2020. Banjir Permintaan Tanaman Hias: Omset Hingga 1 Miliar. Direktorat Jenderal Hortikultura Kementerian Pertanian. <https://hortikultura.pertanian.go.id/?p=6660>,. Diakses 6 Juni 2023.
- Bestiandy, Benny. 2020. Potensi Florikultura Khas Indonesia Bernilai Ekspor Tinggi. Media Indonesia. <https://mediaindonesia.com/ekonomi/360420/potensi-florikultura-khas-indonesia-bernilai-ekspor-tinggi>. Diakses 6 Juni 2023.