

PROJECT AKHIR
**Metode Density Based Scan dalam “Menganalisis Data Customer
Segmentation di dalam Mall”**

Dibuat untuk Memenuhi Tugas Akhir Pertemuan 14

Mata Kuliah IS388 – Data Analyst



Disusun oleh Kelompok 3:

Christopher Darren	00000054804
Christian Alexander	00000054292
Michelle Stephanie	00000055477
Meisha Geovanni Mulin	00000055487
Azzahra Shaffira Wijaya	00000055376

PROGRAM STUDI: SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG

2022

Kata Pengantar

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa karena berkat dan rahmat-Nya kami dapat menyelesaikan pembuatan laporan ini sebagai tugas akhir dan menjadi acuan untuk memperluas wawasan, pengetahuan, dan mengembangkan kemampuan kami dalam menganalisis sebuah data yang berguna bagi kami serta orang banyak.

Tujuan pembuatan laporan ini adalah untuk membantu memberikan informasi dan pengetahuan mengenai penerapan *Machine Learning* khususnya pada metode DBScan (metode untuk clustering). Proyek yang kami buat juga bertujuan untuk memberikan wawasan sederhana kepada para pembaca agar mampu menganalisis hal yang sederhana dalam kehidupan sehari-hari.

Dalam menyelesaikan laporan ini penulis banyak mendapatkan dukungan, bimbingan, pengarahan, dan bantuan dari berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Ibu Ririn Ikana Desanti, S.Kom., M.Kom. selaku Ketua Program Studi Universitas Multimedia Nusantara.
2. Ahmad Faza, S.Kom. selaku Dosen Pengampu Mata Kuliah Data Analyst and sebagai pembimbing yang memberikan arahan dan bimbingan agar laporan ini bisa selesai sebagaimana mestinya.
3. Teman Teman jurusan Sistem Informasi yang memberikan dukungan dan semangat kepada kami sehingga laporan ini bisa selesai dengan baik.

Semoga laporan ini dapat bermanfaat untuk keperluan proyek penulis, pembaca, dan calon peneliti selanjutnya yang akan melakukan penelitian dalam bidang yang serupa. Kami juga meminta maaf sebesar besarnya apabila ada kesalahan dalam penulisan, pengejaan, sistemasi EYD dalam pembuatan laporan ini. Kami juga terbuka terhadap saran dan kritik sehingga bisa menjadi masukan bagi kami dalam meningkatkan penulisan proyek ini kedepannya agar lebih baik.

Tangerang, 20 Desember 2022

Penulis

DAFTAR ISI

Kata Pengantar.....	i
DAFTAR ISI.....	ii
BAB I.....	1
1.1 Latar Belakang.....	1
1.3 Tujuan.....	1
BAB II.....	2
2.1 Definisi Machine Learning.....	2-3
2.2 Definisi DBScan.....	3
2.3 Metode Algoritma.....	3-4
BAB III.....	5-7
BAB IV.....	8
4.1 Visualisasi(Distribusi, dkk, boxplot).....	8-12
4.1.1 Bivariate Analysis.....	12-13
4.2 Pembahasan (DBSCAN).....	14-17
4.3 Kesimpulan.....	17-18
4.4 Saran.....	18
DAFTAR PUSTAKA.....	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manajemen dan pemeliharaan hubungan pelanggan selalu memainkan peran penting dalam memberikan intelijen bisnis kepada organisasi untuk membangun, mengelola, dan mengembangkan hubungan pelanggan jangka panjang yang berharga. Pentingnya memperlakukan pelanggan sebagai aset utama organisasi semakin meningkat nilainya di era sekarang ini. Organisasi - organisasi tertarik untuk berinvestasi dalam pengembangan strategi akuisisi, pemeliharaan, dan pengembangan pelanggan. Kecerdasan bisnis memiliki peran penting dalam memungkinkan perusahaan menggunakan keahlian teknis untuk lebih memahami pelanggan yang lebih baik dan Program untuk penjangkauan. Segmentasi pelanggan dilakukan menggunakan teknik *Clustering* seperti DBscan orang - orang yang memiliki minat yang sama dapat dikelompokkan. Dengan adanya segmentasi pelanggan, dapat membantu tim pemasaran untuk mengenali dan memaparkan berbagai segmen pelanggan yang memiliki pemikiran yang berbeda serta mengikuti strategi pembelian yang berbeda - beda.

1.2 Rumusan Masalah

- Bagaimanakah penerapan DBscan pada dataset mall customer ?

1.3 Tujuan

- Tujuan utama melakukan segmentasi pelanggan adalah untuk melakukan pengelompokan terhadap orang-orang yang memiliki minat yang sama sehingga tim pemasaran dapat membentuk perencanaan pemasaran yang efektif.

BAB II

LANDASAN TEORI

2.1 Definisi Machine Learning

Machine Learning adalah sebuah mesin pembelajaran yang dirancang untuk belajar secara otomatis tanpa instruksi pengguna. Pembelajaran mesin ini didasarkan pada matematika, statistik, penambangan data(mining), dan ilmu lainnya. Bergantung pada kebutuhan keahlian si data analyst sendiri.

Contoh penggunaan Machine Learning adalah:

1. Rekomendasi lagu di spotify berdasarkan kesukaan si pengguna
2. Sugesti untuk menambah teman yang sesuai dengan jaringan si pengguna(Instagram , Facebook, Twitter, dll)
3. Chat Bot yang sering ditemukan di dalam sebuah aplikasi forum kesehatan, toko jual beli online dan lain lain.
4. Security checkup pada sebuah aplikasi ataupun web based untuk segi keamanan

Metode Machine Learning dibagi menjadi 2 teknik yaitu:

- **Supervised Learning**

Supervised learning merupakan pembelajaran yang diawasi dilakukan dengan memberi nama kumpulan data yang digunakan oleh pembelajaran mesin dan mengklasifikasikannya oleh pengembang untuk memungkinkan algoritme mengenali tingkat akurasi kinerja. Dalam metode ini, pembelajaran mesin dipantau pada data berlabel, yang kemudian dipelajari oleh pembelajaran mesin tentang hubungan dan ketergantungan antar data.

- **Unsupervised Learning**

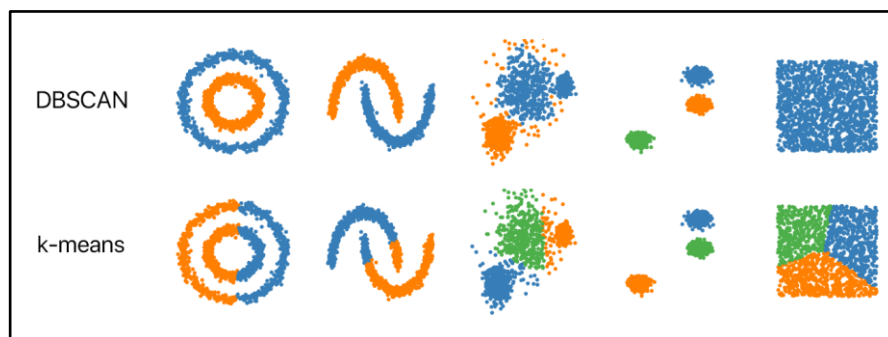
Unsupervised adalah teknik pembelajaran mesin tanpa pengawasan(security). Oleh karena itu, proses dilakukan pada kumpulan data mentah yang tidak berlabel, dan algoritma pembelajaran mesin mencoba mengidentifikasi pola dan hubungan antar data tanpa bantuan pemrogram.

Secara umum, unsupervised learning tidak memerlukan bantuan manusia, sehingga komputer benar-benar mempelajari data dan hubungannya sendiri. Dalam hal ini, kumpulan data tidak diberi nama dan mesin secara komputasi mengidentifikasi pola dalam data. Pembelajaran tanpa pengawasan digunakan untuk memudahkan pengembang membuat keputusan.

2.2 Definisi DBScan

Density Based Scan adalah sebuah metode pengolahan data dengan cara *clustering* berbasis density atau *density based*. Metode clustering merupakan suatu proses yang bertujuan untuk membagi data menjadi kelompok-kelompok tertentu berdasarkan kategori atau tingkat kesamaan data. DBSCAN juga merupakan teknik ML dengan unsupervised machine learning.

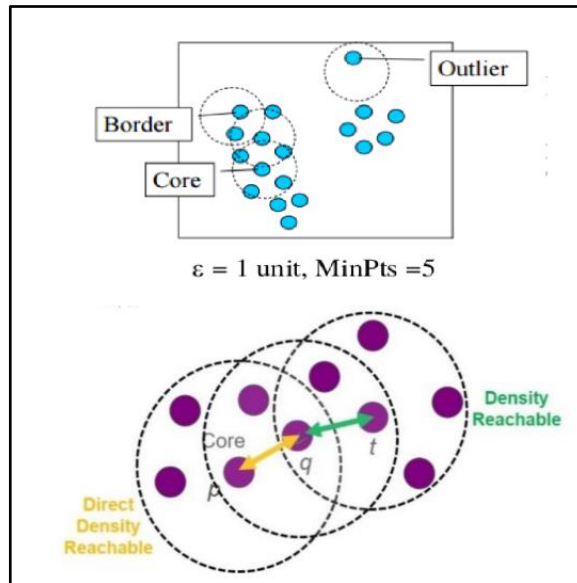
Metode ini sering digunakan untuk menghitung atau mengolah data yang membutuhkan pengelompokan data secara detail. Beberapa contoh data yang dikelola dengan data ini antara lain analisis *data sumber api* dan *analisis titik bencana pada suatu wilayah*.



2.3 Metode Algoritma

Seperti yang sudah diketahui pada penjelasan di atas. Metode DBscan ini menggunakan algoritma density based. Dan ada beberapa parameter yang patut dipertimbangkan yaitu *epsilon* (eps) dan *minimum points* (minPts). Epsilon adalah jarak maksimal antara dua data dalam satu cluster. Sedangkan minimum points adalah banyak data minimal pada jarak epsilon agar terbentuk menjadi cluster. Berikut parameter lain yang harus diperhatikan dalam menggunakan metode Dbscan:

1. Directly density-reachable
2. Density -reachable
3. Core point
4. Border point
5. Outlier

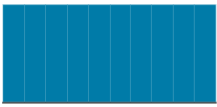

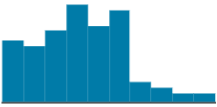
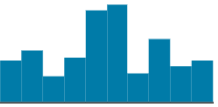


Cara kerja DBScan:

1. Tentukan nilai minimal points serta epsilon yang akan digunakan
2. Pilih data awal “p” secara random
3. Hitung jarak antara data “p” terhadap semua data menggunakan distance Euclidean.
4. Ambil semua pengamatan dengan density-reachable dengan amatan “p”
5. Apabila pengamatan memenuhi nilai epsilon lebih dari jumlah minimal pengamatan dalam 1 group maka pengamatan “p” dikategorikan sebagai *core points* dan group terbentuk
6. Apabila pengamatan “p” adalah border points dan tidak ada amatan yang densitasnya reachable dengan pengamatan “p”, maka bisa dilanjutkan ke pengamatan lainnya
7. Algoritma dbscan akan terus berulang sampai semua titik sudah dikunjungi,(titik q,t ,p)

BAB III

PENGUMPULAN & PENGOLAHAN DATA

CustomerID	Genre		# Age	# Annual Income (k\$)	# Spending Score (...)
	Female Male	56% 44%			
0001	Male		19	15	39
0002	Male		21	15	81
0003	Female		20	16	6
0004	Female		23	16	77
0005	Female		31	17	40
0006	Female		22	17	76
0007	Female		35	18	6
0008	Female		23	18	94
0009	Male		64	19	3
0010	Female		30	19	72
0011	Male		67	19	14

Data yang kami gunakan berasal dari website **kaggle.com** . Data mall customers yang terdiri dari 5 kolom dan 200 baris yang dibagi menjadi id customer (CustomerID) dari 0001 sampai 0200, jenis kelamin (Genre) Male dan Female, umur (Age) dengan range usia 18-70 tahun, pendapatan tahunan (Annual Income) dalam range 15-137\$, dan skor pengeluaran (Spending score) dari 1 sampai 100.

```
In [1]: import pandas as pd
from pylab import rcParams
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

import datetime as dt
import sklearn
from sklearn.cluster import DBSCAN
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from collections import Counter
```


In [2]: `#baca datasetsnya`

```
malls_df = pd.read_csv(r"D:\SEMESTER3\DATAANAYISIS\tugas kelompok\Mall_Customers.
malls_df.head(100)
```

Out[2]:

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...
95	96	Male	24	60	52
96	97	Female	47	60	47
97	98	Female	27	60	50
98	99	Male	48	61	42
99	100	Male	20	61	49

100 rows x 5 columns

In [3]: `malls_df.tail()`

Out[3]:

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

Gambar diatas merupakan Code untuk Membaca dataset

```
In [4]: #kita harus check apakah ada values null di datanya
mall$df.isnull().sum()
```

```
Out[4]: CustomerID      0
Gender      0
Age         0
Annual Income (k$)    0
Spending Score (1-100) 0
dtype: int64
```

```
In [5]: mall$df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CustomerID            200 non-null   int64
1   Gender                200 non-null   object
2   Age                  200 non-null   int64
3   Annual Income (k$)    200 non-null   int64
4   Spending Score (1-100) 200 non-null   int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

```
In [6]: mall$df.describe()
```

```
Out[6]:
```

	CustomerID	Age	Annual Income (k\$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
In [7]: mall$df.shape
```

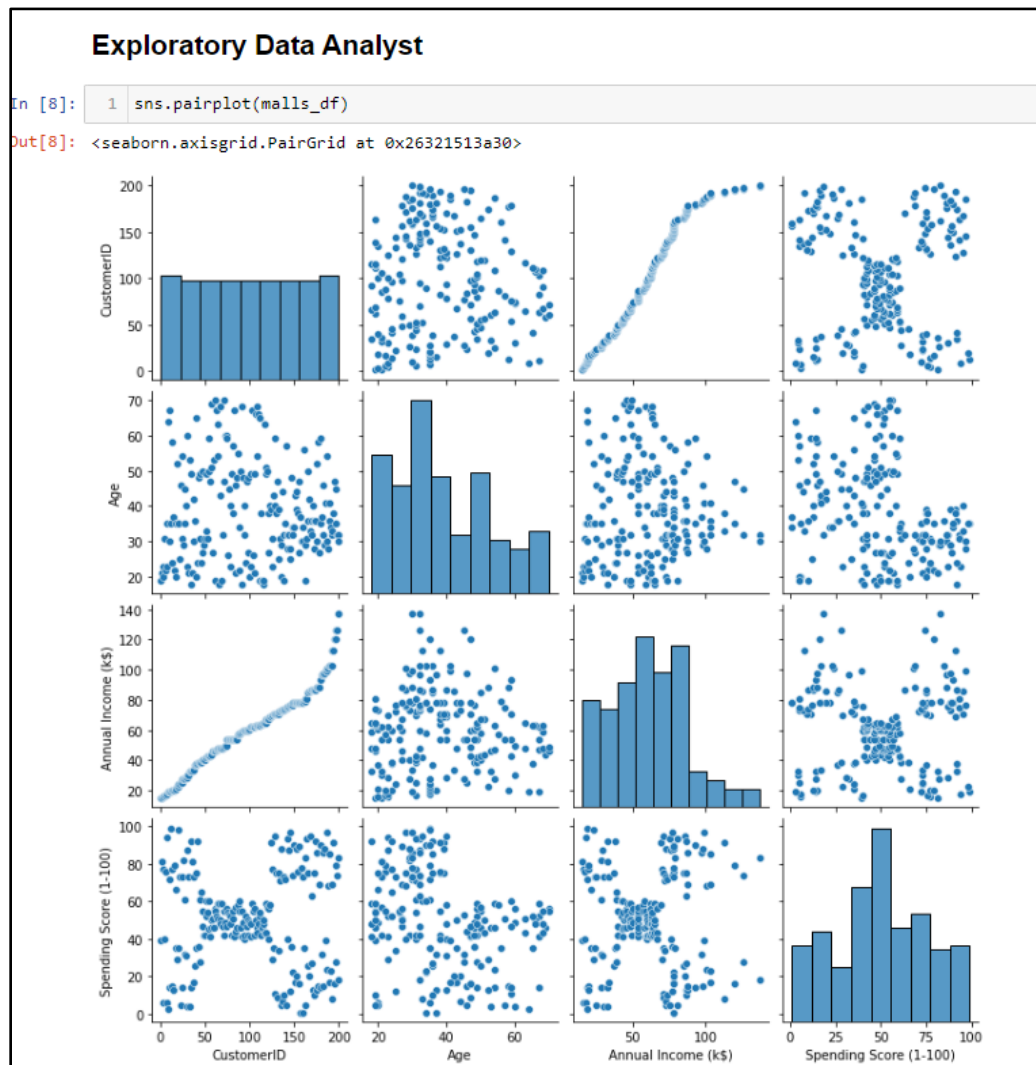
```
Out[7]: (200, 5)
```

Gambar diatas merupakan code yang digunakan untuk mencari data yang memiliki values yang null

BAB IV

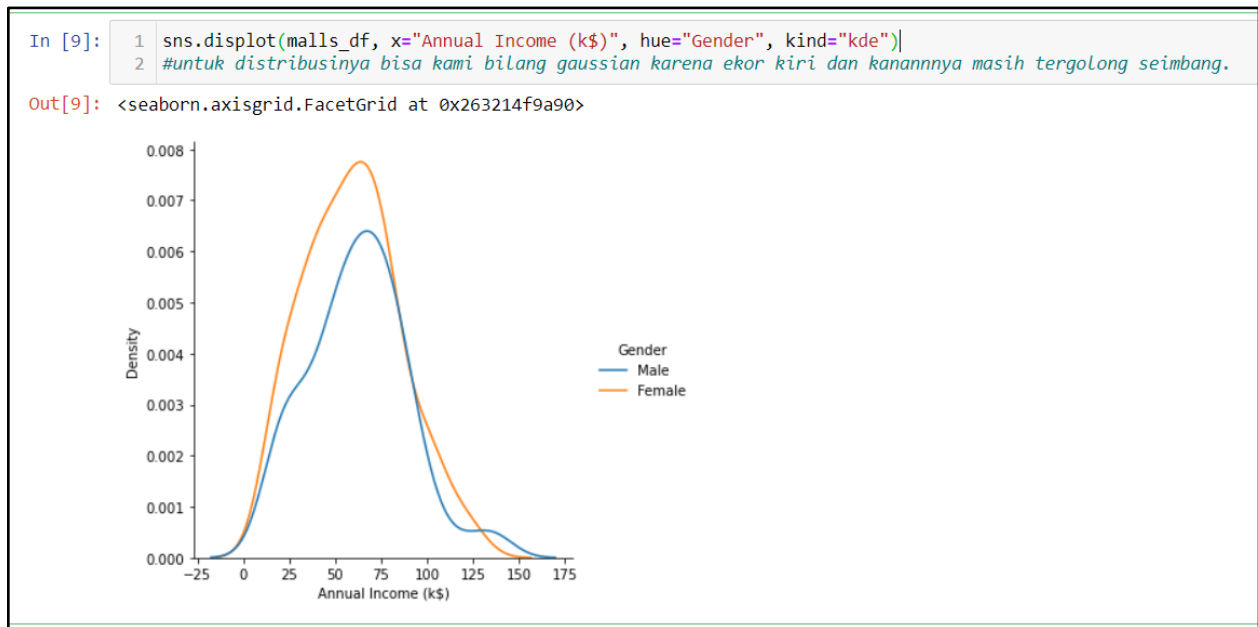
HASIL & PEMBAHASAN

4.1 Visualisasi(Distribusi, dkk, boxplot)



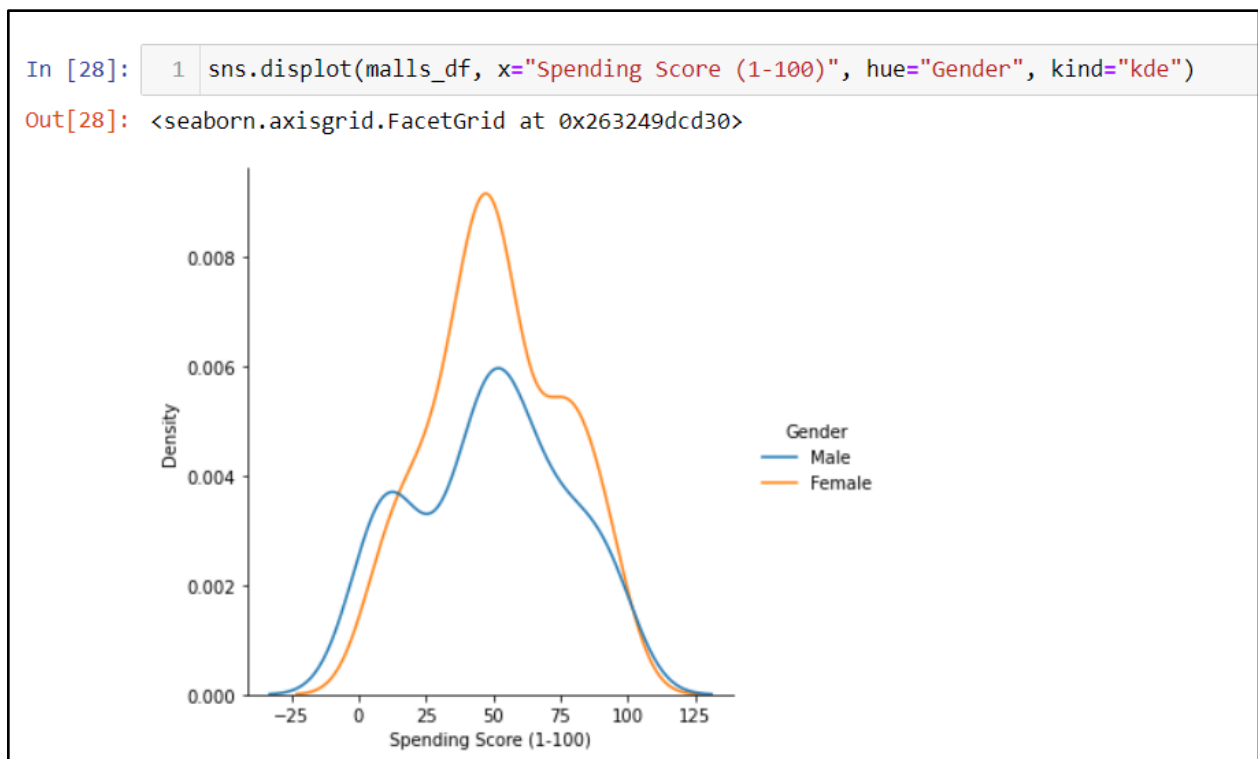
Gambar 4.1 Pairplot Mall Customer Data

Gambar diatas merupakan sns seaborn untuk merangkap semua data didalam 1 gambar penuh dimulai dari scatter plot hingga histogram.



Gambar 4.2 Displot antara Annual Income dengan Gender

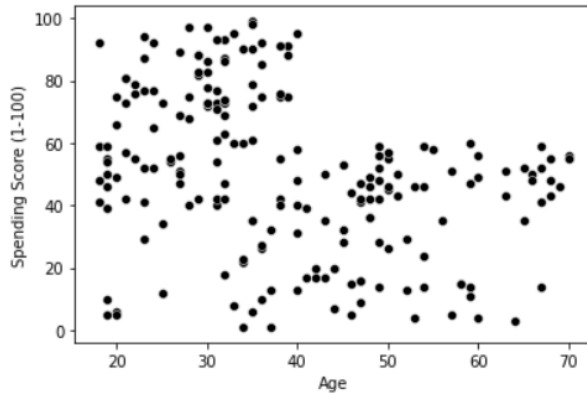
Distribusinya untuk annual income dengan gender bisa kita bilang agak gaussian(datanya normal) karena ekor berbentuk oval.



Gambar 4.3 Displot Annual Spending Score dengan Gender

```
In [11]: 1 sns.scatterplot(data=malls_df, x="Age", y="Spending Score (1-100)", color="black")
         2 #merupakan scatter plot antara umur dengan score spending
```

```
Out[11]: <AxesSubplot:xlabel='Age', ylabel='Spending Score (1-100)'\>
```

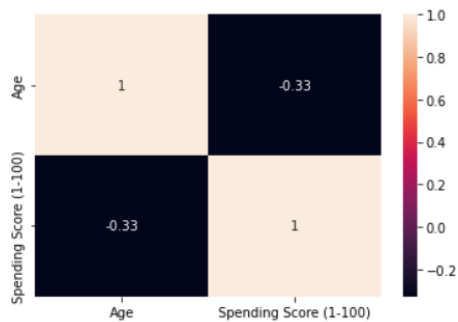


Gambar 4.4 Scatter plot antara *Age* dengan *Spending Score*

Kesimpulannya adalah data tersebut tingkat kerapatannya rendah, beserta tidak ada relationship karena menyebar kemana mana, dan arahnya juga tidak jelas antara positif maupun negatif.

```
In [12]: 1 sns.heatmap(malls_df[['Age', 'Gender', 'Spending Score (1-100)']].corr(), annot=True)
         2 #kesimpulannya adalah bahwa
```

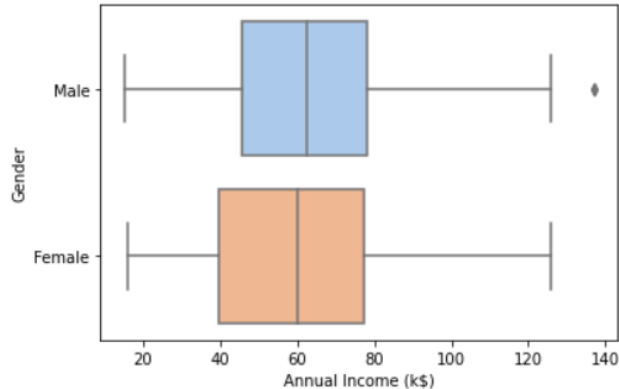
```
Out[12]: <AxesSubplot:\>
```



Gambar 4.5 Heatmap antara korelasi *Age*, *Gender*, dengan *Spending Score*.

```
In [13]: 1 colors = sns.color_palette('pastel')
2 sns.boxplot(x="Annual Income (k$)", y="Gender", data=malls_df, palette=colors)
3 #terdapat 1 outlier pada Gender Male yang berarti ada noise didalamnya
```

```
Out[13]: <AxesSubplot:xlabel='Annual Income (k$)', ylabel='Gender'>
```

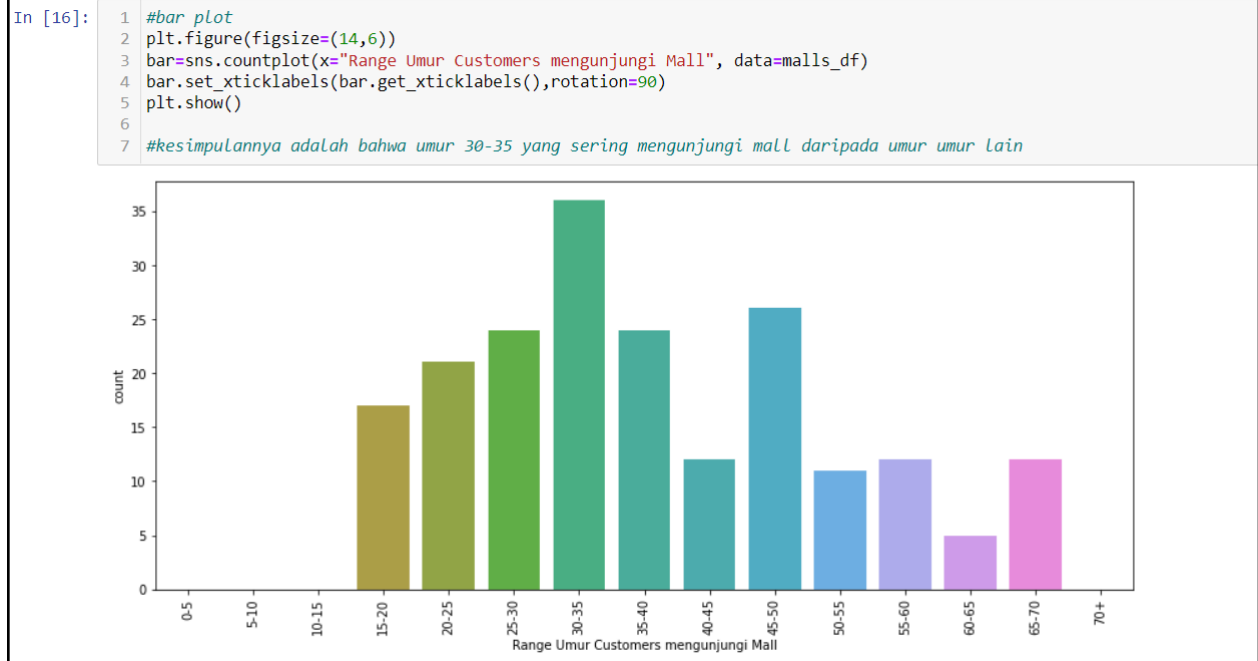


Gambar 4.6 Boxplot antara Annual Income dengan Gender

```
In [15]: 1 #binning umur dari para customers
2 #binning merupakan proses untuk mengelompokkan data menjadi bagian yang lebih kecil, dalam kasus ini kita ingin melihat jangk
3
4 binning = [0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75]
5 range_ = ['0-5', '5-10', '10-15', '15-20', '20-25', '25-30', '30-35',
6           '35-40', '40-45', '45-50', '50-55', '55-60', '60-65', '65-70', '70+']
7
8 malls_df['Range Umur Customers mengunjungi Mall']=pd.cut(malls_df['Age'], binning, labels=range_)
9
```

Binning merupakan proses untuk mengelompokkan data ke dalam bagian-bagian yang lebih kecil yang disebut *bin* berdasarkan kriteria tertentu. Juga merupakan salah satu teknik pra proses data yang akan digunakan untuk meminimalisir kesalahan dalam pengamatan serta terkadang dapat meningkatkan akurasi dari model prediktif.

Binning biasanya digunakan untuk mengelompokkan **data numerik** menjadi beberapa bin agar sebaran data lebih mudah dipahami. Contohnya mengelompokkan usia.



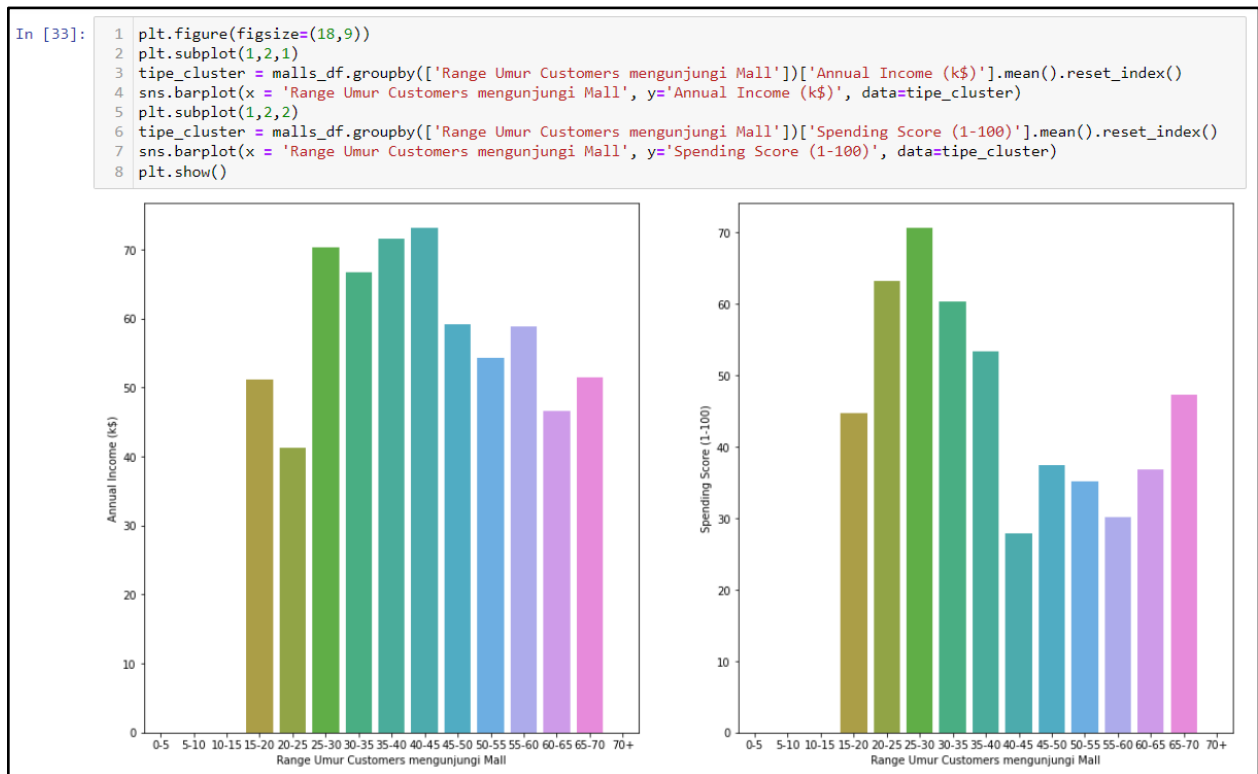
Gambar 4.7 Histogram range umur customer di Mall

Dari data diatas range umur 30-35 adalah pengunjung yang paling sering mengunjungi mall.

4.1.1 Bivariate Analysis



Gambar 4.8 Diagram Batang



Gambar 4.9 Perbandingan histogram Annual Income dengan Spending Score

4.2 Pembahasan (DBSCAN)

DBSCAN clustering untuk identifikasi outliers

train model kita dan identifikasi outliers ¶

```
In [19]: 1 ### DBSCAN clustering untuk identifikasi outliers
2 ##### train model kita dan identifikasi outliers
3
4 df = malls_df.iloc[:, [3,4]].values
```

```
In [20]: 1 df
```



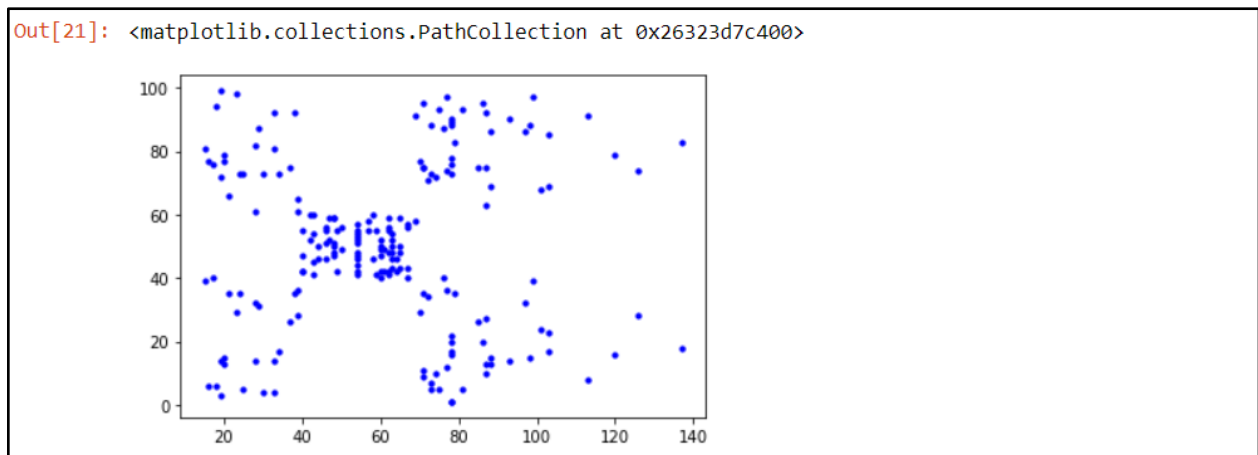
```
Out[35]: array([[ 15, 39],
 [ 15, 81],
 [ 16, 6],
 [ 16, 77],
 [ 17, 40],
 [ 17, 76],
 [ 18, 6],
 [ 18, 94],
 [ 19, 3],
 [ 19, 72],
 [ 19, 14],
 [ 19, 99],
 [ 20, 15],
 [ 20, 77],
 [ 20, 13],
 [ 20, 79],
 [ 21, 35],
 [ 21, 66],
 [ 23, 29],
 [ 23, 22])
```

Inisialisasi DBscan pertama tama kita membuat variable `df` , lalu memasukkan data frame sebelumnya yakni **`malls_df`**. Setelah itu tambahkan code `.iloc[:, [3,4]]` yang artinya semua data array akan di list pada code dibawah ini. Dimulai dari column yang pertama.

```
In [21]: 1 plt.scatter(df[:,0], df[:,1], s=10, c="blue")
          2 #terlihat banyak sekali data yang tersebar dan kita akan lanjut ke proses kmean

Out[21]: <matplotlib.collections.PathCollection at 0x26323d7c400>
```

Kemudian kita tambahkan visualisasi yakni scatter plot untuk melihat seberapa besar cluster yang ada pada mall customer segmentation data. Kita menggunakan warna biru, dan bisa diganti dengan warna bebas lainnya.



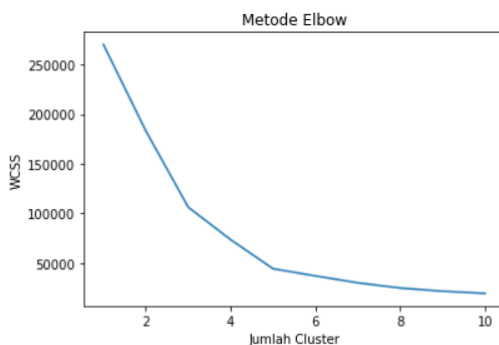
Sebenarnya K Means tidak diperlukan dalam metode dbscan ,namun kita tetap memasukan karena ada WCSS. pada code dibawah ini menggunakan for untuk pengulangan agar di run sebanyak 10 kali. Dan menerapkan code KMeans lalu kita juga membuat sebuah plot agar bisa divisualisasikan dengan benar. Nah pada code ini nantinya akan menggambarkan elbow method yang merupakan

sebuah menentukan jumlah cluster yang akan digunakan pada k-means clustering. Hasilnya akan ada dibawah ini.

```
In [22]: 1 #Kmeans
2 #WCSS merupakan jarak pertambahan squared antara tiap point dan ditengah cluster.nah ketika kita plot WCSS dengan K-value ma
3 #kalau jumlah cluster meningkat maka nilai WCSS akan berkurang. Nilai WCSS paling besar ketika K = 1.
4 wcss = []
5 #ngejalanin jumlah cluster sebanyak 10 kali, dan akan run ke 10kalinya
6 for i in range(1,11):
7     kmeans = KMeans(n_clusters=i,
8                     init = 'k-means++', max_iter=300, n_init=10) #run max 300 tidak bisa lebih
9     kmeans.fit(df)
10    wcss.append(kmeans.inertia_)
11
12 plt.plot(range(1,11), wcss)
13 plt.title("Metode Elbow")
14 plt.xlabel("Jumlah Cluster")
15 plt.ylabel("WCSS")
16 plt.show()
17
18
19 #kesimpulan nilai K kita sekarang adalah 5, dan merupakan angka optimal cluster.
```

```
19 #kesimpulan nilai K kita sekarang adalah 5, dan merupakan angka optimal cluster.
```

```
C:\Users\Darren\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OM
P_NUM_THREADS=1.
  warnings.warn(
```



Elbow method sendiri merupakan metode untuk menentukan jumlah cluster yang tepat melalui persentase hasil perbandingan antara jumlah cluster yang akan membentuk siku pada suatu titik.

Tujuan K-Means pada DBScan:

Membantu dalam pembuatan elbow method untuk DBSCAN sehingga bisa menentukan siku pada 1 titik yang palis pas(dlm hal ini cluster optimal).

```

In [23]: 1 #minimum_samples merupakan sample minimum yang perlu terlihat di dalam lingkaran cluster,dalam hal ini minsamplesnya adalah
          2 #epsilon merupakan radius objek
          3 model = DBSCAN(eps=5, min_samples=5)

In [24]: 1 labels= model.fit_predict(df)

```

Selanjutnya kita masuk ke proses DBScan untuk menemukan samples pada cluster yang sudah dijalankan pada code sebelumnya. Kita sekarang berada di numpy np.unique labels, untuk melihat array yang ada noise atau tidak karena di dataset mall customers segmentation ada noise maka kita mengambil angka -1 sebagai noisenya.

```

In [25]: 1 np.unique(labels)
          2 #-1 bukan cluster, dihasil ini terdapat sebuah noise yakni -1
          3 #angka dibawah merupakan hasil cluster yang di run

Out[25]: array([-1,  0,  1,  2,  3,  4], dtype=int64)

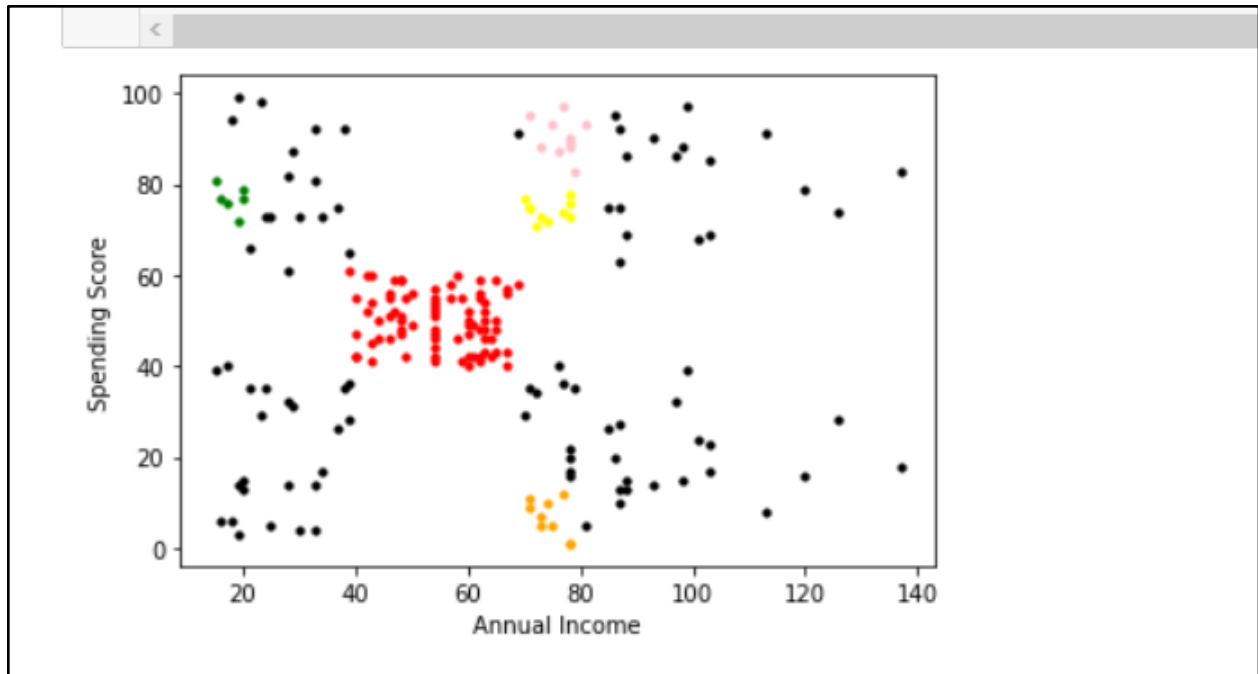
```

Langkah terakhir kita memodifikasi cluster yang sudah dibuat sebelumnya pada bagian atas dengan visualisasi warna agar kelihatan jelas yang mana density tinggi maupun rendah.

```

In [26]: 1 #Visualisasi clusters
          2 plt.scatter(df[labels==0], df[labels==1],s=10, c="black")
          3
          4 plt.scatter(df[labels==0], df[labels==1],s=10, c="green")
          5 plt.scatter(df[labels==1], df[labels==1],s=10, c="red")
          6 plt.scatter(df[labels==2], df[labels==2],s=10, c="yellow")
          7 plt.scatter(df[labels==3], df[labels==3],s=10, c="orange")
          8 plt.scatter(df[labels==4], df[labels==4],s=10, c="pink")
          9 plt.scatter(df[labels==5], df[labels==5],s=10, c="blue")
         10 plt.scatter(df[labels==6], df[labels==6],s=10, c="brown")
         11
         12 plt.xlabel("Annual Income")
         13 plt.ylabel("Spending Score")
         14 plt.show()
         15
         16
         17 #kesimpulan dari clusters ini adalah
         18 #Cluster dengan warna merah merupakan kerapatannya tinggi.
         19 #Cluster hitam bisa dibilang sebagai noise karena tingkat kerapatannya paling rendah dari antara yang lain.
         20 #ciri khas DBSCAN adalah objek sebuah cluster harus lebih rapat daripada cluster dalam 1 area dalam hal ini yang dimaksud ad

```



Terlihat bahwa cluster merah merupakan cluster yang paling rapat densitasnya. Ciri khas dbscan yakni sebuah objek cluster harus lebih rapat daripada cluster dalam 1 area. Dan noise paling keliatan di data ini adalah cluster berwarna hitam(yang mempunyai kerapatan data yang rendah).

4.3 Kesimpulan

Kesimpulan dari proyek ini dengan menerapkan algoritma DBscan adalah

1. DBscan bisa membantu para data analyst untuk mendeteksi adanya noise atau outlier dalam mengelompokkan sebuah data berdasarkan tingkat kepadatan data sehingga memunculkan cluster dengan tingkat kepadatan yang tinggi maupun yang rendah. Kemudian DBScan membutuhkan 2 buah parameter untuk melakukan clustering yakni *eps* dan MinPts(Minimum points). Dalam mall customer kasus ini si data analyst bisa tahu cluster cluster mana saja pengunjung dengan umur, serta spending score serta annual spending yang digambarkan pada cluster cluster di atas.

4.4 Saran

Semoga dengan adanya proyek ini bisa membuka wawasan bagi para calon data analyst dalam melakukan clustering data menggunakan dbscan method. Penulis berharap juga untuk para pembaca yang membaca laporan ini bisa menambah wawasan dalam

pengenalan DBscan dalam clustering sebuah data serta dapat mengimplementasikan terhadap data lain menggunakan DBscan.

DAFTAR PUSTAKA

Raharja, A. D. B. (n.d.). *Machine learning: Pengertian, Cara Kerja, Dan 3 Metodenya!* TALENTS. Retrieved December 8, 2022, from <https://www.ekrut.com/media/apa-itu-machine-learning>

ArcGIS, K. D. P. &. (2022, February 8). *Mengenal algoritma DBSCAN Dan Manfaatnya*. Konsultan Analis Data Penelitian dan Peta Digital. Retrieved December 8, 2022, from <https://patrastatistika.com/mengenal-algoritma-dbscan-dan-manfaatnya/>

Awan, A. A. (2022, August 17). *Implementing DBSCAN in python*. KDnuggets. Retrieved December 8, 2022, from <https://www.kdnuggets.com/2022/08/implementing-dbscan-python.html>

Jagadish, K. (2019, April 29). *Mall customers*. Kaggle. Retrieved December 8, 2022, from <https://www.kaggle.com/datasets/kandij/mall-customers>

Link gambar online

https://miro.medium.com/max/1339/0*xu3GYMsWu9QiKNOo.png