PRESENTATION TEMPLATE

PORTFOLIO DATA ANALYST PROJECT DEPARTMENT HR

NOVITA YOLANDA BARUS
Semarang - Indonesia

RESUME

HELLO!

NOVITA YOLANDA BARUS
MATHEMATIC STUDENT

My name is Novita Yolanda Barus, a final-year student of mathematics at the University of Diponegoro. I have interests in data analytics, data science, and statistics.

Through this portfolio, I would like to share some case studies that I had worked on a while when I was a mentee in a program called Data, Business Analytics & Operations in Ruangguru

Tools:



www.reallygreatsite.com

OUTLINE

Employment development is one of the important responsibility of the HR department. They are responsible for managing employees such as things like attendance of employees, their late comings, gets are totally maintained by HR department

What

Segmenting types of employees into several group according to their performance such as length service, absent hours, age, and distribution ID

Why

Have an accurate anaysis to assess employees performance matters to help increasing quality of employees per distribution centers

Who

Employees of The Look Company

When

Where

The Look Company

How

Analyze the given data by preparing, cleaning, visualizing, and developing data with clustering to see the segmentation among employees

DATA INTRODUCTION

Import Data **Employee** into PGadmin

```
CREATE TABLE public.data_employee
(
        fisrt_name varchar,
        last_name varchar,
        gender varchar,
        age        float,
        length_service float,
        absent_hours float,
        distribution_centers_id int
```

Data Output | Explain | Messages | Notifications

	fisrt_name character varying	last_name character varying	gender character varying	age double precision	length_service double precision	absent_hours double precision	distribution_centers_id integer
1	Gutierrez	Molly	F	32.02881569	6.018478474	36.57730606	5
2	Hardwick	Stephen	M	40.32090167	5.532444578	30.16507231	9
3	Delgado	Chester	M	48.82204661	4.389973118	83.80779766	10
4	Simon	Irene	F	44.59935722	3.081735738	70.02016505	2
5	Delvalle	Edward	M	35.69787561	3.619091448	0	4
6	Jones	Ernie	M	48.44031059	2.717692452	81.83007916	6
7	Buford	Ralph	M	50.75273	10.157918	60.49507152	6
8	Lee	Gregory	M	36.2160312	4.432122862	30.07290192	10
9	Smith	Jerry	M	58.42738025	6.940120524	181.630819	4
10	Beard	Robert	M	39.85398	13.848321	30.66440832	10
11	Mathis	John	M	46.54758052	4.872038321	28.01835332	3
12	Barajas	John	M	15	3.793042148	0	5
13	Leonard	James	M	37.72801116	3.621141838	0	6
14	Davis	Janet	F	30.78519091	4.583328091	34.33444296	1

Import Data **distribution_centers** into PGadmin

```
CREATE TABLE public.distribution_centers
(
        id int PRIMARY KEY,
        name varchar,
        latitude float,
        longitude float
);
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	name character varying	latitude double precision	longitude double precision
1	1	Memphis TN	35.1174	-89.9711
2	2	Chicago IL	41.8369	-87.6847
3	3	Houston TX	29.7604	-95.3698
4	4	Los Angeles CA	34.05	-118.25
5	5	New Orleans LA	29.95	-90.0667
6	6	Port Authority of New York/New Jersey NY/NJ	40.634	-73.7834
7	7	Philadelphia PA	39.95	-75.1667
8	8	Mobile AL	30.6944	-88.0431
9	9	Charleston SC	32.7833	-79.9333
10	10	Savannah GA	32.0167	-81.1167

DATA INTRODUCTION

Join Data Employee and distribution_centers

```
SELECT * FROM
data_employee
JOIN
distribution_centers
ON
data_employee.distribution_centers_id =
distribution_centers.id
```

	fisrt_name character varying	last_name character varying	gender character va	age double precision	length_service double precision	absent_hours double precision	distribution_ce integer	id integer	name character varying	latitude double precision	longitude double precision
1	Gutierrez	Molly	F	32.02881569	6.018478474	36.57730606	5	5	New Orleans LA	29.95	-90.0667
2	Hardwick	Stephen	M	40.32090167	5.532444578	30.16507231	9	9	Charleston SC	32.7833	-79.9333
3	Delgado	Chester	M	48.82204661	4.389973118	83.80779766	10	10	Savannah GA	32.0167	-81.1167
4	Simon	Irene	F	44.59935722	3.081735738	70.02016505	2	2	Chicago IL	41.8369	-87.6847
5	Delvalle	Edward	M	35.69787561	3.619091448	0	4	4	Los Angeles CA	34.05	-118.25
6	Jones	Ernie	M	48.44031059	2.717692452	81.83007916	6	6	Port Authority of New...	40.634	-73.7834
7	Buford	Ralph	M	50.75273	10.157918	60.49507152	6	6	Port Authority of New...	40.634	-73.7834
8	Lee	Gregory	M	36.2160312	4.432122862	30.07290192	10	10	Savannah GA	32.0167	-81.1167
9	Smith	Jerry	M	58.42738025	6.940120524	181.630819	4	4	Los Angeles CA	34.05	-118.25
10	Beard	Robert	M	39.85398	13.848321	30.66440832	10	10	Savannah GA	32.0167	-81.1167
11	Mathis	John	M	46.54758052	4.872038321	28.01835332	3	3	Houston TX	29.7604	-95.3698
12	Baraias	John	M	15	3.793042148	0	5	5	New Orleans LA	29.95	-90.0667

DATA PREPARATION

Import joined data into Python

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import time
     import pprint
```

```
[ ]  # Memanggil data Human Resources yang sudah disimpan dengan nama df_hr
     df_hr = pd.read_csv('Data Join Human Resource.csv')
```

df_hr

	fisrt_name	last_name	gender	age	length_service	absent_hours	distribution_centers_id	id	name	latitude	longitude
0	Gutierrez	Molly	F	32.028816	6.018478	36.577306	5	5	New Orleans LA	29.9500	-90.0667
1	Hardwick	Stephen	M	40.320902	5.532445	30.165072	9	9	Charleston SC	32.7833	-79.9333
2	Delgado	Chester	M	48.822047	4.389973	83.807798	10	10	Savannah GA	32.0167	-81.1167
3	Simon	Irene	F	44.599357	3.081736	70.020165	2	2	Chicago IL	41.8369	-87.6847
4	Delvalle	Edward	M	35.697876	3.619091	0.000000	4	4	Los Angeles CA	34.0500	-118.2500
...	...	...	...	...	...	...	...	...	...	...	...
8331	Coniglio	Bianca	F	46.057544	4.838288	93.665111	9	9	Charleston SC	32.7833	-79.9333
8332	Cox	Jimmie	M	34.455490	2.427274	0.000000	1	1	Memphis TN	35.1174	-89.9711
8333	Hawkins	Mary	F	58.347160	4.009393	176.356940	9	9	Charleston SC	32.7833	-79.9333
8334	Proctor	Theresa	F	43.340616	6.154837	60.321917	7	7	Philadelphia PA	39.9500	-75.1667
8335	Salter	Charles	M	46.192782	5.174722	112.023389	2	2	Chicago IL	41.8369	-87.6847

8336 rows × 11 columns

Variable Explanation

Variable	Description
first_name	First name of employee
last_name	Last name of employee
gender	Employee gender
absent_hours	Total absent hour employee in a year
length_service	Total length service employee
distribution_centers_id	Number of distribution centers The Look Cooperation

DATA CLEANING

Data cleaning is important to do to detect and correct corrupt or inaccurate records from a record set, table, or database refers to identifying, replacing, modifying, or deleting the dirty or coarse data.

Modify variable's name

```
# First_Name dan Last_Name akan digabungkan sehingga menjadi Full_Name dengan fungsi berikut
cols=['fisrt_name', 'last_name']
df_hr['full_name']=
df_hr[cols].apply(lambda row: ' '.join(row.values.astype(str)), axis=1)
```

```
[ ] first_column = df_hr.pop('full_name')
    df_hr.insert(0, 'full_name', first_column)
```

Encoded variable gender

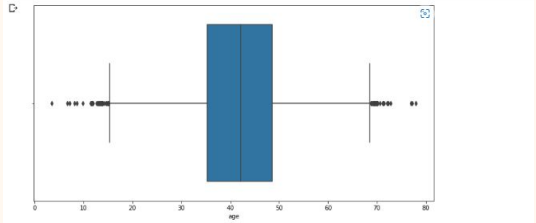
```
[ ] from sklearn.preprocessing import LabelEncoder
    le = LabelEncoder()
    gender_encoded = le.fit_transform(df_hr['gender'])
    print(gender_encoded)
```

```
[0 1 1 ... 0 0 1]
```

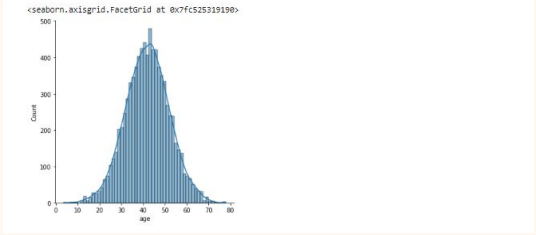
Handling outlier variable Length_service, Absent_Hours, Age

Data Age

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='age', data=df_hr)
plt.show()
```



```
sns.displot(df_hr['age'],kde=True)
```

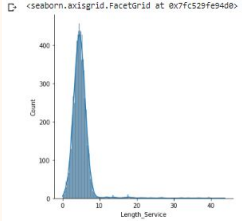


```
df_hr['age'].skew()
```

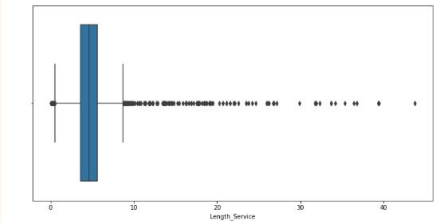
```
-0.019946536896838327
```

Data Length_Service

```
sns.displot(df_hr['Length_Service'],kde=True)
```



```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Length_Service', data=df_hr)
plt.show()
```

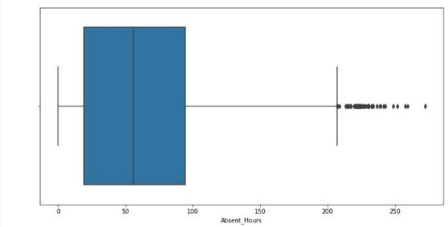


```
df_hr['length_service'].skew()
```

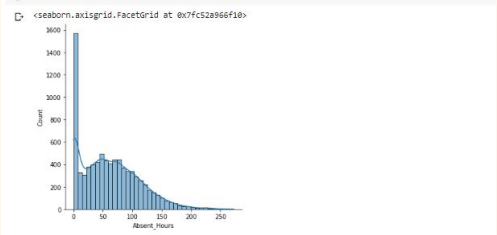
```
5.783156830066989
```

Data Absent_Hours

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Absent_Hours', data=df_hr)
plt.show()
```



```
sns.displot(df_hr['Absent_Hours'],kde=True)
```



```
df_hr['absent_hours'].skew()
```

```
0.6422190359788191
```

Note: Data Length_Service and Absent Hours have **right skewness** greater than 0.5.
Thus, data are not **normal distribution**.

Handling Outlier

Imputing Outlier Length_Service

1

```
def batas_atas(df_hr):
  Q3 = np.quantile(df_hr['Length_Service'], 0.75)
  Q1 = np.quantile(df_hr['Length_Service'], 0.25)
  IQR = Q3 - Q1

  lower_boundaries = Q1 - 1.5 * IQR
  upper_boundaries = Q3 + 1.5 * IQR

  print(f'lower boundaries = {lower_boundaries}')
  print(f'upper boundaries = {upper_boundaries}')
```

```
print('base model :')
batas_atas(df_hr)
```

```
base model :
lower boundaries = 0.5038469635
upper boundaries = 8.695966843499999
```

2

```
[62] def imput3_outlier(df_hr, upper_bound):
       ## kode untuk mencetak outlier dan melihat jumlah outlier yang ada
       outliers = np.where(df_hr.Length_Service > upper_bound)
       print(f'sum of outlier : {np.count_nonzero(df_hr.Length_Service > upper_bound)}')
       print(f'outlier : \n{outliers[0]}\n')

        ## loading
       proses='=============\nProses membersihkan outliers '
       loading='. . . .'
       def message(proses, loading):
         print(proses)
         for i in loading:
           print(i, end='')
           time.sleep(0.5)

         if __name__ == '__main__':
           message(proses, loading)

       ## kode untuk impute outlier
       df_hr.loc[df_hr['Length_Service']>=upper_bound, 'Length_Service']=upper_bound
       print('\n\n==============\nOutlier berhasil di impute!')
```

3

```
print('\n\n==============\nOutlier berhasil di impute!')
```

```
imput3_outlier(df_hr, upper_bound=8.695)
```

```
sum of outlier : 142
outlier :
[   6    9   15   16   19   28   31   35   49   61   67   69   86   87
  102  127  134  136  137  143  144  145  160  164  169  174  185  190
  208  209  216  217  221  227  232  302  308  312  316  320  321  323
  324  350  351  354  361  371  390  401  405  410  415  421  426  435
  437  451  463  465  468  479  482  485  494  500  501  504  518  694
  715  996 1317 1318 1319 1322 1323 1327 1337 1340 1343 1345 1350 1351
 1356 1359 1365 1371 1379 1380 1381 1384 1389 1391 1394 1399 1405 1408
 1412 1413 1440 1447 1457 1460 1462 1468 1469 1474 1479 1484 1485 1486
 1487 1496 1497 1499 1515 1516 1522 1523 1528 1534 1549 1559 1578 2058
 2114 2462 3887 4231 4481 5096 5321 5328 5578 6133 6412 6420 6443 6515
 7009 7445]


==============
Outlier berhasil di impute!
```

```
[64] imput3_outlier(df_hr, upper_bound=8.695)

     sum of outlier : 0
     outlier :
     []
```

4

```
def input3_outlier(df_hr, lower_bound):
  ## kode untuk mencetak outlier dan melihat jumlah outlier yang ada
  outliers = np.where(df_hr.Length_Service < lower_bound)
  print(f'sum of outlier : {np.count_nonzero(df_hr.Length_Service < lower_bound)}')
  print(f'outlier : \n{outliers[0]}\n')

   ## loading
  proses='=============\nProses membersihkan outliers '
  loading='. . . .'
  def message(proses, loading):
    print(proses)
    for i in loading:
      print(i, end='')
      time.sleep(0.5)

    if __name__ == '__main__':
      message(proses, loading)

  ## kode untuk impute outlier
  df_hr.loc[df_hr['Length_Service']<= lower_bound, 'Length_Service']= lower_bound
  print('\n\n==============\nOutlier berhasil di impute!')
```

5

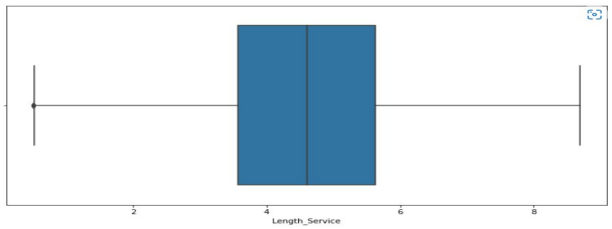
```
[66] input3_outlier(df_hr, lower_bound=0.503)

     sum of outlier : 21
     outlier :
     [  51  746 1033 2096 2460 2661 3129 3234 3404 3581 3895 4058 4243 5509
      6587 6593 6898 7014 7685 7907 8307]


     ==============
     Outlier berhasil di impute!
```

```
input3_outlier(df_hr, lower_bound=0.503)
```

```
sum of outlier : 0
outlier :
[]


==============
Outlier berhasil di impute!
```

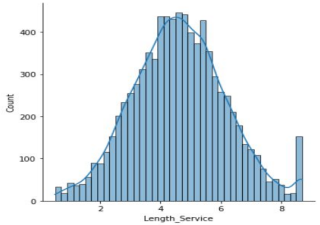
Results

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Length_Service', data=df_hr)
plt.show()
```



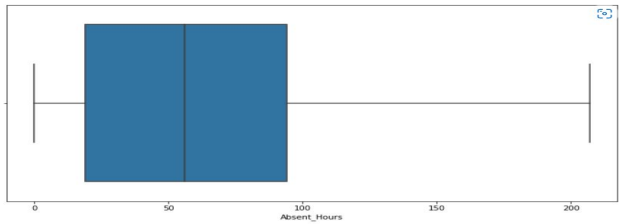

```
sns.displot(df_hr['Length_Service'],kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0062216e50>
```



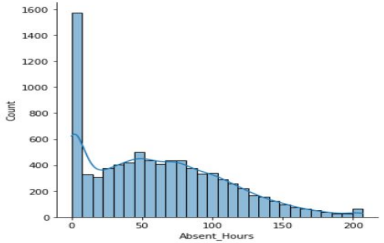
Data Age

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Absent_Hours', data=df_hr)
plt.show()
```



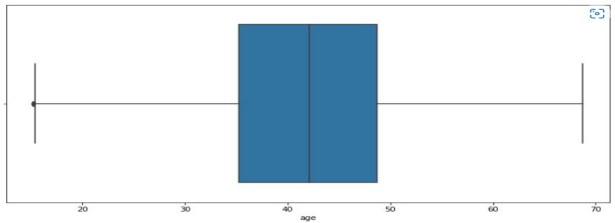

```
sns.displot(df_hr['Absent_Hours'],kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f005b0004d0>
```

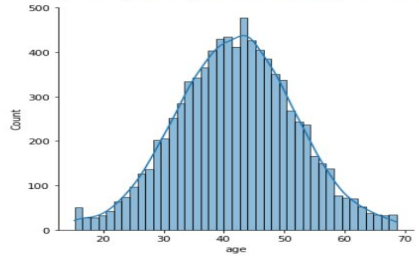


Data Length_Service

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='age', data=df_hr)
plt.show()
```




```
<seaborn.axisgrid.FacetGrid at 0x7f0061ff3a50>
```



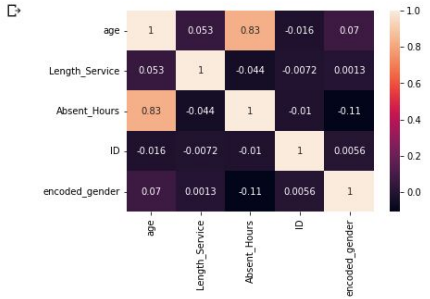
Data Absent_Hours

3 Summary Statistic & Correlation

```
df_hr.describe().T.round(3)
```

	count	mean	std	min	25%	50%	75%	max
age	8336.0	42.007	9.940	3.505	35.299	42.115	48.667	77.938
Length_Service	8336.0	4.783	2.463	0.012	3.576	4.600	5.624	43.735
Absent_Hours	8336.0	61.284	49.038	0.000	19.128	56.006	94.285	272.530
ID	8336.0	5.494	2.872	1.000	3.000	5.000	8.000	10.000
encoded_gender	8336.0	0.506	0.500	0.000	0.000	1.000	1.000	1.000

```
sns.heatmap(df_hr.corr(), annot=True)
plt.show()
```



Melalui summary statistic yang diperlihatkan, bahwa nilai dari summary statistik jelas dilihat pada variabel age, Length_Service, dan Absent_Hour nilai min dan max yang mempunyai gap yang besar, dan nilai mean dan median yang belum sama sehingga bisa dikatakan belum berdistribusi normal.

Dari heatmap juga bisa dilihat bahwa korelasi antar variabel yang paling positive adalah Absent_Hour dan age yaitu sebesar **0.83**

DATA DEVELOPMENT

```
[ ]  x = np.asarray(df_hr.copy())
```

```
[ ]  columns=['age', 'length_service','encoded_gender', 'absent_hours']
```

Make an list named columns consisting of numerical variable taken from employee's data. columns consists of several variable such as **'age', 'length_service', 'encoded_gender', 'absent_hours'**

```
scale = StandardScaler()
x = scale.fit_transform(x)
x_scaled = pd.DataFrame(x, columns=df_hr.columns)
```

```
x_scaled.head()
```

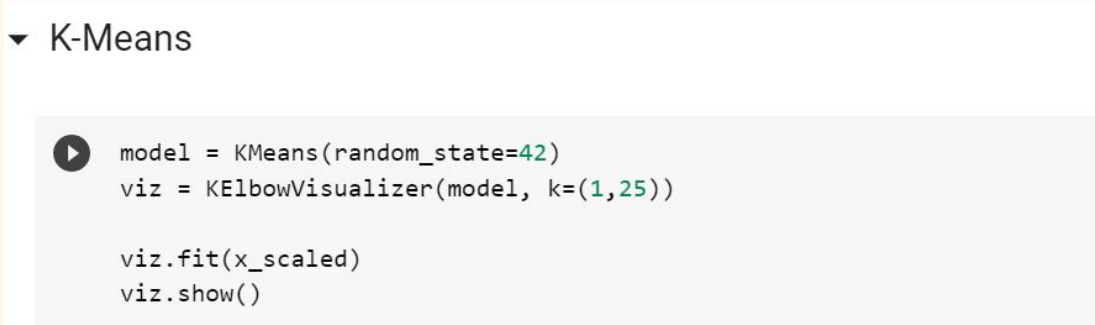
	age	length_service	absent_hours	id	encoded_gender
0	-1.010503	0.889235	-0.505384	-0.171906	-1.011583
1	-0.171163	0.577403	-0.637151	1.220718	0.988549
2	0.689339	-0.155589	0.465168	1.568874	0.988549
3	0.261911	-0.994933	0.181842	-1.216375	-1.011583
4	-0.639114	-0.650174	-1.257021	-0.520062	0.988549

Standarize each variable

Standarization makes each variable to contribute equally. Therefore, average and standar deviation sequentially values 0 and 1

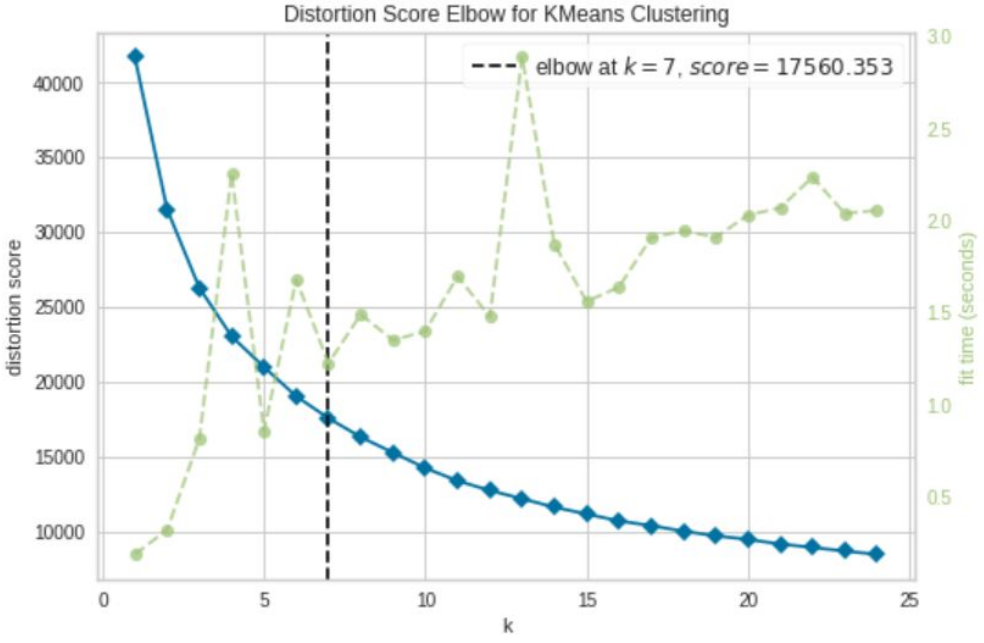
DATA DEVELOPMENT

Clustering Score Visualizer



- The **K-Elbow Visualizer** implements the "**elbow**" method of selecting the optimal number of clusters for **K**-means clustering. **K**-means is a simple unsupervised machine learning algorithm that groups data into a specified number (**k**) of clusters.

- In this case, we will search k-values in between **1** to **25**

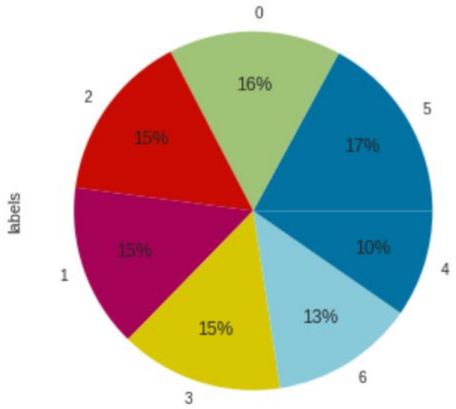



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f99d30ba4d0>
```

Optimal Number of Cluster: k=7

```
x_scaled.labels.value_counts().plot.pie(autopct='%1.0f%%', pctdistance=0.7, labeldistance=1.1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f99cdee7050>
```



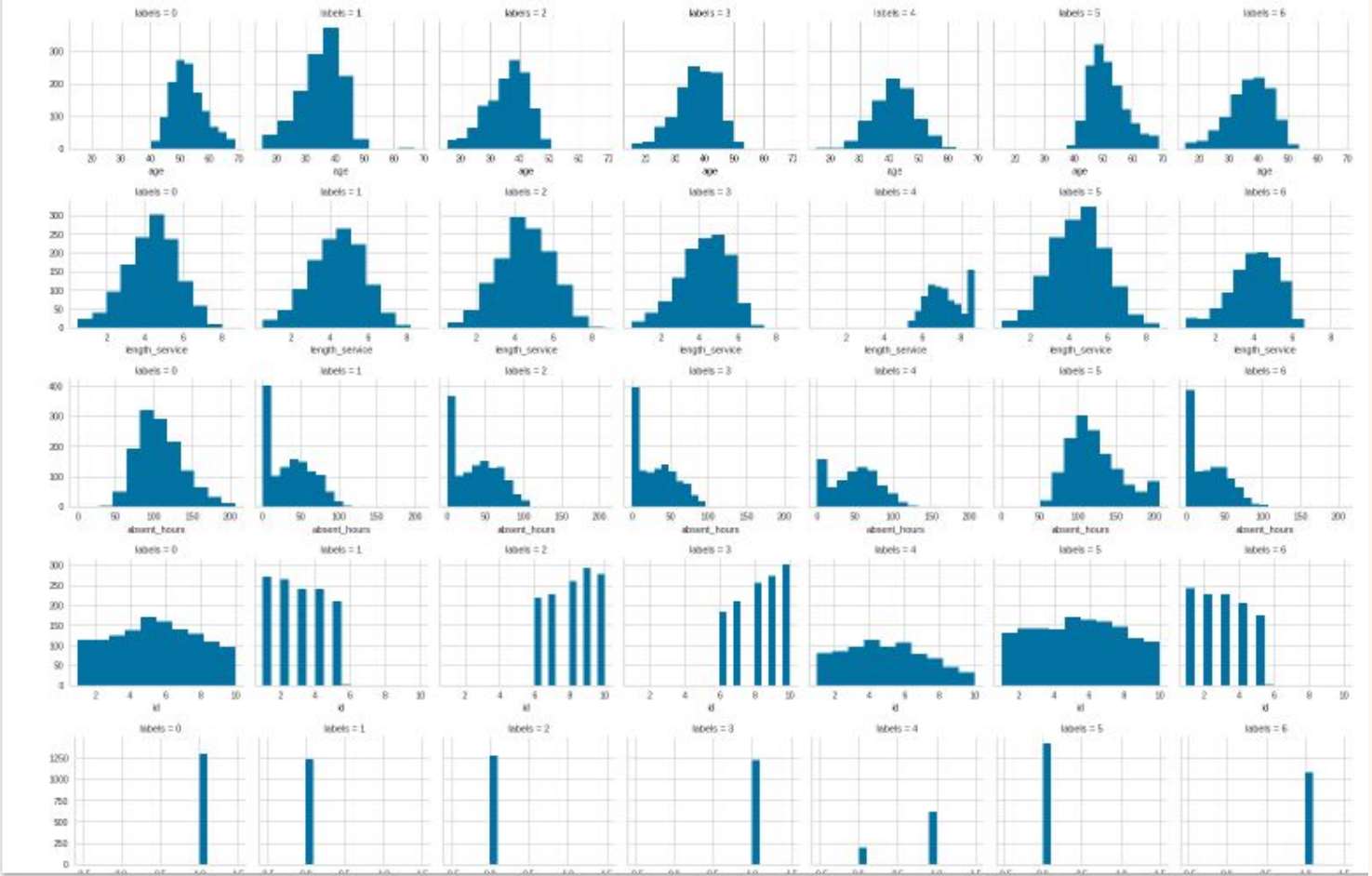
Melalui pie plot yang diperoleh dapat disimpulkan bahwa:

Lable 5 shows the highest

- Label 5 merupakan kelompok cluster yang memiliki jumlah anggota paling banyak yaitu **17%**
- Label 6 merupakan peringkat kedua label yang memiliki kelompok cluster terbesar
- Diikuti oleh label 1, 2, 3 masing masing menghasilkan **15%**
- Label 4 dan 6 berada pada clustering dengan kelompok paling sedikit

Membuat visualisasi untuk menginterpretasikan tiap cluster yang ada

```
for col in df_hr:
    grid= sns.FacetGrid(df_hr, col='labels')
    grid.map(plt.hist, col)
```



```
[ ]  pca = PCA(2)
     x_pca = x_scaled.copy()

     pca.fit(x_pca)
     x_pca = pca.transform(x_pca)
     x_pca.shape
```

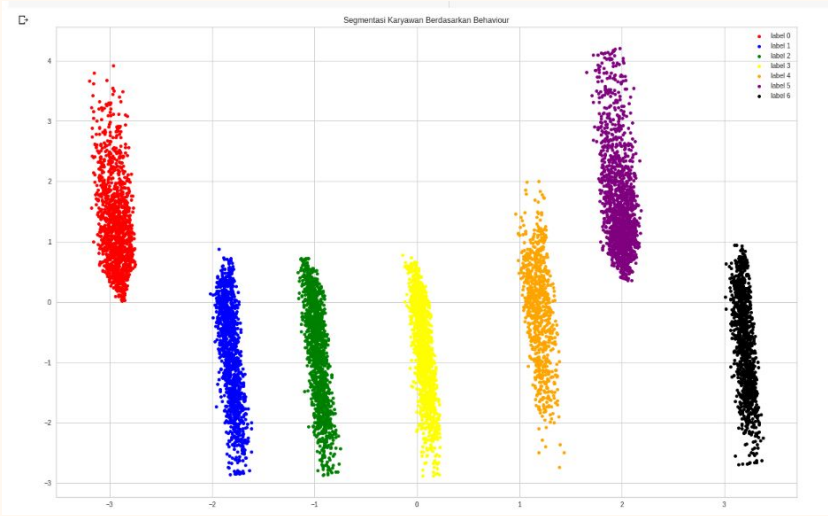
```
     (8336, 2)
```

```
[ ]  x, y = x_pca[:, 0], x_pca[:, 1]

     viz_cluster = pd.DataFrame({'x': x, 'y':y, 'labels':labels})
     groups = viz_cluster.groupby('labels')
```

```
colors = {0: 'red',
          1: 'blue',
          2: 'green',
          3: 'yellow',
          4: 'orange',
          5: 'purple',
          6: 'black',
          7: 'pink'}

names = {0: 'label 0',
         1: 'label 1',
         2: 'label 2',
         3: 'label 3',
         4: 'label 4',
         5: 'label 5',
         6: 'label 6',
         7: 'label 7'}

fig, ax = plt.subplots(figsize=(20, 13))

for name, group in groups:
    ax.plot(group.x, group.y, marker='o', linestyle='', ms=5,
            color=colors[name],label=names[name], mec='none')
    ax.set_aspect('auto')
    ax.tick_params(axis='x',which='both',bottom='off',top='off',labelbottom='off')
    ax.tick_params(axis= 'y',which='both',left='off',top='off',labelleft='off')

ax.legend()
ax.set_title('Segmentasi Karyawan Berdasarkan Behaviour, Umur, ID, dan Gender')
plt.show()
```

Principle Component Analysis

Principal Component Analysis, or PCA, is a **dimensionality-reduction method** that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.



Insight

- In the segmentation cluster label 0 information is obtained that the average age number of employees is 40 - 70 years, dominated by women, the length of work is 1-8 years, the time absent is 50-200 hours, and ID spreads from 1-10
- In the segmentation cluster label 1 information is obtained that the average age number of employees is <20 - 50 years, dominated by men, the length of work is 1-8 years, the time absent is 0 - 100 hours, and ID spread from 1-6
- In the segmentation cluster label 2 information is obtained that average age number of employees is <20 - 50 years, dominated by men, the length of work is 1-9 years, the time absent is 0-110 hours, and ID spread from 6-10
- In the segmentation cluster label 3 information is obtained that the average age number of employees is <20-55 years, dominated by women, the length of work is 1-7 years, the time absent is 0-90 hours, and ID spread from 6-10
- In the segmentation cluster label 4 information is obtained that the average age number of employees is 35-68 years, dominated by woman and few men, the length of work is 5-10 years, the time absent is 0-125 hours, and ID spread from 1-10

Insight

- In the segmentation cluster label 5 information is obtained that the average age number of employees is 38-68 years, dominated by men, the length of work is 0-9 years, the time absent is 50-210 hours, and ID spreads from 1-10
- In the segmentation cluster label 6 information is obtained that the average age number of employees is <20 - 55 years, dominated by women, the length of work is 0-7 years, the time absent is 0 - 100 hours, and ID spread from 1-6

Model Evaluation

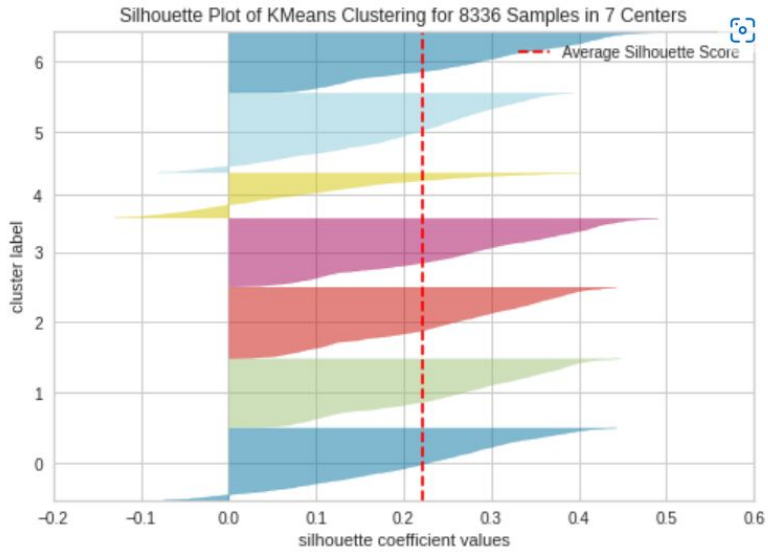
```
print(f'Davies-Bouldin Index = {davies_bouldin_score(x_scaled, labels)}')
print(f'Silhouette Score = {silhouette_score(x_scaled, labels)}')
```

```
Davies-Bouldin Index = 1.064346373652427
Silhouette Score = 0.3913592582495097
```

Untuk melihat seberapa baik model clustering, dapat dievaluasi menggunakan 2 cara berikut:

- **Davis-Bouldin Index**
  The **Davies-Bouldin index** (DBI) is **one of the clustering algorithms evaluation measures.** It is most commonly used to evaluate the goodness of split by a K-Means clustering algorithm for a given number of clusters.
- **Silhouette Coefficient**
  Silhouette coefficient ranges between **−1 and 1**, where a higher silhouette coefficient refers to a model with more coherent clusters. In other words, silhouette coefficients close to +1 means the sample is far away from the neighboring clusters. A value of 0 means that the sample is on or very close to the decision boundary between two neighboring clusters. Finally, negative values indicate that the samples could have potentially been assigned to the wrong cluster

```
viz_sil = SilhouetteVisualizer(kmean, colors='yellowbrick')
viz_sil.fit(x_scaled.drop(columns='labels', axis=1))
viz_sil.show()
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7f99cddc6f50>
```

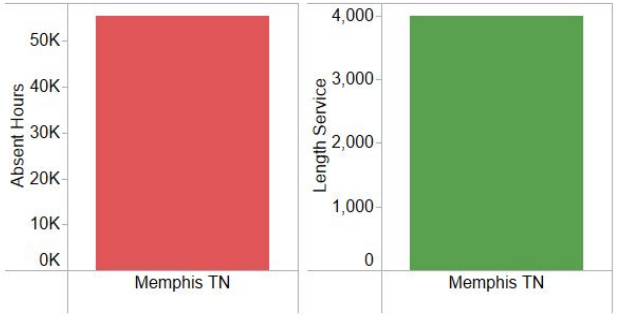
- From the evaluation process of the model above, it can be concluded that the model made is good because the DBI value shows the number 1.064 which is almost close to 0.
- The Silhouette chart shows that all cluster tables have crossed the average silhouette score line, which indicates that our model is good. The coefficient generated by the Silhouette is 0.391 and is not close to minus.The silhouette graph using SilhouetteVisualizer shows that all cluster labels have crossed the average silhouette score line, which indicates that our model is quite good.

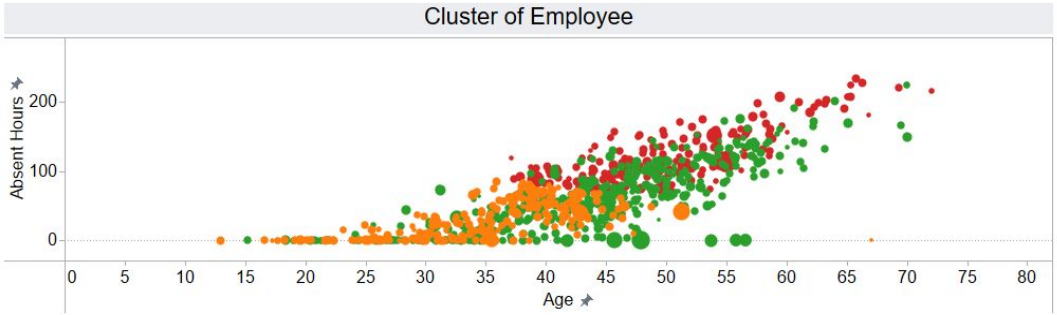
Departement HR Analysis Dashboard

by Novita Yolanda Barus

Id
(All)

Avg of Absent Hour	Avg of Length Service	Avg of Age	Distribution Center ID
66.08	4.758	42.75	1 2 3 4 5 6 7 8 9 10





Notes

Cluster 1: High absent hour
Cluster 2: Medium absent hour
Cluster 3: Low absent hour

Conclusion

- The highest absent rate is in cluster **label 5** with an average of 100 hours and there are a number of employees who have been absent for 200 hours and the average length of work is 5 years. In cluster label 5, distribution centers where employees work are also almost evenly spread from distribution centers 1 - 10 and are mostly located in distribution centers 5. All members of label 5 group are men with an average age of 50 years.
- In cluster label 5, which has the highest number of absentee numbers, it is grouped into groups of workers who have worked for a long time, male workers and old age.

Thank you.