

**UJIAN AKHIR SEMESTER  
MACHINE LEARNING**



Oleh:

Nama : Widi Aning Pangesti

NPM : 41155050190081

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS LANGLANGBUANA  
BANDUNG  
2023**

## Bagian 1

1. Apa itu *Linear* dan *Logistic Regresion* dan apa gunanya?

**Jawab:**

Linear Regression adalah teknik analisis data untuk variable dependen yang memiliki skala data continue guna memprediksi asumsi tambahan yang mengkorelasikan antara variabel independen dan dependen melalui garis yang paling sesuai dari titik data garis lurus.

Logistic Regression adalah teknik analisis data untuk variable dependen yang memiliki skala data kategori guna memprediksi nilai dari salah satu faktor tersebut berdasarkan faktor yang lain (sebab-akibat).

2. Apa itu *Support Vector Machine* dan apa gunanya?

**Jawab:**

*Support Vector Machine* (SVM) adalah salah satu algoritma *machine learning* dengan pendekatan supervised learning yang bekerja dengan mencari *hyperplane* atau fungsi pemisah terbaik untuk memisahkan kelas. Algoritma ini dapat digunakan untuk klasifikasi (SVM *classification*) dan regresi (SVM *regression*).

Tujuan dari SVM adalah untuk membagi dataset ke dalam kelas-kelas untuk menemukan *hyperplane marginal maksimum*.

3. Apa itu *K-Nearest Neighbor* dan apa gunanya?

**Jawab:**

*K-Nearest Neighbor* adalah suatu metode klasifikasi terhadap suatu objek berdasarkan banyaknya data training ( $k$ ) yang memiliki jarak terdekat dengan objek dengan syarat bahwa nilai  $k$  tidak boleh melebihi jumlah data training, harus bernilai ganjil dan lebih dari satu.

*K-Nearest Neighbor* digunakan untuk mengklasifikasi objek baru berdasarkan atribut dan training samples, contohnya algoritma ini digunakan pada aplikasi data mining, *pattern recognition*, *image processing*, dll.

4. Apa itu *Naive Bayes* dan apa gunanya?

**Jawab:**

*Naive Bayes* merupakan suatu algoritma yang banyak dan sering digunakan untuk keperluan klasifikasi atau pengelompokan data. *Naive Bayes* digunakan untuk mengklasifikasikan dokumen teks, seperti teks berita ataupun teks akademis. Sebagai metode machine learning yang menggunakan probabilitas, untuk membuat diagnosis medis secara otomatis, dan mendeteksi atau menyaring spam.

5. Apa itu *Decision Tree* dan apa gunanya?

**Jawab:**

*Decision tree* merupakan algoritma machine learning yang menggunakan seperangkat aturan untuk membuat keputusan dengan struktur seperti pohon yang memodelkan kemungkinan hasil, biaya sumber daya, utilitas dan kemungkinan konsekuensi atau resiko. *Decision Tree* digunakan untuk mengambil suatu keputusan. Untuk bisa mengambil keputusan dengan struktur ini perlu disiapkan beberapa variabel yang akan berfungsi sebagai bahan pertimbangan keputusan yang akan diambil.

6. Apa itu *Random Forest* dan apa gunanya?

**Jawab:**

Random Forest merupakan algoritma *machine learning* yang menggabungkan keluaran dari beberapa *decision tree* untuk mencapai satu hasil. Random Forest membangun beberapa decision tree dan menggabungkannya untuk mendapatkan prediksi yang lebih akurat dan stabil. Keuntungan besar dari *Random Forest* adalah dapat digunakan untuk masalah klasifikasi dan regresi, yang merupakan mayoritas sistem pembelajaran mesin saat ini.

7. Apa itu K-Means dan apa gunanya?

**Jawab:**

K-means merupakan salah satu algoritma machine learning termasuk ke dalam jenis *unsupervised learning*. K-Means berfungsi untuk mengelompokkan dataset yang belum dilabel ke dalam kluster yang berbeda. Algoritma ini dapat menerima data tanpa ada label kategori.

8. Apa itu *Agglomerate Clustering* dan apa gunanya?

**Jawab:**

Agglomerative Clustering merupakan strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu *cluster* yang terpisah kemudian membentuk *cluster* yang semakin membesar. banyaknya *cluster* awal adalah sama dengan banyaknya objek.

9. Apa itu *Apriori Algorithm* dan apa gunanya?

**Jawab:**

Algoritma Apriori merupakan algoritma yang digunakan untuk menghitung aturan asosiasi antar objek. Aturan asosiasi menjelaskan bagaimana dua atau lebih objek terkait satu sama lain. Algoritma Apriori pertama kali diperkenalkan oleh Agrawal dan Shrikant (1994) yang berguna untuk menentukan *frequent itemset* pada sekumpulan data.

10. Apa itu *Self Organizing Map* dan apa gunanya?

**Jawab:**

*Self Organizing Map* merupakan suatu teknik dalam *Neural Network* yang bertujuan untuk melakukan visualisasi data dengan cara mengurangi dimensi data melalui penggunaan self-organizing neural networks sehingga manusia dapat mengerti *high-dimensional* data yang dipetakan dalam bentuk *low-dimensional* data.

## **Bagian 2** Studi Kasus: Membuat model dengan machine learning pada data Liga120192021.csv

1. Buatlah model dengan machine learning pada data Liga120192021.csv !
2. Untuk metode silakan PILIH SALAH SATU: Logistic Regression, Support Vector Machine, K-Nearest Neighbour, K Means. Sebut metode yang dipilih!
3. Buat model perkolom, klasifikasi = 2, cluster = 2.
4. Buat juga hasil pemetaan warna dengan scattered plot !
5. Hasilnya selain dikumpulkan berupa pdf, harus disimpan di Github masing-masing. Berikan URL Github masing-masing!

Dalam menyelesaikan tugas ini, digunakan metode KMeans untuk membuat model dengan machine learning pada data Liga120192021.csv

Coding:

Clustering terhadap colum 'Pass9' dan 'Pass10'

```
# Load Library
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

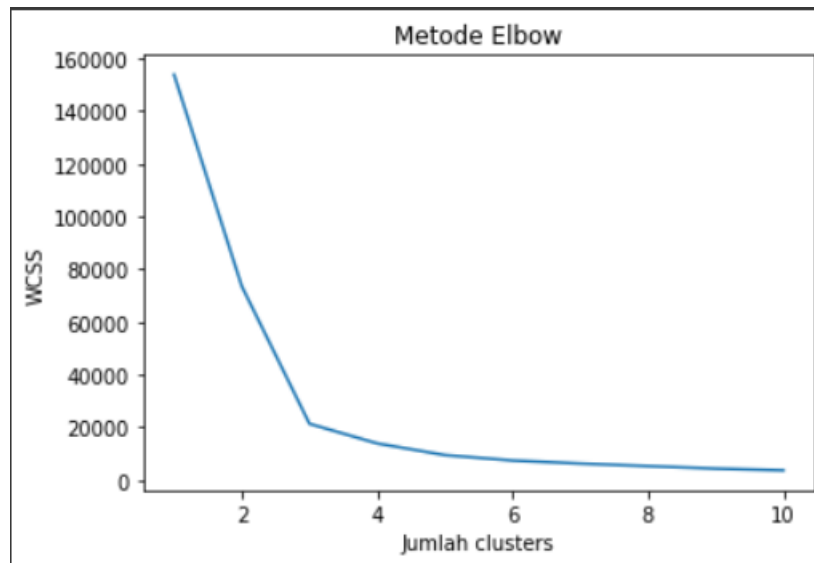
# Load Dataset
dt = pd.read_csv('/content/Liga120192021.csv')
x = dt.iloc[:, [8, 9]].values

# menentukan range cluster
from sklearn.cluster import KMeans
x = dt.iloc[:, [8, 9]].values

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=32)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)

# menampilkan grafik dengan metode Elbow untuk menentukan range
plt.plot(range(1, 11), wcss)
plt.title('Metode Elbow')
plt.xlabel('Jumlah clusters')
plt.ylabel('WCSS')
plt.show()
```

Result:

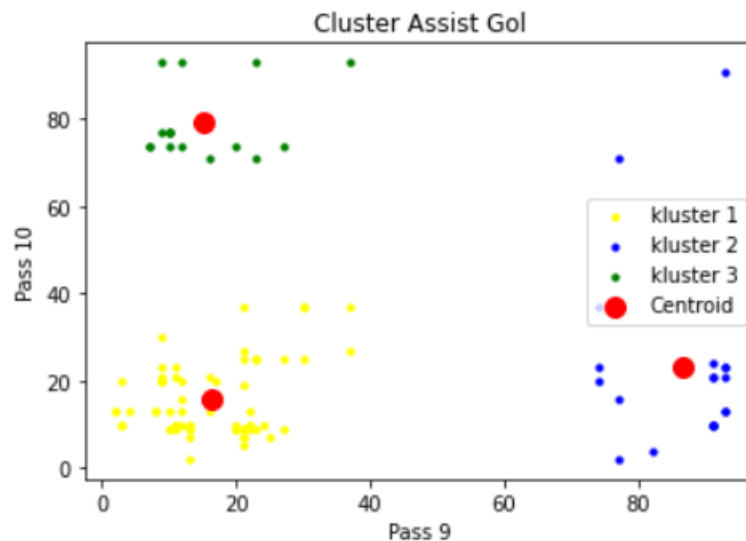


```
# Clustering 2 column dengan range = 3
kmeans = KMeans(n_clusters=3, init = 'k-means++', random_state = 32)
y_kmeans = kmeans.fit_predict(x)

plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s=10, c='yellow', label='kluster 1')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s=10, c='blue', label='kluster 2')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s=10, c='green', label='kluster 3')
plt.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1], s=100, c = 'red', label = 'Centroid')

plt.title('Cluster Assist Gol')
plt.xlabel('Pass 9')
plt.ylabel('Pass 10')
plt.legend()
plt.show()
```

Result:



Coding:

```
# Library
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from matplotlib import pyplot as pyplot
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import warnings
from scipy import stats
warnings.filterwarnings('ignore')

# Load Dataset
dt = pd.read_csv('/content/Liga120192021.csv')
dt
```

```
# Membuat function untuk membuat model KMeans
def createModelBy2Column(index):

    # Mengambil 2 Column berdasarkan index
    new_dt = dt[['Pass{0}'.format(index), 'Pass{0}'.format(index+1)]]
    scaler = StandardScaler()
    scaler.fit(new_dt)
    dt_scaled = scaler.transform(new_dt)
    dt_scaled = pd.DataFrame(dt_scaled)

    # Membuat Prediksi menggunakan K-Means
    km = KMeans(n_clusters=2)
    y_predicted = km.fit_predict(dt_scaled)
```

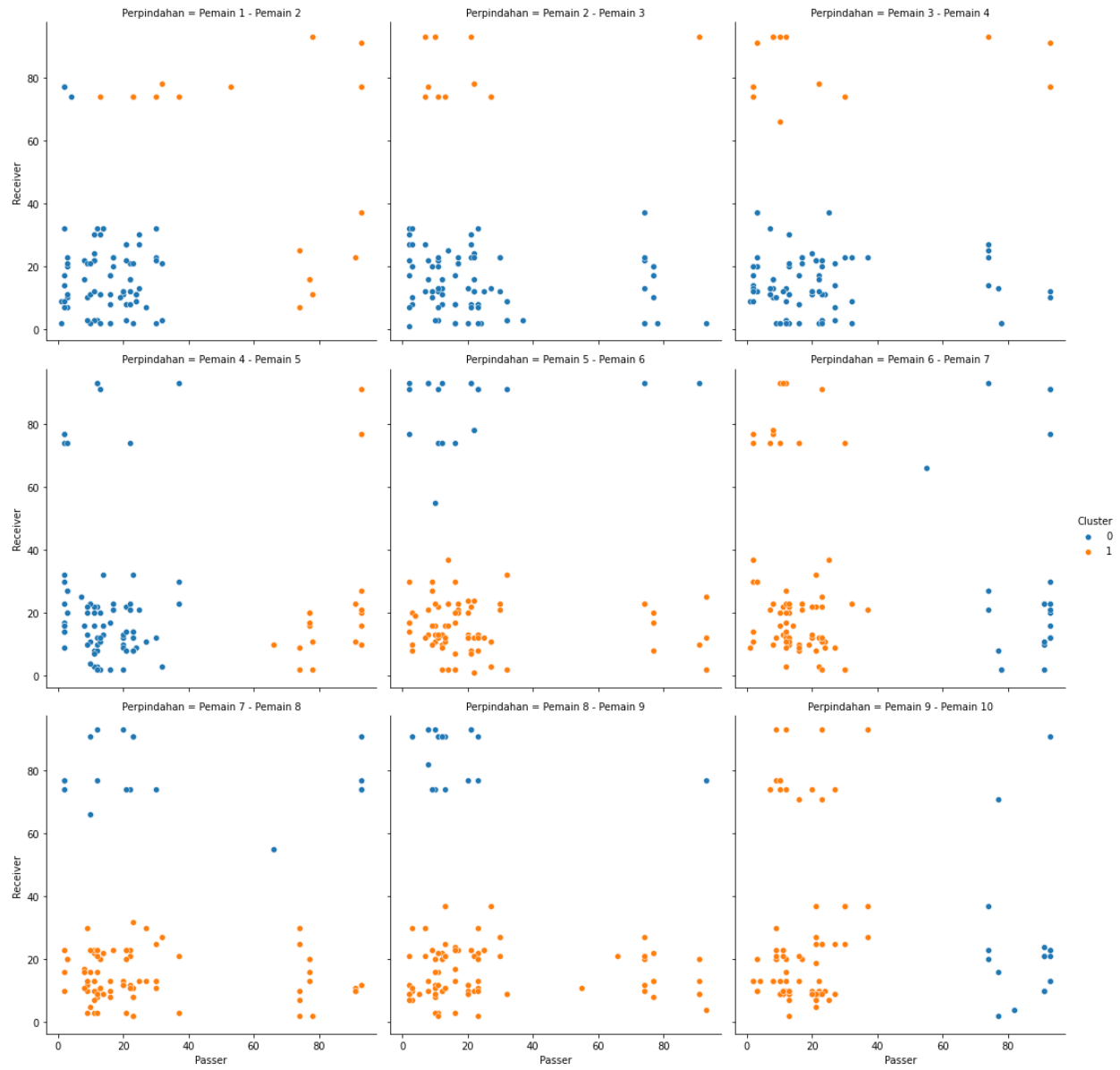
```
# Mengatur ulang Columns
new_dt.loc[:, "Cluster"] = y_predicted
new_dt.loc[:, "Perpindahan"] = 'Pemain {0} - Pemain {1}'.format(index,index+1)
new_dt.loc[:, "Passer"] = new_dt['Pass{0}'.format(index)]
new_dt.loc[:, "Receiver"] = new_dt['Pass{0}'.format(index+1)]
new_dt.drop(['Pass{0}'.format(index), 'Pass{0}'.format(index+1)], axis=1)
return new_dt

results = None

for key in range(len(dt.columns) -1):
    index = key + 1
    result = createModelBy2Column(index)
    if results is None:
        results = result
    else:
        results=pd.concat([results, result])

g = sns.FacetGrid(results, col='Perpindahan', hue = 'Cluster', height=5, col_wrap=3)
g.map(sns.scatterplot, 'Passer', 'Receiver')
g.add_legend()
```

Result:



Github: [https://github.com/widianingp/UAS\\_ML\\_WidiAningPangesti](https://github.com/widianingp/UAS_ML_WidiAningPangesti)