



**LABORATORIUM PEMBELAJARAN ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**

BAB : SELF-ORGANIZING MAPS  
NAMA : ARION SYEMAEL SIAHAAN  
NIM : 225150207111060  
TANGGAL : 06/11/2024  
ASISTEN : ALIFAH KHAIRUNNISA  
ANDHIKA IHSAN CENDEKIA



**A. Praktikum**

1. Buka Google Collaboratory melalui [tautan ini](#).
2. Tulis kode berikut ke dalam setiap *cell* pada *notebook* tersebut.
  - a. Fungsi Self-Organizing Maps

```
import numpy as np

def som(X, lrate, b, max_epoch, n_cluster):
    centroids = np.random.uniform(size=(n_cluster, len(X[0])))
    epoch = 0
    labels = []
    while epoch < max_epoch:
        for x in X:
            d = [sum((w - x) ** 2) for w in centroids]
            min = np.argmin(d)
            centroids[min] += lrate * (x - centroids[min])
            lrate *= b
            epoch += 1
        for x in X:
            d = [sum((w - x) ** 2) for w in centroids]
            min = np.argmin(d)
            labels.append(min)
    return centroids, labels

def draw(X, target, centroids):
    colors = 'rgbcmyk'
    for x, label in zip(X, target):
        plt.plot(x[0], x[1], colors[label] + '.')
    plt.plot(centroids[:, 0], centroids[:, 1], 'kx')
```

**b. Klasterisasi**

```
from sklearn.datasets import make_blobs
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

X, target = make_blobs(n_samples=30, n_features=2,
                        centers=3, random_state=3)
print('Data:\n', X)
centroids, labels = som(X, lrate=.5, b=.5, max_epoch=100,
                        n_cluster=3)
print('\ncentroid terbaru:\n', centroids)
```

```
silhouette = silhouette_score(X, labels)
print('\nSilhouette score:', silhouette)
draw(X, target, centroids)
```

## B. Screenshot

### a. Fungsi Self-Organizing Maps

▼ a) Fungsi *self-organizing maps*

Tulis kode ke dalam *cell* di bawah ini:

```
import numpy as np

def som(X, lrate, b, max_epoch, n_cluster):
    centroids = np.random.uniform(size=(n_cluster, len(X[0])))
    epoch = 0
    labels = []
    while epoch < max_epoch:
        for x in X:
            d = [sum((w - x) ** 2) for w in centroids]
            min = np.argmin(d)
            centroids[min] += lrate * (x - centroids[min])
            lrate *= b
            epoch += 1
        for x in X:
            d = [sum((w - x) ** 2) for w in centroids]
            min = np.argmin(d)
            labels.append(min)
    return centroids, labels

def draw(X, target, centroids):
    colors = 'rgbcmyk'
    for x, label in zip(X, target):
        plt.plot(x[0], x[1], colors[label] + '.')
    plt.plot(centroids[:, 0], centroids[:, 1], 'kx')
```

ARION SYEMAEL SIAHAAN  
225150207111060

### b. Klasterisasi

## ▼ b) Klasterisasi

Tulis kode ke dalam *cell* di bawah ini:

✓  
7s

```
from sklearn.datasets import make_blobs
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

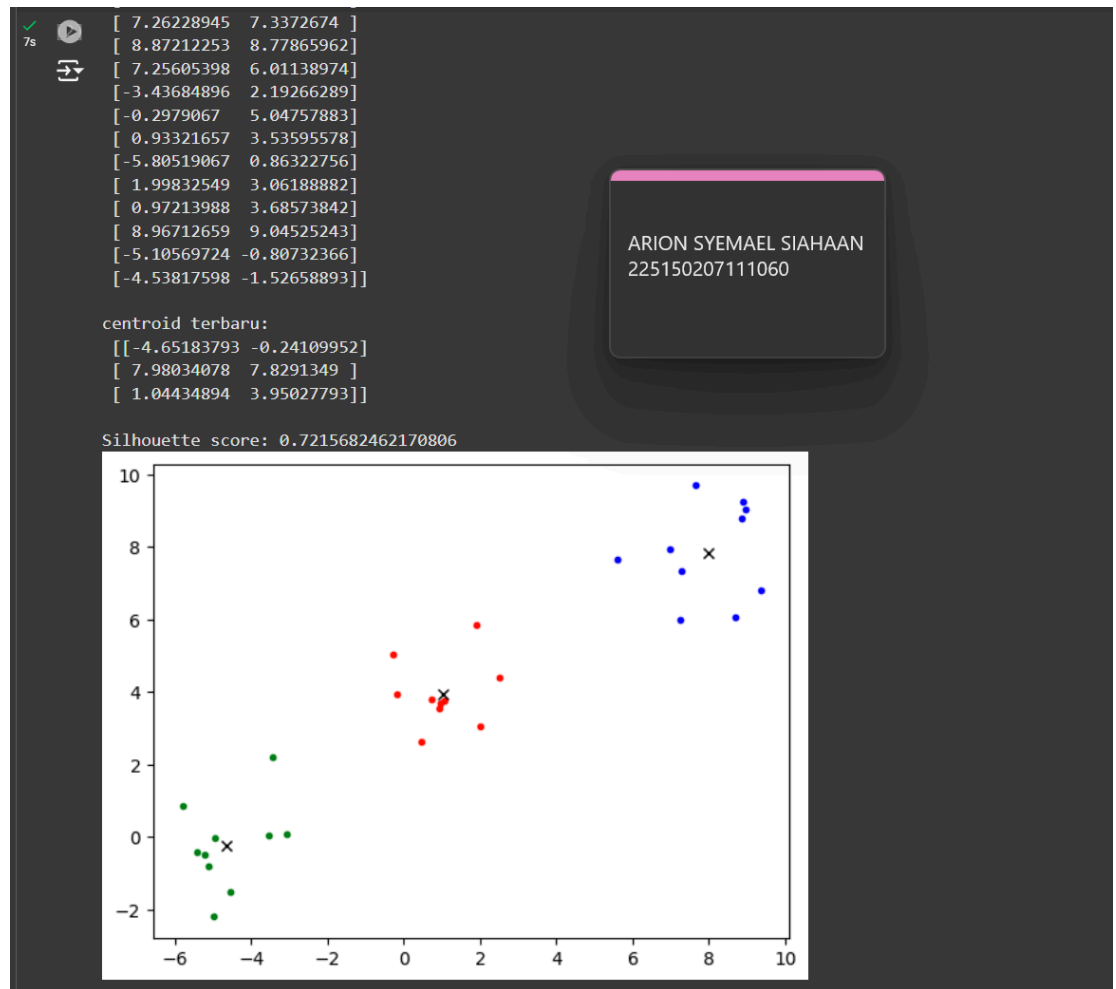
X, target = make_blobs(n_samples=30, n_features=2, centers=3, random_state=3)
print('Data:\n', X)
centroids, labels = som(X, lrate=.5, b=.5, max_epoch=100, n_cluster=3)
print('\ncentroid terbaru:\n', centroids)
silhouette = silhouette_score(X, labels)
print('\nSilhouette score:', silhouette)
draw(X, target, centroids)
```



Data:

```
[[ -3.05792726  0.08463787]
 [ -0.16908848  3.95730655]
 [ -5.20569036 -0.4964411 ]
 [ -4.95074157 -0.01347862]
 [  1.06599169  3.75827904]
 [ -4.98567132 -2.20253107]
 [  0.73856985  3.80819747]
 [  8.9070866   9.2595996 ]
 [  6.98505679  7.95557559]
 [  2.50210641  4.39967272]
 [  9.34648222  6.80756109]
 [ -3.55666026  0.05603874]
 [  0.4705981   2.61647914]
 [  7.66152441  9.70050681]
 [ -5.42602855 -0.40986481]
 [  1.89727609  5.87252952]
 [  5.61068132  7.65809991]
 [  8.70477249  6.06497225]
 [  7.26228945  7.3372674 ]
 [  8.87212253  8.77865962]
 [  7.25605398  6.01138974]
 [ -3.43684896  2.19266289]
 [ -0.2979067   5.04757883]
```

ARION SYEMAEL SIAHAAN  
225150207111060



### C. Analisis

1. Ubah parameter pada kode b menjadi learning rate = 0,05 dan epoch maksimum = 3 lalu jalankan program. Amati gambar hasil klusterisasi dan nilai silhouette yang didapatkan.

```
X, target = make_blobs(n_samples=30, n_features=2, centers=3,
random_state=3)
print('Data:\n', X)
centroids, labels = som(X, lrate=0.05, b=0.5, max_epoch=3,
n_cluster=3)
print('\ncentroids:\n', centroids)
silhouette = silhouette_score(X, labels)
print('\nSilhouette score:', silhouette)
draw(X, target, centroids)
```

Os

▶

```
X, target = make_blobs(n_samples=30, n_features=2, centers=3, random_state=3)
print('Data:\n', X)
centroids, labels = som(X, lrate=0.05, b=0.5, max_epoch=3, n_cluster=3)
print('\ncentroids:\n', centroids)
silhouette = silhouette_score(X, labels)
print('\nSilhouette score:', silhouette)
draw(X, target, centroids)
```

🔗

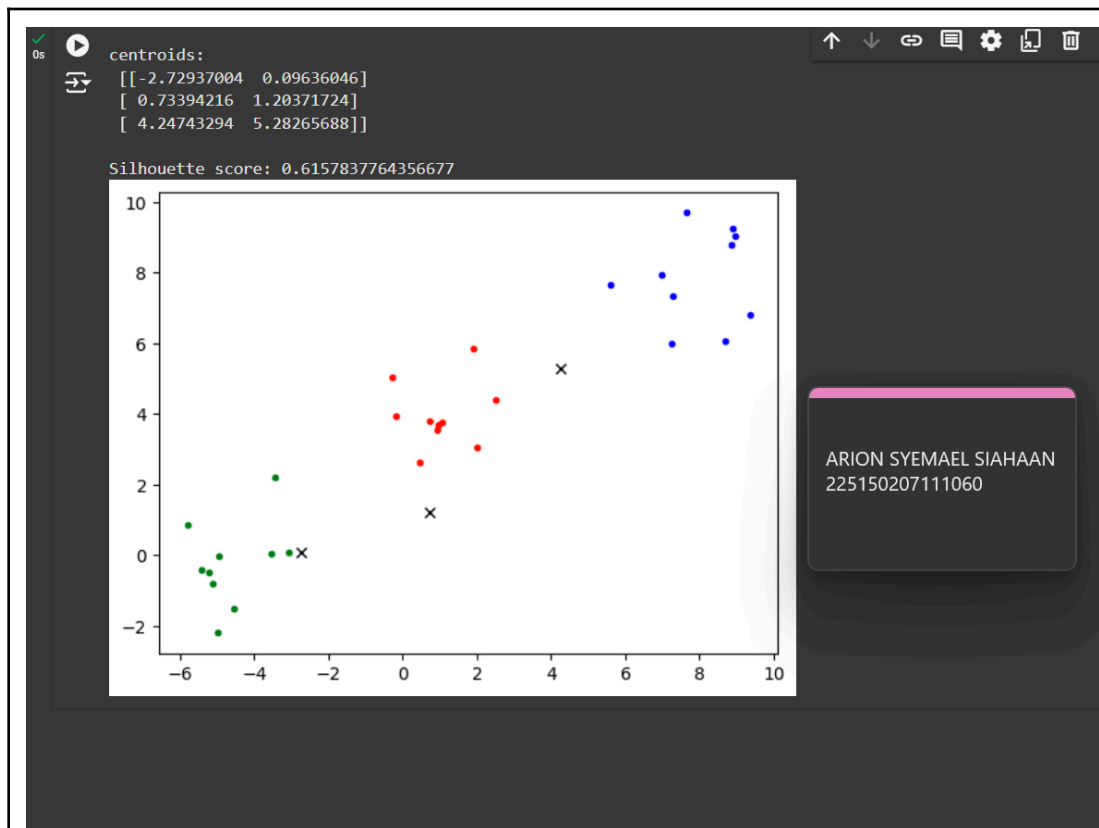
Data:

```
[[ -3.05792726  0.08463787]
 [ -0.16908848  3.95730655]
 [ -5.20569036 -0.4964411 ]
 [ -4.95074157 -0.01347862]
 [  1.06599169  3.75827904]
 [ -4.98567132 -2.20253107]
 [  0.73856985  3.80819747]
 [  8.9070866  9.2595996 ]
 [  6.98505679  7.95557559]
 [  2.50210641  4.39967272]
 [  9.34648222  6.80756109]
 [ -3.55666026  0.05603874]
 [  0.4705981  2.61647914]
 [  7.66152441  9.70050681]
 [ -5.42602855 -0.40986481]
 [  1.89727609  5.87252952]
 [  5.61068132  7.65809991]
 [  8.70477249  6.06497225]
 [  7.26228945  7.3372674 ]
 [  8.87212253  8.77865962]
 [  7.25605398  6.01138974]
 [ -3.43684896  2.19266289]
 [ -0.2979067  5.04757883]
 [  0.93321657  3.53595578]
 [ -5.80519067  0.86322756]
 [  1.99832549  3.06188882]
 [  0.97213988  3.68573842]
 [  8.96712659  9.04525243]
 [ -5.10569724 -0.80732366]
 [ -4.53817598 -1.52658893]]
```

ARION SYEMAEL SIAHAAN

225150207111060

centroids:



Setelah dilakukan perubahan parameter dengan nilai learning rate 0,05 dan epoch maksimum 3, nilai silhouette score menurun menjadi 0,2961. Penurunan ini menunjukkan bahwa kualitas klusterisasi melemah, dengan centroid yang tidak berada tepat di tengah masing-masing kluster. Learning rate dan jumlah epoch yang rendah mempengaruhi konvergensi, sehingga posisi centroid hanya mendekati pusat kluster secara kasar dan belum optimal. Hal ini menyebabkan centroid tidak sepenuhnya merepresentasikan struktur kluster, sehingga kluster tidak terbentuk dengan sempurna, dan kualitas klusterisasi pun terganggu.

#### D. Kesimpulan

Self-Organizing Maps (SOM) adalah metode jaringan saraf tiruan yang berfungsi untuk memetakan data berdimensi tinggi ke representasi berdimensi lebih rendah, biasanya dua dimensi. Hasil dari proses ini disebut sebagai "peta" yang secara otomatis menata diri tanpa memerlukan pengawasan. SOM bekerja dengan mengelompokkan data ke dalam beberapa kluster, mirip dengan cara korteks serebral pada otak kita memisahkan fungsi sensorik seperti penglihatan, pendengaran, dan penciuman. Dalam SOM, neuron-neuron dengan keluaran yang mirip akan cenderung berada pada area yang berdekatan. Centroid dalam SOM memiliki peran utama dalam proses pengelompokan, karena tiap centroid menjadi pusat dari kluster dan

menentukan data mana yang berada dalam kelompok tertentu berdasarkan kedekatannya.

Pada fungsi SOM dalam kode, library numpy diimpor sebagai np untuk memudahkan operasi numerik. Fungsi utama som memiliki beberapa parameter: X (data input), lr (learning rate), b (bias), max\_epoch (jumlah maksimum epoch), dan n\_cluster (jumlah kluster). Pertama, fungsi ini menginisialisasi centroid secara acak, memulai epoch pada nilai nol, dan mendefinisikan array labels untuk menyimpan label dari tiap data. Proses pembaruan centroid terjadi dalam loop while. Selama epoch belum mencapai max\_epoch, setiap data diperiksa untuk menemukan centroid terdekatnya. Berdasarkan jarak tersebut, centroid diperbarui, dan epoch terus ditambah. Ketika epoch mencapai max\_epoch, perulangan berhenti, dan data diberi label sesuai dengan centroid terdekat yang sudah dihitung.