In [5]:

```
pip install pygad
```

Collecting pygadNote: you may need to restart the kernel to use updated packages.

  Downloading pygad-3.0.1-py3-none-any.whl (67 kB)
                                             0.0/68.0 kB ? eta -:--:--
     ------   10.2/68.0 kB ? eta -:--:--
     ----------------   30.7/68.0 kB 660.6 kB/s eta 0:00:01
     ---------------------   41.0/68.0 kB 281.8 kB/s eta 0:00:01
     ------------------------------------ 68.0/68.0 kB 370.0 kB/s eta 0:00:00
Collecting cloudpickle (from pygad)
  Downloading cloudpickle-2.2.1-py3-none-any.whl (25 kB)
Requirement already satisfied: matplotlib in c:\users\svijayalakshmi\appdata\local\pro
grams\python\python311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\svijayalakshmi\appdata\local\programs
\python\python311\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\svijayalakshmi\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\svijayalakshmi\appdata\local\p
rograms\python\python311\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\svijayalakshmi\appdata\lo
cal\programs\python\python311\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\svijayalakshmi\appdata\lo
cal\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\svijayalakshmi\appdata\loca
l\programs\python\python311\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\svijayalakshmi\appdata\local
\programs\python\python311\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\svijayalakshmi\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\svijayalakshmi\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\svijayalakshmi\appdata\local\progr
ams\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib->pygad)
(1.16.0)
Installing collected packages: cloudpickle, pygad
Successfully installed cloudpickle-2.2.1 pygad-3.0.1

In [47]:

```
import numpy
import matplotlib.pyplot
import pygad
```

In [48]:

```python
cluster1_num_samples = 10
cluster1_x1_start = 0
cluster1_x1_end = 5
cluster1_x2_start = 2
cluster1_x2_end = 6
cluster1_x1 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x1 = cluster1_x1 * (cluster1_x1_end - cluster1_x1_start) + cluster1_x1_start
cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x2_start

cluster2_num_samples = 10
cluster2_x1_start = 10
cluster2_x1_end = 15
cluster2_x2_start = 8
cluster2_x2_end = 12
cluster2_x1 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x1 = cluster2_x1 * (cluster2_x1_end - cluster2_x1_start) + cluster2_x1_start
cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x2_start
```

In [49]:

```python
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[49]:

```
array([[ 1.83174911,  4.10730303],
       [ 2.11578439,  5.2125791 ],
       [ 1.5156703 ,  4.85047576],
       [ 4.36017569,  2.74417868],
       [ 0.74402825,  4.57077987],
       [ 2.61606724,  4.50472895],
       [ 4.2029517 ,  2.38858208],
       [ 4.70363733,  3.95728384],
       [ 0.56939275,  2.88562526],
       [ 0.75701584,  3.86799548],
       [13.18958296, 10.67333822],
       [11.04502646,  9.5337572 ],
       [12.77167313,  9.89906935],
       [10.61786228, 11.16242696],
       [11.96170064, 10.90483515],
       [13.32795872,  8.13685395],
       [14.10869613, 11.37483659],
       [11.22431108,  9.51565487],
       [10.07148386,  9.65055708],
       [12.5185522 ,  8.64144722]])
```

In [50]:

```python
matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```
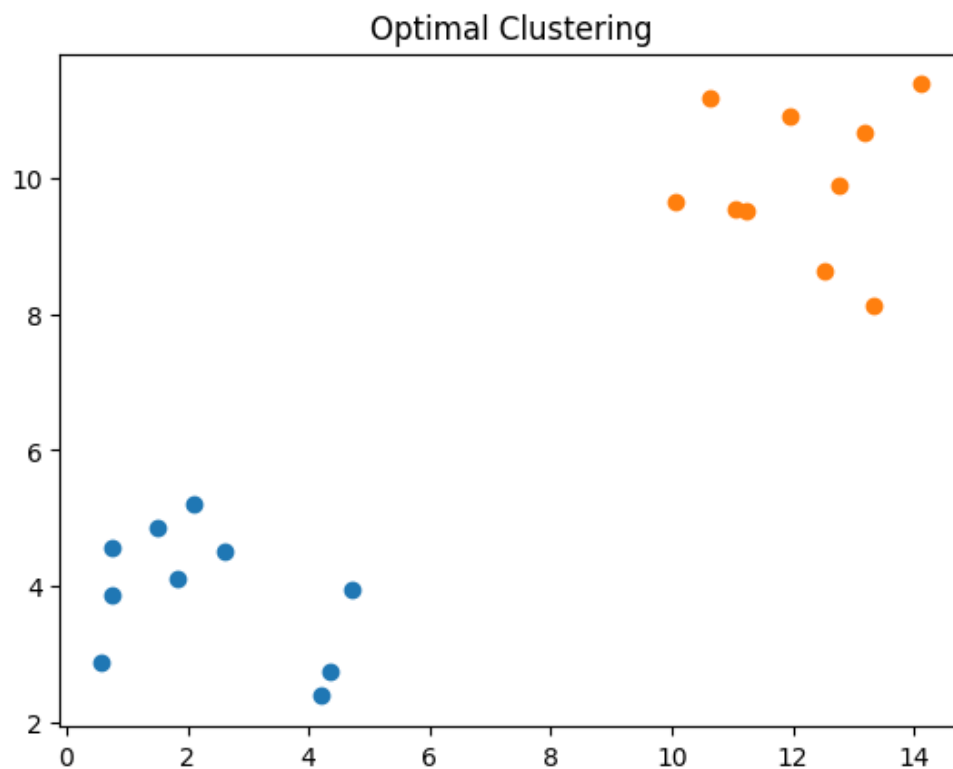

In [52]:

```python
def euclidean_distance(X, Y):
    return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

In [53]:

```python
def cluster_data(solution, solution_idx):
    global num_cluster, data
    feature_vector_length = data.shape[1]
    cluster_centers = []
    all_clusters_dists = []
    clusters = []
    clusters_sum_dist = []

    for clust_idx in range(num_clusters):
        cluster_centers.append(solution[feature_vector_length*clust_idx:feature_vector_length*(clust_
        cluster_center_dists = euclidean_distance(data, cluster_centers[clust_idx])
        all_clusters_dists.append(numpy.array(cluster_center_dists))
    cluster_centers = numpy.array(cluster_centers)
    all_clusters_dists = numpy.array(all_clusters_dists)

    cluster_indices = numpy.argmin(all_clusters_dists, axis=0)
    for clust_idx in range(num_clusters):
        clusters.append(numpy.where(cluster_indices == clust_idx)[0])

        if len(clusters[clust_idx]) == 0:
            clusters_sum_dist.append(0)
        else:
            clusters_sum_dist.append(numpy.sum(all_clusters_dists[clust_idx, clusters[clust_idx]]))
    clusters_sum_dist = numpy.array(clusters_sum_dist)
    return cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist
```

In [54]:

```python
def fitness_func(ga_instance,solution, solution_idx):
    _, _, _, _, clusters_sum_dist = cluster_data(solution, solution_idx)
    fitness = 1.0 / (numpy.sum(clusters_sum_dist) + 0.00000001)
    return fitness
```

In [55]:

```python
num_clusters = 2
num_genes = num_clusters * data.shape[1]

ga_instance = pygad.GA(num_generations=100,
                       sol_per_pop=10,
                       num_parents_mating=5,
                       init_range_low=-6,
                       init_range_high=20,
                       keep_parents=2,
                       num_genes=num_genes,
                       fitness_func=fitness_func,
                       suppress_warnings=True)

ga_instance.run()
```

In [56]:

```python
best_solution, best_solution_fitness, best_solution_idx = ga_instance.best_solution()
print("Best solution is {bs}".format(bs=best_solution))
print("Fitness of the best solution is {bsf}".format(bsf=best_solution_fitness))
print("Best solution found after {gen} generations".format(gen=ga_instance.best_solution_generation)
```
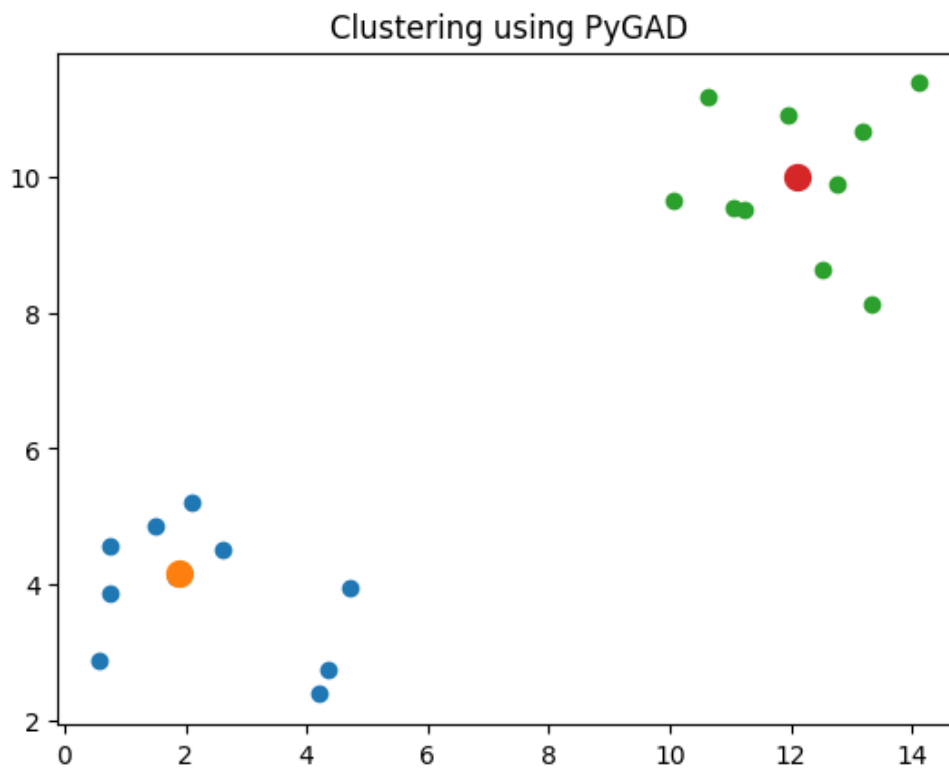
```
Best solution is [ 1.90870677  4.16827636 12.1144465   9.99614103]
Fitness of the best solution is 0.03266617392014203
Best solution found after 62 generations
```

In [57]:

```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist=cluster_data(best_
```

In [58]:

```python
for cluster_idx in range(num_clusters):
    cluster_x = data[clusters[cluster_idx], 0]
    cluster_y = data[clusters[cluster_idx], 1]
    matplotlib.pyplot.scatter(cluster_x, cluster_y)
    matplotlib.pyplot.scatter(cluster_centers[cluster_idx, 0], cluster_centers[cluster_idx, 1], linew
matplotlib.pyplot.title("Clustering using PyGAD")
matplotlib.pyplot.show()
```


In [ ]: