```
import numpy as np


points = np.array([[0.1, 0.6],
                   [0.15, 0.71],
                   [0.08, 0.9],
                   [0.16, 0.85],
                   [0.2, 0.3],
                   [0.25, 0.5],
                   [0.24, 0.1],
                   [0.3, 0.2]])


m1 = np.array([0.1, 0.6])
m2 = np.array([0.3, 0.2])


def euclidean_distance(x1, y1, x2, y2):
    return np.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)


# K-means clustering
def kmeans(points, m1, m2):
    clusters = {}
    while True:
        for point in points:
            dist_to_m1 = euclidean_distance(point[0], point[1], m1[0], m1[1])
            dist_to_m2 = euclidean_distance(point[0], point[1], m2[0], m2[1])
            if dist_to_m1 < dist_to_m2:
                if 'C1' in clusters:
                    clusters['C1'].append(point)
                else:
                    clusters['C1'] = [point]
            else:
                if 'C2' in clusters:
                    clusters['C2'].append(point)
                else:
                    clusters['C2'] = [point]

        new_m1 = np.mean(clusters['C1'], axis=0)
        new_m2 = np.mean(clusters['C2'], axis=0)

        if np.array_equal(new_m1, m1) and np.array_equal(new_m2, m2):
            break

        m1 = new_m1
        m2 = new_m2
        clusters.clear()

    return clusters, m1, m2


clusters, updated_m1, updated_m2 = kmeans(points, m1, m2)

# Results
print("Clusters:", clusters)
print("Updated m1:", updated_m1)
print("Updated m2:", updated_m2)
```

```
Clusters: {'C1': [array([0.1, 0.6]), array([0.15, 0.71]), array([0.08, 0.9 ]), array([0.16, 0.85]), array([0.25, 0.5 ])], 'C2': [arr
Updated m1: [0.148 0.712]
Updated m2: [0.24666667 0.2       ]
```

```
point_p6 = np.array([0.25, 0.5])


for cluster, points in clusters.items():
    if np.in1d(point_p6, points).any():
        print("Point P6 belongs to cluster", cluster)
```

```
Point P6 belongs to cluster C1
```

```
if 'C2' in clusters:
    print("Population of cluster around m2:", len(clusters['C2']))
else:
    print("Cluster around m2 is empty")
```

```
Population of cluster around m2: 3
```

```
print("Updated m1:", updated_m1)
print("Updated m2:", updated_m2)
```

```
Updated m1: [0.148 0.712]
Updated m2: [0.24666667 0.2       ]
```

Start coding or generate with AI.