Clustering Implementation

Using k-mean algorithm

In [1]:
```
from clustering_practice import *
```

Generating random clusters

We create a random number of clusters, consisted of random number of points in a space

In [2]:
```
x_max=10
y_max=10
z_max=10
radius=1
min_groups=3
max_groups=7
k= 2

points = gerenate_random_vectors(x_max,y_max,z_max,radius,min_groups,max_groups
```

```
Random Vector Function Complete. 425 numbers of points was created in 7 group
s.
```

The mean-k algorithm

Performed for k=2 to 10 to find the best clustering

In [3]:
```
clusters, k = clustering_k(points,2)
```

```
-------------
for k= 2 :  Iterations 3 -- Obj Val 24.476
-------------
for k= 3 :  Iterations 14 -- Obj Val 18.644
-------------
for k= 4 :  Iterations 3 -- Obj Val 10.650
-------------
for k= 5 :  Iterations 3 -- Obj Val 7.322
-------------
for k= 6 :  Iterations 4 -- Obj Val 15.466
-------------
for k= 7 :  Iterations 13 -- Obj Val 8.297
-------------
for k= 8 :  Iterations 4 -- Obj Val 7.079
-------------
for k= 9 :  Iterations 10 -- Obj Val 16.229
-------------
for k= 10 :  Iterations 11 -- Obj Val 8.219
```

In [4]:
```
print("The best clustering k is", str(k))
```

```
The best clustering k is 8
```

Plotting

Based on the best k value

In [5]:
```
fig = plt.figure()
ax=plt.axes(projection="3d")

for cluster in clusters:
        plot_points(cluster.point_list, plt, ax)

plt.show()
```