In [1]:
```
import numpy as np
import pandas as pd
import pulp
import itertools
import gmaps
import googlemaps
import matplotlib.pyplot as plt
import json
import time


f = open('pelanggan.json')
data = json.load(f)
# get customer demand and location data
df = pd.DataFrame(data)
print(df)

# get distance data
distance = np.loadtxt('distance.txt')
# print(distance)

# customer count ('0' is depot)
customer_count = 16

# the number of vehicle
vehicle_count = 10

# the capacity of vehicle
vehicle_capacity = 250

# fix random seed
np.random.seed(seed=777)
```

| | name | latitude | longitude | demand |
|---|---|---|---|---|
| 0 | Depo-Gas Station - Pertamina - 34-15316 | -6.293370 | 106.648566 | 0 |
| 1 | Customer 1 | -6.308553 | 106.653698 | 46 |
| 2 | Customer 2 | -6.306463 | 106.653170 | 26 |
| 3 | Customer 3 | -6.313126 | 106.652578 | 26 |
| 4 | Customer 4 | -6.314943 | 106.652226 | 27 |
| 5 | Customer 5 | -6.312494 | 106.646621 | 16 |
| 6 | Customer 6 | -6.301097 | 106.642021 | 56 |
| 7 | Customer 7 | -6.303528 | 106.641497 | 26 |
| 8 | Customer 8 | -6.314158 | 106.652149 | 30 |
| 9 | Customer 9 | -6.309986 | 106.653968 | 56 |
| 10 | Customer 10 | -6.300227 | 106.648346 | 65 |
| 11 | Customer 11 | -6.296925 | 106.647788 | 46 |
| 12 | Customer 12 | -6.301284 | 106.641789 | 33 |
| 13 | Customer 13 | -6.303443 | 106.641643 | 17 |
| 14 | Customer 14 | -6.309696 | 106.641600 | 39 |
| 15 | Customer 15 | -6.313765 | 106.643033 | 16 |
| 16 | Customer 16 | -6.312273 | 106.660423 | 53 |
| 17 | Customer 17 | -6.312349 | 106.658869 | 31 |
| 18 | Customer 18 | -6.316018 | 106.653213 | 35 |
| 19 | Customer 19 | -6.318031 | 106.654294 | 64 |
| 20 | Customer 20 | -6.321640 | 106.654045 | 11 |
| 21 | Customer 21 | -6.296962 | 106.646390 | 13 |
| 22 | Customer 22 | -6.300260 | 106.647090 | 36 |
| 23 | Customer 23 | -6.303233 | 106.646141 | 33 |
| 24 | Customer 24 | -6.304129 | 106.650905 | 61 |
| 25 | Customer 25 | -6.296907 | 106.637953 | 42 |
| 26 | Customer 26 | -6.294232 | 106.640760 | 38 |
| 27 | Customer 27 | -6.234953 | 106.652488 | 22 |
| 28 | Customer 28 | -6.236532 | 106.656668 | 58 |
| 29 | Customer 29 | -6.223981 | 106.651458 | 57 |
| 30 | Customer 30 | -6.230960 | 106.627495 | 41 |
| 31 | Customer 31 | -6.259449 | 106.622465 | 23 |
| 32 | Customer 32 | -6.263024 | 106.619555 | 43 |
| 33 | Customer 33 | -6.272375 | 106.642678 | 48 |
| 34 | Customer 34 | -6.277135 | 106.664548 | 70 |
| 35 | Customer 35 | -6.297099 | 106.670204 | 50 |
| 36 | Customer 36 | -6.307720 | 106.675834 | 15 |
| 37 | Customer 37 | -6.316209 | 106.648896 | 67 |
| 38 | Customer 38 | -6.315513 | 106.650969 | 40 |
| 39 | Customer 39 | -6.316921 | 106.646948 | 43 |
| 40 | Customer 40 | -6.313991 | 106.649574 | 17 |
| 41 | Customer 41 | -6.219898 | 106.623173 | 43 |
| 42 | Customer 42 | -6.220871 | 106.615671 | 66 |
| 43 | Customer 43 | -6.279793 | 106.667599 | 48 |
| 44 | Customer 44 | -6.283248 | 106.668483 | 22 |
| 45 | Customer 45 | -6.296213 | 106.664227 | 11 |
| 46 | Customer 46 | -6.292639 | 106.663978 | 66 |
| 47 | Customer 47 | -6.292579 | 106.677626 | 41 |
| 48 | Customer 48 | -6.279082 | 106.665420 | 36 |
| 49 | Customer 49 | -6.277922 | 106.632067 | 19 |
| 50 | Customer 50 | -6.284739 | 106.626531 | 35 |

In [2]:
```
start_time = time.time()
# solve with pulp
for vehicle_count in range(1,vehicle_count+1):

    # definition of LpProblem instance
    problem = pulp.LpProblem("CVRP", pulp.LpMinimize)

    # definition of variables which are 0/1
    x = [[[pulp.LpVariable("x%s_%s_%s"%(r,i,j), cat="Binary") if i != j else None for j in range(customer_count)] for i in range(
    # add objective function
    problem += pulp.lpSum(distance[i][j] * x[r][i][j] if i != j else 0
                          for j in range(customer_count)
                          for i in range (customer_count)
                          for r in range(vehicle_count))
    # constraints
    # foluma (2)
    for j in range(1, customer_count):
        problem += pulp.lpSum(x[r][i][j] if i != j else 0
                              for r in range(vehicle_count)
                              for i in range(customer_count)) == 1
    # foluma (3)
    for r in range(vehicle_count):
        problem += pulp.lpSum(x[r][0][j] for j in range(1,customer_count)) == 1
```

```
    problem += pulp.lpSum(x[r][i][0] for i in range(1,customer_count)) == 1

    # foluma (4)
    for r in range(vehicle_count):
        for j in range(customer_count):
            problem += pulp.lpSum(x[r][i][j] if i != j else 0
                                  for i in range(customer_count)) -  pulp.lpSum(x[r][j][i] for i in range(customer_count)) == 0

    #foluma (5)
    for r in range(vehicle_count):
        problem += pulp.lpSum(df.demand[j] * x[r][i][j] if i != j else 0 for i in range(customer_count) for j in range (1,custome

     # fomula (6)
    subtours = []
    for i in range(2,customer_count):
         subtours += itertools.combinations(range(1,customer_count), i)

    for s in subtours:
        problem += pulp.lpSum(x[r][i][j] if i !=j else 0 for i, j in itertools.permutations(s,2) for r in range(vehicle_count)) <


    # print vehicle_count which needed for solving problem
    # print calculated minimum distance value
    if problem.solve() == 1:
        print('Vehicle Requirements:', vehicle_count)
        print('Moving Distance:', pulp.value(problem.objective))
        break

#run your code
print("--- %s seconds ---" % (time.time() - start_time))
```

```
Vehicle Requirements: 3
Moving Distance: 20628.0
--- 2192.8745658397675 seconds ---
```

In [3]:
```
print(df)
# visualization : plotting on google maps
fig = gmaps.figure()
layer = []
color_list = ["red","blue","green"]

for r in range(vehicle_count):
    for i in range(customer_count):
        for j in range(customer_count):
            if i != j and pulp.value(x[r][i][j]) == 1:
                layer.append(gmaps.directions.Directions(
                    (df.latitude[i],df.longitude[i]),
                    (df.latitude[j],df.longitude[j]),
                    mode='car',stroke_color=color_list[r],stroke_opacity=1.0, stroke_weight=5.0))

for i in range(len(layer)):
    fig.add_layer(layer[i])

fig
```

| | name | latitude | longitude | demand |
|---|---|---|---|---|
| 0 | Depo-Gas Station - Pertamina - 34-15316 | -6.293370 | 106.648566 | 0 |
| 1 | Customer 1 | -6.308553 | 106.653698 | 46 |
| 2 | Customer 2 | -6.306463 | 106.653170 | 26 |
| 3 | Customer 3 | -6.313126 | 106.652578 | 26 |
| 4 | Customer 4 | -6.314943 | 106.652226 | 27 |
| 5 | Customer 5 | -6.312494 | 106.646621 | 16 |
| 6 | Customer 6 | -6.301097 | 106.642021 | 56 |
| 7 | Customer 7 | -6.303528 | 106.641497 | 26 |
| 8 | Customer 8 | -6.314158 | 106.652149 | 30 |
| 9 | Customer 9 | -6.309986 | 106.653968 | 56 |
| 10 | Customer 10 | -6.300227 | 106.648346 | 65 |
| 11 | Customer 11 | -6.296925 | 106.647788 | 46 |
| 12 | Customer 12 | -6.301284 | 106.641789 | 33 |
| 13 | Customer 13 | -6.303443 | 106.641643 | 17 |
| 14 | Customer 14 | -6.309696 | 106.641600 | 39 |
| 15 | Customer 15 | -6.313765 | 106.643033 | 16 |
| 16 | Customer 16 | -6.312273 | 106.660423 | 53 |
| 17 | Customer 17 | -6.312349 | 106.658869 | 31 |
| 18 | Customer 18 | -6.316018 | 106.653213 | 35 |
| 19 | Customer 19 | -6.318031 | 106.654294 | 64 |
| 20 | Customer 20 | -6.321640 | 106.654045 | 11 |
| 21 | Customer 21 | -6.296962 | 106.646390 | 13 |
| 22 | Customer 22 | -6.300260 | 106.647090 | 36 |
| 23 | Customer 23 | -6.303233 | 106.646141 | 33 |
| 24 | Customer 24 | -6.304129 | 106.650905 | 61 |
| 25 | Customer 25 | -6.296907 | 106.637953 | 42 |
| 26 | Customer 26 | -6.294232 | 106.640760 | 38 |
| 27 | Customer 27 | -6.234953 | 106.652488 | 22 |
| 28 | Customer 28 | -6.236532 | 106.656668 | 58 |
| 29 | Customer 29 | -6.223981 | 106.651458 | 57 |
| 30 | Customer 30 | -6.230960 | 106.627495 | 41 |
| 31 | Customer 31 | -6.259449 | 106.622465 | 23 |
| 32 | Customer 32 | -6.263024 | 106.619555 | 43 |
| 33 | Customer 33 | -6.272375 | 106.642678 | 48 |
| 34 | Customer 34 | -6.277135 | 106.664548 | 70 |
| 35 | Customer 35 | -6.297099 | 106.670204 | 50 |
| 36 | Customer 36 | -6.307720 | 106.675834 | 15 |
| 37 | Customer 37 | -6.316209 | 106.648896 | 67 |
| 38 | Customer 38 | -6.315513 | 106.650969 | 40 |
| 39 | Customer 39 | -6.316921 | 106.646948 | 43 |
| 40 | Customer 40 | -6.313991 | 106.649574 | 17 |
| 41 | Customer 41 | -6.219898 | 106.623173 | 43 |
| 42 | Customer 42 | -6.220871 | 106.615671 | 66 |
| 43 | Customer 43 | -6.279793 | 106.667599 | 48 |
| 44 | Customer 44 | -6.283248 | 106.668483 | 22 |
| 45 | Customer 45 | -6.296213 | 106.664227 | 11 |
| 46 | Customer 46 | -6.292639 | 106.663978 | 66 |
| 47 | Customer 47 | -6.292579 | 106.677626 | 41 |
| 48 | Customer 48 | -6.279082 | 106.665420 | 36 |
| 49 | Customer 49 | -6.277922 | 106.632067 | 19 |
| 50 | Customer 50 | -6.284739 | 106.626531 | 35 |

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

```
[directions layer] You are not allowed to use Google's directions service
```

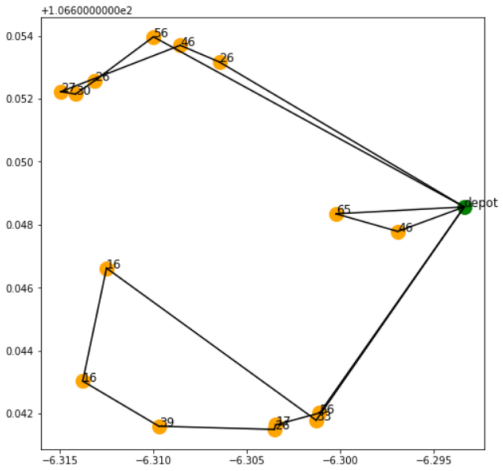
```
[directions layer] You are not allowed to use Google's directions service
```

In [4]:

```
# visualization : plotting with matplolib
plt.figure(figsize=(8,8))
for i in range(customer_count):
    if i == 0:
        plt.scatter(df.latitude[i], df.longitude[i], c='green', s=200)
        plt.text(df.latitude[i], df.longitude[i], "depot", fontsize=12)
    else:
        plt.scatter(df.latitude[i], df.longitude[i], c='orange', s=200)
        plt.text(df.latitude[i], df.longitude[i], str(df.demand[i]), fontsize=12)

for k in range(vehicle_count):
    for i in range(customer_count):
        for j in range(customer_count):
            if i != j and pulp.value(x[k][i][j]) == 1:
                plt.plot([df.latitude[i], df.latitude[j]], [df.longitude[i], df.longitude[j]], c="black")

plt.show()
```




In [5]:

```
res = [None] * vehicle_count
for k in range(vehicle_count):
    res[k] = [None] * customer_count
    for i in range(customer_count):
        for j in range(customer_count):
            if i != j and pulp.value(x[k][i][j]) == 1:
#                 print("i:"+str(i)+" ,j:"+str(j)+" ,k:"+str(k))
                res[k][i] = j
# print(res)
for k in range(vehicle_count):
    capacity = vehicle_capacity
    rts = "Vechicle "+str(k)+" Routes = 0["+str(capacity)+"] --> "
    c_rts = res[k][0]
    while(c_rts != 0):
        capacity -= df.iloc[c_rts,3]
        rts += str(c_rts)+"["+str(capacity)+"] --> "
        c_rts = res[k][c_rts]
    rts+= "0"
    print(rts)
```

```
Vechicle 0 Routes = 0[250] --> 12[217] --> 5[201] --> 15[185] --> 14[146] --> 7[120] --> 13[103] --> 6[47] --> 0
Vechicle 1 Routes = 0[250] --> 9[194] --> 3[168] --> 8[138] --> 4[111] --> 1[65] --> 2[39] --> 0
Vechicle 2 Routes = 0[250] --> 10[185] --> 11[139] --> 0
```

In [ ]: