

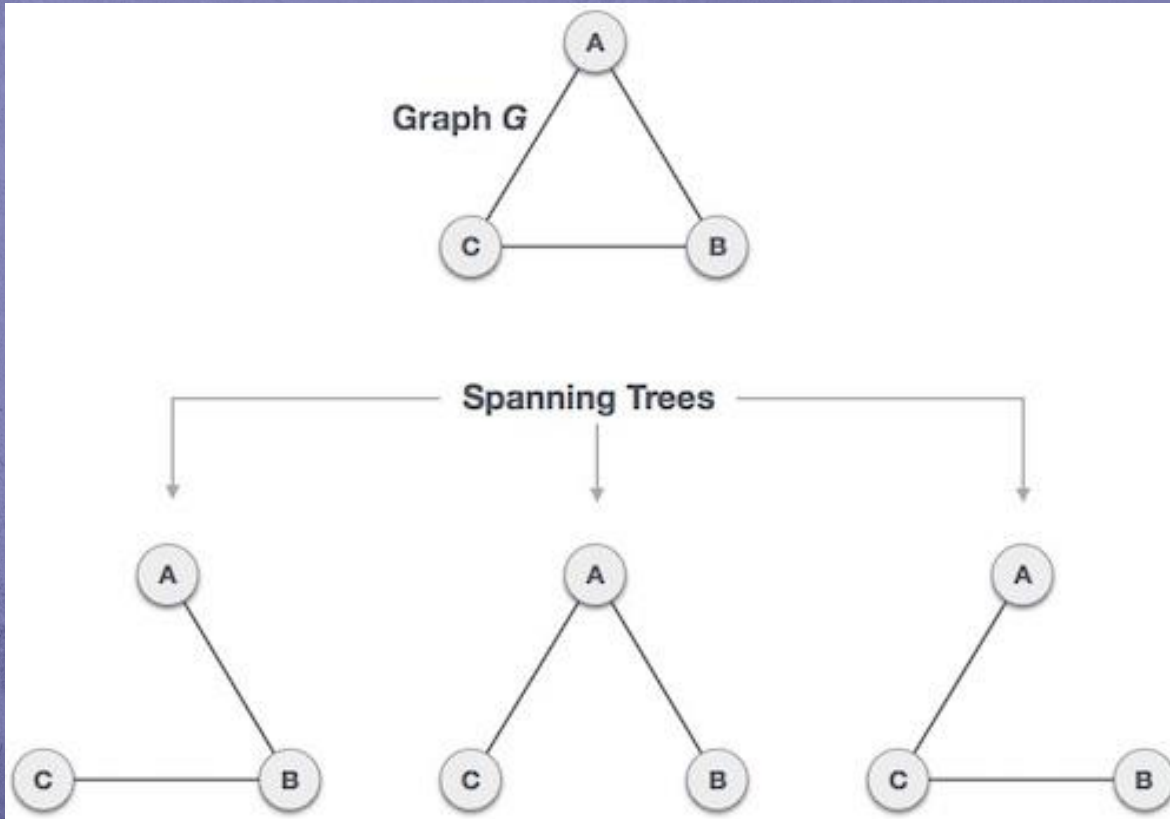
ALGORITMA PEMROGRAMAN II

NAMA: ROHMATULLOH FADHILAH

NIM : 212410103026

KELAS : ALGO II E

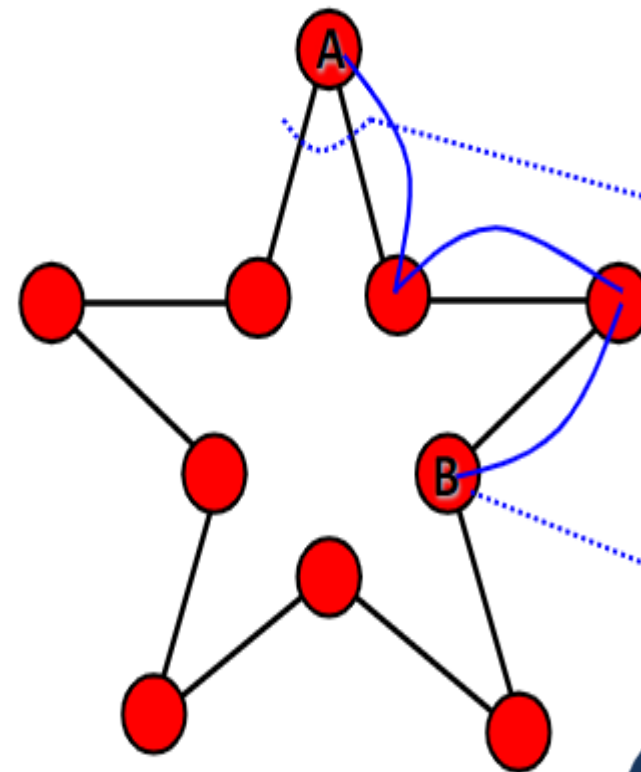
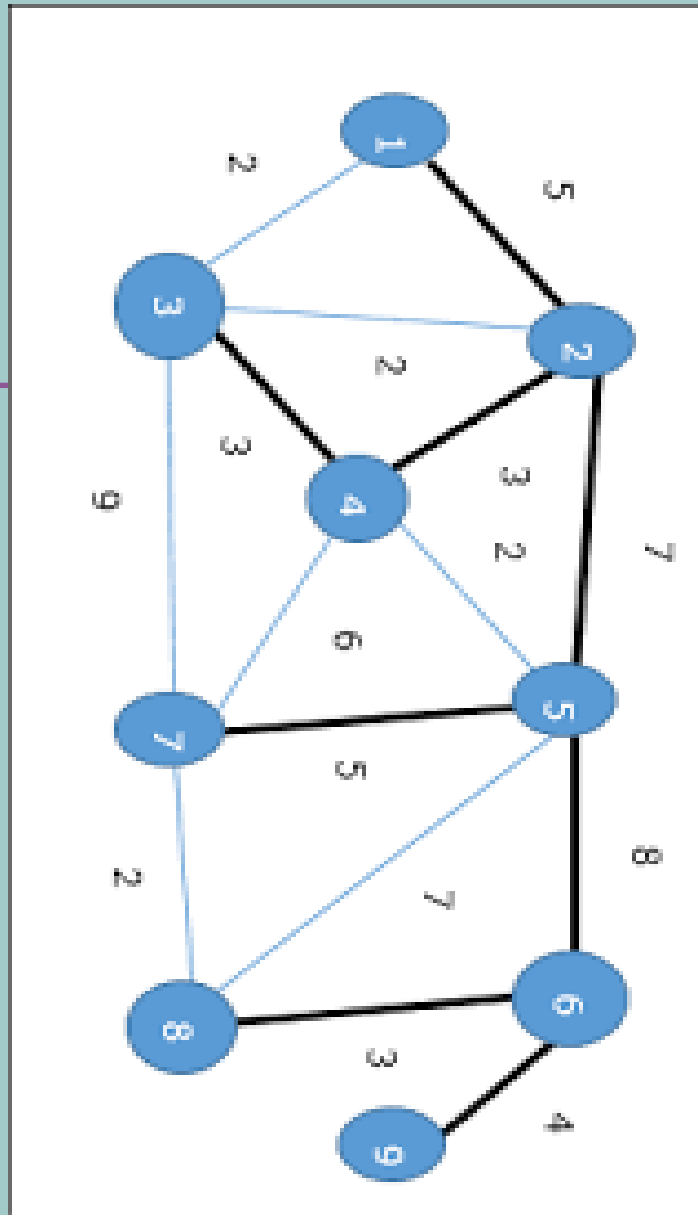
SPANNING TREE



Spanning Tree merupakan subset dari graf G yang memiliki semua simpul yang ditutupi dengan jumlah tepi yang seminimal mungkin.

Dapat ditarik kesimpulan bahwa setiap graf G terhubung dan tidak berarah memiliki setiap graf G terhubung dan tidak berarah setidaknya memiliki satu spanning yang merentang. Graf terputus tidak memiliki pohon merentang

karena tidak dapat direntang ke semua simpulnya.



Derajat = jumlah koneksi dengan titik

Derajat titik A = 2

Jarak = panjang lintasan terpendek pada dua titik

Jarak A ke B = 3

Diameter = jarak maksimum antara sembarang pasangan titik

Graf dengan diameter 5

MODULE

Math digunakan untuk
perhitungan matematika

Tkinter digunakan untuk
membuat tampilan layer GUI

```
1  ✓ import math
2    import tkinter as tk
3
```

PENETUAN TITIK

Code di samping merupakan sebuah fungsi yang sebuah titik sesuai dengan inputan yang di masukkan melalui file testcase nya. Dimana nanti file tascase yang berbentuk str akan di uraikan menjadi beberapa str lalu di casting menjadi int agar bisa di lakukan perhitungan dalam penentuan titiknya

Pada line 12 dan 13 adalah sebuah code yang berguna untuk membuat sebuah oval sesuai dengan ukura yang di minta.

```
6 def inptitik(n):
7     case_titik = []
8     for noUrut in range(n):
9         titik = input()
10        x= int(titik.split(' ')[0])
11        y= int(titik.split(' ')[1])
12        canvas.create_oval(x-10,y+10,x+10,y-10,fill='green')
13        canvas.create_text(x,y,text=noUrut)
14        case_titik.append([x,y])
15
16    return case_titik
17
```

MEMBUAT JARAK DAN MENCARI NILAI MINIMUM .

```
17
18 def jarak(titik1,titik2):
19     result = math.sqrt(math.pow(titik1[0]-titik2[0],2) + math.pow(titik1[1]-titik2[1],2))
20     return result
21
22 def min2d (list2d,extractIndex):
23     lengthSet = []
24     for index in list2d:
25         lengthSet.append( (index[extractIndex]) )
26     minimum = min(lengthSet)
27     for i in list2d:
28         if i[extractIndex] == minimum:
29             return i
```

PENGIMPEMENTASIAN DARI FUNGSI-SUNGSI SEBELUMNYA

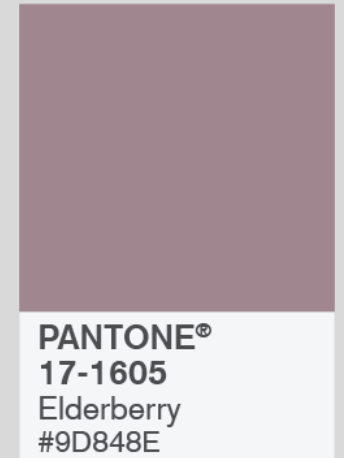
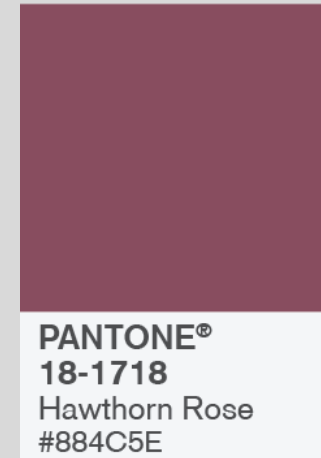
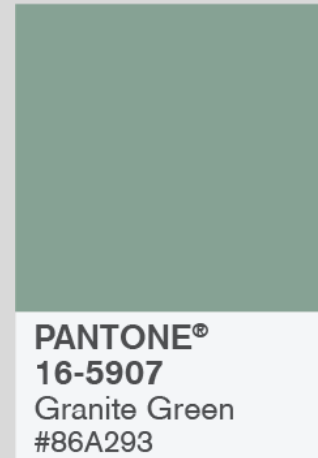
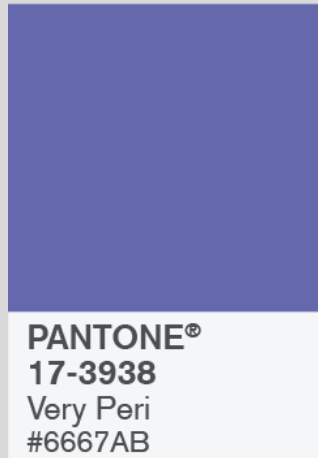
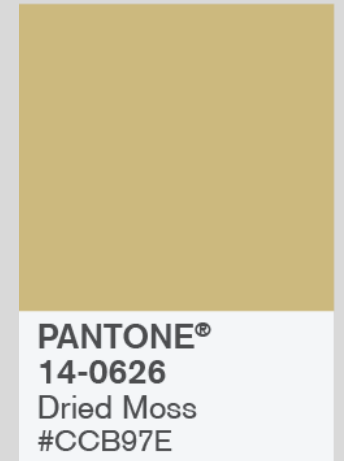
```
31
32 def MinSpanningTree (case):
33     titik =[]
34     cnVertikal = [case[0]] #permulaan titik pertama
35     discnVertikal = case[1:] #sisanya disconnected akan tersimpan disini
36     #[ [V1,V2,length] ] titik
37     x = len(case)-1
38     while x > 0:
39         temp1_titik = []
40         for cnver in cnVertikal: #cn = connected
41             temp2_titik = []
42
43             for dcver in discnVertikal : #dc = disconnected
44                 length = jarak(cnver, dcver)
45                 temp2_titik.append([cnver, dcver, length])
46
47             # mencari jarak terdekat dari kombinasi cnver
48             mintitik = min2d(temp2_titik,2)
49             temp1_titik.append(mintitik)
50
51             #memilih jarak terdekat diantara titik lainnya
52             short_titik = min2d(temp1_titik,2)
53             #extract ver yg dc dari list, index dc = 1
54             shortestver = short_titik[1]
55
56             #memindahkan titik yg dc menjadi cn
57             cnVertikal.append(shortestver)
58             discnVertikal.remove(shortestver)
59
60             titik.append(min2d(temp1_titik,2))
61
62             x-=1
63     return titik
```


PENETUAN UKURAN WINDOWS

```
65
66  cnv = tk.Canvas(window, height=1000, width=1000)
67  cnv.pack()
68  x = int(input())
69
70  case = inptitik(x)
71  titik = MinSpanningTree(case)
72
73  for i in range(len(titik)):
74      cnv.create_text((titik[i][0][0]+titik[i][1][0])/2,(titik[i][0][1]+titik[i][1][1])/2, text=i+1)
75      cnv.create_line(titik[i][0],titik[i][1])
76
77
78  window.mainloop()
79
```


TERIMA KASIH

BALANCING ACT



WELLSPRING



PANTONE®
16-6127
Greenbrier
#48996B



PANTONE®
16-2037
Foliage
#759F51



PANTONE®
18-0135
Treetop
#436A2F



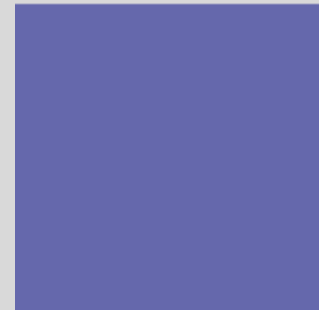
PANTONE®
14-0647
Celery
#CFBF54



PANTONE®
17-0949
Chai Tea
#B3832F



PANTONE®
18-3533
Dewberry
#8C5896



PANTONE®
17-3938
Very Peri
#6667AB



PANTONE®
14-4809
Eggshell Blue
#A1CAC9

THE STAR OF THE SHOW



PANTONE®
19-4007
Anthracite
#29282D



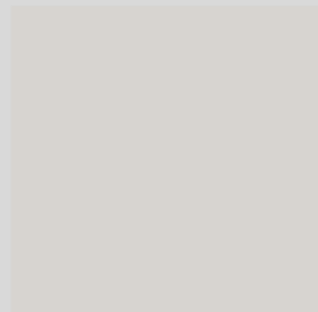
PANTONE®
18-3908
Volcanic Glass
#625C60



PANTONE®
16-1312
Deep Taupe
#7C6560



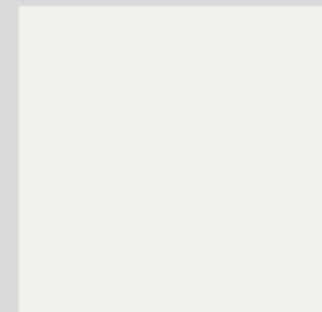
PANTONE®
16-1105
Plaza Taupe
#AEA392



PANTONE®
13-0002
White Sand
#D8D4D0



PANTONE®
17-1115
Petrified Oak
#8E7961

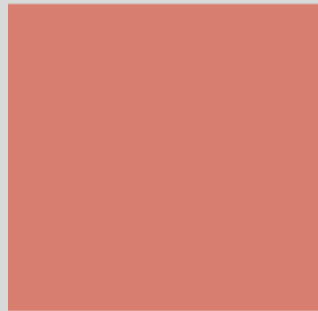


PANTONE®
11-4201
Cloud Dance
#F0EDE8

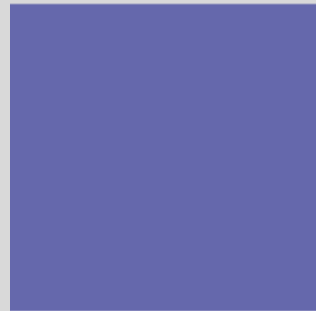


PANTONE®
17-3938
Very Peri
#6667AB

AMUSEMENTS



PANTONE®
17-1341
Tawny Orange
#D77E6F



PANTONE®
17-3938
Very Peri
#6667AB



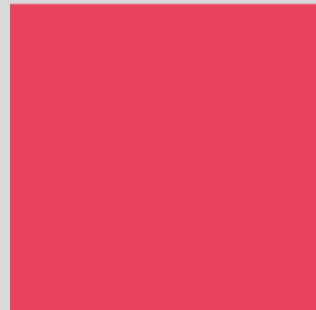
PANTONE®
15-1040
Iced Coffee
#B38F6A



PANTONE®
18-2133
Pink Flambé
#D75078



PANTONE®
15-2718
Fuchsia Pink
#E288B6



PANTONE®
17-1755
Paradise Pink
#E9445D



PANTONE®
13-0932
Cornsilk
#EEC272



PANTONE®
16-4411
Tourmaline
#85A0A9

USE THESE COLORS IN ANY POWERPOINT PRESENTATION

1. Select a shape or text box border. When you do that, the **Shape Format** tab appears.

***Tip:** To change multiple shapes or text boxes, click the first shape or text box, and then press and hold Ctrl while you click the other shapes or text boxes.*

2. On the **Shape Format** tab, select **Shape Fill > More Fill Colors**.
3. In the **Colors** box, select the **Custom** tab.
4. Enter the **Hex** value of the color you want to use.

