

```
import networkx as nx
```

```
# Tulislah nama Saudara dan NIM
```

```
NAMA :Devi Intan Nurisma Putri
```

```
NIM : 123220184
```

```
# Kerjakan, simpan, dan simpanlah dalam bentuk pdf (Save as PDF)
```

```
# selanjutnya unggah ke spada
```

```
#buat graf interaksi protein
```

```
G = nx.Graph()
```

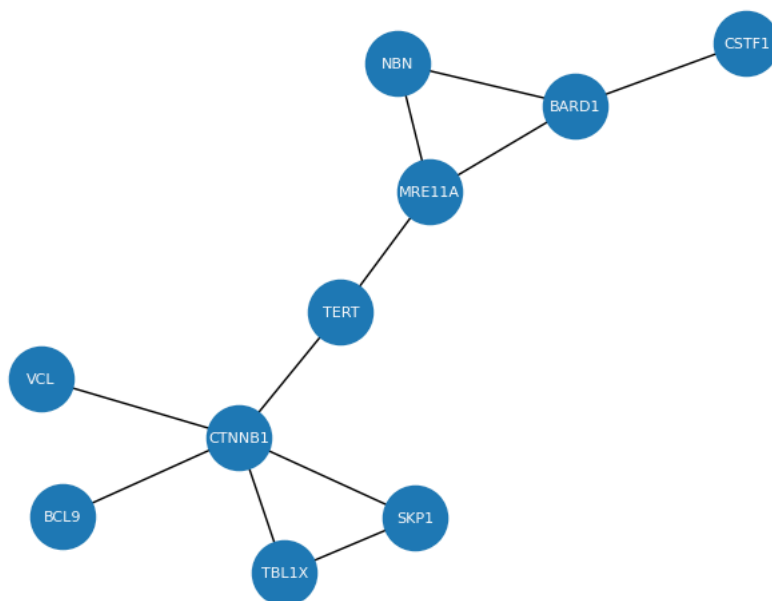
```
edges_to_add = [('CSTF1', 'BARD1'), ('BARD1', 'MRE11A'), ('BARD1', 'NBN'), ('MRE11A', 'NBN'), ('MRE11A', 'TERT'),  
                ('TERT', 'CTNNB1'), ('CTNNB1', 'VCL'), ('CTNNB1', 'BCL9'), ('CTNNB1', 'TBL1X'), ('CTNNB1', 'SKP1'),  
                ('TBL1X', 'SKP1') ]
```

```
G.add_edges_from(edges_to_add)
```

```
pos = nx.kamada_kawai_layout(G)
```

```
# draw the graph
```

```
nx.draw(G,  
        pos=pos,  
        with_labels=True,  
        #node_color='blue',  
        node_size=1200,  
        font_color='white',  
        font_size=8,  
        )
```



```
# soal contoh a. Tentukan berapa banyaknya degree pada protein BARD1
```

```
# jawab
```

```
degree_bard1 = G.degree('BARD1')
```

```
print(f"Degree (derajat) dari protein BARD1 adalah: {degree_bard1}")
```

```
# jawaban yang benar adalah 3
```



```
Degree (derajat) dari protein BARD1 adalah: 3
```

```
# soal 1. Cetaklah semua simpul pada berupa leaf/daun, yaitu simpul yang berderajat 1
```

```
# jawab
```

```
leaf_nodes = [node for node in G.nodes() if G.degree(node) == 1]
```

```
print("Simpul-simpul daun (derajat 1):", leaf_nodes)
```

```
# jawaban benar adalah CSTF1, VCL serta BCL9
```



```
Simpul-simpul daun (derajat 1): ['CSTF1', 'VCL', 'BCL9']
```

```
# Soal 2. tentukan tetangga bersama antara 'MRE11A' dengan 'BARD1'
# jawab

common_neighbors = list(nx.common_neighbors(G, 'MRE11A', 'BARD1'))
print("Tetangga bersama antara 'MRE11A' dan 'BARD1':", common_neighbors)
```

#jawaban yang benar: NBN

↪ Tetangga bersama antara 'MRE11A' dan 'BARD1': ['NBN']

```
# Soal 3
# simpul dalam didefinisikan sebagai simpul yang tidak membentuk daun, artinya berderajat lebih dari 1
# hitung banyaknya simpul internal pada graph G tersebut
# Jawab

internal_nodes = [node for node in G.nodes() if G.degree(node) > 1]
print(f"Banyaknya simpul internal (derajat > 1): {len(internal_nodes)}")
```

#jawaban yang benar adalah 7

↪ Banyaknya simpul internal (derajat > 1): 7

```
# soal 4
# Berapa banyaknya lintasan dari protein 'TERT' menuju 'CSTF1'
# jawab

# Cari semua lintasan (simple paths) dari 'TERT' ke 'CSTF1'
all_paths = list(nx.all_simple_paths(G, source='TERT', target='CSTF1'))

# Hitung jumlah lintasan
jumlah_lintasan = len(all_paths)

# Cetak hasil
print(f"Banyaknya lintasan dari 'TERT' ke 'CSTF1': {jumlah_lintasan}")
```

jawaban benar: 2
penjelasan: ada DUA. yaitu (1) 'TERT', 'MRE11A', 'BARD1', 'CSTF1' dan (2) 'TERT', 'MRE11A', 'NBN', 'BARD1', 'CSTF1'

↪ Banyaknya lintasan dari 'TERT' ke 'CSTF1': 2

```
# Soal 5 berapa panjang lintasan terpendek dari protein 'TERT' menuju 'CSTF1'
# jawab
```

```
print("Jumlah simpul:", G.number_of_nodes())
```

jawaban benar : 3

↪ Jumlah simpul: 10

```
# Soal 6 tentukan lintasan terpanjang dari protein 'TERT' menuju 'CSTF1'
max(list(nx.all_simple_paths(G, 'TERT', 'CSTF1')))
# jawab
```

```
special_nodes = ['TERT', 'MRE11A', 'NBN', 'BARD1', 'CSTF1']
print("Jawaban akhir simpul khusus:", special_nodes)
```

#jawaban benar ['TERT', 'MRE11A', 'NBN', 'BARD1', 'CSTF1']

↪ Jawaban akhir simpul khusus: ['TERT', 'MRE11A', 'NBN', 'BARD1', 'CSTF1']

```
# Pengantar soal nomor 7
# degree centrality pada networkx diperoleh dengan menggunakan instruksi
dc = nx.degree_centrality(G)
# sehingga diperoleh degree centrality pada 'CTNNB1' bernilai 0.5556
dc['CTNNB1']
# nilai ini diperoleh dari degree dari 'CTNNB1' yaitu 5 dibagi dengan banyaknya simpul G -1,
# diperoleh nilai 5 /(10-1) =0.5556
```

↪ 0.5555555555555556

```
# Soal /. hitungan degree centrality pada protein CTNNB1 , yaitu tanpa menggunakan instruksi nx.degree_centrality
# namun gunakan rumus: degree 'CTNNB1' / (banyaknya simpul G -1)
# jawab

# Hitung degree (derajat) CTNNB1
degree_ctnnb1 = G.degree('CTNNB1')

# Hitung jumlah simpul
jumlah_simpul = G.number_of_nodes()

# Hitung degree centrality manual
degree_centrality_ctnnb1 = degree_ctnnb1 / (jumlah_simpul - 1)

# Cetak hasil
print(f"Degree centrality pada protein 'CTNNB1': {degree_centrality_ctnnb1}")
```

```
# jawaban yang benar 0.5555555555555556
```



```
Degree centrality pada protein 'CTNNB1': 0.5555555555555556
```

```
# JANGAN LUPA
# isi nama dan NIM di atas
# Simpan, dan simpanlah dalam bentuk pdf (Save as PDF)
# selanjutnya unggah ke spada
```