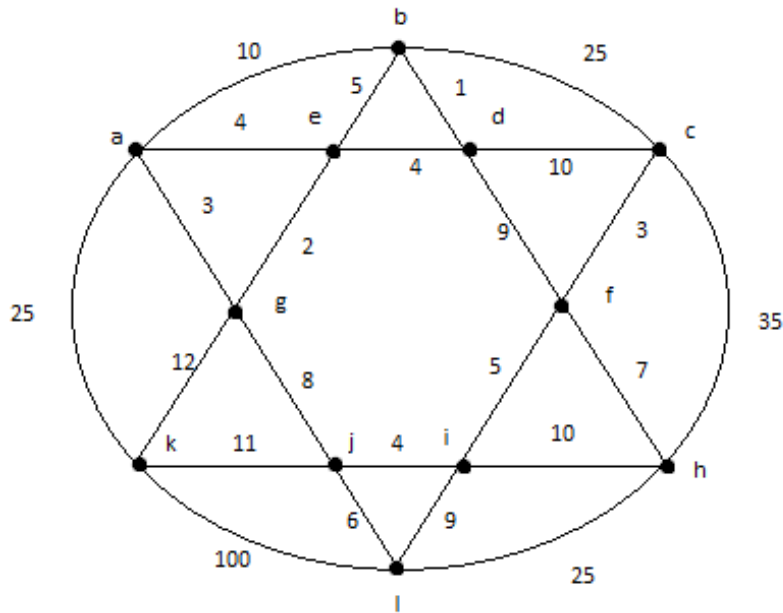
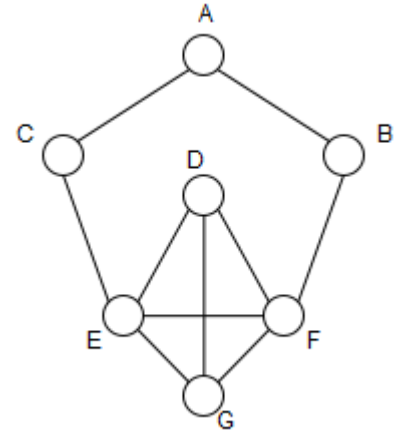


Solusi Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Graf, Pohon, dan Kompleksitas Algoritma  
 Dosen: Rinaldi Munir, Harlili  
 Senin, 2 Desember 2016  
 Waktu: 90 menit

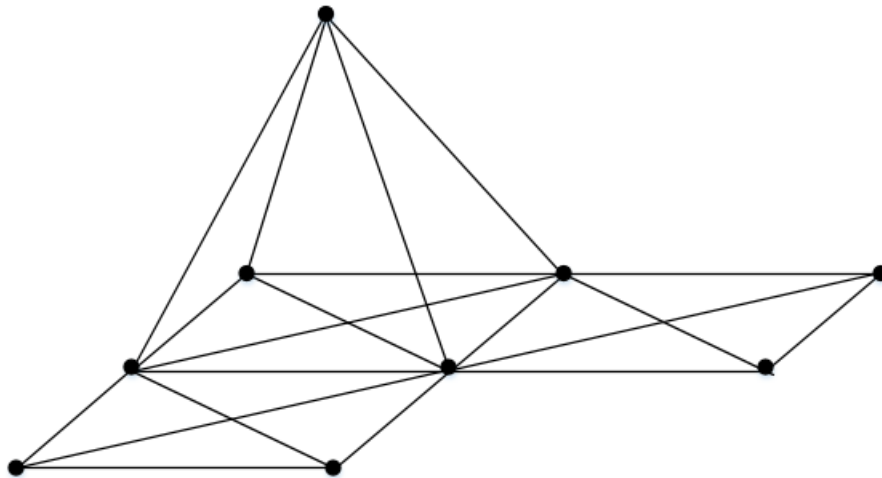
Graf untuk soal nomor 1



Graf untuk soal nomor 8



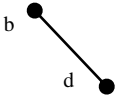
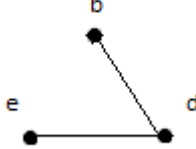
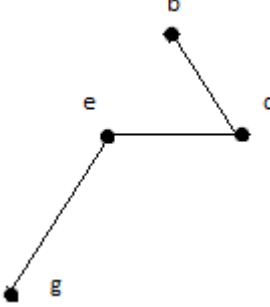
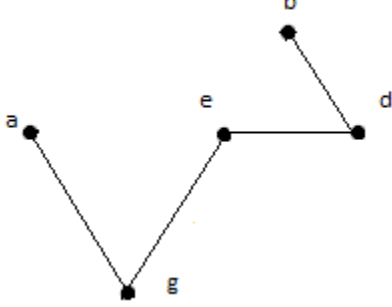
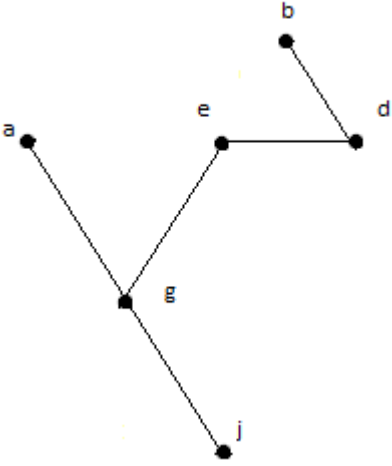
Graf untuk soal nomor 7



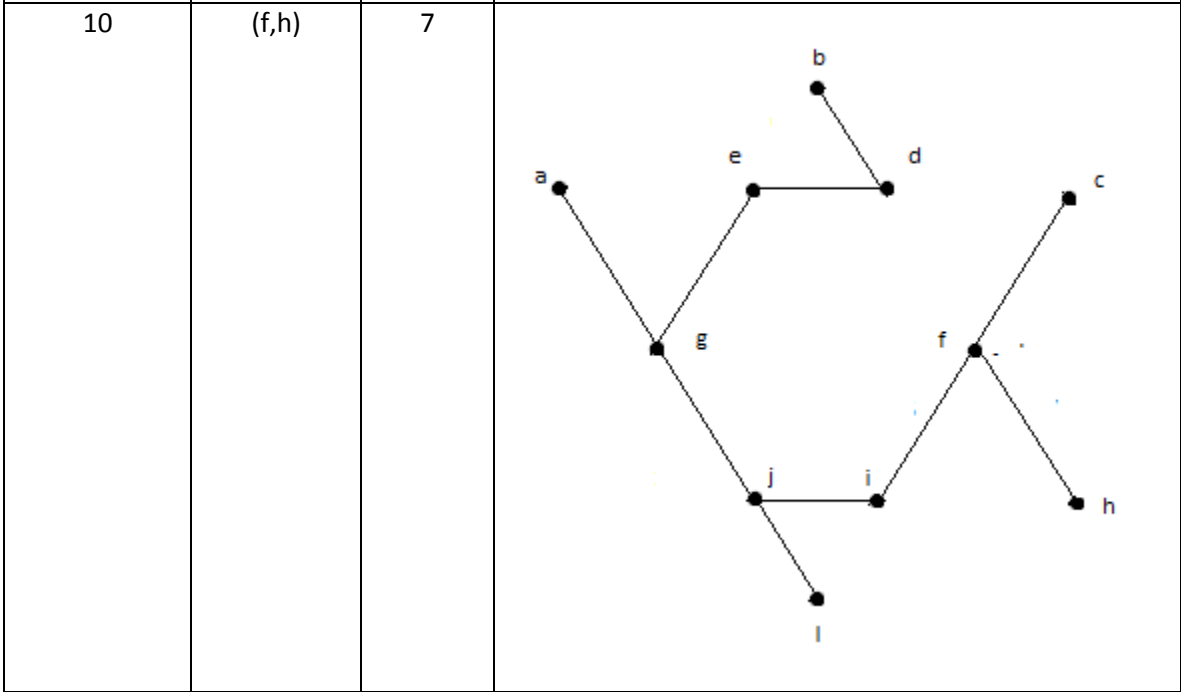
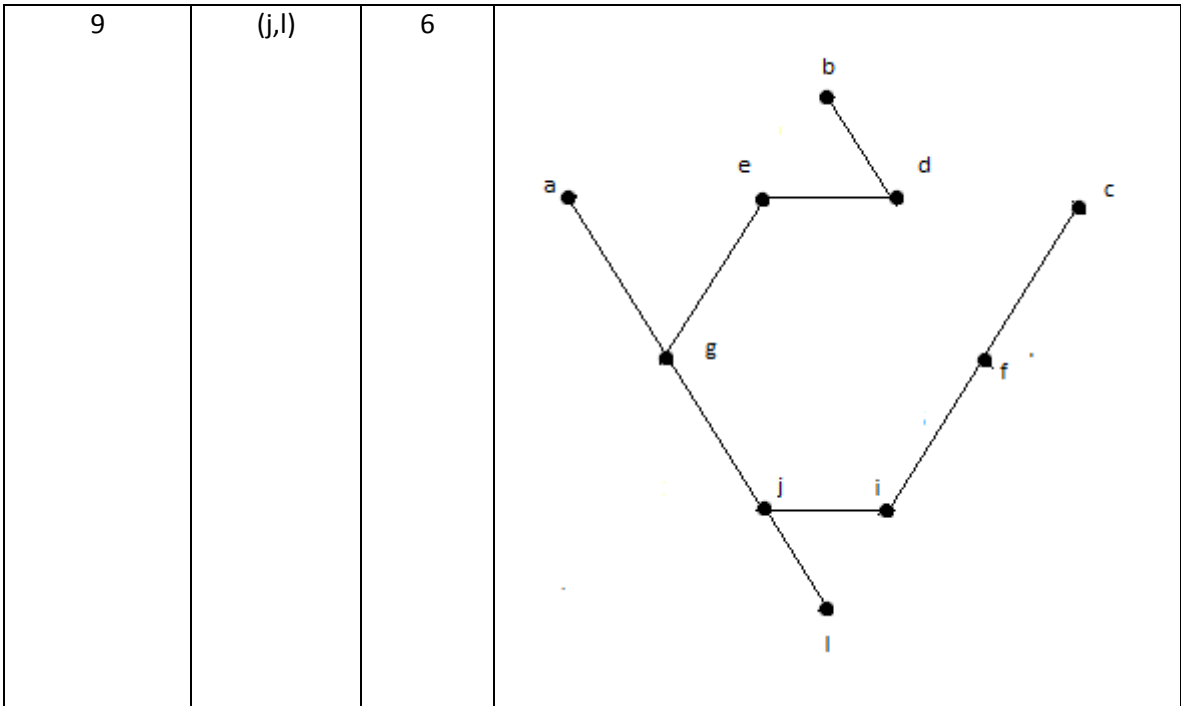
1. Carilah pohon merentang minimum dari graf soal nomor 1 dan tentukan bobot nya !

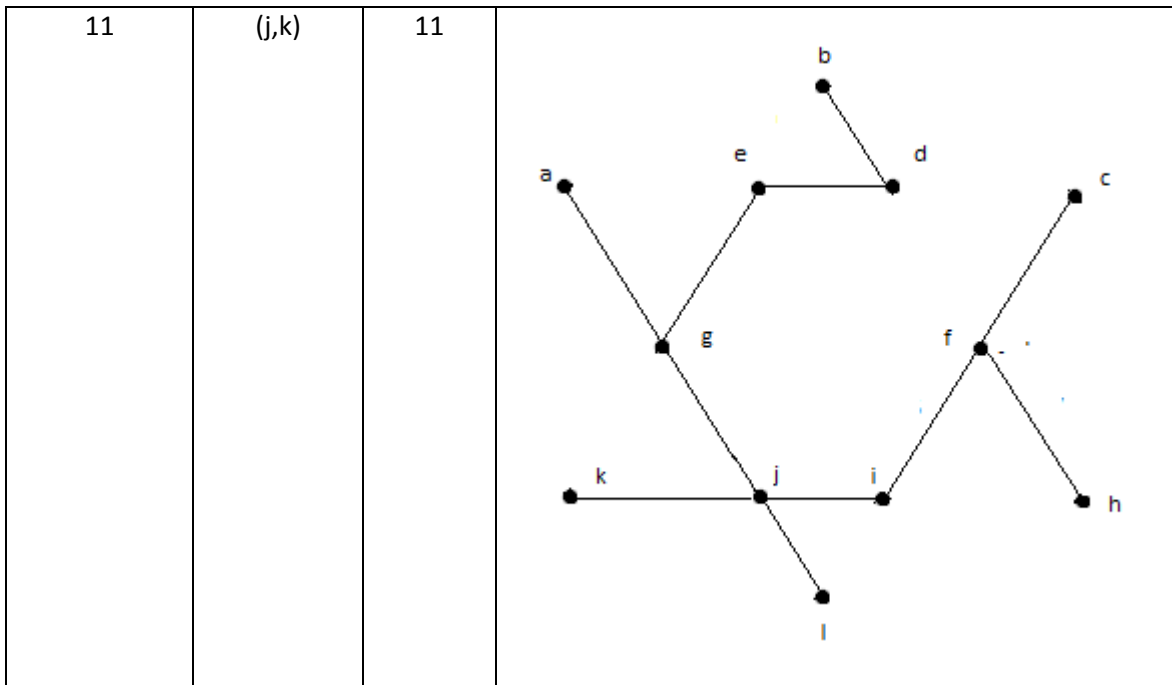
Penyelesaian:

Berikut adalah tabel langkah pembentukan pohon :

Langkah ke -	Sisi	Bobot	Pohon rentang
1	(b,d)	1	
2	(d,e)	4	
3	(e,g)	2	
4	(g,a)	3	
5	(g,i)	8	

6	(j,i)	4	
7	(i,f)	5	
8	(f,c)	3	



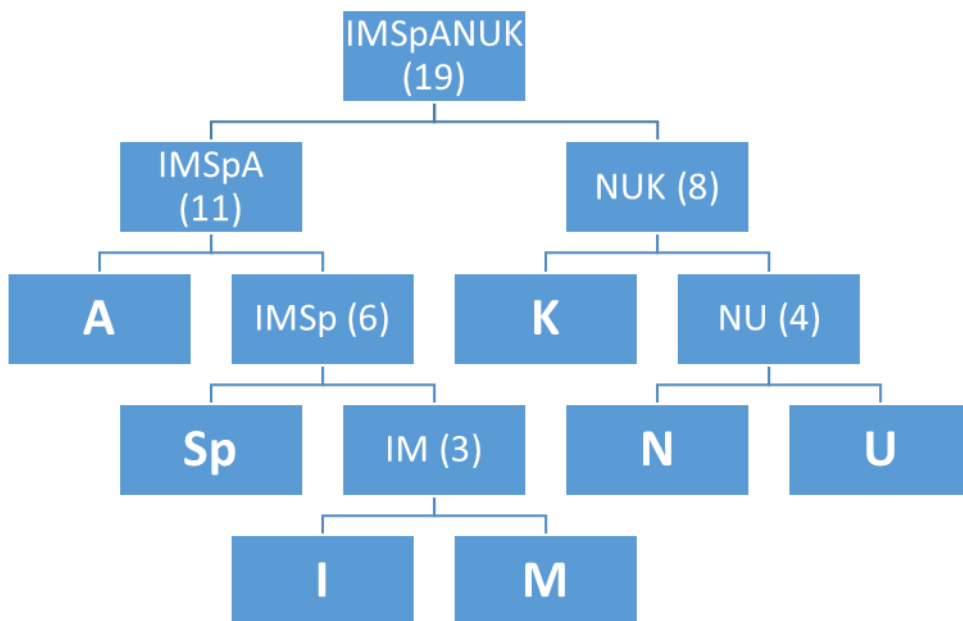


Total bobot = 54

2. Terdapat suatu pesan (string) yang disimpan di komputer : “AKU KAMU MAKAN IKAN”. Setiap karakter (termasuk spasi) secara *default* berukuran 1 byte (8 bit). Namun pesan tersebut dapat dimampatkan dengan algoritma Huffman sehingga ukurannya di memory menjadi lebih kecil.
- Hitunglah frekuensi kemunculan setiap karakter (termasuk spasi), lalu gambarkan pohon Huffmannya.
  - Tentukan kode Huffman untuk setiap karakter dan hitung ukuran bit yang dihasilkan jika pesan diubah menjadi kode Huffman (hitung ukuran bitnya saja).
  - Apa yang akan terjadi jika pemampatan dengan algoritma Huffman dilakukan pada sebuah string yang terdiri dari semua jenis karakter yang mungkin (terdapat 256 jenis karakter, 1 byte per karakter) dan jumlah kemunculan setiap jenis karakter adalah sama? Bagaimana ukuran bit string setelah dan sebelum pemampatan?

Penyelesaian:

- I=1, M=2, N=2, U=2, Sp=3, K=4, A=5



- Kode huffman

A	00
K	10
Sp	010
N	110
U	111
I	0110
M	0111

"AKU KAMU MAKAN IKAN"

$$\begin{aligned} \text{Ukuran bit} &= (2 + 2 + 3) + 3 + (2 + 2 + 4 + 3) + 3 + (4 + 2 + 2 + 2 + 3) + 3 + (4 + 2 + 2 + 3) \\ &= 7 + 3 + 11 + 3 + 13 + 3 + 11 = 51 \end{aligned}$$

c. Tidak terjadi pemampatan, ukuran string setelah dan sebelum pemampatan adalah sama besar.

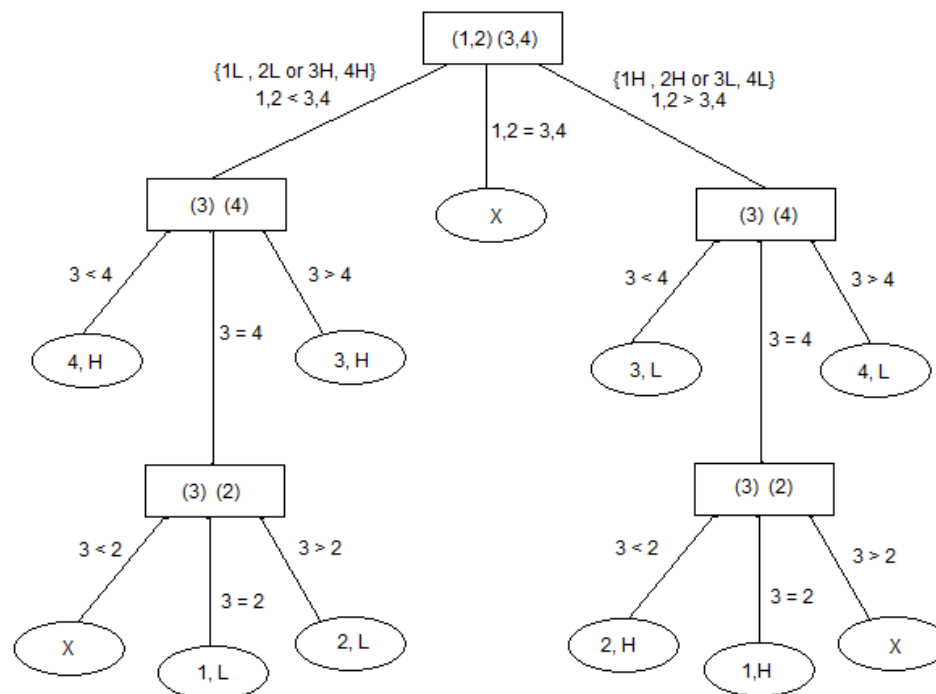
3. Diberikan 4 buah koin yang identik antara satu dengan yang lainnya, namun ternyata satu di antaranya adalah koin yang palsu. Koin yang palsu memiliki berat yang berbeda dengan koin yang asli, namun tidak diketahui apakah koin palsu tersebut lebih berat / lebih ringan daripada yang asli. Untuk menentukan mana yang palsu, diberikan sebuah timbangan, namun hanya dapat digunakan sebanyak 3 kali penimbangan. Dengan menggunakan decision tree, tentukan semua kemungkinan koin yang palsu berdasarkan penimbangan, dan apakah koin palsu tersebut lebih berat / lebih ringan dari yang asli.

Penyelesaian:

Penimbangan pertama

Penimbangan kedua

Penimbangan ketiga



4. Diketahui fungsi kompleksitas beberapa algoritma :

- (i)  $f(n) = 7^n + n^2$
- (ii)  $g(n) = 2n^2 + n \log n$
- (iii)  $h(n) = n + 5$

Hitunglah notasi O (big-oh) dari hasil operasi berikut :

a.  $f(n) + g(n) + h(n)$

b.  $f(n) \times g(n) \times h(n)$

Penyelesaian:

$f(n) = O(7^n)$

$g(n) = O(n^2)$

$h(n) = O(n)$

Maka :

a.  $f(n) + g(n) + h(n) = O(\max(7^n, n^2, n)) = O(7^n)$

b.  $f(n) \times g(n) \times h(n) = O(7^n \times n^2 \times n) = O(n^3 7^n)$

5. Perhatikan potongan kode C berikut :

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
        a = a + j;
    }
}
for (k = 0; k < N; k++) {
    b = b + k;
}
```

Tentukan kompleksitas waktu T(n) dan kompleksitas waktu asimtotik dari algoritma diatas, berikan langkah / penjelasan singkat bagaimana anda bisa menentukan jawaban anda!

Penyelesaian:

$O(N * N)$  atau  $O(N^2)$

Loop pertama adalah loop yang bersifat *nested*, operasi pada variabel a dilakukan N kali dan diulang sebanyak N kali lagi. ( $O(N^2)$ )

Loop kedua pada variabel b merupakan *single loop*, operasi pada variabel b dilakukan N kali sehingga  $O(N)$ .

$O(N^2) + O(N) = O(N^2)$

6. Diberikan waktu proses T(n) untuk menyelesaikan sebuah masalah dengan algoritma tertentu.

T(n)	O(n)
$0.01n + 100n^2 + 100000$	
$100n \log n + n^3 + 10000n$	
$2n + n^{1.1} + 0.5n^{1.25}$	
$0.0003 \log n + 1000 \log(\log(n))$	
$0.001 n \log n + n (\log n)^2$	
$n^2 \log n + n (\log n)^2$	

Ubah ekspresi tersebut menjadi notasi O dan urutkan dari yang tercepat.

Penyelesaian:

T(n)	O(n)
$0.01n + 100n^2 + 100000$	$O(n^2)$
$100n \log n + n^3 + 10000n$	$O(n^3)$
$2n + n^{1.1} + 0.5n^{1.25}$	$O(n^{1.25})$
$0.0003 \log n + 1000 \log(\log(n))$	$O(\log n)$
$0.001 n \log n + n (\log n)^2$	$O(n (\log n)^2)$
$n^2 \log n + n (\log n)^2$	$O(n^2 \log n)$

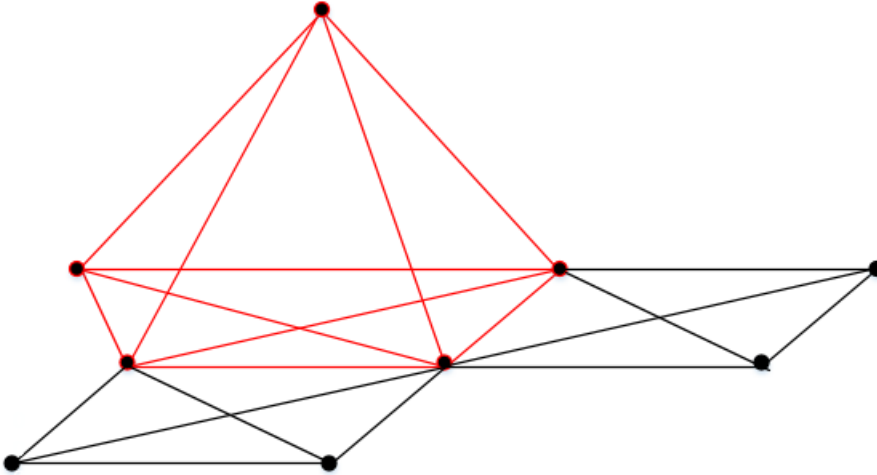
Hasil Pengurutan :

$0.0003 \log n + 1000 \log (\log(n))$ ,  $0.001 n \log n + n (\log n)^2$ ,  $2n + n^{1.1} + 0.5n^{1.25}$ ,  $0.01n + 100n^2 + 100000$ ,  $n^2 \log n + n (\log n)^2$ ,  $100n \log n + n^3 + 10000n$

7. Apakah graf soal nomor 7 merupakan graf planar? Berikan alasannya berdasarkan teorema Kuratowski!

Penyelesaian:

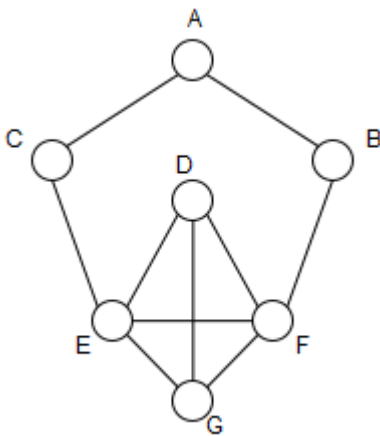
Tidak planar karena mengandung upagraf  $K_5$ .



8. Berdasarkan graf soal nomor 8, tentukan dan berikan penjelasan:

- Apakah graf merupakan graf Euler, Semi – Euler, atau bukan keduanya?
- Gambarkan sirkuit/lintasan Euler dan Hamilton dari graf tersebut jika ada!

Penyelesaian:



- Graf tersebut merupakan graf Semi Euler. Graf ini bukan graf Euler karena terdapat simpul berderajat ganjil (G dan D). Graf ini semi euler karena jumlah simpul berderajat ganjil berjumlah tepat 2.
- Contoh sirkuit Hamilton: E-C-A-B-F-D-G-E  
Contoh lintasan Euler: D-E-C-A-B-F-G-D-F-E-G