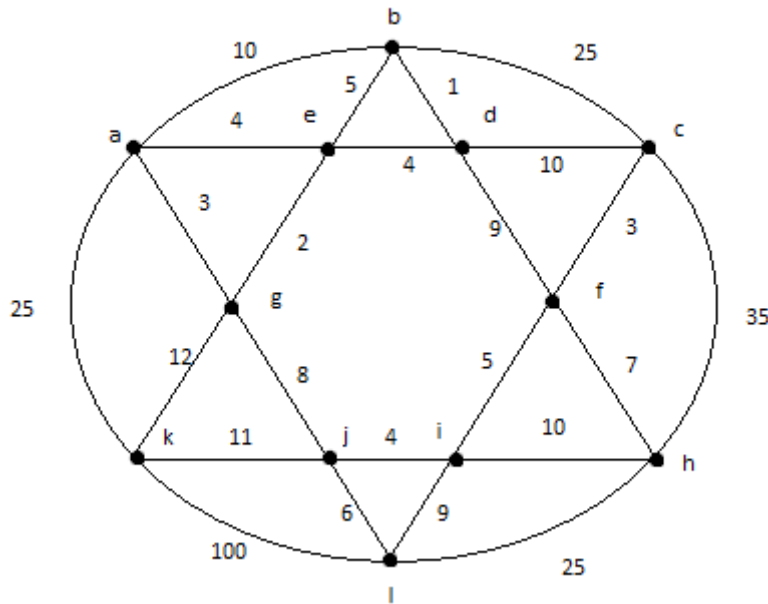


## Soal dan Jawaban Materi Graf, Pohon, dan Kompleksitas Algoritma

### POHON

- Ubahlah graf berikut ini dengan menggunakan algoritma prim agar menjadi pohon merentang minimum dan tentukan bobotnya !



- 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.
- 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.
- 
- 
- Buatlah pohon biner dari ekspresi aritmatik berikut ini (*preorder*)  $+*/-A+BCM+F^{\wedge}OPD$   
Lalu nyatakan ekspresi tersebut ke dalam bentuk notasi lainnya (2 notasi)
- Terdapat string : TEKNIKINFORMATIKA

- a. Gambarkan pohon Huffman dengan terlebih dahulu menghitung frekuensi dan peluang setiap karakter dari string diatas.
  - b. Tentukan representasi bit dari kata KAIN berdasarkan pohon Huffman yang dibikin pada bagian a.
7. Gambarkan pohon pencarian biner dari data – data berikut:
- a. 42, 10, 50, 41, 3, 18, 39, 47, 43, 49
  - b. “kota”, “rumah”, “desa”, “suku”, “tanah”, “bangunan”, “negara”, “dusun”, “danau”, “domba”
8. 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.
- a. Hitunglah frekuensi kemunculan setiap karakter (termasuk spasi), lalu gambarkan pohon Huffmannya.
  - b. Tentukan kode Huffman untuk setiap karakter dan hitung ukuran bit yang dihasilkan jika pesan diubah menjadi kode Huffman (hitung ukuran bitnya saja).
  - c. 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?

## KOMPLEKSITAS ALGORITMA

1. 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 dari algoritma diatas, berikan langkah / penjelasan singkat bagaimana anda bisa menentukan jawaban anda !

2. 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.

3. Perhatikan function berikut !

```
procedure BubbleSort (input/output s : TabelInt, input n : integer)
{ Mengurutkan tabel s[1..N] sehingga terurut menaik dengan metode }
```

### Deklarasi

```
    i      : integer      { pencacah untuk jumlah langkah }
    k      : integer      { pencacah, untuk pengapungan pada setiap
langkah }
    temp   : integer      { peubah bantu untuk pertukaran }
```

### Algoritma:

```
    for i ← n-1 downto 1 do
        for k ← 1 to i do
            if s[k+1] < s[k] then
```

```
        {pertukarkan s[k] dengan s[k+1]}  
        temp ← s[k]  
        s[k] ← s[k+1]  
        s[k+1] ← temp  
    endif  
endfor  
endfor
```

Tentukan jumlah pertukaran untuk kasus terburuk. Setelah itu, tentukan kompleksitas waktu asimptotik dalam notasi O besar.

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

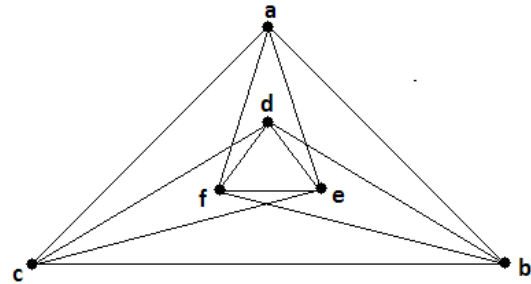
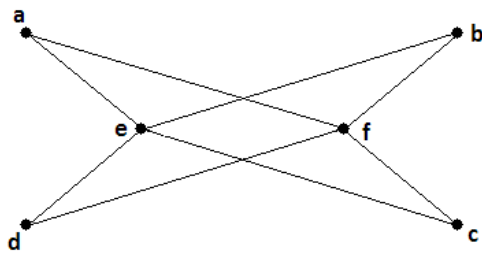
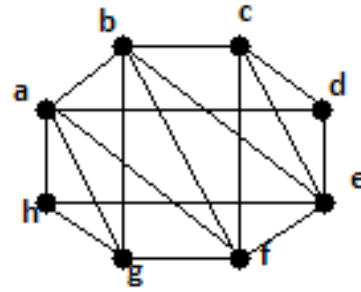
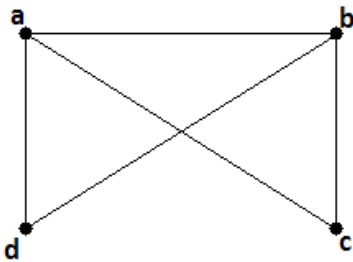
b.  $f(n) \times g(n) \times h(n)$

## GRAF

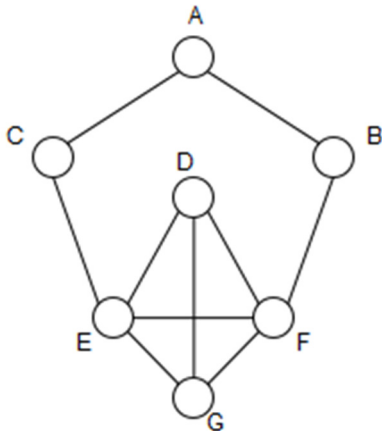
1. Sudoku adalah permainan ketangkasan berupa matriks berukuran 9x9 yang setiap sel nya dapat diisi angka yang unik pada setiap kolom, baris dan persegi nya.

Deskripsikanlah bagaimana cara anda mengaplikasikan pewarnaan graf pada permainan Sudoku ini dan tentukan berapa jumlah warna minimum yang digunakan ! ( **tidak perlu menggambar graf nya** )

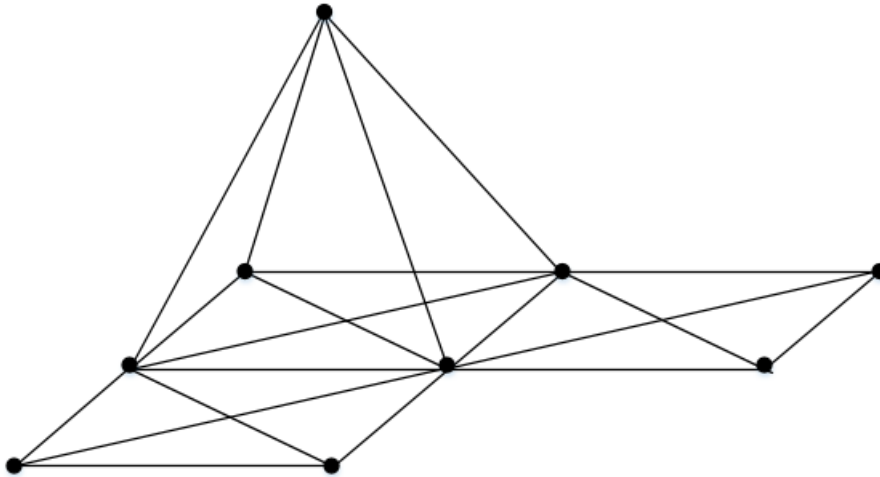
2. Tentukan apakah graf dibawah ini planar atau tidak. Jika ya, tunjukan graf planarnya tanpa jalur yang bersilangan.



3. Berdasarkan graf pada gambar di bawah ini, tentukan dan berikan penjelasan:

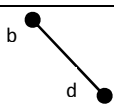
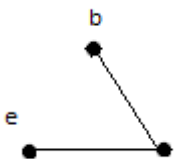
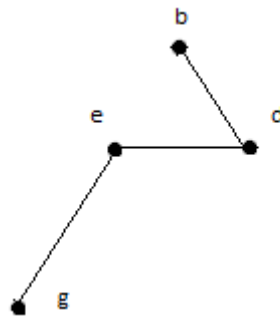
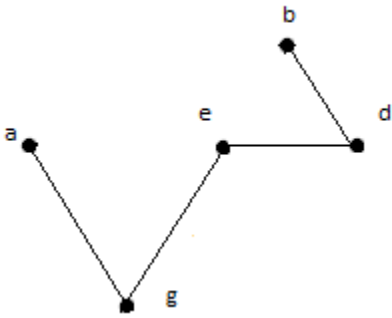


- a. Apa yang dimaksud dengan Sirkuit Euler dan Sirkuit Hamilton?
  - b. Apakah graf merupakan graf Euler, Semi – Euler, atau bukan keduanya?
  - c. Gambarkan sirkuit/lintasan Euler dan Hamilton dari graf tersebut jika ada!
4. Buktikan dengan teori graf pernyataan dibawah :
- a. Jika ada  $n$  orang menghadiri sebuah pesta dan salaing bersalaman (tapi tidak pada diri sendiri) maka pada akhirnya ada paling sedikit dua orang berjabat tangan dengan orang yang sama lagi.
  - b. Graf dapat dibuat dengan 5 simpul dengan derajat masing-masing simpul adalah 4, 4, 3, 2, 3
- Berikan juga gambar graf untuk masing-masing soal diatas.
5. Apakah graf berikut merupakan graf planar? Berikan alasannya berdasarkan teorema Kuratowski!



## Solusi

1. 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,j)	8	
6	(j,i)	4	
7	(i,f)	5	



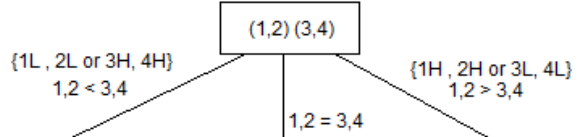
8	(f,c)	3	
9	(j,l)	6	

10	(f,h)	7	
11	(j,k)	11	

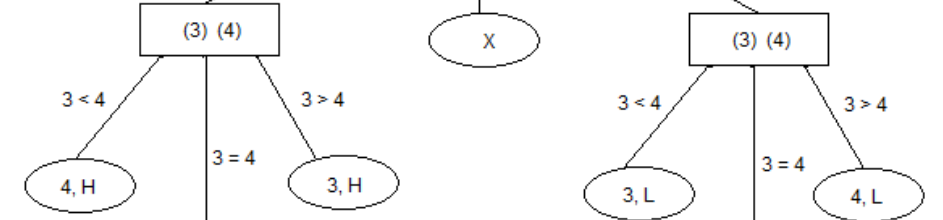
Total bobot = 54

2. Jawaban :

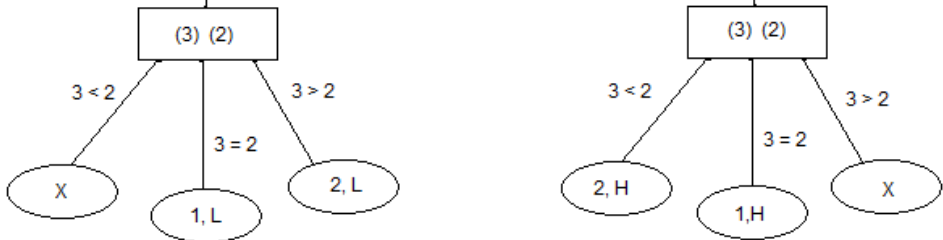
Penimbangan pertama



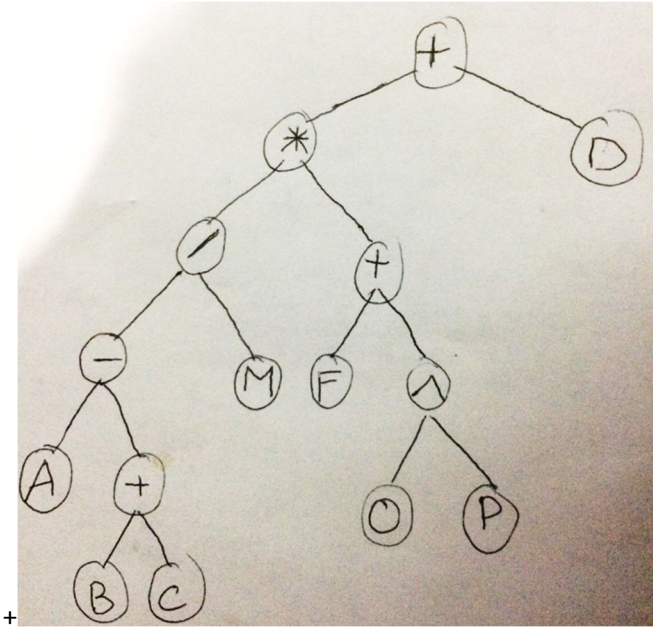
Penimbangan kedua



Penimbangan ketiga



3. Inorder :  $A-B+C/M*F+O^P+D$   
 Postorder :  $ABC+-M/FOP^+*D+$

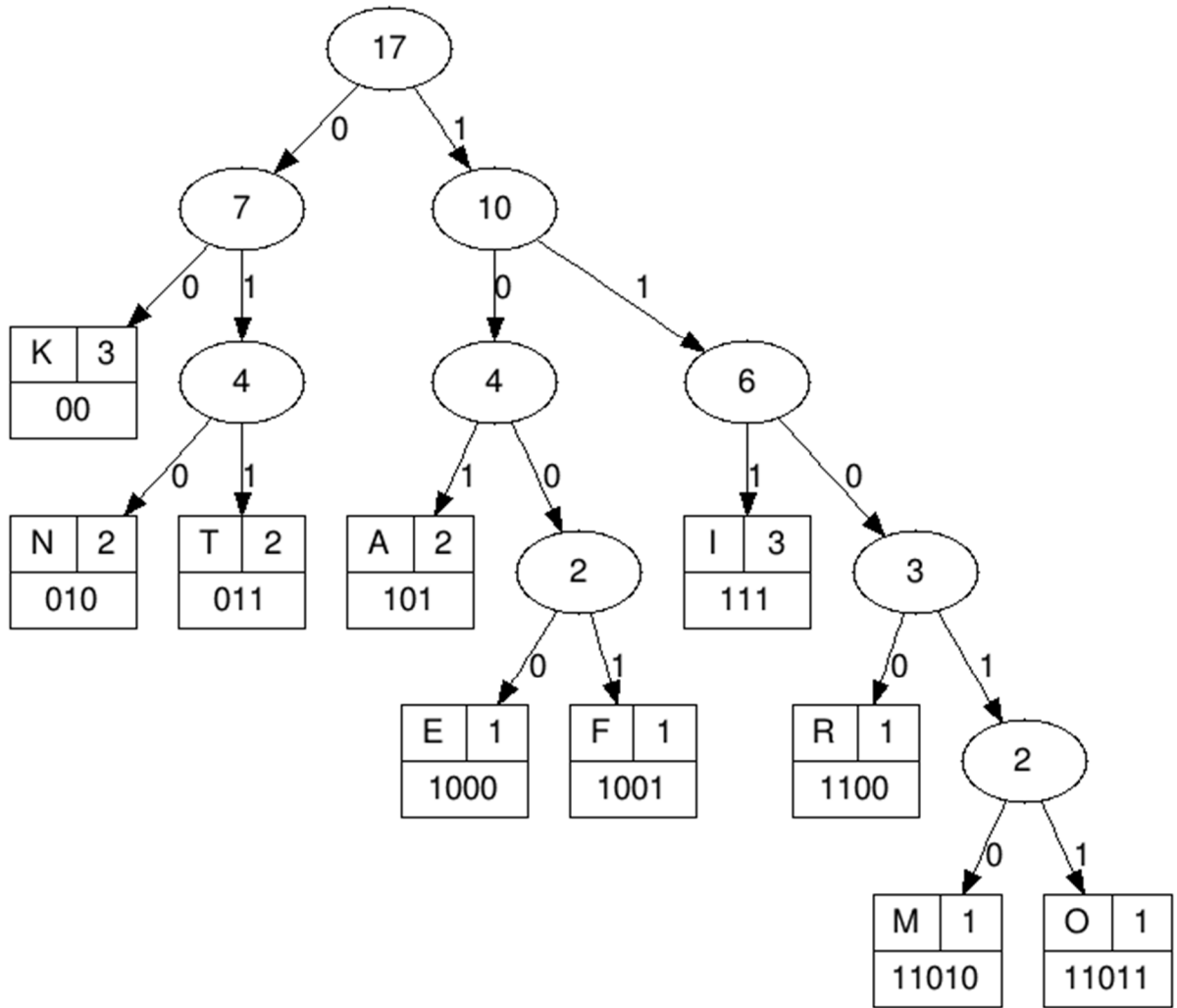


4. Solusi:  
 a.

Simbol	Kekerapan	Peluang
--------	-----------	---------

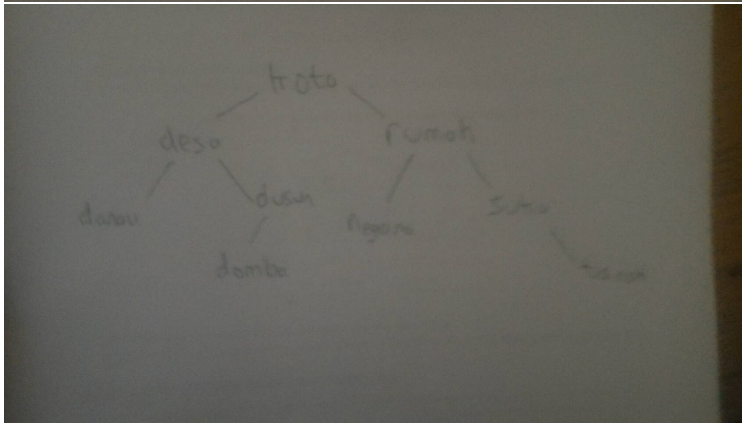
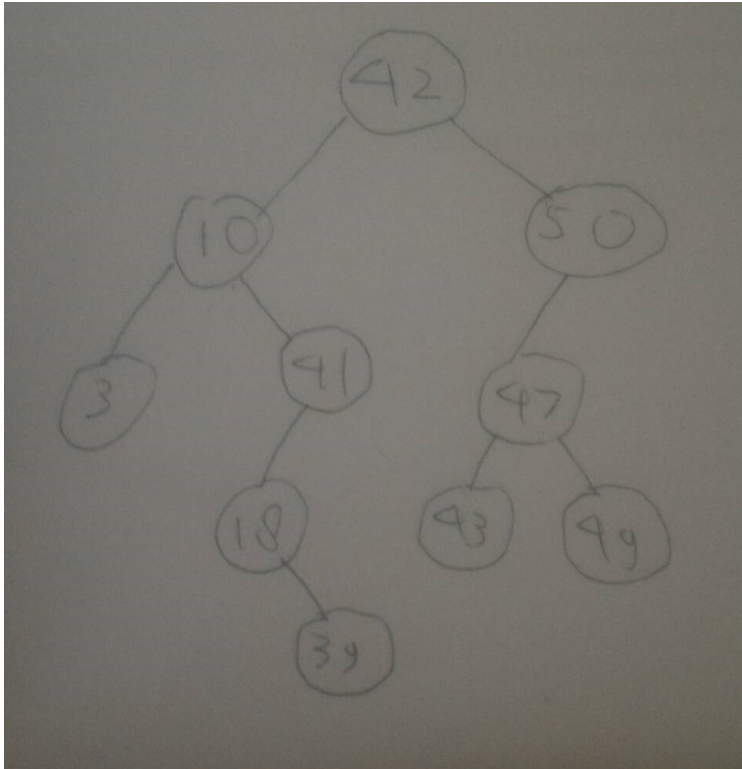
A	2	2/17
I	3	3/17
K	3	3/17
N	2	2/17
T	2	2/17
E	1	1/17
F	1	1/17
R	1	1/17
M	1	1/17
O	1	1/17

Contoh pohon Huffman yang mungkin:



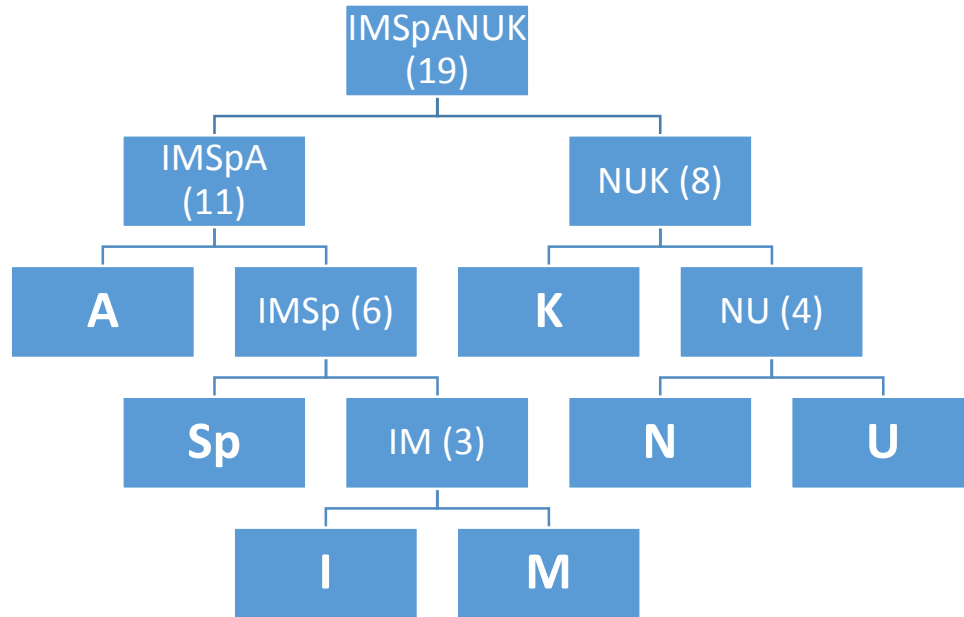
- b. K=00  
 A=101  
 I=111  
 N=010

5. Solusi



6. Jawaban :

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



b. 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.

7.  $O(N * N)$  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  $O(N)$ .

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

8. Solusi :

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 :

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

9. Jumlah pertukaran (kasus terburuk) :  $n(n - 1)/2$

Kompleksitas :  $O(n^2)$ .

10. Jawaban :

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

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

$$h(n) = O(n)$$

Maka :

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

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

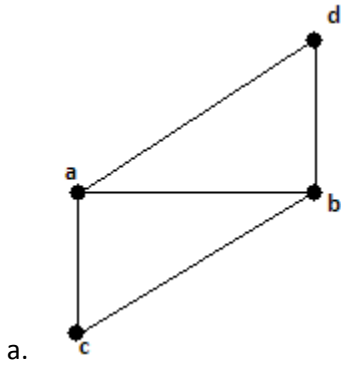


11. Berikut langkah – langkahnya :

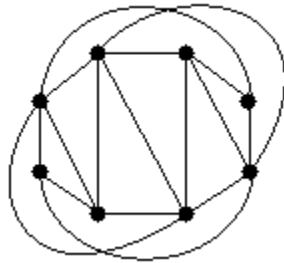
- a. Asumsikan simpul adalah sel pada kotak Sudoku, sehingga ada 81 buah simpul
- b. Hubungkan simpul dengan simpul lainnya yang nilainya tidak boleh sama ( baris, kolom dan kotak)
- c. Warnai setiap simpul dengan aturan tidak ada simpul yang bertetangga memiliki warna yang sama

Didapat ada 9 warna minimum yang digunakan.

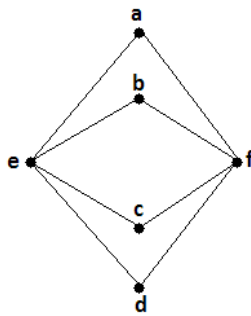
12. Solusi

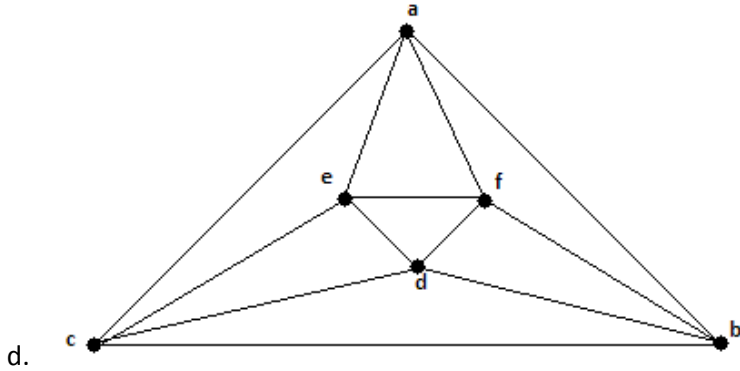


b. Tidak planar



c.





13. Solusi

- a. Sirkuit Euler adalah Sirkuit yang melewati masing – masing sisi tepat satu kali  
Sirkuit Hamilton adalah Sirkuit yang melewati masing – masing simpul tepat satu kali
- b. 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.
- c. Contoh graf Hamilton: E-C-A-B-F-D-G-E  
Contoh graf Euler: D-E-C-A-B-F-G-D-F-E-G

14. Jawaban :

- a. Misalkan orang direpresentasikan sebagai simpul dan jabat tangan sebagai sisi maka dapat dibuktikan dengan menggunakan **teorema jabat tangan** “jumlah derajat semua simpul pada suatu graf adalah genap, yaitu dua kali jumlah sisi graf tersebut”.  
Karena jumlah derajat suatu simpul selalu berjumlah genap maka minimal terdapat 2 simpul yang memiliki derajat yang sama.
- b. Dapat dibentuk graf karena jumlah derajat adalah genap.

15. Jawab :

Tidak planar karena mengandung upagraf  $K_5$ .

