

## SOLUSI KUIS MATDIS 4

1. Di labtek V terdapat 25 telephone. Apakah mungkin menghubungkan telephone-telephone tersebut sehingga **setiap telephone terkoneksi dengan 7 telephone lainnya?** (Taufiq)

### Jawaban

Jika setiap telephone harus terkoneksi dengan 7 telephone lainnya, maka

Setiap node memiliki derajat 7.

Total derajat semua simpul =  $25 \times 7 = 175$  (25 node, dengan masing-masing node memiliki derajat 7)

Padahal, berdasarkan lemma jabat tangan

**Lemma Jabat Tangan.** Jumlah derajat semua simpul pada suatu graf adalah genap, yaitu dua kali jumlah sisi pada graf tersebut.

Dengan kata lain, jika  $G = (V, E)$ , maka  $\sum_{v \in V} d(v) = 2|E|$

Total derajat semua simpul haruslah genap.

Karena pada graf ini total derajat semua simpulnya bernilai 175 (ganjil), **Maka graf ini tidak mungkin dibentuk.**

2. Diberikan sebuah data penting berisi “STRUKTUR DISKRIT” dalam sebuah *script*. Data tersebut kemudian di-*compressed* menggunakan metode Huffman. (gill)
  - a. Tentukan kode huffman setiap karakter dari data teks tersebut (termasuk spasi)!
  - b. Tentukan panjang pesan dalam bit setelah pengkodean Huffman!

### Jawaban:

Akan dilakukan pencacahan banyak setiap karakter unik yang ada pada data “STRUKTUR DISKRIT”. Hasil pencacahan setiap karakter unik adalah sebagai berikut.

S = 2

T = 3

R = 3

U = 2

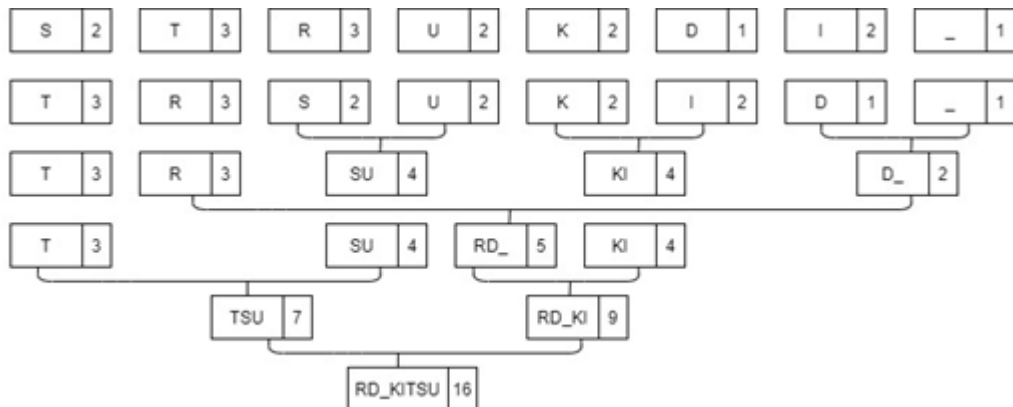
K = 2

D = 1

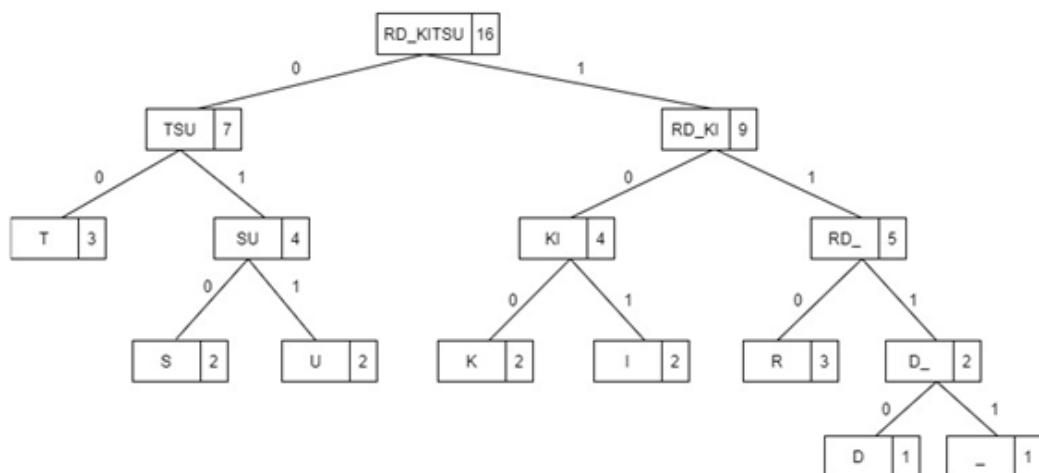
I = 2

\_ = 1

Berikut ini adalah pembentukan pohon huffman dan hasil pohon huffman yang terbentuk.



Pohon Huffman untuk setiap karakter unik dari “STRUKTUR DISKRIT”.

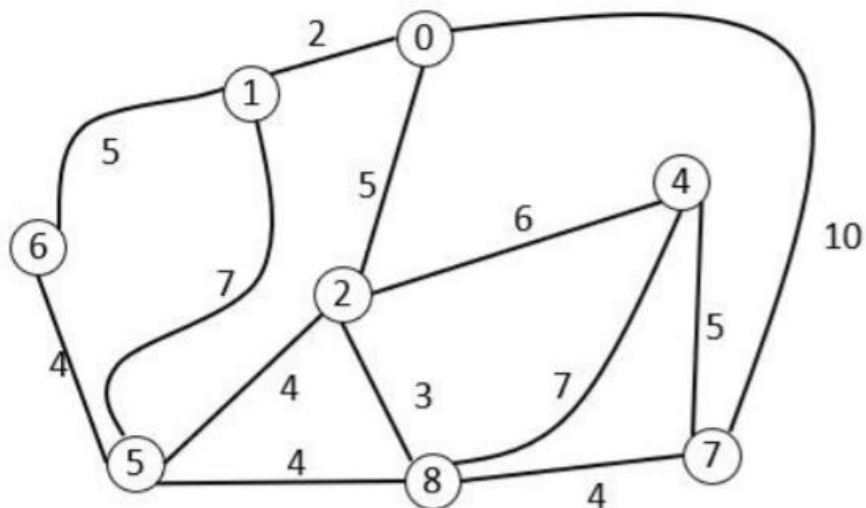


Simbol	Frekuensi	Peluang	Kode Huffman
T	3	3/16	00
S	2	2/16	010
U	2	2/16	011
K	2	2/16	100
I	2	2/16	101
R	3	3/16	110
D	1	1/16	1110
–	1	1/16	1111



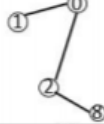
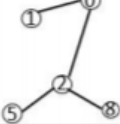
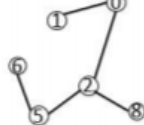
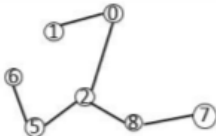
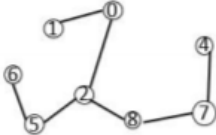
b. Panjang pesan dalam bit setelah pengkodean Huffman adalah

$$(3 \times 2) + (2 \times 3) + (2 \times 3) + (2 \times 3) + (2 \times 3) + (3 \times 3) + (1 \times 4) + (1 \times 4) = 47 \text{ bit}$$

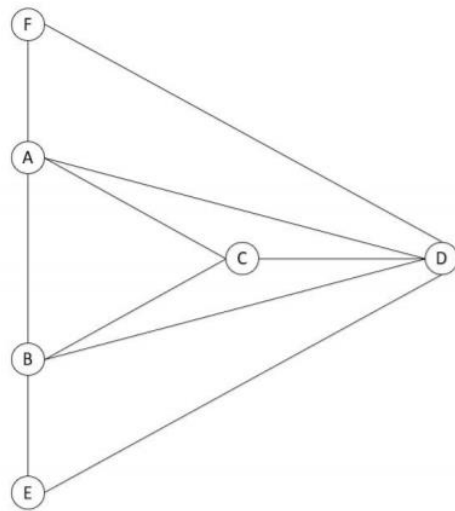
3. Bentuklah *minimum spanning tree* dari graf berikut dengan menggunakan algoritma prim dan tentukan bobot totalnya. Jika terdapat sisi dengan bobot yang sama, utamakan sisi dengan jumlah simpul terkecil. Algoritma dimulai dari simpul 0. (*Nanda*)



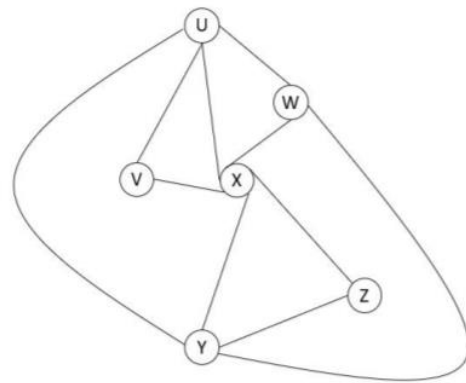
**Jawaban:**

Langkah	Sisi	Bobot Total	Pohon Rentang
1	(0,1)	2	
2	(0,2)	7	
3	(2,8)	10	
4	(2,5)	14	
5	(5,6)	18	
6	(7,8)	22	
7	(4,7)	26	

4. Cari tahu apakah kedua graf di bawah ini isomorfik atau tidak! Jika tidak, sebutkan alasannya. Jika iya, tunjukkan simpul-simpul yang berkorespondensi. **(Nanda)**



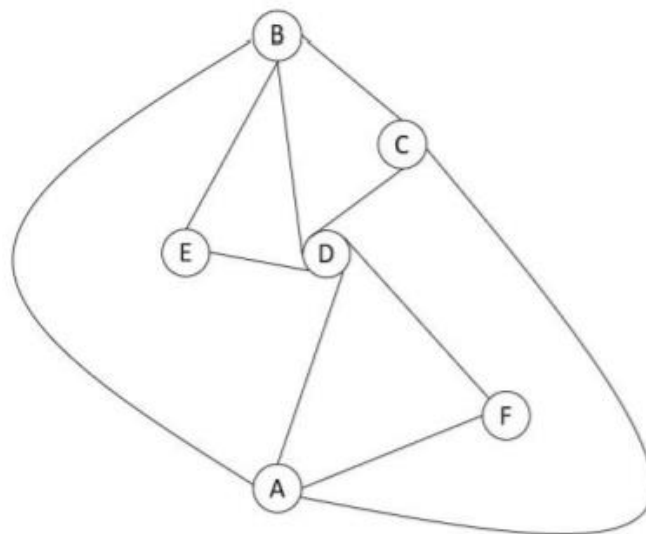
Graf 1



Graf 2

**Jawaban:**

Ya, kedua graf isomorfik. Berikut merupakan graf 1 yang memiliki bentuk seperti graf 2 untuk menunjukkan bahwa kedua graf isomorfik.



5. Tentukan kompleksitas waktu dari algoritma yang ditulis dalam bahasa c++ berikut ini. Proses yang dihitung waktunya hanya di bagian “sum += j”, untuk proses lainnya abaikan waktunya (*Taufiq*)

```

int i = n;
int sum = 0;
while (i > 0){
    for (int j = 0; j<i; ++j){
        sum += j;
    }
    i = i / 2    //pembagian dengan pembulatan ke bawah
}

```

Note: asumsikan n merupakan integer kelipatan 2

A. Tentukan  $T(n)$

B. Tentukan dalam notasi Big-O

### **Jawaban**

Iterasi ke-1,  $i = n$ , jumlah operasi penambahan sebanyak  $n$  kali

Iterasi ke-2,  $i = n/2$ , jumlah operasi penambahan sebanyak  $n/2$  kali

·  
·  
·

Iterasi ke- $k$ ,  $i = 1$ , jumlah operasi penambahan sebanyak 1 kali

sehingga, jumlah operasi penambahan seluruhnya adalah

$= n + n/2 + n/4 + \dots + 2 + 1$  (deret geometri sepanjang  $k$ , dimana  $n = 2^k$ )

karena  $n$  kelipatan 2)

$= n(1 - (1/2)^k) / 1 - 1/2$

$= 2n - n/(2^k)$

$= 2n - 1$  (karena  $n = 2^k$ )

Jadi,  $T(n) = 2n - 1$

Dan, notasi Big-O nya adalah  $O(n)$

6. Tentukan apakah pernyataan kompleksitas algoritma berikut ini BENAR atau SALAH. Jika SALAH, berikan pernyataan yang benar. **(Fu)**
- Diberikan  $T(n) = 5n$  dan  $T(n) = 5n$ , maka pernyataan  $T(n) + T(n) = O(n)$  adalah benar
  - Diberikan  $T(n) = 2 + n + 4n$ , maka pernyataan  $T(n) = O(n)$  adalah benar
  - Diberikan  $T(n) = 2 + n + 4n$ , maka pernyataan  $T(n) = O(\log n)$  adalah benar
  - Diberikan  $T(n) = 5n$  dan  $T(n) = 5n$ , maka pernyataan  $T(n)T(n) = O(n)$  adalah benar
  - Diberikan  $T(n) = 3 + 6 + 9 + \dots + 3n$  dan  $T(n) = 2 + n + 4n$ , maka pernyataan  $T(n) = O(n) = T(n)$  adalah salah.

Jawaban:

- SALAH  
 $O(f(n)) + O(g(n)) = O(\max\{f(n), g(n)\})$ , sehingga  $T(n) = O(n)$  dan  $T(n) = O(n)$ , sehingga  $T(n) + T(n) = O(n)$
- BENAR  
 $T(n) = 2 + n + 4n = O(n)$  secara definisi memenuhi pernyataan  $T(n) = O(f(n))$  jika terdapat  $C$  dan  $n$  sedemikian sehingga  $T(n) \leq C \cdot f(n)$  untuk  $n \leq n \rightarrow$  tidak menyiratkan seberapa atas fungsi  $f$  itu
- SALAH  
 $T(n) = 2 + n + 4n = O(n)$ , karena  $\log n < n$  maka dapat dikatakan pernyataan  $T(n) = O(\log n)$  adalah salah
- BENAR  
 $T(n)T(n) = O(f(n))O(g(n)) = O(f(n)g(n))$  sehingga  $T(n)T(n) = O(n \cdot n) = O(n)$
- SALAH  

$$T(n) = 3 + 6 + 9 + \dots + 3n$$

$$= 3(1 + 2 + 3 + \dots + n) \leq 3(n + n + n + \dots + n) \text{ untuk } n \geq 1$$

$$= 3n = O(n)$$

$$T(n) = O(n) \rightarrow T(n) \text{ yang merupakan polinom derajat } m \text{ memiliki Big-O notation } O(n)$$

Maka, dapat dikatakan bahwa  $T(n) = O(n) = T(n)$ .

