

Program Studi Teknik Informatika  
.....  
Sekolah Teknik Elektro dan Informatika  
.....  
Institut Teknologi Bandung

Nama  
  
NIM

T.tangan:.....

Solusi Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Graf, Pohon, dan Kompleksitas Algoritma  
Dosen: Rinaldi Munir, Harlili, Fariska Zakhralativa, Nur Ulfa Maulidevi  
Kamis, 3 Desember 2020  
Waktu: 60 menit

0. Tulis ulang pernyataan berikut (jika **tidak** menuliskannya, maka ujian tidak akan diperiksa, nilai langsung 0): "Saya menyatakan bahwa saya mengerjakan kuis ini dengan sejujur-jujurnya, tanpa bantuan orang lain dan tanpa menggunakan cara yang tidak dibenarkan. Apabila pada kemudian hari diketahui saya mengerjakan kuis ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF22120 Semester 1 2020/2021. "  
(Nilai: 2)

1. Di labtek V terdapat 25 pesawat telepon. Apakah mungkin menghubungkan telepon-telepon tersebut sehingga **setiap telepon terkoneksi dengan 7 telepon lainnya?**  
(Nilai: 10)

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 dimampatkan menggunakan metode Huffman.  
(Nilai: 20)
- 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.

$$S = 2$$

$$T = 3$$

$$R = 3$$

$$U = 2$$

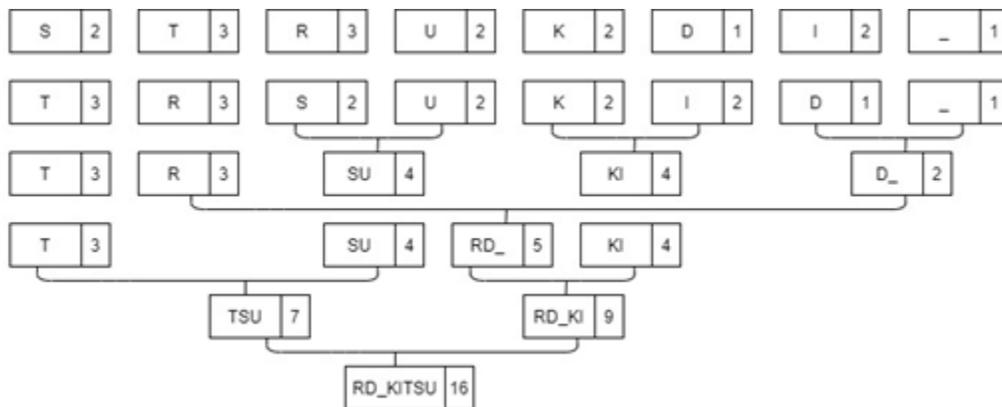
$$K = 2$$

$$D = 1$$

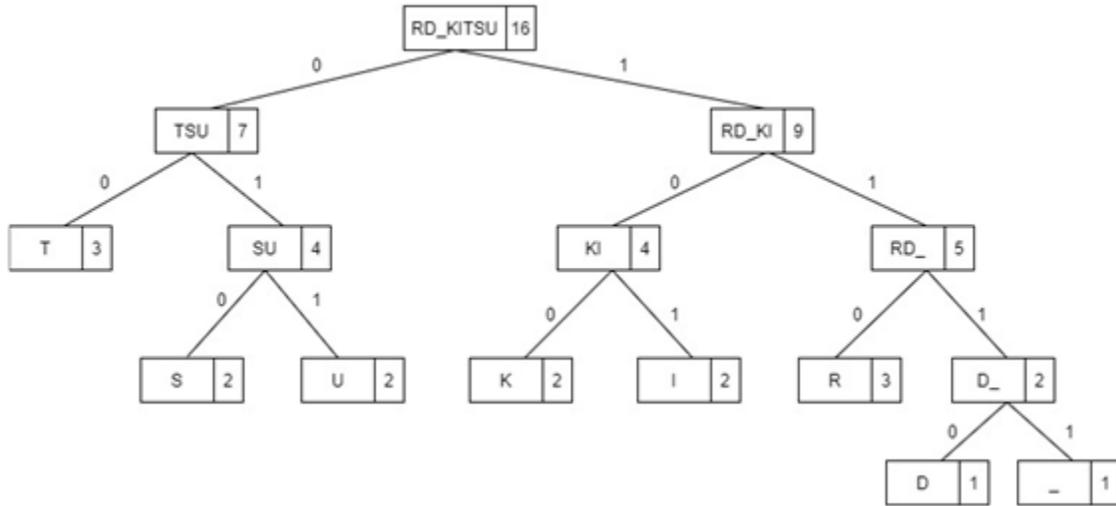
$$I = 2$$

$$\_ = 1$$

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



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

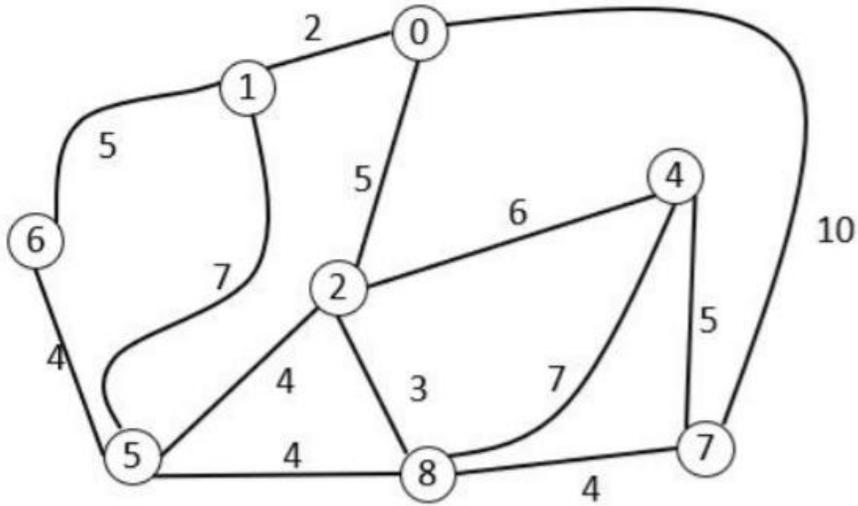


Simbol	Frekuensi	Peluang	Kode Huffman
T	3	3/16	00
S	2	2/16	010
U	2	2/16	011
K	2	2/16	100
I	2	2/16	101
R	3	3/16	110
D	1	1/16	1110
-	1	1/16	1111

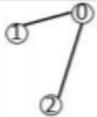
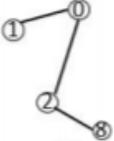
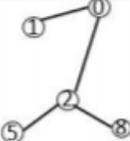
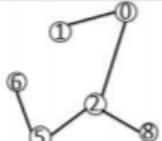
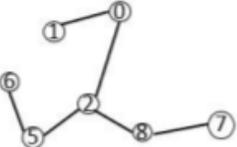
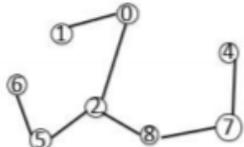
b. Panjang pesan dalam bit setelah pengkodean Huffman adalah

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

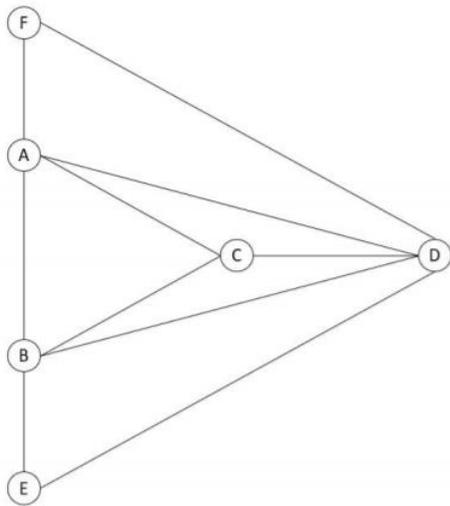
3. Bentuklah *minimum spanning tree* dari graf berikut dengan menggunakan Algoritma Prim dan tentukan bobot totalnya. Jika terdapat sisi dengan bobot yang sama, utamakan sisi dengan nomor simpul terkecil. (Nilai: 20)



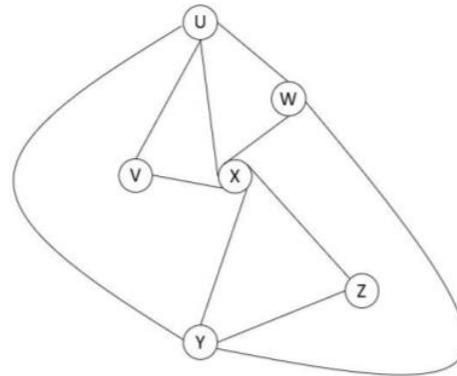
Jawaban:

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. (Nilai: 15)



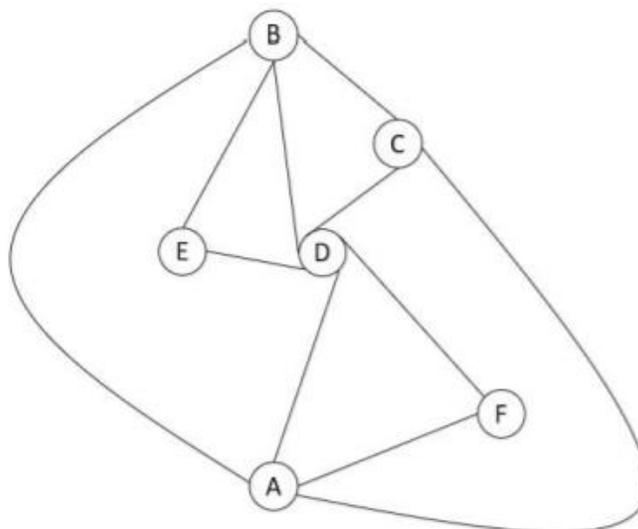
Graf 1



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 (Nilai: 20)

```
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

- Tentukan T(n)
- Tentukan dalam notasi Big-O, Big-Omega, dan Big-Tetha

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

$$= n + n/2 + n/4 + \dots + 2 + 1 \text{ (deret geometri sepanjang } k, \text{ dimana } n = 2^k \text{ karena } n \text{ kelipatan } 2)$$

$$\begin{aligned} &= \frac{n(1 - (1/2)^k)}{1 - 1/2} \\ &= 2n - n/(2^k) \\ &= 2n - 1 \quad (\text{karena } n = 2^k) \end{aligned}$$

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

Dan, notasi Big-O, , Big-Omega, dan Big-Tethanya adalah  $O(n)$ ,  $\Omega(n)$ , dan  $\Theta(n)$

6. Tentukan apakah pernyataan kompleksitas algoritma berikut ini BENAR atau SALAH. Jika SALAH, berikan pernyataan yang benar. (Nilai: 15)
- Diberikan  $T_1(n) = 5n$  dan  $T_2(n) = 5n^2$ , maka pernyataan  $T_1(n) + T_2(n) = O(n)$  adalah benar
  - Diberikan  $T(n) = 2 + n + 4n^2$ , maka pernyataan  $T(n) = O(n^3)$  adalah benar
  - Diberikan  $T_1(n) = 5n$  dan  $T_2(n) = 5n^2$ , maka pernyataan  $T_1(n)T_2(n) = O(n^3)$  adalah benar
  - Diberikan  $T_1(n) = 3 + 6 + 9 + \dots + 3n$  dan  $T_2(n) = 2 + n + 4n^2$ , maka pernyataan  $T_1(n) = O(n^2) = T_2(n)$  adalah salah.

Jawaban:

a. SALAH

$O(f(n)) + O(g(n)) = O(\max\{f(n), g(n)\})$ , sehingga  $T_1(n) = O(n)$  dan  $T_2(n) = O(n^2)$ , sehingga  $T_1(n) + T_2(n) = O(n^2)$

b. BENAR

$T(n) = 2 + n + 4n^2 = O(n^3)$  secara definisi memenuhi pernyataan  $T(n) = O(f(n))$  jika terdapat  $C$  dan  $n_0$  sedemikian sehingga  $T(n) \leq C \cdot f(n)$  untuk  $n \leq n_0 \rightarrow$  tidak menyiratkan seberapa atas fungsi  $f$  itu

c. BENAR

$T_1(n)T_2(n) = O(f(n))O(g(n)) = O(f(n)g(n))$  sehingga  $T_1(n)T_2(n) = O(n \cdot n^2) = O(n^3)$

d. SALAH

$$\begin{aligned} T_1(n) &= 3 + 6 + 9 + \dots + 3n \\ &= 3(1 + 2 + 3 + \dots + n) \leq 3(n + n + n + \dots + n) \text{ untuk } n \geq 1 \\ &= 3n^2 = O(n^2) \end{aligned}$$

$T_2(n) = O(n^2) \rightarrow T(n)$  yang merupakan polinom derajat  $m$  memiliki Big-O notation  $O(n^m)$

Maka, dapat dikatakan bahwa  $T_1(n) = O(n^2) = T_2(n)$ .