

Nama : Renaldi

NIM : 1910101001

Penggunaan Algoritma Greedy dalam menyelesaikan masalah

***) Latihan Knapsack Problem**

1) Berikut ini ada 4 objek dengan berat dan profit sebagai berikut:

$w_1=5, p_1=50$

$w_2=20, p_2=140$

$w_3=10, p_3=60$

$w_4=12, p_4=60$

pilihlah objek yang dapat dimasukkan dalam knapsack dengan kapasitas $k=30$ dengan algoritma greedy untuk knapsack problem dan fractional knapsack problem

knapsack problem (0/1)

i	W_i	P_i	P_i/W_i	by W	by P	by P/W
1	5	50	10	1	0	1
2	20	140	7	0	1	1
3	10	60	6	1	1	0
4	12	60	5	1	0	0
			total W	27	30	25
			total P	170	200	190

optimum

Urutan barang masuk kedalam tas dan perhitungan total W dan total P

note: format $x(y)$ dengan x adalah i dan y adalah kapasitas barang dalam tas sekarang

by W: 1(5), 3(15), 4(27) tak ada barang yang bisa diisi lagi, kapasitas tidak cukup

$$\text{total } W = 1 \times 5 + 0 \times 20 + 1 \times 10 + 1 \times 12 = 27$$

$$\text{total } P = 1 \times 50 + 0 \times 140 + 1 \times 60 + 1 \times 60 = 170$$

by P: 2(20),3(30) tas penuh

$$\text{total } W = 0 \times 5 + 1 \times 20 + 1 \times 10 + 0 \times 12 = 30$$

$$\text{total } P = 0 \times 50 + 1 \times 140 + 1 \times 60 + 0 \times 60 = 200$$

by P/W: 1(5), 2(25) tak ada barang yang bisa diisi lagi, kapasitas tidak cukup

$$\text{total } W = 1 \times 5 + 1 \times 20 + 0 \times 10 + 0 \times 12 = 25$$

$$\text{total } P = 1 \times 50 + 1 \times 140 + 0 \times 60 + 0 \times 60 = 190$$

fractional knapsack problem

i	W_i	P_i	P_i/W_i	by W	by P	by P/W
1	5	50	10	1	0	1
2	20	140	7	$\frac{3}{20}$	1	1
3	10	60	6	1	1	$\frac{1}{2}$
4	12	60	5	1	0	0
			total W	30	30	30
			total P	191	200	220

optimum

Urutan barang masuk kedalam tas dan perhitungan total W dan total P

note: format x(y) dengan x adalah i dan y adalah kapasitas barang dalam tas sekarang

by W: 1(5), 3(15), 4(27) ,<butuh 3 untuk mencapai 30 maka $\frac{3}{20} \times 20$ > ,2(30)

$$\text{total } W = 1 \times 5 + \frac{3}{20} \times 20 + 1 \times 10 + 1 \times 12 = 30$$

$$\text{total } P = 1 \times 50 + \frac{3}{20} \times 140 + 1 \times 60 + 1 \times 60 = 191$$

by P: 2(20),3(30) tas penuh

$$\text{total } W = 0 \times 5 + 1 \times 20 + 1 \times 10 + 0 \times 12 = 30$$

$$\text{total } P = 0 \times 50 + 1 \times 140 + 1 \times 60 + 0 \times 60 = 200$$

by P/W: 1(5), 2(25) <butuh 5 untuk mencapai 30 maka $\frac{5}{10} \times 10$ atau $\frac{1}{2} \times 10$ >, 3(30)

$$\text{total } W = 1 \times 5 + 1 \times 20 + \frac{1}{2} \times 10 + 0 \times 12 = 30$$

$$\text{total } P = 1 \times 50 + 1 \times 140 + \frac{1}{2} \times 60 + 0 \times 60 = 220$$

2) Berikut ini ada 8 objek dengan berat dan profit sebagai berikut :

$w_1=3, p_1=10$ $w_5=1, p_5=3$
 $w_2=5, p_2=9$ $w_6=9, p_6=12$
 $w_3=1, p_3=5$ $w_7=6, p_7=9$
 $w_4=2, p_4=8$ $w_8=4, p_8=15$

pilihlah objek yang dapat dimasukkan dalam knapsack dengan kapasitas $k=15$ dengan algoritma greedy untuk knapsack problem dan fractional knapsack problem

knapsack problem (1/0)

i	W_i	P_i	P_i/W_i	by W	by P	by P/W
1	3	10	3,33	1	0	1
2	5	9	1,8	0	0	0
3	1	5	5	1	0	1
4	2	8	4	1	1	1
5	1	3	3	1	0	1
6	9	12	1,33	0	1	0
7	6	9	1,5	0	0	0
8	4	15	3,75	1	1	1
			total W	11	15	11
			total P	41	35	41

Urutan barang masuk kedalam tas dan perhitungan total W dan total P

note: format $x(y)$ dengan x adalah i dan y adalah kapasitas barang dalam tas sekarang

by W : 3(1), 5(2), 4(4), 1(7), 8(11) tak ada barang yang bisa diisi lagi, kapasitas tidak cukup

$$\text{total W} = 1 \times 3 + 0 \times 5 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 0 \times 9 + 0 \times 6 + 1 \times 4 = 11$$

$$\text{total P} = 1 \times 10 + 0 \times 9 + 1 \times 5 + 1 \times 8 + 1 \times 3 + 0 \times 12 + 0 \times 9 + 1 \times 15 = 41$$

by P: 8(4), 6(13), <barang selanjutnya yang dimasukkan seharusnya barang ke 1 namun karena kapasitas tidak cukup maka diseleksi barang berikutnya dan seterusnya hingga menemui barang yang bisa dimasukkan> 4(15) tas penuh

$$\text{total W} = 0 \times 3 + 0 \times 5 + 0 \times 1 + 1 \times 2 + 0 \times 1 + 1 \times 9 + 0 \times 6 + 1 \times 4 = 15$$

$$\text{total P} = 0 \times 10 + 0 \times 9 + 0 \times 5 + 1 \times 8 + 0 \times 3 + 1 \times 12 + 0 \times 9 + 1 \times 15 = 35$$

by P/W: 3(1), 4(3), 8(7), 1(10), 5(11) tak ada barang yang bisa diisi lagi, kapasitas tidak cukup

$$\text{total } W = 1 \times 3 + 0 \times 5 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 0 \times 9 + 0 \times 6 + 1 \times 4 = 11$$

$$\text{total } P = 1 \times 10 + 0 \times 9 + 1 \times 5 + 1 \times 8 + 1 \times 3 + 0 \times 12 + 0 \times 9 + 1 \times 15 = 41$$

fractional knapsack problem

i	W_i	P_i	P_i/W_i	by W	by P	by P/W
1	3	10	3,33	1	$\frac{2}{3}$	1
2	5	9	1,8	$\frac{4}{5}$	0	$\frac{4}{5}$
3	1	5	5	1	0	1
4	2	8	4	1	0	1
5	1	3	3	1	0	1
6	9	12	1,33	0	1	0
7	6	9	1,5	0	0	0
8	4	15	3,75	1	1	1
			total W	15	15	15
			total P	48,2	33,67	48,2

Urutan barang masuk kedalam tas dan perhitungan total W dan total P

note: format x(y) dengan x adalah i dan y adalah kapasitas barang dalam tas sekarang

by W : 3(1), 5(2), 4(4), 1(7), 8(11) <butuh 4 untuk mencapai 15 maka $\frac{4}{5} \times 5 >$, 2(15) tas penuh

$$\text{total } W = 1 \times 3 + \frac{4}{5} \times 5 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 0 \times 9 + 0 \times 6 + 1 \times 4 = 15$$

$$\text{total } P = 1 \times 10 + \frac{4}{5} \times 9 + 1 \times 5 + 1 \times 8 + 1 \times 3 + 0 \times 12 + 0 \times 9 + 1 \times 15 = 48,2$$

by P: 8(4), 6(13), <butuh 2 untuk mencapai 15 maka $\frac{2}{3} \times 3 >$ 1(15) tas penuh

$$\text{total } W = \frac{2}{3} \times 3 + 0 \times 5 + 0 \times 1 + 0 \times 2 + 0 \times 1 + 1 \times 9 + 0 \times 6 + 1 \times 4 = 15$$

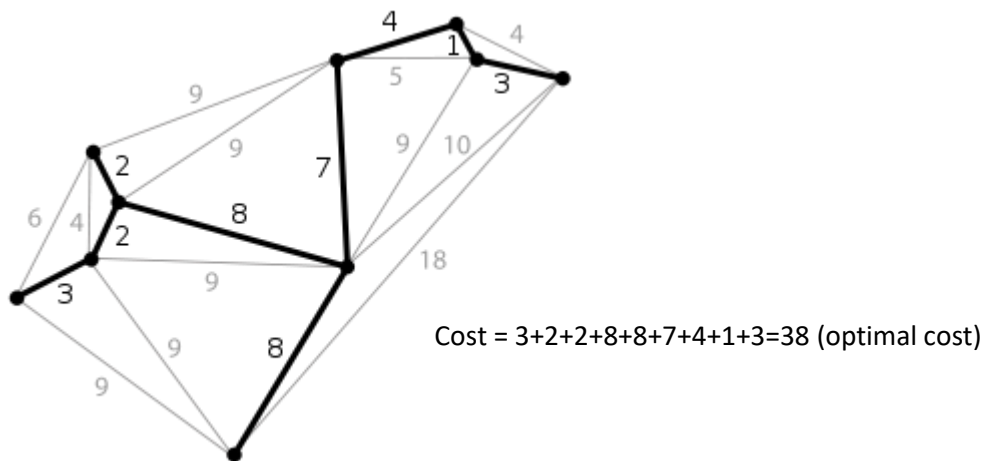
$$\text{total } P = \frac{2}{3} \times 10 + 0 \times 9 + 0 \times 5 + 0 \times 8 + 0 \times 3 + 1 \times 12 + 0 \times 9 + 1 \times 15 = 33,67$$

by P/W: 3(1), 4(3), 8(7), 1(10), 5(11) <butuh 4 untuk mencapai 15 maka $\frac{4}{5} \times 5$ >, 2(15) tas penuh

$$\text{total } W = 1 \times 3 + \frac{4}{5} \times 5 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 0 \times 9 + 0 \times 6 + 1 \times 4 = 15$$

$$\text{total } P = 1 \times 10 + \frac{4}{5} \times 9 + 1 \times 5 + 1 \times 8 + 1 \times 3 + 0 \times 12 + 0 \times 9 + 1 \times 15 = 48,2$$

A. Minimum Spanning Tree Problem



Permasalahan *Minimum Spanning Tree* adalah bagaimana cara menghubungkan semua titik tanpa ada nya jalur melingkar (cycle) , dengan beban (weight/cost) semimum mungkin.

Permasalahan *Minimum Spanning Tree* bisa diselesaikan dengan Algoritma Kruskal dan Algoritma Prim. kedua algoritma tersebut menggunakan teknik algoritma Greedy

Algoritma Kruskal untuk menyelesaikan minimum spanning tree problem :

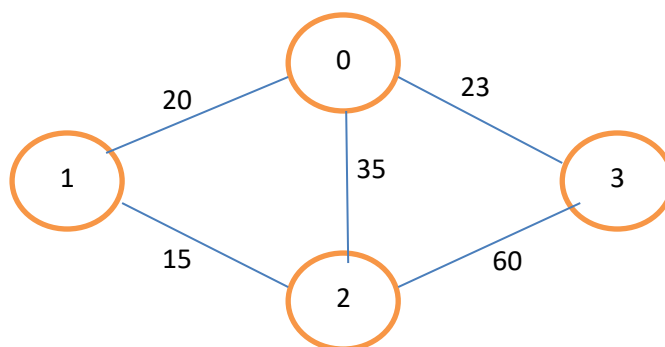
1. Sort all the edges in non-decreasing order of their weight.
2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.
3. Repeat step#2 until there are $(V-1)$ edges in the spanning tree. (V =total vertices)

Algoritma Prim untuk menyelesaikan minimum spanning tree problem:

- 1) Create a set *mstSet* that keeps track of vertices already included in MST.
- 2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.
- 3) While *mstSet* doesn't include all vertices
 -a) Pick a vertex *u* which is not there in *mstSet* and has minimum key value.
 -b) Include *u* to *mstSet*.
 -c) Update key value of all adjacent vertices of *u*. To update the key values, iterate through all adjacent vertices. For every adjacent vertex *v*, if weight of edge *u-v* is less than the previous key value of *v*, update the key value as weight of *u-v*

contoh masalah :

1)



buat minimum spanning tree dari grafik diatas menggunakan algoritma kruskal dan algoritma prim

Jawab:

Algoritma Kruskal

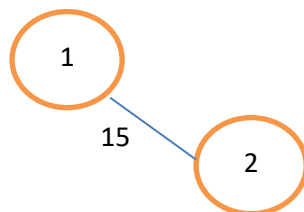
grafik diatas memiliki 4 titik itu berarti minimum spanning tree yang terbentuk akan memiliki $(4-1) = 3$ edge

Sorting garis berdasarkan weight secara ascending

Weight	Edge
15	1-2
20	0-1
23	0-3
35	0-2
60	2-3

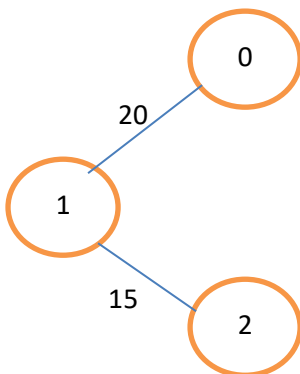
pilih semua garis satu per satu dari garis dengan weight terkecil

1. pilih edge 1-2 , tidak ada cycle yang terbentuk , masukkan



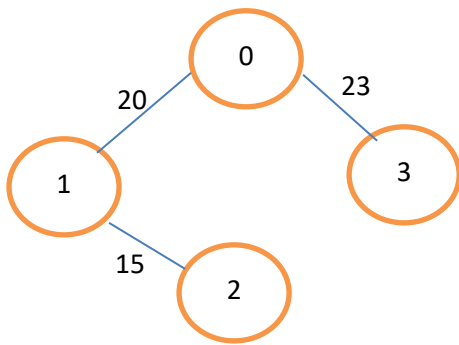
Edge yang terbentuk =1

2. pilih edge 0-1 , tidak ada cycle yang terbentuk , masukkan



Edge yang terbentuk =2

3. pilih edge 0-3, tidak ada cycle yang terbentuk, masukkan



Edge yang terbentuk = 3
 algoritma berhenti disini.

cost yang didapatkan dari algoritma greedy :

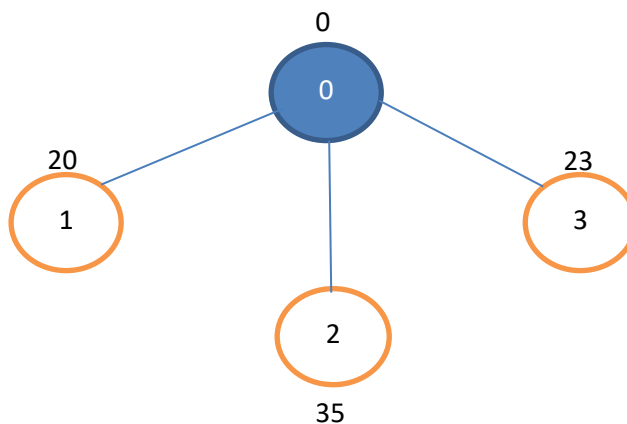
cost = $15 + 20 + 23 = 58$ (optimal cost)

Algoritma Prim

Inisialisasi pertama kali

mstSet = {0, INF, INF, INF}

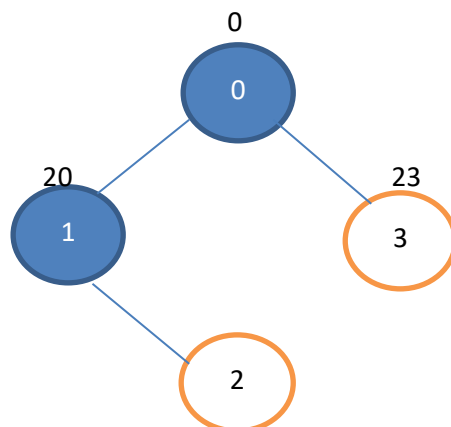
mst = {1, 2, 3}



Pilih titik yang belum masuk kedalam mstSet dengan keyvalue terkecil, titik 1 memiliki keyvalue terkecil, maka titik 1 dimasukkan ke dalam mstSet, kemudian munculkan cabang dari titik 1. update keyvalue apabila keyvalue dari cabang baru lebih kecil dari keyvalue sebelumnya

mstSet = {0, 1, INF, INF}

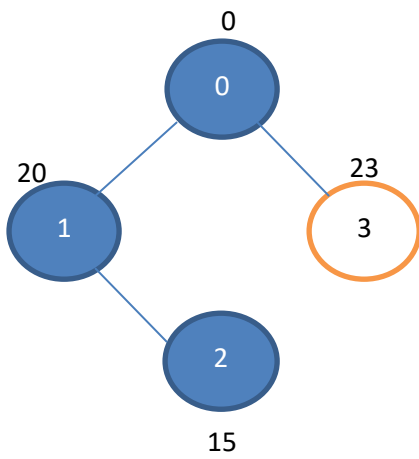
mst = {2, 3}



keyvalue titik 2 di update karena value 1-2 (15) lebih kecil daripada value 0-2 (35).

ulangi langkah sebelumnya hingga mstSet sudah terisi semua titik

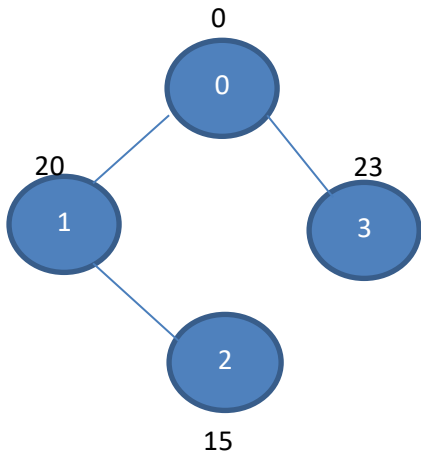
UPDATED 15



mstSet={0,1,2,INF}

mst={3}

cabang titik 2-3 tidak dimunculkan
karena keyvalue 2-3 (60) lebih besar
dari key value sebelumnya (23)



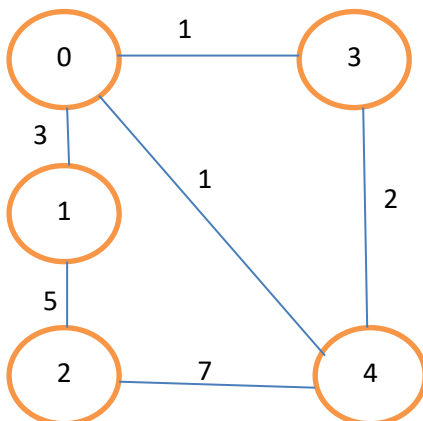
mstSet={0,1,2,3}

mst={}

ini penyelesaian minimum spanning tree menggunakan
prim's algorithm

cost=0+15+20+23=58 (optimum cost)

2)



buat minimum spanning tree dari grafik diatas menggunakan algoritma kruskal
dan prim

Jawab:

Algoritma Kruskal

Grafik diatas memiliki 5 titik , itu berarti minimum spanning tree yang terbentuk akan memiliki $(5-1) = 4$ edge

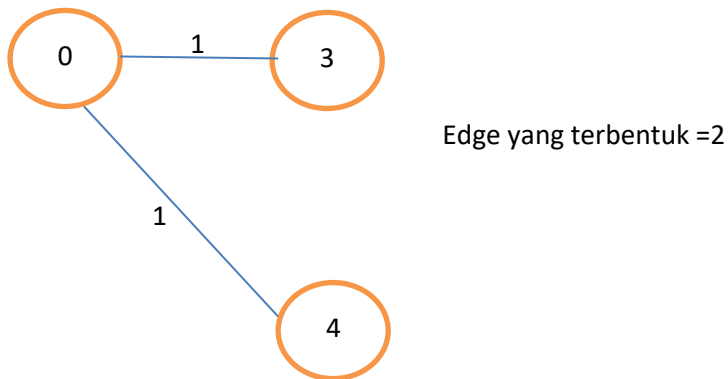
Sorting garis berdasarkan weight secara ascending

weight	edge
1	0-3
1	0-4
2	3-4
3	0-1
5	1-2
7	2-4

1.pilih edge 0-3, tidak ada cycle yang terbentuk, masukkan

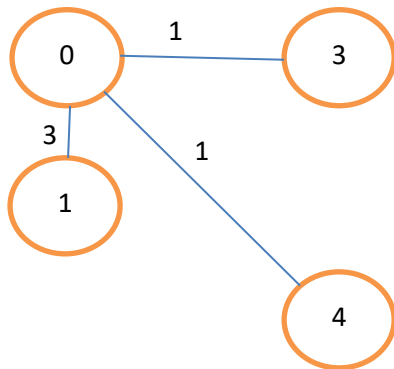


2.pilih edge 0-4, tidak ada cycle yang terbentuk, masukkan



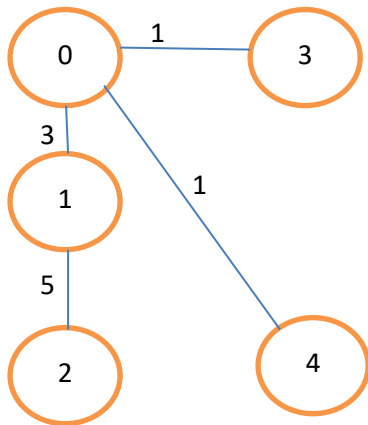
3.pilih edge 3-4, jika dimasukkan akan membentuk sebuah cycle, tidak dimasukkan

4. pilih edge 0-1, tidak ada cycle yang terbentuk, masukkan



Edge yang terbentuk =3

5. pilih edge 1-2, tidak ada cycle yang terbentuk, masukkan



Edge yang terbentuk =4

algoritma berhenti disini

cost yang didapatkan dari algoritma kruskal :

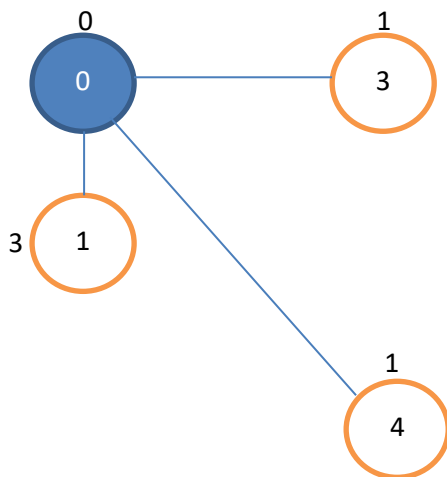
cost=1+1+3+5=10 (optimal cost)

Algoritma Prim

Inisialisasi awal

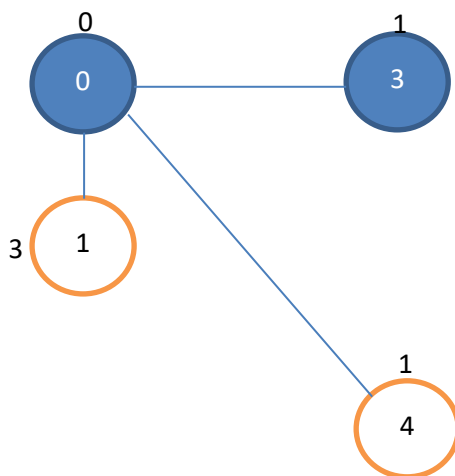
(1) mstSet = {0, INF, INF, INF, INF}

mst = {1, 2, 3, 4}



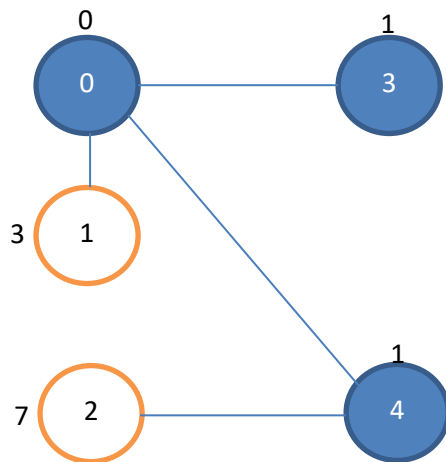
(2) mstSet = {0, 3, INF, INF, INF}

mst = {1, 2, 4}



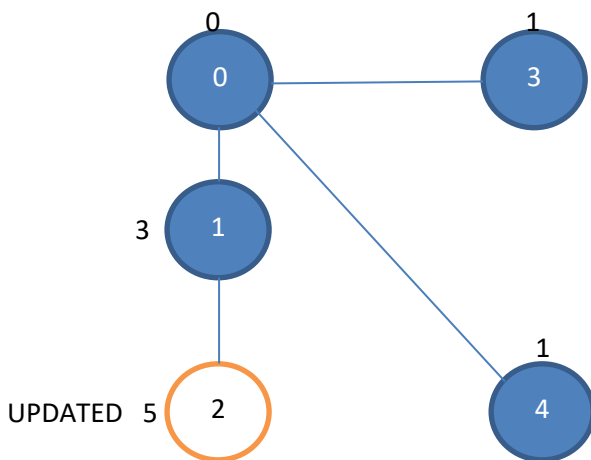
(3) mstSet = {0, 3, 4, INF, INF}

mst = {1,2}



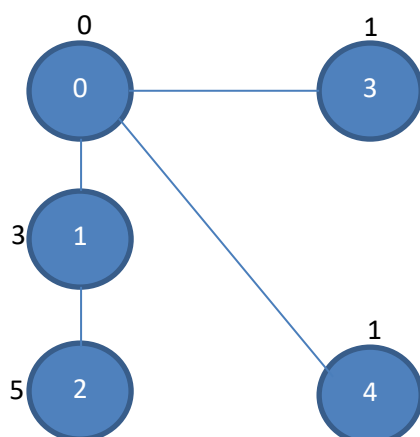
(4) mstSet = {0, 3, 4, 1, INF}

mst = {2}



(5) mstSet = {0, 3, 4, 1, 2}

mst = {}



ini penyelesaian minimum spanning tree menggunakan prim's algorithm

cost=0+1+1+3+5=10 (optimal cost)

B. Data Compression

Kompresi adalah proses pengubahan data yang berupa kumpulan karakter menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu transmisi data. Data compression bisa dilakukan dengan prinsip kode Huffman. Prinsip kode Huffman adalah mengganti karakter yang paling sering muncul didalam data dengan kode yang lebih pendek, sedangkan karakter yang lebih jarang muncul dikodekan dengan kode yang lebih panjang

Algoritma Greedy bisa diterapkan untuk membentuk kode prefiks yang optimal pada kode Huffman.

Langkah-langkah pembentukkan pohon Huffman dengan algoritma Greedy adalah sebagai berikut :

1. Baca semua karakter di dalam data untuk menghitung frekuensi kemunculan setiap karakter. Setiap karakter penyusun data dinyatakan sebagai pohon bersimpul tunggal. Dan setiap simpul ini di-*assign* dengan frekuensi kemunculan karakter tersebut.
2. Terapkan strategi greedy dengan menggabungkan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon penyusunnya.
3. Ulangi langkah dua sampai hanya tersisa satu buah pohon Huffman. Agar pemilihan dua pohon yang akan digabungkan berlangsung dengan cepat, maka semua pohon yang ada selalu teratur menaik berdasarkan frekuensi.
4. Baca kembali karakter-karakter di dalam data, kodekan setiap karakter dengan kode Huffman yang bersesuaian

Contoh Masalah :

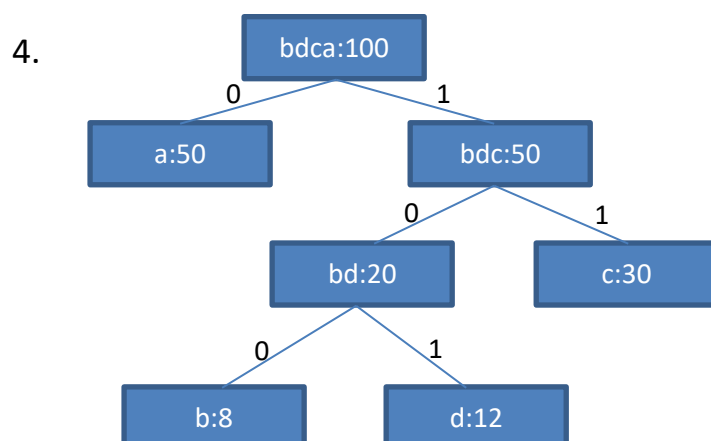
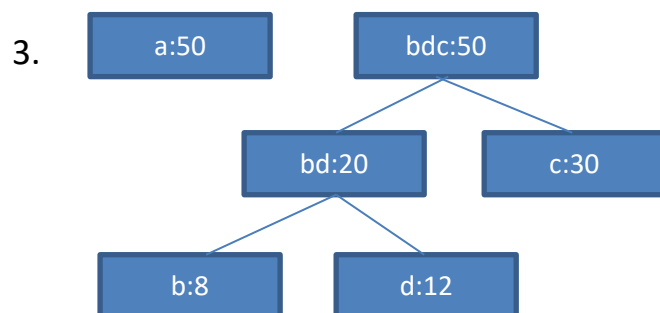
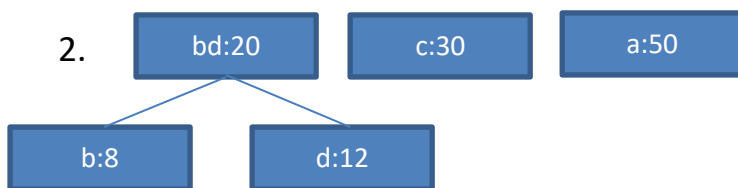
1) Terdapat data dengan panjang 100 karakter dan disusun oleh huruf a,b,c,d dengan kemunculan frekuensi setiap huruf sebagai berikut:

Karakter	Frekuensi
a	50%
b	8%
c	30%
d	12%

buatlah kode Huffman dengan algoritma Greedy dan tentukan presentase kompresi data, jika digunakan 3 bit untuk mengkodekan setiap karakter sebelum pengompresan!

Jawab:

Langkah-langkah pembentukan pohon huffman



Karakter	Frekuensi	Kode
a	50%	0 (1 bit)
b	8%	100 (3 bit)
c	30%	11 (2 bit)
d	12%	101 (3 bit)

ukuran data sebelum di kompres : $3 \times 100 = 300$ bit

ukuran data setelah di kompres:

$((0,5 \times 1) + (0,08 \times 3) + (0,3 \times 2) + (0,12 \times 3)) \times 100 = 170$ bit

data yang di kompres sebesar :

$$\frac{(300-170)}{300} \times 100\% = 43,33\%$$

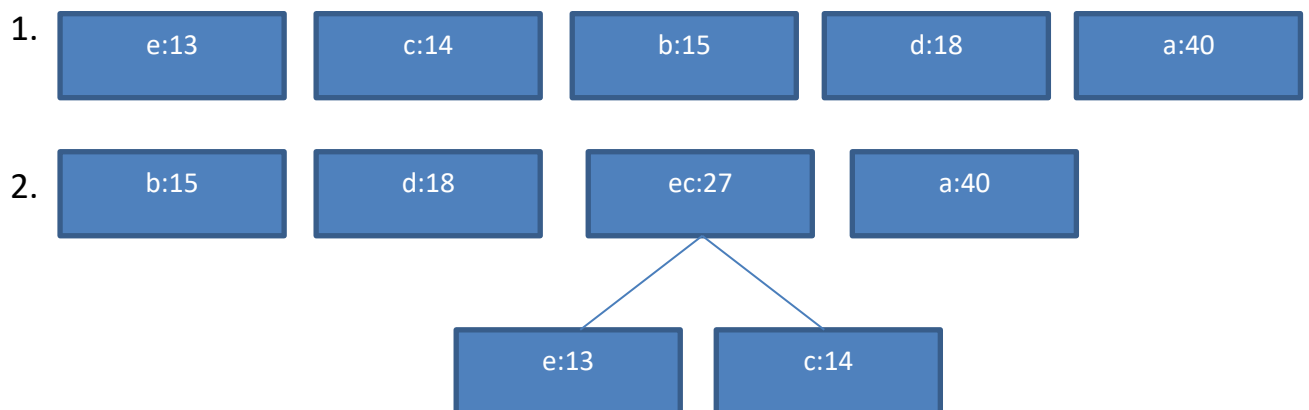
2) terdapat data dengan panjang 500 karakter dan disusun oleh huruf a,b,c,d,e dengan kemunculan frekuensi setiap huruf sebagai berikut:

Karakter	Frekuensi
a	40%
b	15%
c	14%
d	18%
e	13%

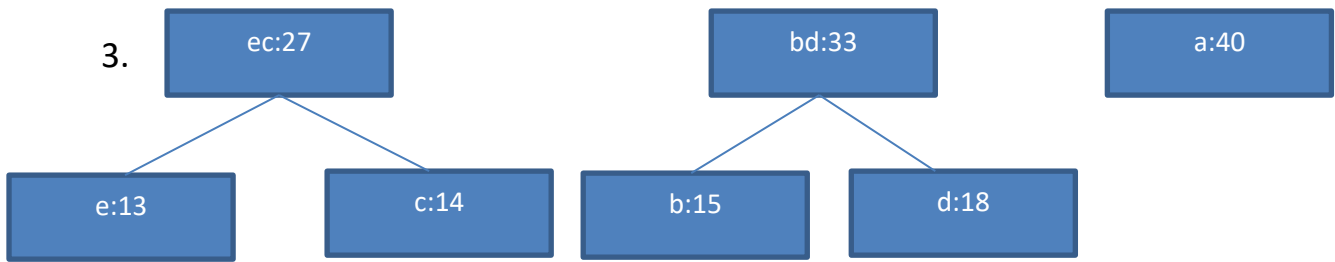
buatlah kode Huffman dengan algoritma Greedy dan tentukan presentase kompresi data, jika digunakan 3 bit untuk mengkodekan setiap karakter sebelum pengompresan!

Jawab:

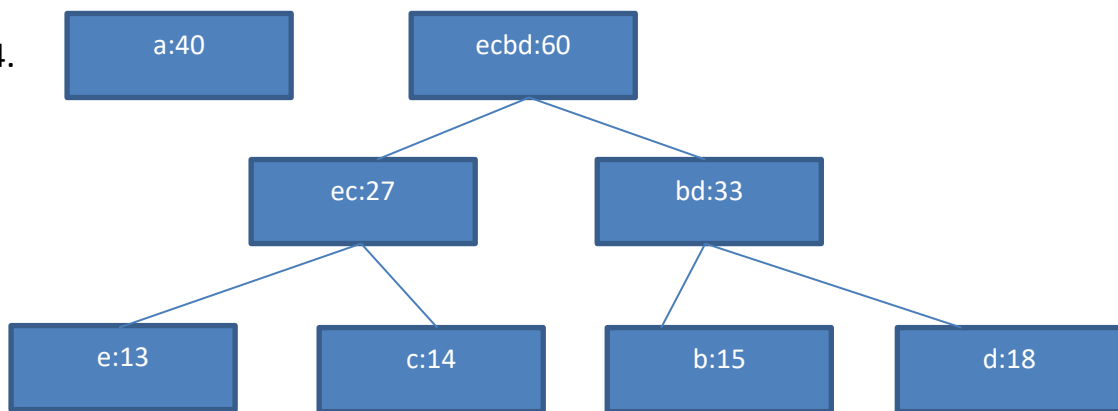
Langkah-langkah pembentukan pohon Huffman



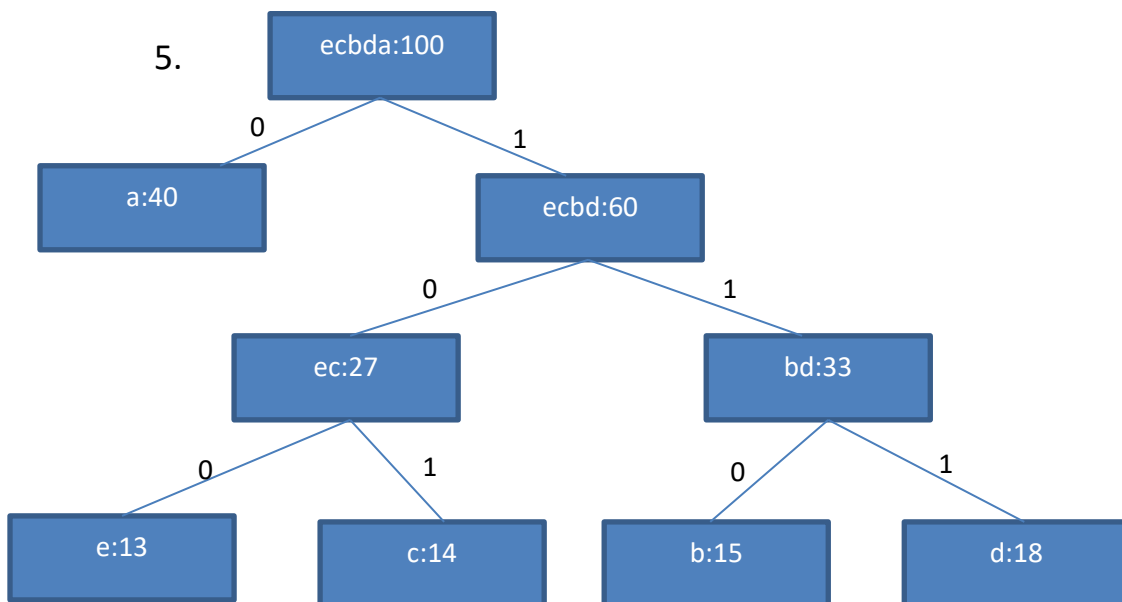
3.



4.



5.



Karakter	Frekuensi	Kode
a	40%	0 (1 bit)
b	15%	110 (3 bit)
c	14%	101 (3 bit)
d	18%	111 (3 bit)
e	13%	100 (3 bit)

ukuran data sebelum di kompres : $3 \times 500 = 1500$ bit

ukuran data setelah di kompres:

$$((0,4 \times 1) + (0,15 \times 3) + (0,14 \times 3) + (0,18 \times 3) + (0,13 \times 3)) \times 500 = 1100 \text{ bit}$$

data yang di kompres sebesar:

$$\frac{(1500-1100)}{1500} \times 100\% = 26,67\%$$