

Greedy Technique (2)

Assignment problem Using Greedy

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

- Greedy by Person : for each person find the job which cost is minimum
[P1, J2] : cost 2
[P2, J3] : cost 3
[P3, J1] : cost 5
[P4, J4] : cost 4
Total Cost = 14
- Greedy by Job : For each Job find the person who cost is minimum
[J1, P3] : cost 5
[J2, P1] : cost 2
[J3, P2] : cost 3
[J4, P4] : cost 4
Total Cost = 14

Assignment problem using Exhaustive Search

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

assume solution denoted by n tuple X
 $=\{X_1, X_2, X_3, X_4\}$

List of possible solution :

X	Cost
{P1,P2,P3,P4}	$9+4+1+4 = 18$
{P2,P1,P3,P4}	$6+2+1+4 = 13$
{P3,P2,P1,P4}	$5+4+7+4 = 20$
{P4,P2,P3,P1}	$7+4+1+8 = 20$
{P1,P3,P2,P4}	$9+8+3+4 = 24$
{P1,P4,P3,P2}	$9+6+1+7 = 23$
.....	

Job Scheduling with deadlines

- Problem :
 - Ada n buah *job* yang akan dikerjakan oleh sebuah mesin;
 - tiap *job* diproses oleh mesin selama 1 satuan waktu dan tenggat waktu (*deadline*) setiap *job* i adalah $d_i \geq 0$;
 - *job* i akan memberikan keuntungan sebesar p_i jika dan hanya jika *job* tersebut diselesaikan tidak melebihi tenggat waktunya;

- Bagaimana memilih *job-job* yang akan dikerjakan oleh mesin sehingga keuntungan yang diperoleh dari pengerjaan itu maksimum?

- Fungsi obyektif persoalan ini:

$$\text{Maksimasi } F = \sum_{i \in J} p_i$$

- Solusi layak: himpunan J yang berisi urutan *job* yang sedemikian sehingga setiap *job* di dalam J selesai dikerjakan sebelum tenggat waktunya.
- Solusi optimum ialah solusi layak yang memaksimumkan F .

Contoh 7. Misalkan A berisi 4 *job* ($n = 4$):

$$(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$$

$$(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$$

Mesin mulai bekerja jam 6.00 pagi.

<i>Job</i>	Tenggat (d_i)	Harus selesai sebelum pukul
1	2 jam	8.00
2	1 jam	7.00
3	2 jam	8.00
4	1 jam	7.00

Solusi dgn Exhaustive Search

- Cari himpunan bagian (*subset*) *job* yang layak dan memberikan total keuntungan terbesar

<u>Barisan <i>job</i></u>	<u>Keuntungan (<i>F</i>)</u>	<u>Keterangan</u>
{}	0	layak
{1}	50	layak
{2}	10	layak
{3}	15	layak
{4}	30	layak
{1, 2}	-	tidak layak
{1, 3}	65	layak
{1, 4}	-	tidak layak
{2, 1}	60	layak
{2, 3}	25	layak
{2, 4}	-	tidak layak
{3, 1}	65	layak
{3, 2}	-	tidak layak
{3, 4}	-	tidak layak
{4, 1}	80	layak (Optimum!)
{4, 2}	-	tidak layak
{4, 3}	45	layak

Solusi dgn Greedy

Strategi : Pada setiap langkah, pilih *job i* dengan p_i yang terbesar untuk menaikkan nilai fungsi obyektif F .

Contoh: $(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$

$(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

Langkah	J	$F = \sum p_i$	Keterangan
0	$\{\}$	0	-
1	$\{1\}$	50	layak
2	$\{4, 1\}$	$50 + 30 = 80$	layak
3	$\{4, 1, 3\}$	-	tidak layak
4	$\{4, 1, 2\}$	-	tidak layak

Solusi optimal: $J = \{4, 1\}$ dengan $F = 80$.

```
function JobSchedulling1(input C : himpunan_job) → himpunan_job  
{ Menghasilkan barisan job yang akan diproses oleh mesin }
```

Deklarasi

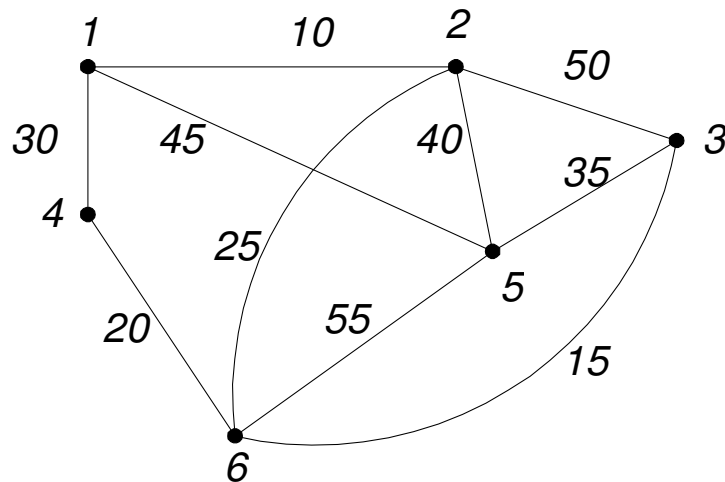
```
i : integer  
J : himpunan_job    { solusi }
```

Algoritma

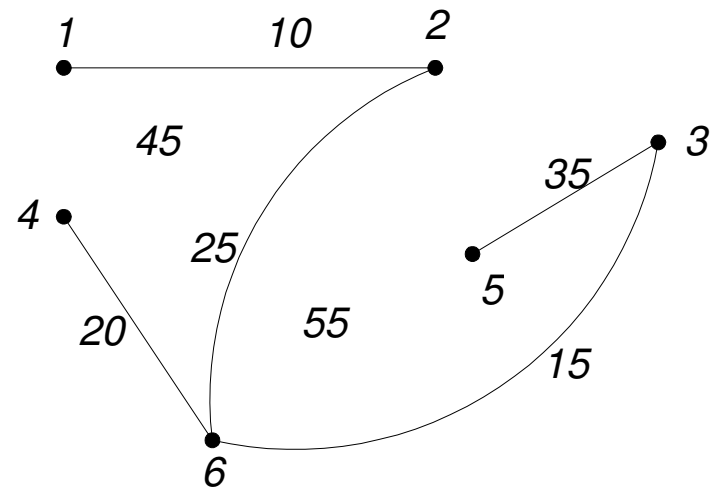
```
J ← {}  
while C ≠ {} do  
    i ← job yang mempunyai p[i] terbesar  
    C ← C - {i}  
    if (semua job di dalam J ∪ {i} layak) then  
        J ← J ∪ {i}  
    endif  
endwhile  
{ C = {} }  
return J
```

Kompleksitas algoritma greedy : $O(n^2)$.

Minimum Spanning Tree



(a) Graf $G = (V, E)$



(b) Pohon merentang minimum

Minimum Spanning Tree

- *Prim's Algorithm*
 - Pada setiap langkah, pilih sisi dari graf G yang mempunyai bobot minimum dengan syarat sisi tersebut tetap terhubung dengan pohon merentang minimum T yang telah terbentuk.
 - Kompleksitas waktu asimptotik: $O(n^2)$.
- *Kruskal's Algorithm*
 - Pada setiap langkah, pilih sisi dari graf G yang mempunyai bobot minimum tetapi sisi tersebut tidak membentuk sirkuit di T .
 - Kompleksitas asimptotik $O(|E| \log |E|)$ dengan $|E|$ adalah jumlah sisi di dalam graf G .

Prim's Algorithm

$F := \{\}$

$Y := \{v_1\}$

While the instance is not solved do

 select a vertex in $V - Y$ that is nearest to Y

 add the vertex to Y

 add the edge to F

 if $Y = V$ then

 the instance is solved

End;

Kruskal Algorithm

$F := \{\}$

Create disjoint subsets of V , one for each vertex

Sort the edges in E in increasing order

While the instance is not solved do

 select next edge

 if the edge connect two vertices in disjoint subset then

 merge the subset

 add the edge to F

 if all the subset are merged then

 the instance is solved

End;

Algoritma Djikstraa

Persoalan:

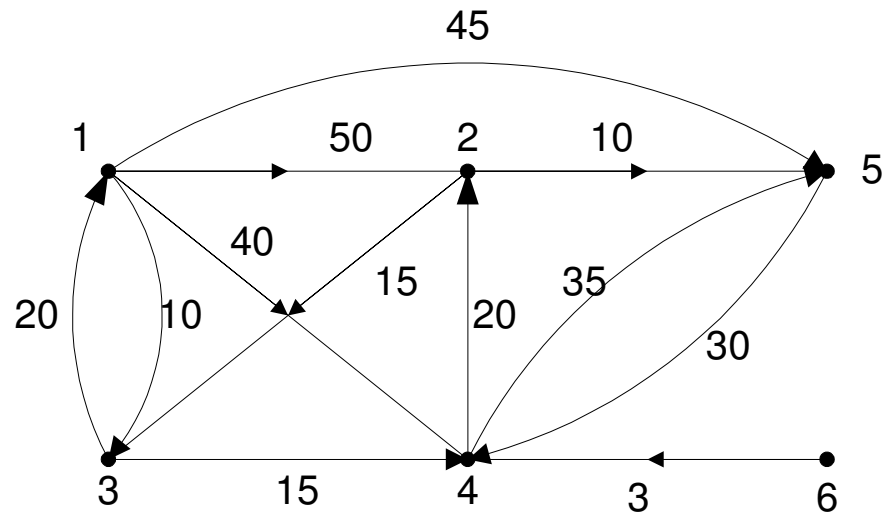
Diberikan graf berbobot $G = (V, E)$. Tentukan lintasan terpendek dari sebuah simpul asal a ke setiap simpul lainnya di G .

Asumsi yang kita buat adalah bahwa semua sisi berbobot positif.

Strategi *greedy*:

Lintasan dibentuk satu per satu. Lintasan berikutnya yang dibentuk ialah lintasan yang meminimumkan jumlah jaraknya.

Contoh 8.



Simpul asal	Simpul tujuan	Lintasan terpendek	Jarak
1	3	$1 \rightarrow 3$	10
1	4	$1 \rightarrow 3 \rightarrow 4$	25
1	2	$1 \rightarrow 3 \rightarrow 4 \rightarrow 2$	45
1	5	$1 \rightarrow 5$	45
1	6	tidak ada	-

Next

Devide and Conquer