

Program Dinamis

(Bagian 1)

Bahan Kuliah IF2251 Strategi Algoritmik

Oleh: Rinaldi Munir

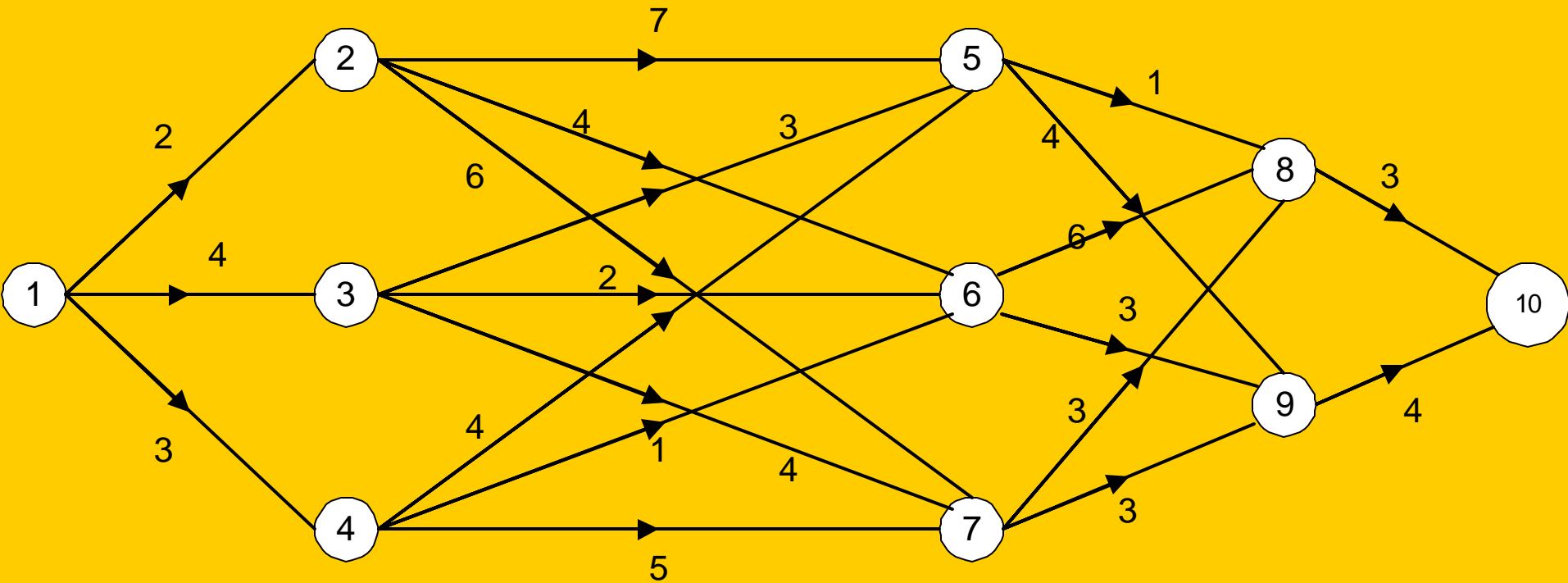
Program Dinamis

- **Program Dinamis** (*dynamic programming*):
 - metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*)
 - sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Pada penyelesaian persoalan dengan metode ini:

1. terdapat sejumlah berhingga pilihan yang mungkin,
2. solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya,
3. kita menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Tinjau graf di bawah ini. Kita ingin menemukan lintasan terpendek dari 1 ke 10.



Prinsip Optimalitas

- Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan **Prinsip Optimalitas**.
- Prinsip Optimalitas: *jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal.*

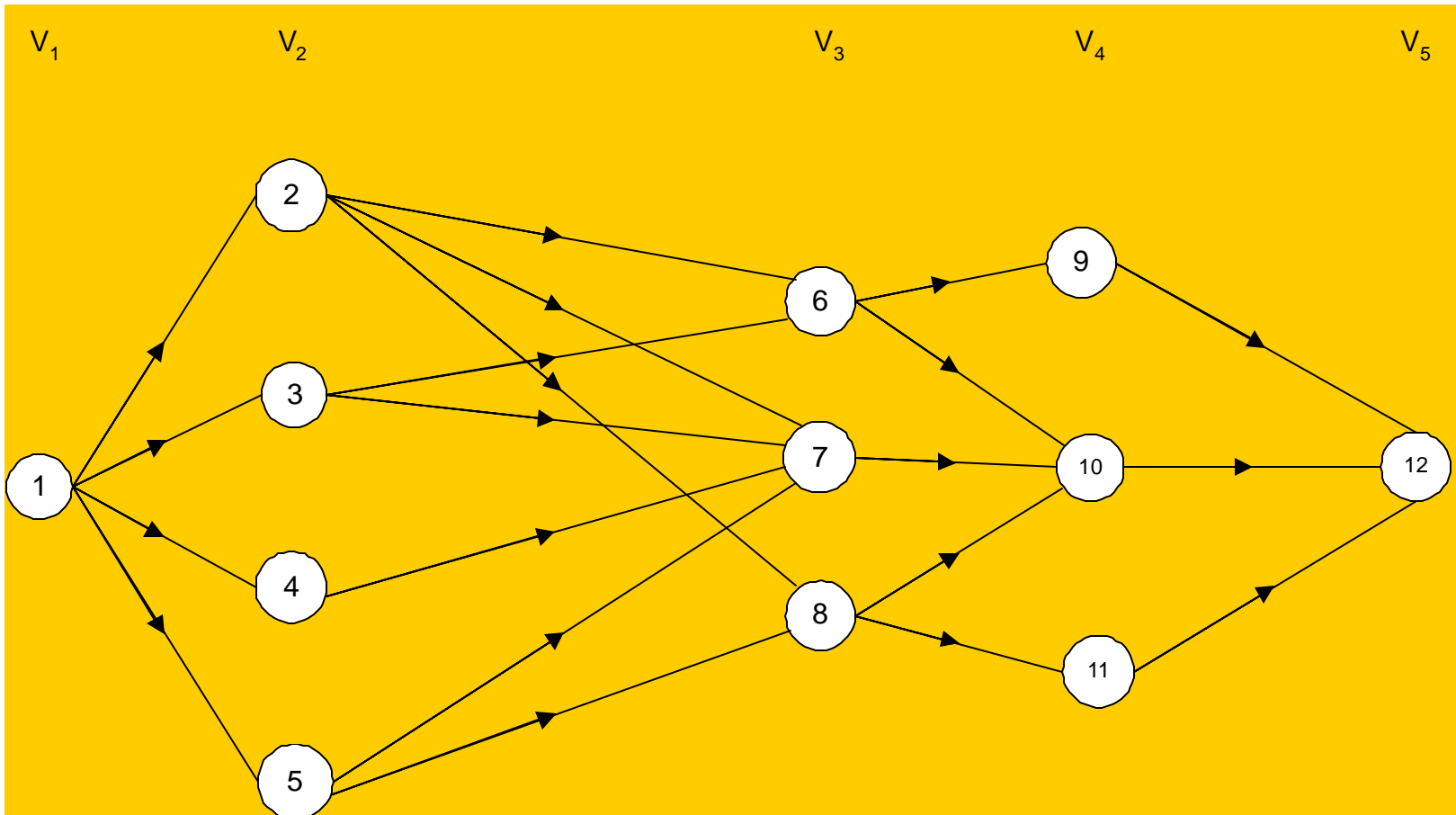
- Prinsip optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap $k + 1$, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal.
- ongkos pada tahap $k + 1 =$
(ongkos yang dihasilkan pada tahap k) +
(ongkos dari tahap k ke tahap $k + 1$)

- Dengan prinsip optimalitas ini dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya.
- Pada metode *greedy* hanya satu rangkaian keputusan yang pernah dihasilkan, sedangkan pada metode program dinamis lebih dari satu rangkaian keputusan. Hanya rangkaian keputusan yang memenuhi prinsip optimalitas yang akan dihasilkan.

Karakteristik Persoalan Program Dinamis

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut.

Graf multistage (*multistage graph*). Tiap simpul di dalam graf tersebut menyatakan status, sedangkan V_1, V_2, \dots menyatakan tahap.



3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.

6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasikan keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Dua pendekatan PD

- Dua pendekatan yang digunakan dalam PD: maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*).

- Misalkan x_1, x_2, \dots, x_n menyatakan peubah (*variable*) keputusan yang harus dibuat masing-masing untuk tahap 1, 2, ..., n .
Maka,
 1. Program dinamis maju. Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n . Runtunan peubah keputusan adalah x_1, x_2, \dots, x_n .

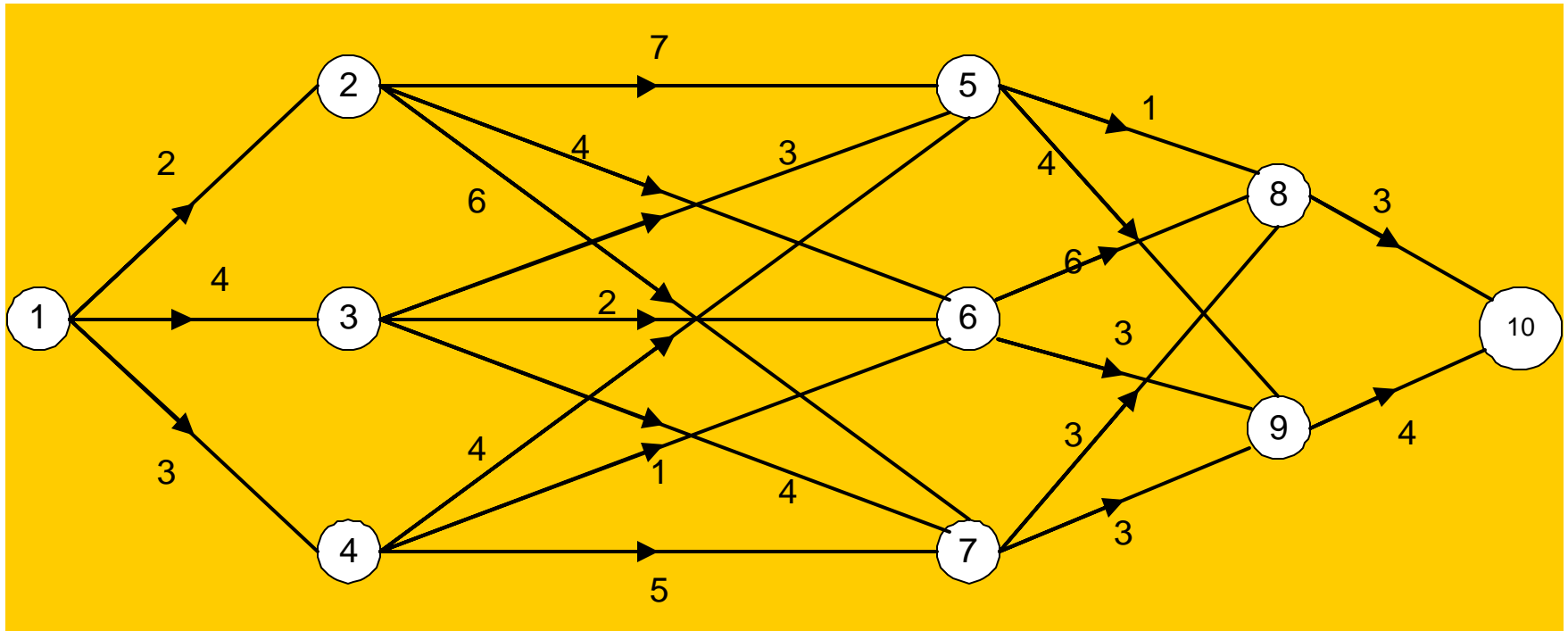
2. Program dinamis mundur. Program dinamis bergerak mulai dari tahap n , terus mundur ke tahap $n - 1$, $n - 2$, dan seterusnya sampai tahap 1. Runtunan peubah keputusan adalah x_n, x_{n-1}, \dots, x_1 .

Langkah-langkah Pengembangan Algoritma Program Dinamis

1. Karakteristikkan struktur solusi optimal.
2. Definisikan secara rekursif nilai solusi optimal.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Konstruksi solusi optimal.

Lintasan Terpendek (*Shortest Path*)

- Tentukan lintasan terpendek dari simpul 1 ke simpul 10:



Penyelesaian dengan Program Dinamis Mundur

- Misalkan x_1, x_2, \dots, x_4 adalah simpul-simpul yang dikunjungi pada tahap k ($k = 1, 2, 3, 4$).
- Maka rute yang dilalui adalah
$$1 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 ,$$
yang dalam hal ini $x_4 = 10$.

Pada persoalan ini,

- *Tahap* (k) adalah proses memilih simpul tujuan berikutnya (ada 4 tahap).
- *Status* (s) yang berhubungan dengan masing-masing tahap adalah simpul-simpul di dalam graf.

Relasi rekurens berikut menyatakan lintasan terpendek dari status s ke x_4 pada tahap k :

$$f_4(s) = c_{sx_4} \quad (\text{basis})$$

$$f_k(s) = \min_{x_k} \{c_{sx_k} + f_{k+1}(x_k)\}, \quad (\text{rekurens})$$

$$k = 1, 2, 3$$

Keterangan:

- a. x_k : peubah keputusan pada tahap k ($k = 1, 2, 3$).
- b. c_{sx_k} : bobot (*cost*) sisi dari s ke x_k
- c. $f_k(s, x_k)$: total bobot lintasan dari s ke x_k
- d. $f_k(s)$: nilai minimum dari $f_k(s, x_k)$

Tujuan program dinamis mundur: mendapatkan $f_1(1)$ dengan cara mencari $f_4(s), f_3(s), f_2(s)$ terlebih dahulu.

Tahap 4:

$$f_4(s) = c_{sx_4}$$

s	Solusi Optimum	
	$f_4(s)$	x_4^*
8	3	10
9	4	10

Catatan: x_k^* adalah nilai x_k yang meminimumkan $f_k(s, x_k)$.

Tahap 3:

$$f_3(s) = \min_{x_3} \{c_{sx_3} + f_4(x_3)\}$$

$s \backslash x_3$	$f_3(s, x_3) = c_{s,x_3} + f_4(x_3)$		Solusi Optimum	
	8	9	$f_3(s)$	x_3^*
5	4	8	4	8
6	9	7	7	9
7	6	7	6	8

Tahap 2:

$$f_2(s) = \min_{x_2} \{c_{sx_2} + f_3(x_2)\}$$

$s \backslash x_2$	$f_2(s, x_2) = c_{s,x_2} + f_3(x_2)$			Solusi Optimum	
	5	6	7	$f_2(s)$	x_2^*
2	11	11	12	11	5 atau 6
3	7	9	10	7	5
4	8	8	11	8	5 atau 6

Tahap 1:

$$f_1(s) = \min_{x_1} \{c_{sx_1} + f_2(x_1)\}$$

$x_1 \backslash s$	$f_1(s, x_1) = c_{s,x_1} + f_2(x_1)$			Solusi Optimum	
	2	3	4	$f_1(s)$	x_1^*
1	13	11	11	11	3 atau 4

Solusi optimum dapat dibaca pada tabel di bawah ini:

	x_1	x_2	x_3	x_4	Panjang Lintasan Terpendek
1	3	5	8	10	11
		5	8	10	11
		6	9	10	11

Jadi ada tiga lintasan terpendek dari 1 ke 10, yaitu

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 10$$

$$1 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 10$$

$$1 \rightarrow 4 \rightarrow 6 \rightarrow 9 \rightarrow 10$$

Panjang ketiga lintasan tersebut sama, yaitu 11.

Penganggaran Modal

(Capital Budgeting)

- Sebuah perusahaan berencana akan mengembangkan usaha (proyek) melalui ketiga buah pabrik (*plant*) yang dimilikinya. Setiap pabrik diminta mengirimkan proposal (boleh lebih dari satu) ke perusahaan untuk proyek yang akan dikembangkan. Setiap proposal memuat total biaya yang dibutuhkan (c) dan total keuntungan (*revenue*) yang akan diperoleh (R) dari pengembangan usaha itu. Perusahaan mengalokasikan Rp 5 milyar untuk alokasi dana bagi ketiga pabriknya itu.

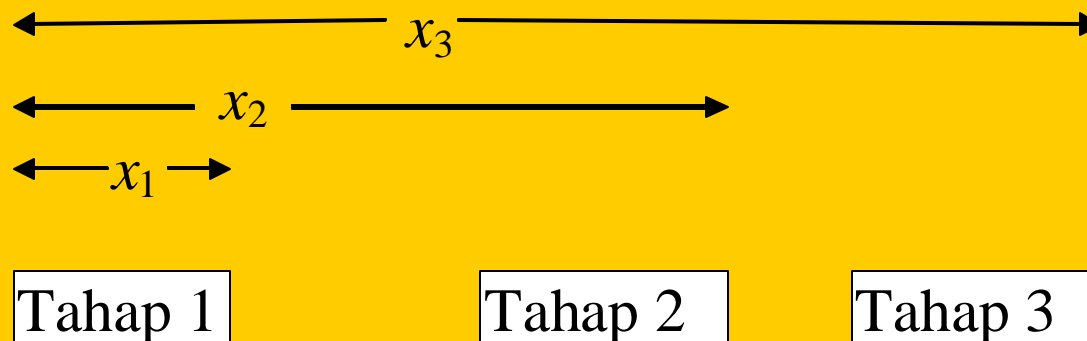
- Tabel berikut meringkaskan nilai c dan R untuk masing-masing proposal proyek. Proposal proyek bernilai-nol sengaja dicantumkan yang berarti tidak ada alokasi dana yang diberikan untuk setiap pabrik. Tujuan Perusahaan adalah memperoleh keuntungan yang maksimum dari pengalokasian dana sebesar Rp 5 milyar tersebut. Selesaikan persoalan ini dengan program dinamis.

Peubah status yang terdapat pada tahap 1, 2, dan 3:

$x_1 = \sum$ modal yang dialokasikan pada tahap 1

$x_2 = \sum$ modal yang dialokasikan pada tahap 1 dan 2

$x_3 = \sum$ modal yang dialokasikan pada tahap 1, 2, dan 3



Kemungkinan nilai-nilai untuk x_1 dan x_2 adalah 0, 1, 2, 3, 4, 5 (milyar), sedangkan nilai untuk x_3 adalah 5

Proyek	Pabrik 1		Pabrik 2		Pabrik 3	
	c_1	R_1	c_2	R_2	c_3	R_3
1	0	0	0	0	0	0
2	1	5	2	8	1	3
3	2	6	3	9	-	-
4	-	-	4	12	-	-

Penyelesaian dengan Program Dinamis Maju.

Misalkan,

$R_k(p_k)$ = keuntungan dari alternatif p_k pada tahap k

$f_k(x_k)$ = keuntungan optimal dari tahap 1, 2, ..., dan k yang diberikan oleh status x_k

Relasi rekurens keuntungan optimal:

$$f_1(x_1) = \max_{\substack{\text{feasible} \\ \text{proposal } - p_1}} \{R_1(p_1)\} \quad (\text{basis})$$

$$f_k(x_k) = \max_{\substack{\text{feasible} \\ \text{proposal } - p_k}} \{R_k(p_k) + f_{k-1}(x_{k-1})\} \quad (\text{rekurens})$$
$$k = 2, 3$$

Catatan:

1. $x_{k-1} = x_k - c_k(p_k)$

$c(p_k)$ adalah biaya untuk alternatif p_k pada tahap k .

2. Proposal p_k dikatakan layak (*feasible*) jika biayanya, $c(p_k)$, tidak melebihi nilai status x_k pada tahap k .

Relasi rekurens keuntungan optimal menjadi

$$f_1(x_1) = \max_{c_1(p_1) \leq x_1} \{R_1(p_1)\} \quad (\text{basis})$$

$$f_k(x_k) = \max_{c_k(p_k) \leq x_k} \{R_k(p_k) + f_{k-1}[x_k - c_k(p_k)]\} \quad (\text{rekurens})$$
$$k = 2, 3$$

Tahap 1

$$f_1(x_1) = \max_{\substack{c_1(p_1) \leq x_1 \\ p_1=1,2,3}} \{R_1(p_1)\}$$

x_1	$R_1(p_1)$			Solusi Optimal	
	$p_1 = 1$	$p_1 = 2$	$p_1 = 3$	$f_1(x_1)$	p_1^*
0	0	-	-	0	1
1	0	5	-	5	2
2	0	5	6	6	3
3	0	5	6	6	3
4	0	5	6	6	3
5	0	5	6	6	3

Tahap 2

$$f_2(x_2) = \max_{\substack{c_2(p_2) \leq x_2 \\ p_2=1,2,3,4}} \{R_2(p_2) + f_1[(x_2 - c_2(p_2))]\},$$

x_2	$R_2(p_2) + f_1[(x_2 - c_2(p_2))]$				Solusi Optimal	
	$p_2 = 1$	$p_2 = 2$	$p_2 = 3$	$p_2 = 4$	$f_2(x_2)$	p_2^*
0	$0 + 0 = \mathbf{0}$	-	-	-	0	1
1	$0 + 5 = \mathbf{5}$	-	-	-	5	1
2	$0 + 6 = 6$	$8 + 0 = \mathbf{8}$	-	-	8	2
3	$0 + 6 = 6$	$8 + 5 = \mathbf{13}$	$9 + 0 = 9$	-	13	2
4	$0 + 6 = 6$	$8 + 6 = \mathbf{14}$	$9 + 5 = \mathbf{14}$	$12 + 0 = 12$	14	2 atau 3
5	$0 + 6 = 6$	$8 + 6 = 14$	$9 + 6 = 15$	$12 + 5 = \mathbf{17}$	17	4

Tahap 3

$$f_3(x_3) = \max_{\substack{c_3(p_3) \leq x_3 \\ p_3=1,2}} \{R_3(p_3) + f_2[(x_3 - c_3(p_3))]\},$$

x_3	$R_3(p_3) + f_2[(x_3 - c_3(p_3))]$		Solusi Optimal	
	$p_3 = 1$	$p_3 = 2$	$f_3(x_3)$	p_3^*
5	$0 + 17 = \mathbf{17}$	$3 + 14 = \mathbf{17}$	17	1 atau 2

Rekonstruksi solusi:

x_3	p_3^*	x_2	p_2^*	x_1	p_1^*	(p_1^*, p_2^*, p_3^*)
	1	$(5 - 0 = 5)$	4	$(5 - 4 = 1)$	2	$(2, 4, 1)$
1			2	$(4 - 2 = 2)$	3	$(3, 2, 2)$
	2	$(5 - 1 = 4)$	3	$(4 - 3 = 1)$	3	$(2, 3, 2)$

Integer (1/0) Knapsack

Pada persoalan ini,

1. Tahap (k) adalah proses memasukkan barang ke dalam karung (*knapsack*) (ada 3 tahap).
2. Status (y) menyatakan kapasitas muat karung yang tersisa setelah memasukkan barang pada tahap sebelumnya.

Dari tahap ke-1, kita masukkan objek ke-1 ke dalam karung untuk setiap satuan kapasitas karung sampai batas kapasitas maksimumnya. Karena kapasitas karung adalah bilangan bulat, maka pendekatan ini praktis.

- Misalkan ketika memasukkan objek pada tahap k , kapasitas muat karung sekarang adalah $y - w_k$.
- Untuk mengisi kapasitas sisanya, kita menerapkan prinsip optimalitas dengan mengacu pada nilai optimum dari tahap sebelumnya untuk kapasitas sisa $y - w_k$ (yaitu $f_{k-1}(y - w_k)$).

Penyelesaian dengan Program Dinamis

- Tahap (k) adalah proses mengalokasikan dana untuk setiap pabrik (ada 3 tahap, tiap pabrik mendefinisikan sebuah tahap).
- Status (x_k) menyatakan jumlah modal yang dialokasikan pada pada setiap tahap (namun terikat bersama semua tahap lainnya).
- Alternatif (p) menyatakan proposal proyek yang diusulkan setiap pabrik. Pabrik 1, 2, dan 3 masing-masing memiliki 3, 4 dan 2 alternatif proposal.

- Selanjutnya, kita bandingkan nilai keuntungan dari objek pada tahap k (yaitu p_k) plus nilai $f_{k-1}(y - w_k)$ dengan keuntungan pengisian hanya $k - 1$ macam objek, $f_{k-1}(y)$.
- Jika $p_k + f_{k-1}(y - w_k)$ lebih kecil dari $f_{k-1}(y)$, maka objek yang ke- k tidak dimasukkan ke dalam karung, tetapi jika lebih besar, maka objek yang ke- k dimasukkan.

- Relasi rekurens untuk persoalan ini adalah

$$f_0(y) = 0, \quad y = 0, 1, 2, \dots, M \quad (\text{basis})$$

$$f_k(y) = -\infty, \quad y < 0 \quad (\text{basis})$$

$$f_k(y) = \max_{k=1, 2, \dots, n} \{f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}, \quad (\text{rekurens})$$

- $f_k(y)$ adalah keuntungan optimum dari persoalan 0/1 *Knapsack* pada tahap k untuk kapasitas karung sebesar y .
- $f_0(y) = 0$ adalah nilai dari persoalan *knapsack* kosong (tidak ada persoalan *knapsack*) dengan kapasitas y ,
- $f_k(y) = -\infty$ adalah nilai dari persoalan *knapsack* untuk kapasitas negatif. Solusi optimum dari persoalan 0/1 *Knapsack* adalah $f_n(M)$.

Contoh: $n = 3$

$M = 5$

Barang ke- i	w_i	p_i
1	2	65
2	3	80
3	1	30

Tahap 1:

$$f_1(y) = \max\{f_0(y), p_1 + f_0(y - w_1)\}$$
$$= \max\{f_0(y), 65 + f_0(y - 2)\}$$

y			Solusi Optimum	
	$f_0(y)$	$65 + f_0(y - 2)$	$f_1(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$-\infty$	0	(0, 0, 0)
1	0	$-\infty$	0	(0, 0, 0)
2	0	65	65	(1, 0, 0)
3	0	65	65	(1, 0, 0)
4	0	65	65	(1, 0, 0)
5	0	65	65	(1, 0, 0)

Tahap 2:

$$f_2(y) = \max\{f_1(y), p_2 + f_1(y - w_2)\}$$
$$= \max\{f_1(y), 80 + f_1(y - 3)\}$$

y			Solusi Optimum	
	$f_1(y)$	$80 + f_1(y - 3)$	$f_2(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$80 + (-\infty) = -\infty$	65	(1, 0, 0)
3	65	$80 + 0 = \mathbf{80}$	80	(0, 1, 0)
4	65	$80 + 0 = \mathbf{80}$	80	(0, 1, 0)
5	65	$80 + 65 = \mathbf{145}$	145	(1, 1, 0)

Tahap 3:

$$\begin{aligned} f_3(y) &= \max\{f_2(y), p_3 + f_2(y - w_3)\} \\ &= \max\{f_2(y), 30 + f_2(y - 1)\} \end{aligned}$$

y	Solusi Optimum			
	$f_2(y)$	$30 + f_2(y - 1)$	$f_3(y)$	(x_1^*, x_2^*, x_3^*)
0	0	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$30 + 0 = 30$	65	(1, 0, 0)
3	80	$30 + 65 = \mathbf{95}$	95	(1, 0, 1)
4	80	$30 + 80 = \mathbf{110}$	110	(0, 1, 1)
5	145	$30 + 80 = 110$	145	(1, 1, 0)

Solusi optimum $X = (1, 1, 0)$ dengan $\sum p = f = 145$.