Program Dinamis (*Dynamic Programming*)

Bahan Kuliah IF2211 Strategi Algoritma

Oleh: Rinaldi Munir

Program Studi Teknik Informatika STEI-ITB

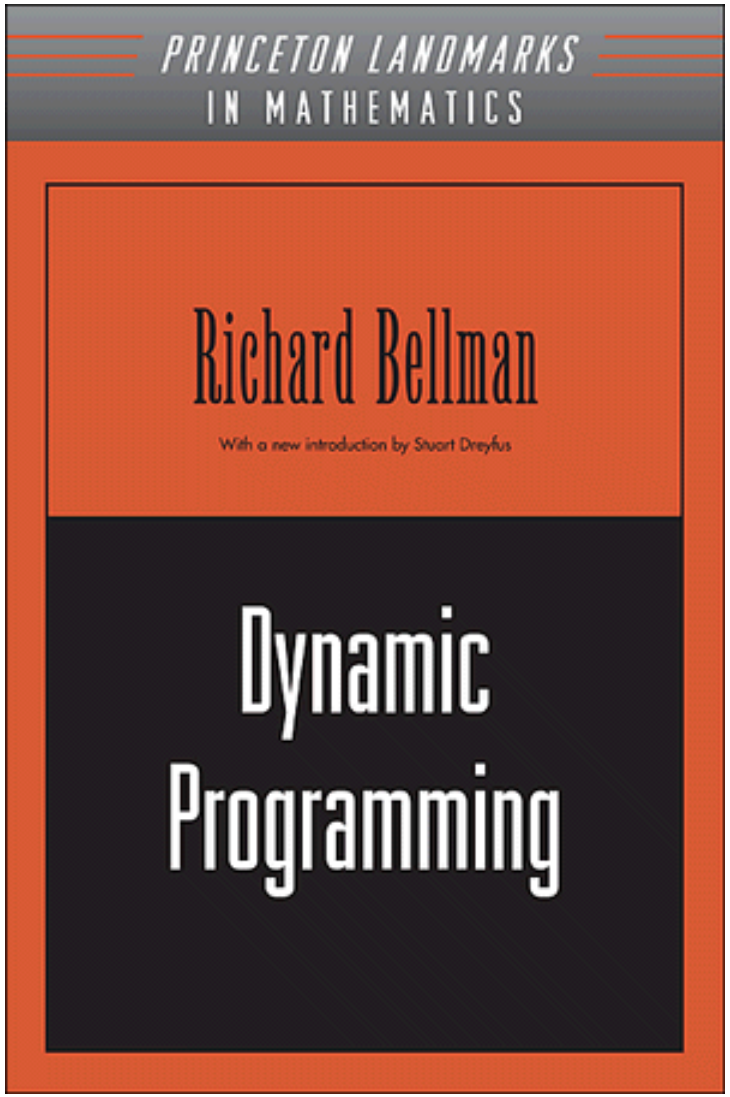
Program Dinamis

- **Program Dinamis** (*dynamic programming*):

  - metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (*stage*)

  - sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

- Istilah “program dinamis” muncul karena perhitungan solusi menggunakan tabel-tabel.


PRINCETON LANDMARKS
IN MATHEMATICS
Richard Bellman
With a new introduction by Stuart Dreyfus
Dynamic
Programming

Karakteristik penyelesaian persoalan dengan Program Dinamis:

1. terdapat sejumlah berhingga pilihan yang mungkin,

2. solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya,

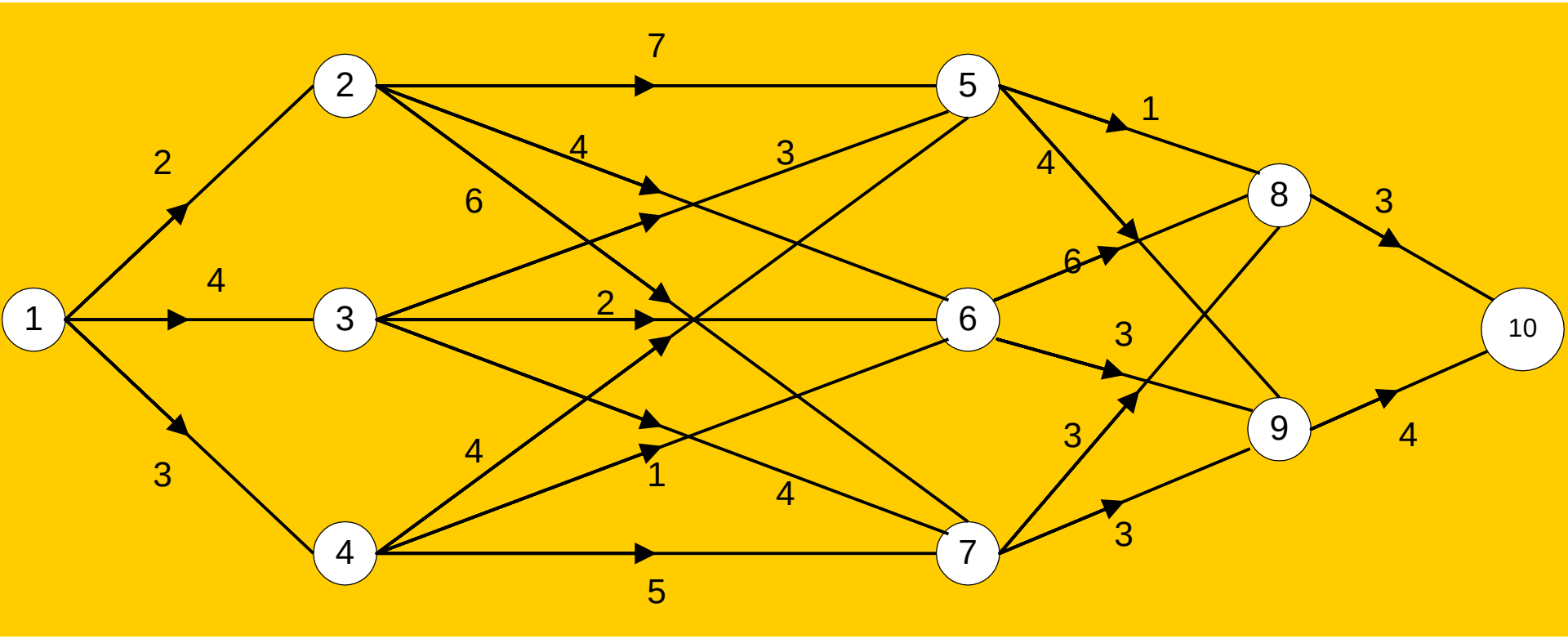
3. kita menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

- Perbedaan Algoritma *Greedy* dengan Program Dinamis:

  → Greedy: hanya satu rangkaian keputusan yang dihasilkan

  → Program dinamis: lebih dari satu rangkaian keputusan yang dipertimbangkan.

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



Greedy: 1 – 2 – 6 – 9 – 10 dengan *cost* = 2 + 4 + 3 + 4 = 13

Program Dinamis: akan dijelaskan kemudian

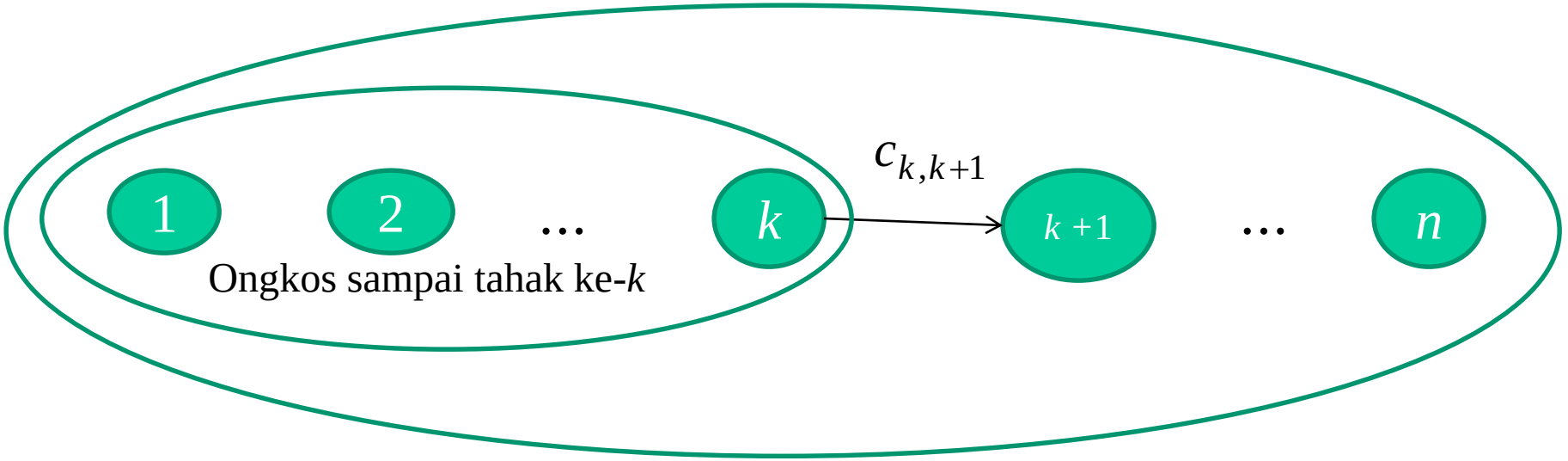
Prinsip Optimalitas

- 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)

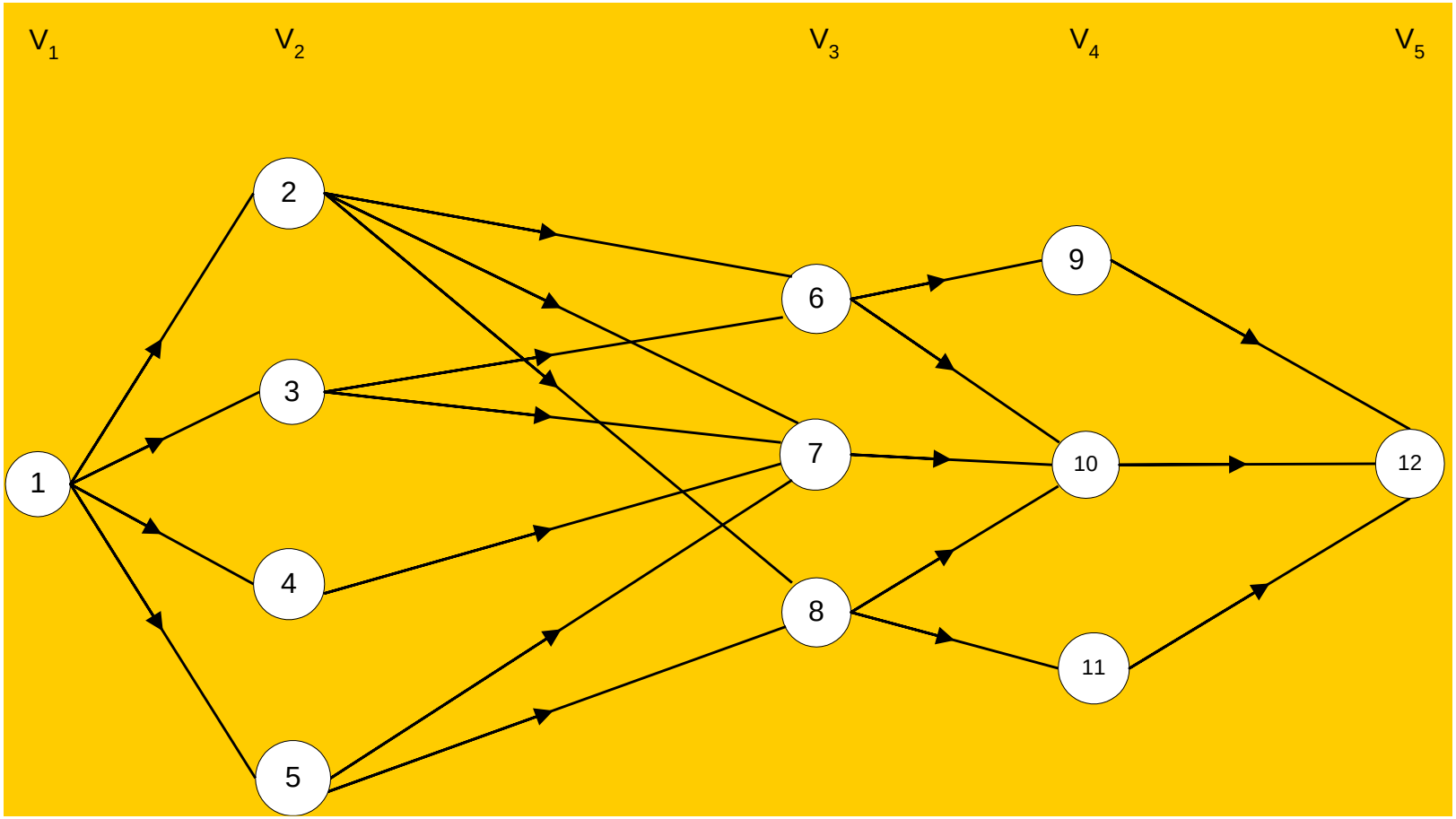


Karakteristik Persoalan Program Dinamis

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 multitahap (*multistage graph*). Tiap simpul di dalam graf tersebut menyatakan status, sedangkan V_1, V_2, … 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:
  1. PD maju (*forward* atau *up-down*)
  2. PD mundur (*backward* atau *bottom-up*).

Misalkan $x_1, x_2, \ldots, x_n$ 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 $x_1, x_2, \ldots, x_n$.

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 $x_n, x_{n-1}, \ldots, x_1$.

Prinsip optimalitas pada PD maju:

ongkos pada tahap k +1 = (ongkos yang dihasilkan pada tahap k) + (ongkos dari tahap k ke tahap k + 1)

$k = 1, 2, \ldots, n - 1$

Prinsip optimalitas pada PD mundur:

ongkos pada tahap k = (ongkos yang dihasilkan pada tahap k + 1) + (ongkos dari tahap k + 1 ke tahap k)

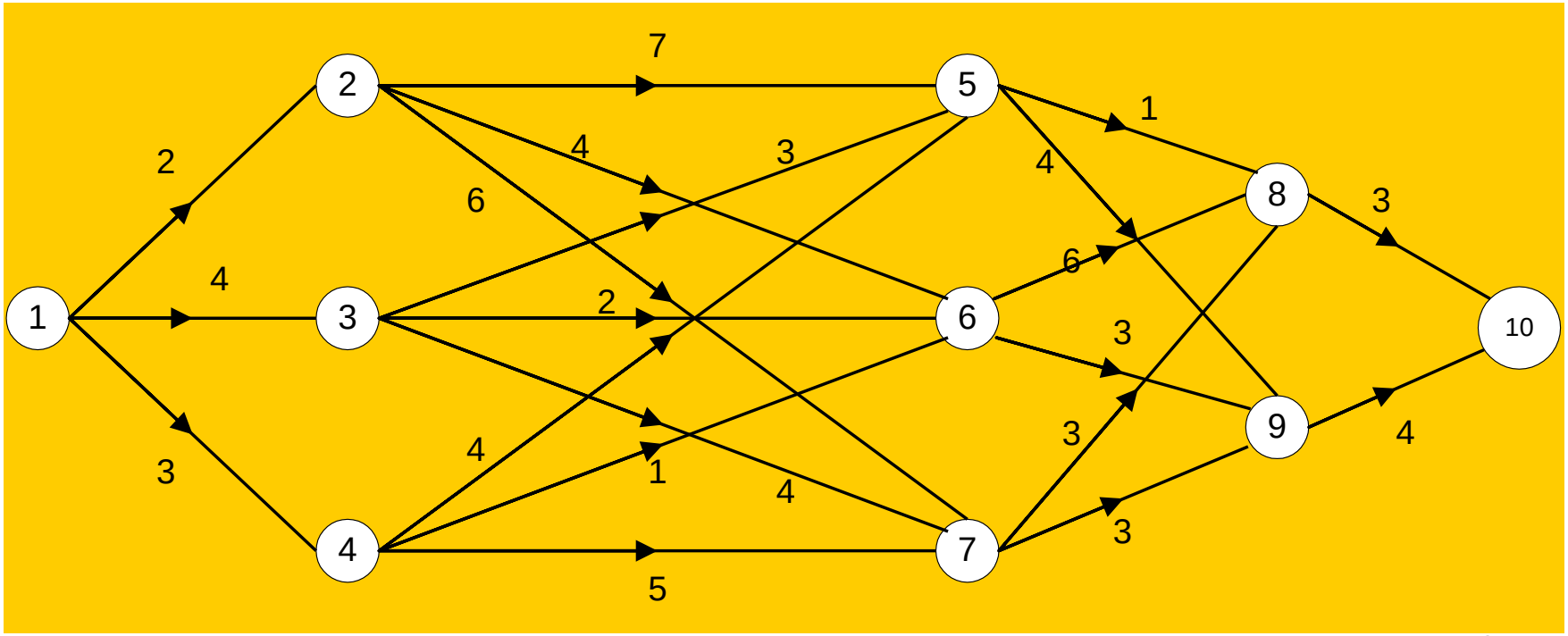
$k = n, n - 1, \ldots, 1$

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 Maju

- Misalkan $x1$, $x2$, …, $x4$ adalah simpul-simpul yang dikunjungi pada tahap k (k = 1, 2, 3, 4).

- Maka rute yang dilalui adalah

  $x1 \rightarrow x2 \rightarrow x3 \rightarrow x4 \rightarrow 10$,

  yang dalam hal ini $x1 = 1$.

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:

$$f_1(s) = c_{x_1 s} \qquad \text{(basis)}$$

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

$k = 2, 3, 4$

Keterangan:

a. x_k : peubah keputusan pada tahap k ($k = 2, 3, 4$).

b. c_{sx_k} : bobot (*cost*) sisi dari s ke x_k

c. $f_k(s)$: nilai minimum dari $f_k(x_k, s)$

d. $f_k(x_k, s)$: total bobot lintasan dari ke x_k ke s

Tujuan program dinamis maju: mendapatkan $f_4(10)$ dengan cara mencari $f_1(s)$, $f_2(s)$, $f_3(s)$ terlebih dahulu.

Tahap 1:

$$f_1(s) = c_{x1s}$$

s	Solusi Optimum	
	$f_1(s)$	x_1^*
2	2	1
3	4	1
4	3	1

Catatan: x_k^* adalah nilai x_k yang meminimumkan f_s.

Tahap 2:

$$f_2(s) = \min_{x_2} \{c_{x_2 s} + f_1(x_2)\}$$

x_2 / s	$f_2(x_2,s) = c_{x2,s} + f_1(x_2)$			Solusi Optimum	
	2	3	4	$f_2(s)$	x_2^*
5	9	7	7	7	3 atau 4
6	6	6	4	4	4
7	8	8	8	8	2, 3, 4

Tahap 3:

$$f_3(s) = \min_{x_3} \{c_{x_3 s} + f_2(x_3)\}$$

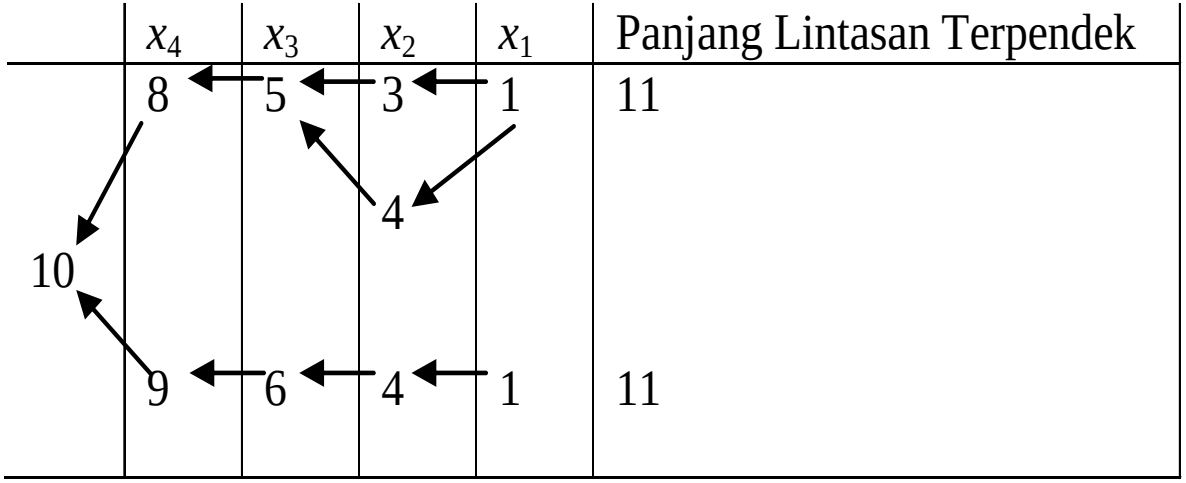
x_2 / s	$f_2(x_3, s) = c_{x3,s} + f_2(x_3)$			Solusi Optimum	
	5	6	7	$f_3(s)$	x_3^*
8	8	10	11	8	5
9	11	7	11	7	6

Tahap 4:

$$f_4(s) = \min_{x_4} \{c_{x_4 s} + f_3(x_4)\}$$

s \ x_1	$f_1(x_4, s) = c_{x4,s} + f_3(x_4)$		Solusi Optimum	
	8	9	$f_4(s)$	x_4^*
10	11	11	11	8 atau 9

Solusi optimum dapat dibaca pada tabel di bawah ini:



Jadi ada tiga lintasan terpendek dari 1 ke 10, yaitu

$1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 10$
$1 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 10$
$1 \rightarrow 4 \rightarrow 6 \rightarrow 9 \rightarrow 10$

yang mana 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 menganggarkan 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 ntuk setiap pabrik. Tujuan Perusahaan adalah memperoleh keuntungan yang maksimum dari pengalokasian dana sebesar Rp 5 milyar tersebut. Selesaikan persoalan ini dengan program dinamis.

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

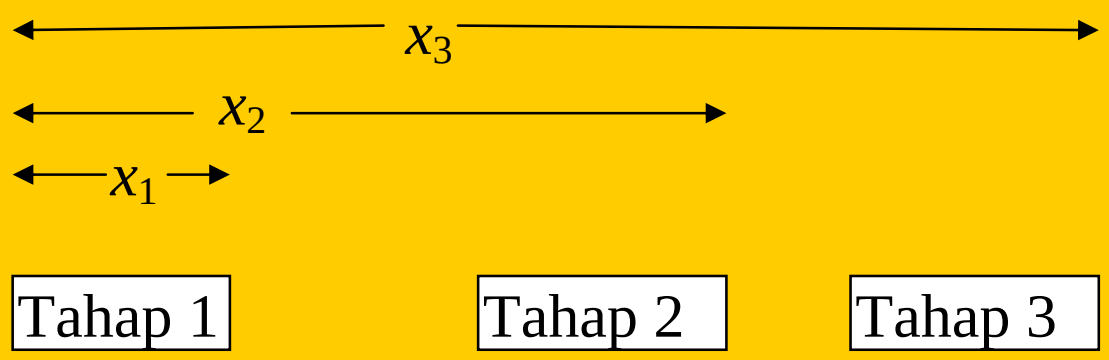
- Tahap (k) adalah proses mengalokasikan dana untuk setiap pabrik (ada 3 tahap, tiap pabrik mendefinisikan sebuah tahap).

- 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.

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

x_1 = ∑ modal yang dialokasikan pada tahap 1
x_2 = ∑ modal yang dialokasikan pada tahap 1 dan 2
x_3 = ∑ 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

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:

$$f_1(x_1) = \max_{\substack{feasible \\ proposal\_p_1}} \{R_1(p_1)\} \qquad \text{(basis)}$$

$$f_k(x_k) = \max_{\substack{feasible \\ proposal\_p_k}} \{R_k(p_k) + f_{k-1}(x_{k-1})\} \qquad \text{(rekurens)}$$

$$k = 2, 3$$

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

$$f_1(x_1) = \max_{c_1(p_1) \le x_1} \{R_1(p_1)\} \qquad \text{(basis)}$$

$$f_k(x_k) = \max_{c_k(p_k) \le x_k} \{R_k(p_k) + f_{k\text{-}1}[x_k - c_k(p_k)]\} \qquad \text{(rekurens)}$$

$$k = 2, 3$$

Tahap 1

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

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

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

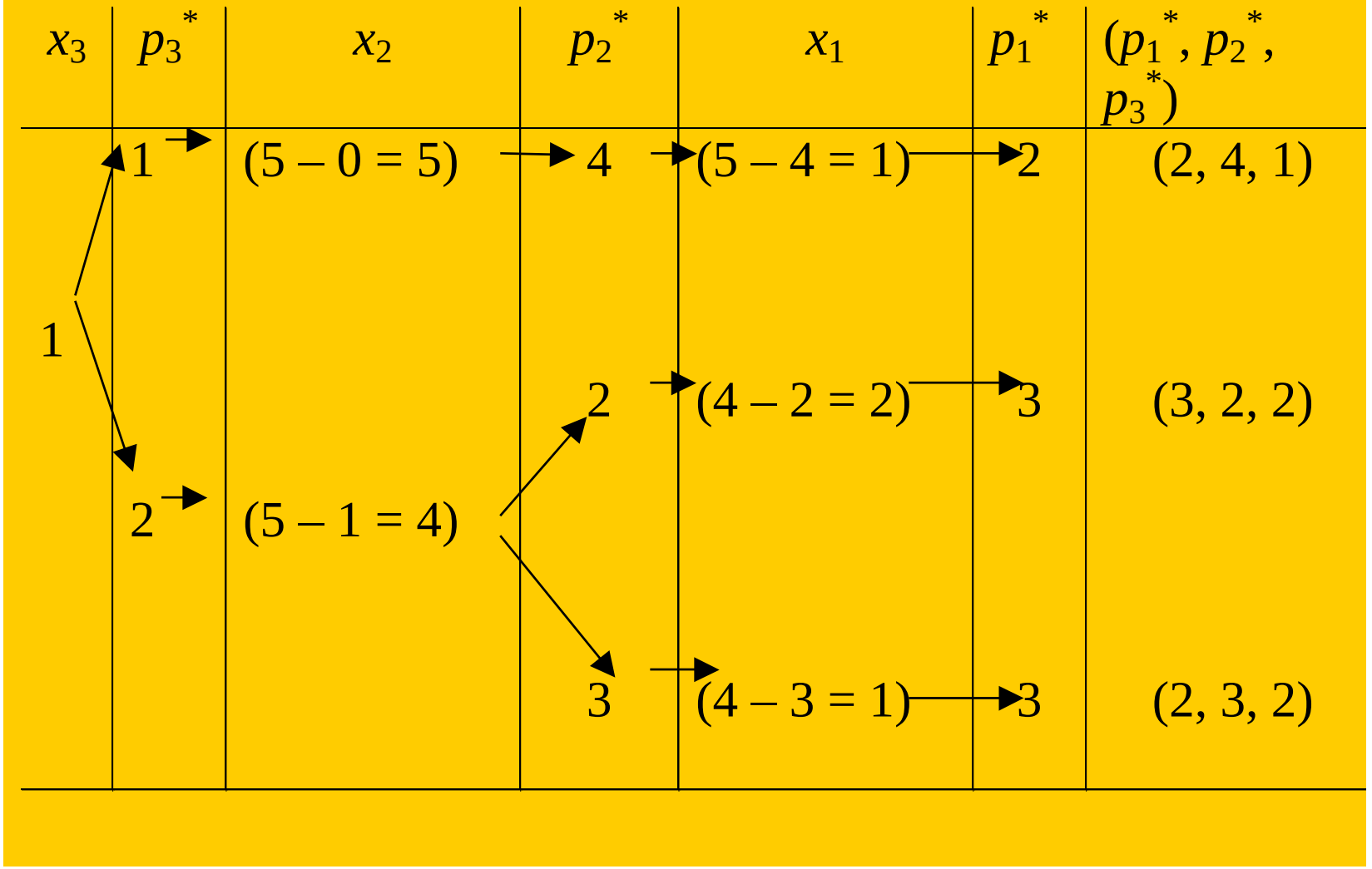
x_2	$R_2(p_2) + f_1[(x_2 - c_2(p_2)]$				Solusi Optimal	
	$p_2 = 1$	$p_2 = 2$	$p_2 = 3$	$p_2 = 4$	$f_2(x_2)$	p_2^*
0	0 + 0 = **0**	-	-	-	0	1
1	0 + 5 = **5**	-	-	-	5	1
2	0 + 6 = 6	8 + 0 = **8**	-	-	8	2
3	0 + 6 = 6	8 + 5 = **13**	9 + 0 = 9	-	13	2
4	0 + 6 = 6	8 + 6 = **14**	9 + 5 = **14**	12 + 0 = 12	14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

Tahap 3

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

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 = **17**	3 + 14 = **17**	17	1 atau 2

Rekonstruksi solusi:



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 $f_{k\text{-}1}(y - w_k)$).

- Selanjutnya, kita bandingkan nilai keuntungan dari objek pada tahap k (yaitu pk) 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

$f_0(y) = 0, \quad y = 0, 1, 2, \ldots, M$ (basis)

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

$f_k(y) = \max\{f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}$, (rekurens)

$k = 1, 2, \ldots, n$

- $f_k(y)$ 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 *knapscak*) 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	$f_0(y)$	$65 + f_0(y-2)$	Solusi Optimum	
			$f_1(y)$	(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	$f_1(y)$	$80 + f_1(y-3)$	Solusi Optimum $f_2(y)$	Solusi Optimum (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	$80 + 0 =$ **80**	80	(0, 1, 0)
4	65	$80 + 0 =$ **80**	80	(0, 1, 0)
5	65	$80 + 65 =$ **145**	145	(1, 1, 0)

Tahap 3:

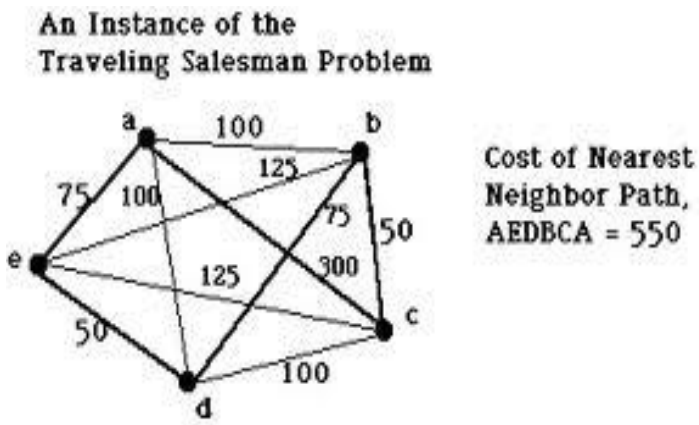
$$f_3(y) = \max\{f_2(y),\ p_3 + f_2(y - w_3)\}$$
$$= \max\{f_2(y),\ 30 + f_2(y - 1)\}$$

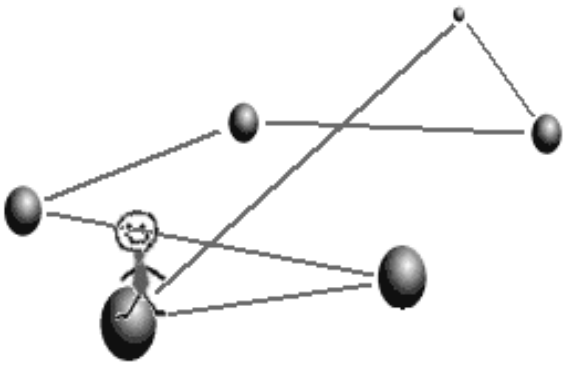
y			Solusi Optimum	
	$f_2(y)$	$30 + f_2(y-1)$	$f_3(y)$	(x_1^*, x_2^*, x_3^*)
0	**0**	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
1	**0**	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
2	**65**	30 + 0 = 30	65	(1, 0, 0)
3	80	30 + 65 = **95**	95	(1, 0, 1)
4	80	30 + 80 = **110**	110	(0, 1, 1)
5	**145**	30 + 80 = 110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

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

Travelling Salesperson Problem (TSP)

- Diberikan sejumlah kota dan diketahui jarak antar kota. Tentukan tur terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.





- Misalkan $G = (V, E)$ adalah graf lengkap berarah dengan sisi-sisi yang diberi harga $c_{ij} > 0$.

- Misalkan $|V| = n$ dan $n > 1$. Setiap simpul diberi nomor 1, 2, …, n.

- Asumsikan perjalanan (tur) dimulai dan berakhir pada simpul 1.

- Setiap tur pasti terdiri dari sisi (1, k) untuk beberapa $k \in V - \{1\}$ dan sebuah lintasan dari simpul k ke simpul 1.

- Lintasan dari simpul k ke simpul 1 tersebut melalui setiap simpul di dalam $V - \{1, k\}$ tepat hanya sekali.

- Prinsip Optimalitas: jika tur tersebut optimal maka lintasan dari simpul k ke simpul 1 juga menjadi lintasan k ke 1 **terpendek** yang melalui simpul-simpul di dalam $V - \{1, k\}$.

- Misalkan $f(i, S)$ adalah bobot lintasan terpendek yang berawal pada simpul i, yang melalui semua simpul di dalam S dan berakhir pada simpul 1.

- Nilai $f(1, V - \{1\})$ adalah bobot tur terpendek.

Hubungan rekursif:

$$f(1, V - \{1\}) = \min_{2 \le k \le n} \{c_{1k} + f(k, V - \{1, k\})\} \qquad (1)$$

Dengan merampatkan persamaan (1), diperoleh

$$f(i, \varnothing) = c_{i,1} \quad , \quad 2 \le i \le n \qquad \text{(basis)}$$

$$f(i, S) = \min_{j \in S} \{c_{ij} + f(j, S - \{j\})\} \quad \text{(rekurens)} \quad (2)$$

- Gunakan persamaan (2) untuk memperoleh $f(i, S)$ untuk $|S| = 1$, $f(i, S)$ untuk $|S| = 2$, dan seterusnya sampai untuk $|S| = n - 1$.

Tinjau persoalan TSP untuk $n = 4$:

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

Tahap 1: $f(i, \varnothing) = c_{i,1}$, $2 \le i \le n$

Diperoleh:

$f(2, \varnothing) = c_{21} = 5;$
$f(3, \varnothing) = c_{31} = 6;$
$f(4, \varnothing) = c_{41} = 8;$

Tahap 2:

$$f(i,S) = \min_{j \in S}\{c_{ij} + f(j, S - \{j\})\} \quad \text{untuk} \; |S| = 1$$

Diperoleh:

$f(2, \{3\}) = \min\{c_{23} + f(3, \varnothing)\} = \min\{9 + 6\} = \min\{15\} = 15$

$f(2, \{4\}) = \min\{c_{24} + f(4, \varnothing)\} = \min\{10 + 8\} = \min\{18\} = 18$

$f(3, \{2\}) = \min\{c_{32} + f(2, \varnothing)\} = \min\{13 + 5\} = \min\{18\} = 18$

$f(3, \{4\}) = \min\{c_{34} + f(4, \varnothing)\} = \min\{12 + 8\} = \min\{20\} = 20$

$f(4, \{2\}) = \min\{c_{42} + f(2, \varnothing)\} = \min\{8 + 5\} = \min\{13\} = 13$

$f(4, \{3\}) = \min\{c_{43} + f(3, \varnothing)\} = \min\{9 + 6\} = \min\{15\} = 15$

Tahap 3:

$$f(i,S) = \min_{j \in S}\{c_{ij} + f(j, S - \{j\})\}$$

untuk $|S| = 2$ dan $i \neq 1$, $1 \notin S$ dan $i \notin S$.

Diperoleh:

$f(2, \{3, 4\}) = \min\{c_{23} + f(3, \{4\}), c_{24} + f(4, \{3\})\}$
$= \min\{9 + 20, 10 + 15\}$
$= \min\{29, 25\} \quad = 25$

$f(3, \{2, 4\}) = \min\{c_{32} + f(2, \{4\}), c_{34} + f(4, \{2\})\}$
$= \min\{13 + 18, 12 + 13\}$
$= \min\{31, 25\} \quad = 25$

$f(4, \{2, 3\}) = \min\{c_{42} + f(2, \{3\}), c_{43} + f(3, \{2\})\}$
$= \min\{8 + 15, 9 + 18\}$
$= \min\{23, 27\} = 23$

Dengan menggunakan persamaan (1) diperoleh:

$$f(1, \{2, 3, 4\}) = \min\{c_{12} + f(2, \{3, 4\}), c_{13} + f(3, \{2, 4\}), c_{14} + f(4, \{2, 3\})\}$$
$$= \min\{10 + 25, 15 + 25, 20 + 23\}$$
$$= \min\{35, 40, 43\} = 35$$

Jadi, bobot tur yang berawal dan berakhir di simpul 1 adalah 35.

Menentukan lintasan yang dilalui

- Tinjau pada setiap $f(i, S)$ nilai j yang meminimumkan persamaan (2)
- Misalkan $J(i, S)$ adalah nilai yang dimaksudkan tersebut. Maka, $J(1, \{2, 3, 4\}) = 2$. Jadi, tur mulai dari simpul 1 selanjutnya ke simpul 2.
- Simpul berikutnya dapat diperoleh dari $f(2, \{3, 4\})$, yang mana $J(2, \{3, 4\}) = 4$. Jadi, simpul berikutnya adalah simpul 4.
- Simpul terakhir dapat diperoleh dari $f(4, \{3\})$, yang mana $J(4, \{3\}) = 3$. Jadi, tur yang optimal adalah 1, 2, 4, 3, 1 dengan bobot (panjang) = 35.