

Program Dinamis (Dynamic Programming)



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.

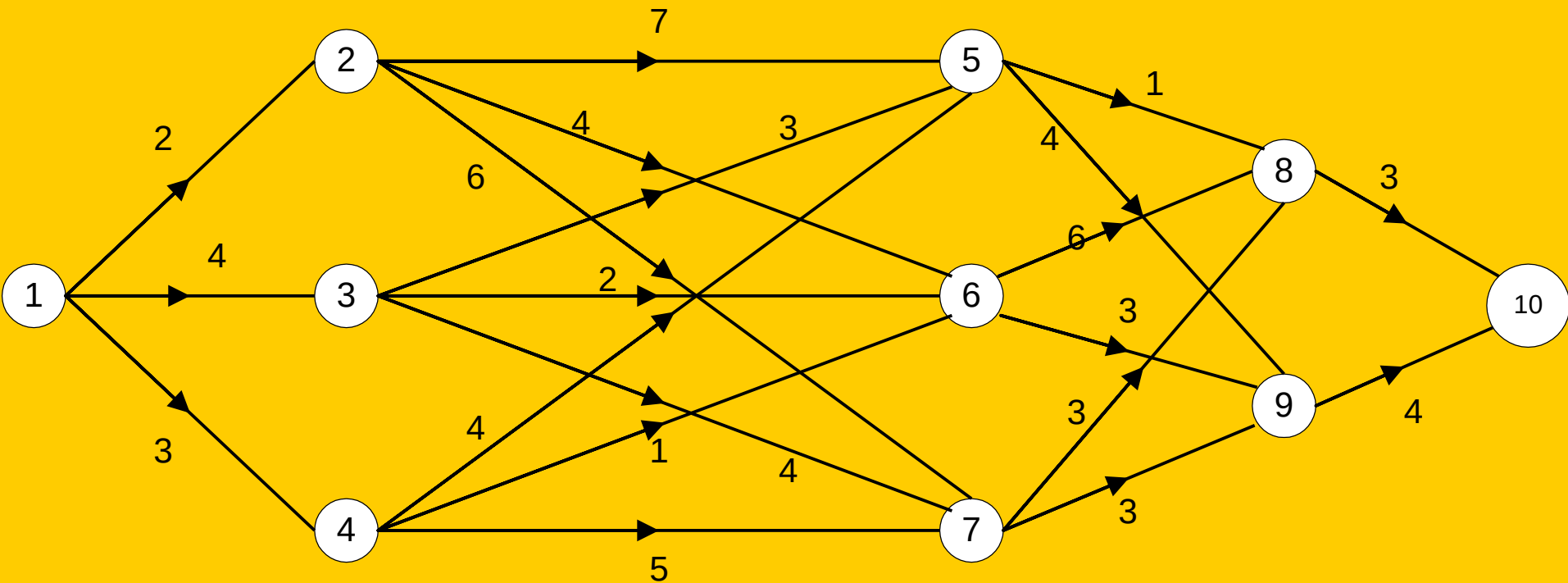


Pada penyelesaian persoalan dengan metode ini:

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.

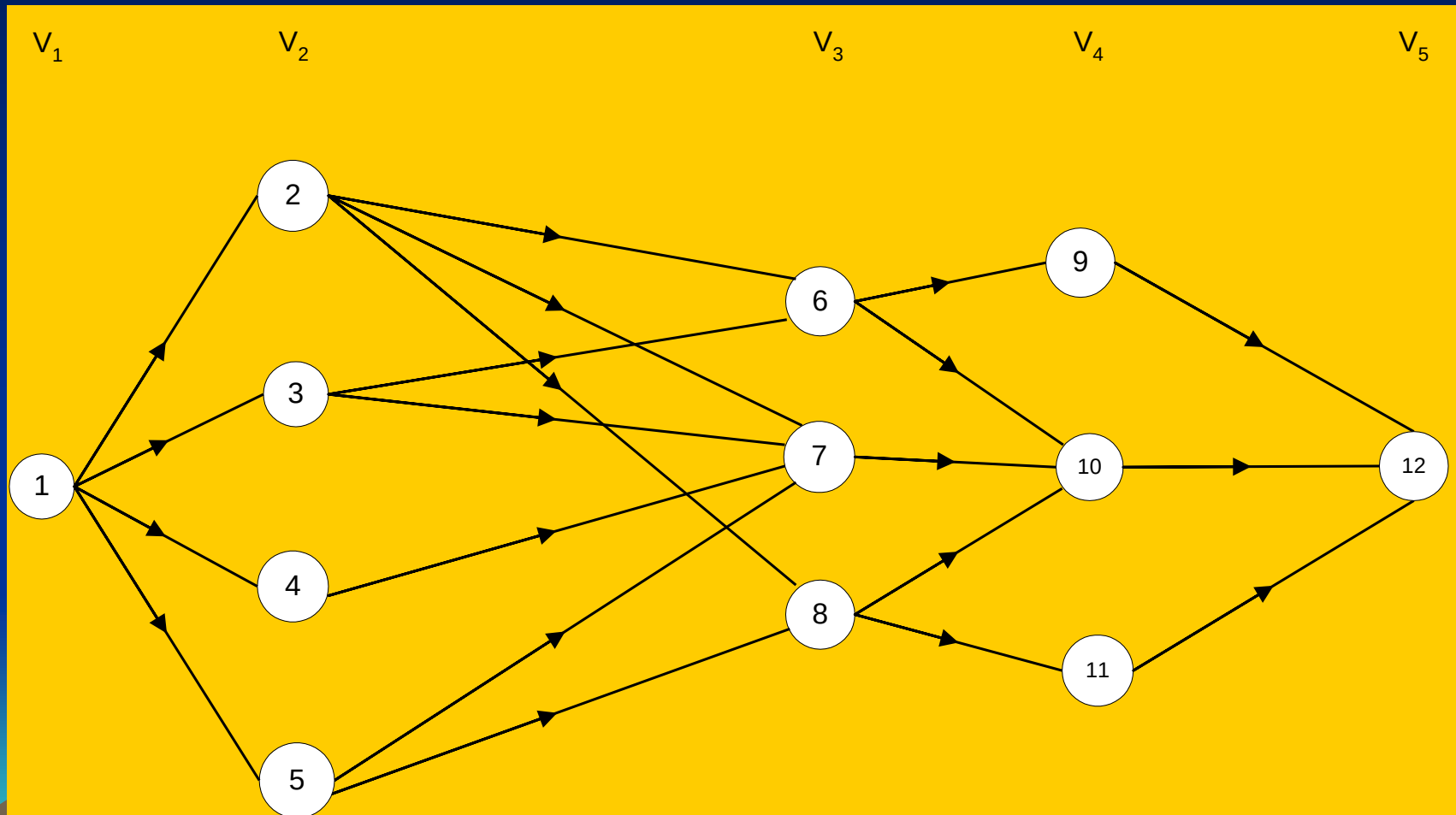


Karakteristik Persoalan Program Dinamis

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



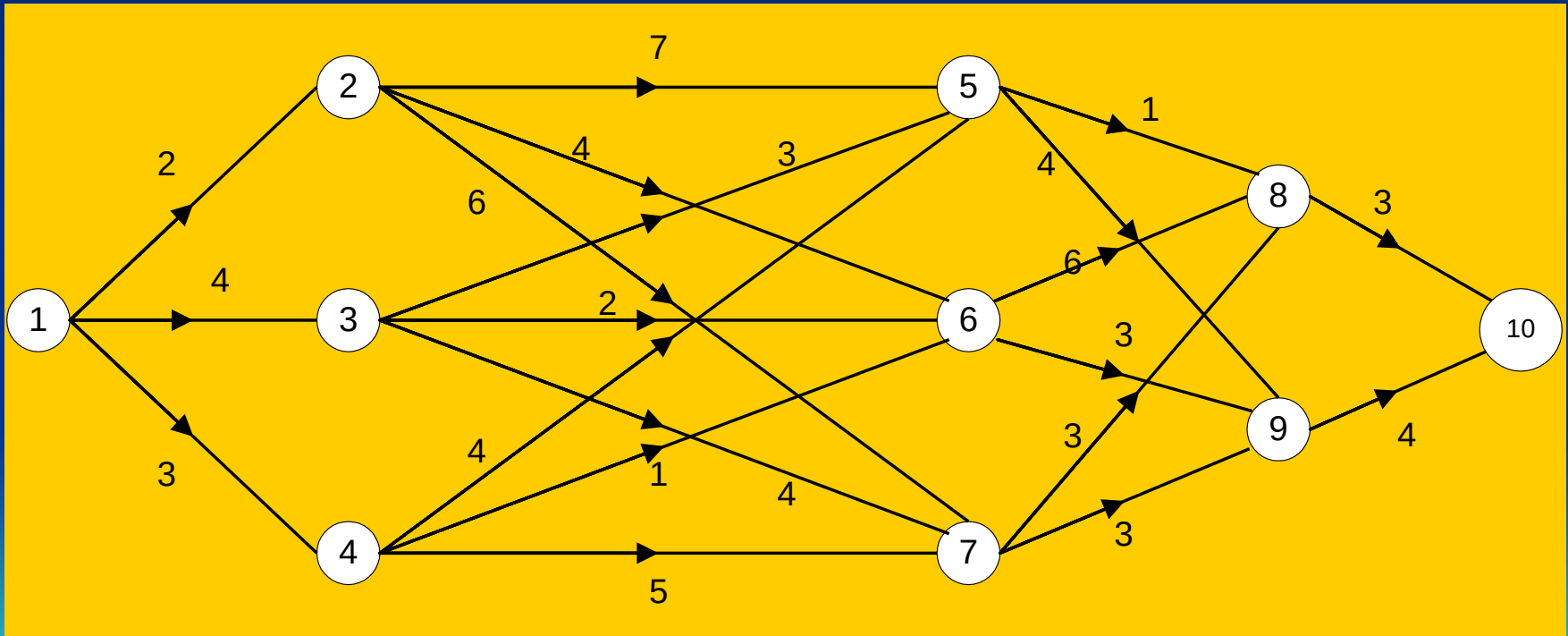
Langkah-langkah Pengembangan Algoritma Program Dinamis

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:

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

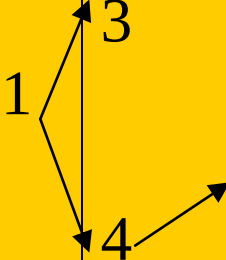
$s \backslash x_2$	$f_2(s, x_2) = c_{s,x_2} + f_3(x_2)$			Solusi Optimum	
	5	6	7	$f_2(s)$	x_2^*
2	11	11	12	11	5 atau 6
3	7	9	10	7	5
4	8	8	11	8	5 atau 6

Tahap 1:

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

$\begin{matrix} x_1 \\ s \end{matrix}$	$f_1(s, x_1) = c_{s,x_1} + f_2(x_1)$			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
	3	5	8	10	11
	4	5	8	10	11
	4	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.