

OPERATIONAL RESEARCH II

Agustina Eunike, ST., MT., MBA.
Industrial Engineering – University of Brawijaya



DYNAMIC PROGRAMMING

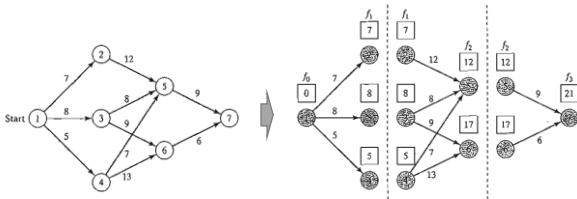


- Pemrograman Dinamis (dynamic programming / DP)
 - Prosedur matematis yang dirancang untuk memperbaiki efisiensi perhitungan masalah pemrograman matematis tertentu dengan menguraikannya menjadi bagian-bagian masalah yang lebih kecil.
 - Penyederhanaan perhitungan.
 - Menjawab masalah dalam **tahap – tahap (stages)**, dengan setiap tahap meliputi satu variabel optimasi.
 - Perhitungan di tahap yang berbeda dihubungkan melalui **perhitungan rekursif**, untuk menjamin bahwa pemecahan layak untuk tiap tahap juga layak untuk keseluruhan masalah.

DYNAMIC PROGRAMMING



- Contoh DP pada *Shortest Route*:



DYNAMIC PROGRAMMING



- Contoh DP pada *Shortest Route*:
 - Pemecahan menjadi tahap-tahap (*stages*):
 - Tahap 1 (*stage 1*)
 - Shortest distance from node 1 to node 2 = 7 miles (*from node 1*)
 - Shortest distance from node 1 to node 3 = 8 miles (*from node 1*)
 - Shortest distance from node 1 to node 4 = 5 miles (*from node 1*)

DYNAMIC PROGRAMMING



- Contoh DP pada *Shortest Route*:
 - Pemecahan menjadi tahap-tahap (*stages*):
 - Tahap 2 (*stage 2*)

$$\begin{aligned} \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node 5} \end{array} \right) &= \min_{i=2,3,4} \left\{ \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node } i \end{array} \right) + \left(\begin{array}{l} \text{Distance from} \\ \text{node } i \text{ to node 5} \end{array} \right) \right\} \\ &= \min \left\{ \begin{array}{l} 7 + 12 = 19 \\ 8 + 8 = 16 \\ 5 + 7 = 12 \end{array} \right\} = 12 \text{ (from node 4)} \end{aligned}$$

$$\begin{aligned} \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node 6} \end{array} \right) &= \min_{i=3,4} \left\{ \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node } i \end{array} \right) + \left(\begin{array}{l} \text{Distance from} \\ \text{node } i \text{ to node 6} \end{array} \right) \right\} \\ &= \min \left\{ \begin{array}{l} 8 + 9 = 17 \\ 5 + 13 = 18 \end{array} \right\} = 17 \text{ (from node 3)} \end{aligned}$$

DYNAMIC PROGRAMMING



- Contoh DP pada *Shortest Route*:
 - Pemecahan menjadi tahap-tahap (*stages*):
 - Tahap 3 (*stage 3*)

$$\begin{aligned} \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node 7} \end{array} \right) &= \min_{i=5,6} \left\{ \left(\begin{array}{l} \text{Shortest distance} \\ \text{to node } i \end{array} \right) + \left(\begin{array}{l} \text{Distance from} \\ \text{node } i \text{ to node 7} \end{array} \right) \right\} \\ &= \min \left\{ \begin{array}{l} 12 + 9 = 21 \\ 17 + 6 = 23 \end{array} \right\} = 21 \text{ (from node 5)} \end{aligned}$$

DYNAMIC PROGRAMMING



- Prinsip dasar pada perhitungan DP:
 - Masalah dapat dibagi menjadi **tahap-tahap (stages)**, dengan **keputusan kebijakan** yang dibuat pada tiap tahap.
 - Tiap tahap mempunyai **state** yang berhubungan dengan kondisi awal tahap. *State* merupakan informasi yang diperlukan pada tahapan (*stage*) untuk membuat keputusan optimal.
 - Efek keputusan kebijakan pada setiap tahap menggambarkan *perubahan state saat ini menjadi state lain pada awal tahap berikutnya*.
 - Prosedur penyelesaian dirancang untuk menemukan **kebijakan optimal** dari *keseluruhan masalah*, yang menunjukkan keputusan kebijakan mana yang optimal pada setiap tahap untuk setiap *state* yang mungkin.
 - Pada *state* saat ini, *kebijakan optimal untuk langkah selanjutnya independen* terhadap pilihan *states* (keadaan-keadaan) sebelumnya. Hal di atas disebut sebagai **principle of optimality**.

DYNAMIC PROGRAMMING



- Rumus Rekursif Maju (Forward Recursive):**

$$f_i^*(s_i) = \min_{\text{all feasible } (x_1, \dots, x_n)} \{c_{x_{i-1}, x_i} + f_{i-1}(x_{i-1})\},$$

$$i = 1, 2, \dots, n$$

- Rumus Rekursif Mundur (Backward Recursive):**

$$f_i^*(s_i) = \min_{\text{all feasible } (x_1, \dots, x_n)} \{c_{x_i, x_{i+1}} + f_{i+1}(x_{i+1})\},$$

$$i = 1, 2, \dots, n$$

DYNAMIC PROGRAMMING



– Keterangan:

- n = jumlah *stages*
- i = label untuk *stage* saat ini ($i = 1, 2, 3, \dots, n$)
- s_i = *state* saat ini pada *stage* i
- x_i = variabel keputusan pada *stage* i
- $f_i^*(s_i)$ = kontribusi *stage* 1, 2, ..., $i-1$ (untuk forward), atau $i, i+1, \dots, n$ (untuk backward).

DYNAMIC PROGRAMMING

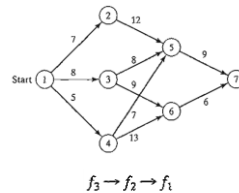


- Contoh *Backward Recursive* pada Shortest Route (di atas):

– Stage 3:

$$f_4(x_4) = 0 \text{ for } x_4 = 7$$

x_3	$d(x_3, x_4)$		Optimum solution	
	$x_4 = 7$		$f_3(x_3)$	x_4^*
5	9		9	7
6	6		6	7



DYNAMIC PROGRAMMING



- Contoh *Backward Recursive* pada Shortest Route (di atas):

– Stage 2:

x_2	$d(x_2, x_3) + f_3(x_3)$		Optimum solution	
	$x_3 = 5$	$x_3 = 6$	$f_2(x_2)$	x_3^*
2	$12 + 9 = 21$	—	21	5
3	$8 + 9 = 17$	$9 + 6 = 15$	15	6
4	$7 + 9 = 16$	$13 + 6 = 19$	16	5

DYNAMIC PROGRAMMING

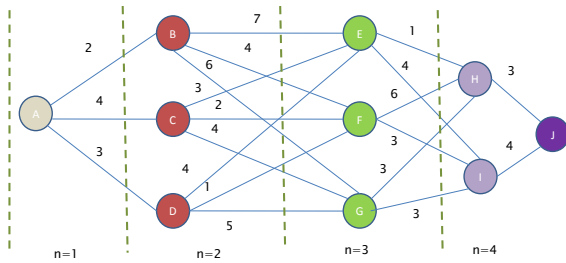


- Contoh *Backward Recursive* pada Shortest Route (di atas):

– Stage 1:

x_1	$d(x_1, x_2) + f_2(x_2)$			Optimum solution	
	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$f_1(x_1)$	x_2^*
1	$7 + 21 = 28$	$8 + 15 = 23$	$5 + 16 = 21$	21	4

Contoh 1 : Rute Terpendek



Alternatif keputusan yang Dapat diambil pada Setiap Tahap

Tahap 1

	B	C	D
A	2	4	3

Tahap 3

	H	I
E	1	4
F	6	3
G	3	3

Tahap 2

	E	F	G
B	7	4	6
C	3	2	4
D	4	1	5

Tahap 4

	J
H	3
I	4

Tahap(n), State(S_n), X_n(Alternatif Keputusan)

Pada tahap ke-2 (n = 2)

- o Terdapat tiga state(S₂) yaitu B, C dan D, orang yang melakukan perjalanan tersebut mungkin sedang berada di kota B, C ataupun D.
- o ketika sedang berada di kota B(S₂= B) orang tersebut bisa memilih untuk meneruskan perjalanan melalui E atau F atau G.
- o E, F dan G merupakan X₃ → alternatif keputusan yang dapat diambil ketika berada pada state tertentu

Tahap 2

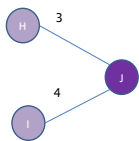
	E	F	G
B	7	4	6
C	3	2	4
D	4	1	5

Pemecahan Masalah

- Pendekatan jangka pendek yaitu memilih rute(jalur) dengan biaya termurah pada setiap tahap tidak menjamin diperolehnya biaya yang minimal secara keseluruhan.
 - A-B-F-I-J → Biaya total 13
 - Dengan programa dinamis bisa diperoleh solusi optimal dengan biaya total = 11
- Pencarian solusi optimal harus dilakukan dengan mempertimbangkan seluruh biaya.

n=4 → tahap terakhir

- Ketika berada di kota H atau S₄=H, hanya terdapat satu pilihan yaitu menuju ke kota tujuan akhir yaitu kota J (X₄=J).
- Biaya menuju ke kota J adalah 3.



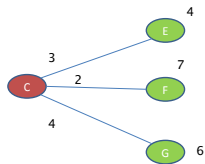
s	f ₄ [*] (S)	x ₄ [*]
H	3	J
I	4	J

n=3

- Ketika berada di kota E atau S₃=E, terdapat dua pilihan (alternatif keputusan) yaitu menuju ke kota H atau ke kota I → X₃= H atau X₃= I.
 - Jika memilih menuju ke kota H biayanya adalah 4.
 - Jika memilih menuju ke kota I biayanya adalah 8.
 - Biaya ini dihitung pada tahap sebelumnya n=4.
- Sehingga jika sedang berada pada kota E maka yang harus dipilih adalah menuju ke kota H

f ₃ (s, x ₃) = c _{s,x₃} + f ₄ [*] (x ₃)		f ₃ [*] (s)		x ₃ [*]
	H	I		
E	4	8	4	H
F	9	7	7	I
G	6	7	6	H

n=2

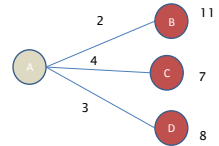


$$f_4(s, x_2) = c_{sx_2} + f_3^*(x_2)$$

	E	F	G	$f_4^*(s)$	x_2^*
B	11	11	12	11	E atau F
C	7	9	10	7	E
D	8	8	11	8	E atau F

n=1

Kota B merupakan kota yang terdekat dengan kota tujuan awal (kota A), akan tetapi memilih kota yang terdekat TIDAK selalu menghasilkan solusi yang terbaik jika tidak memperhatikan masalah secara keseluruhan.
 → Pada kasus ini memilih kota yang terdekat justru memberikan solusi yang paling buruk.



$$f_1(s, x_1) = c_{sx_1} + f_2^*(x_1)$$

	B	C	D	$f_1^*(s)$	x_1^*
A	13	11	11	11	C atau D

Penentuan Solusi

- Pada tahap 1 diperoleh hasil bahwa kalau dari kota A harus menuju ke kota C atau D dan jangan ke kota B
- Setelah sampai di kota C dan D kemudian kemana?
- Berdasarkan hasil tahap 2 kalau dari kota C maka harus ke kota E dan kalau dari kota D maka harus ke kota E atau F.

Tahap 1

$$f_1(s, x_1) = c_{sx_1} + f_2^*(x_1)$$

	B	C	D	$f_1^*(s)$	x_1^*
A	13	11	11	11	C atau D

Tahap 2

$$f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$$

	E	F	G	$f_2^*(s)$	x_2^*
B	11	11	12	11	E atau F
C	7	9	10	7	E
D	8	8	11	8	E atau F

Penentuan Solusi

Tahap 2

$$f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$$

	E	F	G	$f_2^*(s)$	x_2^*
B	11	11	12	11	E atau F
C	7	9	10	7	E
D	8	8	11	8	E atau F

Berdasarkan hasil tahap 3:

- dari kota E harus dilanjutkan menuju kota H dan
- dari kota F harus menuju kota I.

Tahap 3

$$f_3(s, x_3) = c_{sx_3} + f_4^*(x_3)$$

	H	I	J	$f_3^*(s)$	x_3^*
E	8	4	4	4	H
F	9	7	7	7	I
G	6	7	6	6	H

Penentuan Solusi

Berdasarkan hasil tahap 3:

- jika sudah sampai di kota E maka perjalanan harus dilanjutkan menuju ke kota H.
- jika sudah sampai di kota F maka perjalanan harus dilanjutkan menuju ke kota I.

$$f_3^*(s)$$

S	$f_3^*(S)$	x_3^*
H	3	J
I	4	J

$$f_4(s, x_3) = c_{sx_3} + f_5^*(x_3)$$

	H	I	J	$f_4^*(s)$	x_3^*
E	4	4	4	4	H
F	9	7	7	7	I
G	6	7	6	6	H

Tahap 3

Penentuan Solusi

Berdasarkan hasil tahap 4:

- jika sudah sampai di kota H maka perjalanan harus dilanjutkan menuju ke kota J.
- jika sudah sampai di kota I maka perjalanan harus dilanjutkan menuju ke kota J.

Tahap 4

$$f_5^*(s)$$

S	$f_5^*(S)$	x_5^*
H	3	J
I	4	J

Solusi

Tahap 1

$f_1(s, x_1) = c_{11}x_1 + f_2^*(s_1)$				
B	C	D	$f_1^*(s)$	x_1
A	13	11	11	11

Tahap 2

$f_2(s, x_2) = c_{21}x_2 + f_3^*(s_2)$					
E	F	G	$f_2^*(s)$	x_2	
B	11	11	12	11	E atau F
C	7	9	10	7	E
D	8	8	11	8	E atau F

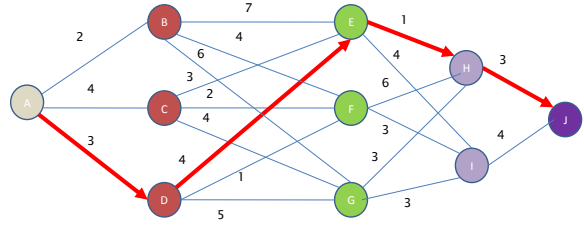
Tahap 3

$f_3(s, x_3) = c_{31}x_3 + f_4^*(s_3)$					
H	I	J	$f_3^*(s)$	x_3	
E	8	4	8	4	H
F	9	7	7	7	H
G	6	7	6	6	H

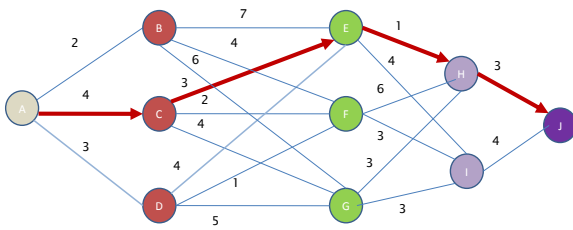
Tahap 4

$f_4^*(S)$		
S	$f_4^*(S)$	x_4
H	3	J
I	4	J

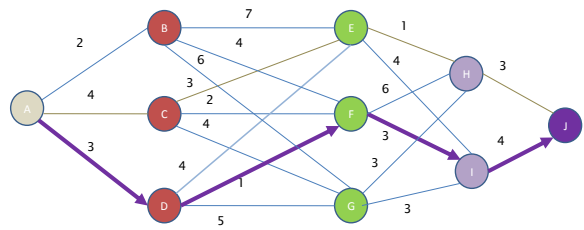
Solusi 1



Solusi 2



Solusi 3



Soal Latihan 1

- Sebuah perusahaan menghasilkan dua macam barang. Waktu kerja yang tersedia adalah 430 menit.
 - Barang 1 memerlukan waktu 2 menit per unit dan barang 2 memerlukan waktu 1 menit per unit.
 - Tidak terdapat pembatasan jumlah barang 1 yang dapat diproduksi akan tetapi untuk barang 2 terbatas sampai 230 unit.
 - Keuntungan barang 1 adalah Rp. 20000 per unit dan barang 2 adalah Rp. 50000 per unit.
- Carilah penyelesaian optimalnya dengan menggunakan Program dinamis.

Soal Latihan 2

Sebuah perusahaan ingin menentukan kebijaksanaan penggantian yang optimal dari peralatan yang saat ini berumur 3 tahun untuk keperluan pemakaian 4 tahun yang akan datang (n = 4) yaitu sampai awal tahun ke 5.

Tabel berikut adalah data dari persoalan. Perusahaan menentukan peralatan yang berumur 6 tahun harus diganti. Harga peralatan baru adalah \$ 100.000. Selesaikan permasalahan ini dengan program dinamis !

Umur, t (tahun)	Pendapatan, r(t) (\$)	Biaya operasi, c(t) (\$)	Nilai sisa, s(t) (\$)
0	20.000	200	-
1	19.000	600	80.000
2	18.500	1200	60.000
3	17.200	1500	50.000
4	15.500	1700	30.000
5	14.000	1800	10.000
6	12.200	2200	5.000

Soal Latihan 3

Sebuah kontraktor memperkirakan kebutuhan tenaga kerja yang diperlukan untuk 5 minggu yang akan datang berturut-turut sebanyak 5, 7, 8, 4, dan 6 orang.

Kelebihan tenaga kerja memberikan biaya sebesar \$ 300 per orang per minggu sedangkan untuk menambah tenaga kerja diperlukan biaya tetap sebesar \$ 400 dan biaya variabel sebesar \$ 200 per orang per minggu.

Selesaikan permasalahan ini dengan programa dinamis !

Soal Latihan 4

Kapal dengan kapasitas angkut 4 ton akan dimuati 1 unit atau lebih dari tiga macam barang. Pada tabel di bawah ini diberikan berat per unit (w ton) dan nilai per unit (r ribu \$) dari barang i . Berapa jumlah setiap barang akan dimuat ke dalam kapal yang memberikan pendapatan total yang maksimal?

Barang i	w_i	r_i
1	2	31
2	3	47
3	1	14

Soal Latihan 5

PT. Y memproduksi transformator besar untuk keperluan jaringan listrik. Permintaan yang harus dipenuhi untuk bulan-bulan mendatang pada tabel berikut :

Bulan	Mei	Juni	Juli	Agustus
Jumlah Permintaan	30	40	20	30

Transformator harus dikirim pada akhir bulan. Misalnya permintaan 30 unit transformator untuk bulan Mei harus dikirim pada akhir bulan Mei. Biaya produksi transformator tergantung pada jumlah unit yang diproduksi seperti digambarkan pada tabel

Jumlah Produksi (unit)	0	10	20	30	40
Biaya Produksi (\$ 000)	0	7	9	10	11

Kapasitas gudang untuk menyimpan transformator adalah 30 unit. Transformator yang tidak dikirim dalam bulan yang sama dapat disimpan dalam gudang dengan biaya \$ 100 per unit per bulan. Persediaan pada awal bulan Mei adalah 20 unit dan persediaan pada awal bulan September ditetapkan sebesar 0 unit. Untuk praktisnya transformator hanya dapat diproduksi, dikirim dan disimpan dalam jumlah kelipatan 10 unit.

Soal Latihan 6

- Seseorang akan melakukan perjalanan dengan kereta kuda dari kota 1 menuju kota 10 melalui beberapa kota dari kota 2 sampai kota 9, melewati daerah berbahaya seperti pada gambar 1. Dia dapat memilih beberapa rute, misalnya melalui kota 1-2-5-8-10 atau 1-3-7-9-10. Setiap penggal jalan mempunyai bahaya yang diukur sebagai besar asuransi jiwa $r(x,d)$ dari kota x ke kota d . Bahaya dari rute tertentu dari kota 1 ke kota 10 diukur dengan besarnya asuransi yang dibayar sepanjang rute tersebut. Biaya asuransi yang harus dibayar dicantumkan pada penggal garis pada gambar 1.

Soal Latihan 7

- PT. X memutuskan untuk menambah enam orang tenaga penjualan pada bagian penjualannya. Tenaga kerja yang baru akan ditempatkan pada keempat daerah pemasarannya. Tabel berikut menunjukkan kenaikan penjualan yang diharapkan (dalam \$ 000) dari setiap daerah pemasaran, tergantung dari jumlah tenaga kerja yang ditugaskan pada daerah tersebut. Tentukan alokasi optimal dari keenam tenaga penjualan baru di keempat daerah pemasaran tersebut.

Daerah Pemasaran	1	2	3	4	5	6
1	80	120	140	140	140	140
2	40	80	120	160	180	200
3	120	160	160	160	160	160
4	40	60	80	80	80	80

Kesimpulan

- Model programa dinamis menggambarkan proses pengambilan keputusan proses bertahap.
- Masalah programa dinamis menghasilkan model matematis yang berbeda antara kasus yang satu dengan yang lain.
- Untuk memformulasikan masalah programa dinamis harus mengerti dengan baik mengenai konsep dari status, tahap, alternatif keputusan, keputusan yang optimal, dan alasan penggunaan fungsi rekursi.



Review Question

- Kapankah metode program dinamis bisa digunakan untuk menentukan solusi suatu permasalahan?
- Apakah yang dimaksud dengan:
 - State
 - Tahap
 - Variabel Keputusan
 - Fungsi Rekursif
- Kenapa dalam metode program dinamis tidak mempunyai struktur model matematis yang pasti sehingga untuk masalah yang beda akan menghasilkan struktur model yang berbeda juga?

DETERMINISTIC DYNAMIC PROGRAMMING

DETERMINISTIC DP

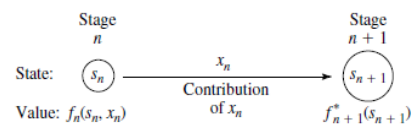


- *State* pada *stage* berikutnya ditentukan oleh *state* dan kebijakan keputusan pada *stage* saat ini.
- Pengkategorian:
 - Fungsi tujuan
 - Minimasi atau Maksimasi
 - *Set of states*
 - Variabel Diskrit
 - Variabel Kontinyu
 - Vektor

DETERMINISTIC DP



- Struktur dasar:



PROBABILISTIC DP



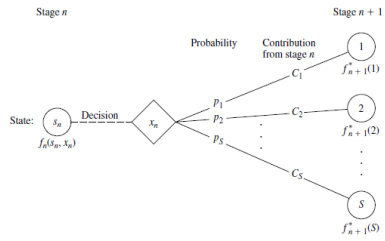
- *State* pada *stage* berikutnya mengikuti suatu distribusi peluang.
- Distribusi peluang ditentukan oleh *stage* saat ini.
- Untuk permasalahan yang sederhana dapat digambarkan menggunakan *decision tree*
- Lebih kompleks dibandingkan deterministic DP.

PROBABILISTIC DYNAMIC PROGRAMMING

PROBABILISTIC DP



- Struktur dasar:



References

- Frederick Hillier and Gerald J. Lieberman, Introduction to Operations Research, Holden Day Ltd, San Francisco, 1997.
- Taha, Hamdy, Operation Research : An Introduction, Macmillan Publishing Company, New York, 1997.