

# Penyelesaian Masalah Capital Rationing Prasarana Jalan dengan Dynamic Programming

Azwar Tamim

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Jl. Ganeca 10 Bandung  
e-mail: if16109@students.if.itb.ac.id

## ABSTRAK

Capital rationing merupakan salah satu contoh permasalahan optimasi dalam dunia industri. Sebagai contoh adalah permasalahan untuk mengoptimasi keuntungan dari modal yang dimiliki. Misalnya, dalam hal pembangunan jalan. Ada berbagai metode yang dapat digunakan untuk mencari solusi yang optimal. Dalam makalah ini, penulis mencoba menyajikan penggunaan salah satu metode pemrograman matematis, yaitu dynamic programming.

Dynamic programming pertama kali digunakan oleh Richard Bellman untuk mendeskripsikan proses pemecahan masalah dimana seseorang harus mengambil setiap keputusan terbaik pada setiap langkahnya. Optimisasi dengan dynamic programming dicapai dengan memilih setiap keputusan terbaik (optimal) dalam setiap tahap. Pada persoalan dalam makalah ini, keputusan terbaik dipilih dalam setiap tahap keputusan pembangunan jalan.

**Kata kunci:** capital rationing, optimasi, dynamic programming.

## 1. PENDAHULUAN

Pengembangan metode penyelesaian masalah telah berkembang di zaman ini menggunakan berbagai metode atau algoritma. Salah satu permasalahan yang banyak dihadapi adalah permasalahan dalam bidang industri. Misalnya, dalam hal pengalokasian anggaran untuk pembangunan jalan.

Manusia berusaha sedapat mungkin mengeluarkan sedikit biaya dan mendapatkan keuntungan yang sebesar-besarnya sebagaimana yang tertera dalam prinsip ekonomi. Untuk itu, dikembangkanlah berbagai metode optimasi dalam menghadapi permasalahan-permasalahan tersebut. Salah satu metode yang cukup populer adalah dynamic programming. Kata “programming” dalam istilah

dynamic programming sebenarnya tidak ada kaitannya dengan computer programming. Kata itu berasal dari kata pemrograman matematis yang bersinonim dengan arti kata optimasi.

Misalkan akan dibangun sembilan buah jalan dengan alokasi dana sebanyak 31,2 miliar rupiah. Suatu perusahaan yang akan membangun jalan-jalan tersebut memutuskan bagaimana alokasi dana tersebut berdasarkan nilai optimal dari biaya investasi ( $c$ ) dan nilai benefit cost ratio ( $v$ ) dari pembangunan jalan yang dilakukan. Tabel berikut ini menggambarkan kedua nilai tersebut.

Jalan	Biaya Investasi (Rp)	Benefit Cost Ratio
Jalan 1	4.505.359.375	1.01874
Jalan 2	2.703.215.625	1.33241
Jalan 3	3.604.287.500	1.05584
Jalan 4	3.604.287.500	1.82289
Jalan 5	6.307.503.125	1.05382
Jalan 6	6.307.503.125	1.06233
Jalan 7	1.802.143.750	4.61721
Jalan 8	1.802.143.750	3.05994
Jalan 9	6.307.503.125	1.0306
Modal	31.500.000.000	

Gambar 1. Biaya investasi dan benefit cost ratio jalan.

## 2. METODE PENCARIAN SOLUSI

Persoalan ini mirip dengan persoalan integer knapsack karena hanya mengambil keputusan apakah jalan dibangun atau tidak (1 atau 0). Persoalan ini dicoba diselesaikan dengan metode dynamic programming maju sebagai berikut.

Terdapat sembilan tahap untuk menyelesaikan masalah ini. Pada setiap tahap dilakukan perhitungan apakah jalan tersebut dibangun atau tidak berdasarkan perbandingan nilai benefit cost ratio kumulatif dari tahap-tahap sebelumnya seperti digambarkan pada gambar berikut.

y	f0(y)	p 1 + f0(y - 4.505,359,375)	f1(y)
0	0	~	0
1,802,143,750	0	~	0
1,802,143,750	0	~	0
2,703,215,625	0	~	0
3,604,287,500	0	~	0
3,604,287,500	0	~	0
3,604,287,500	0	~	0
4,505,359,375	0	1.01874	1.01874
4,505,359,375	0	1.01874	1.01874
4,505,359,375	0	1.01874	1.01874
5,406,431,250	0	1.01874	1.01874

Gambar 2. Tahap 1 dynamic programming.

Pada tahap ini diambil keputusan apakah jalan 1 dibangun atau tidak. Keputusan didapat dengan membandingkan benefit cost ratio apabila jalan 1 dibangun  $f_0(y)$  dan tidak dibangun ( $p_1 + f_0(y - w_1)$ ). Apabila benefit cost ratio setelah jalan 1 dibangun lebih besar dibandingkan tidak dibangun, maka jalan 1 dibangun. Begitu pula dengan tahap kedua sebagai berikut.

y	f1(y)	p 2 + f1(y - 2.703,215,625)	f2(y)
0	0	~	0
1,802,143,750	0	~	0
1,802,143,750	0	~	0
2,703,215,625	0	1.33241	1.33241
3,604,287,500	0	1.33241	1.33241
3,604,287,500	0	1.33241	1.33241
3,604,287,500	0	1.33241	1.33241
4,505,359,375	1.01874	1.33241	1.33241
4,505,359,375	1.01874	1.33241	1.33241
4,505,359,375	1.01874	1.33241	1.33241
5,406,431,250	1.01874	1.33241	1.33241

Gambar 3. Tahap 2 dynamic programming.

Perhitungan ini dilakukan sampai tahap kesembilan pada gambar berikut.

y	f8(y)	p 9 + f8(y - 6.307,503,125)	f9(y)
0	0	~	0
1,802,143,750	4.61721	~	4.61721
1,802,143,750	4.61721	~	4.61721
2,703,215,625	7.67715	~	7.67715
3,604,287,500	7.67715	~	7.67715
3,604,287,500	7.67715	~	7.67715
3,604,287,500	7.67715	~	7.67715
4,505,359,375	7.67715	~	7.67715
4,505,359,375	7.67715	~	7.67715
4,505,359,375	7.67715	~	7.67715
5,406,431,250	7.67715	~	7.67715

Gambar 4. Tahap 9 dynamic programming, bagian atas.

sampai bagian akhir baris dimana biaya pembangunan jalan total tidak melebihi batas modal 31,5 miliar rupiah.

29,735,371,875	14.00444	13.98122	14.00444
29,735,371,875	14.00444	13.98122	14.00444
29,735,371,875	14.00444	13.98122	14.00444
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318
30,636,443,750	15.02318	14.99996	15.02318

Gambar 5. Tahap 9 dynamic programming, bagian bawah.

Pada setiap tahap, metode dynamic programming melakukan perhitungan benefit cost ratio kumulatif dari tahap-tahap sebelumnya agar dapat digunakan untuk mengambil keputusan apakah suatu jalan dibangun atau tidak. Benefit cost ratio yang terbaik yang terpilih menjadi acuan bagi tahap selanjutnya untuk memutuskan apakah akan membangun jalan selanjutnya atau tidak berdasarkan perbandingan benefit cost ratio yang didapat.

Setelah semua tahap diselesaikan, kita akan mendapatkan keputusan jalan mana saja yang akan dibangun. Pada permasalahan ini, diperlihatkan pada gambar di bawah keputusan mengenai jalan mana saja yang akan dibangun yang memiliki benefit cost ratio tertinggi tanpa melebihi modal yang dimiliki. (Gambar dibawah ini adalah potongan kanan dari gambar 5 di atas)

+ f8(y - 6.307,503,125)	f9(y)	[x1 x2 x3 x4 x5 x6 x7 x8 x9]
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
13.98122	14.00444	011111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110
14.99996	15.02318	111111110

Gambar 6. Tahap 9 dynamic programming, keputusan jalan yang dibangun.

Penggunaan metode dynamic programming ini, mampu menghemat waktu perhitungan karena perhitungan yang dilakukan berulang-ulang disimpan ke dalam sebuah memori untuk diakses jika hasil perhitungan tersebut dibutuhkan kemudian. Pada persoalan ini, perhitungan

nilai  $fX(y)$ , dimana  $X$  adalah 1..9 tidak perlu dilakukan berulang-ulang sebagaimana jika dilakukan dengan algoritma brute force karena hasil perhitungan ini telah disimpan di memori dari hasil perhitungan sebelumnya.

Pada hasil keputusan di atas, terlihat bahwa jalan yang dibangun adalah Jalan 1 sampai dengan Jalan 8, sedangkan Jalan 9 tidak dibangun. Hasil ini merupakan hasil optimal dengan benefit cost ratio 15.02318 dan dana yang dikeluarkan untuk membangun jalan sejumlah 30,636,443,750 rupiah yang tidak lebih dari jumlah modal yang dianggarkan, yaitu 31,5 miliar rupiah.

### 3. PERUMUSAN DAN PERBANDINGAN METODE

Pada penjabaran metode di atas, tahap adalah proses memilih jalan yang akan dibangun, status ( $y$ ) menyatakan jumlah dana yang digunakan untuk membangun jalan.

Ketika memilih membangun jalan pada tahap 1, modal sekarang adalah  $y-w_1$ . Untuk menggunakan modal sisanya kita menerapkan prinsip optimalitas dengan mengacu pada nilai optimum dari tahap sebelumnya untuk modal sisa  $y-w_0$  (yaitu  $f_0(y-w_1)$ ). Begitu juga untuk tahap-tahap selanjutnya.

Variabel  $f_1(y)$  adalah keuntungan (benefit cost ratio) optimum pada tahap 1 untuk modal sebesar  $y$ . Begitu juga untuk  $f_2(y)$ ,  $f_3(y)$ , ...,  $f_9(y)$ .  $fX(y) = 0$  adalah nilai dari persoalan knapsack kosong dengan kapasitas  $y$ .  $fX(y) = -\infty$  adalah nilai dari persoalan knapsack untuk modal negatif.

Jika dibandingkan dengan algoritma brute force, dynamic programming mampu memotong waktu komputasi dengan mengorbankan (memanfaatkan) memori agar program tidak melakukan perhitungan yang sama berulang-ulang. Apabila menggunakan algoritma brute force, maka program akan memeriksa semua kemungkinan pembangunan jalan dan mengakibatkan waktu proses berjalan lama.

### IV. KESIMPULAN

Dari penulisan makalah ini, penulis menyimpulkan beberapa hal:

1. Dynamic programming dapat digunakan untuk menyelesaikan persoalan optimasi, dalam hal ini persoalan optimasi industri.
2. Dynamic programming dapat memotong waktu komputasi yang diperlukan algoritma brute force dalam menyelesaikan permasalahan ini.
3. Dynamic programming menggunakan cukup besar ruang memori apabila status modal yang digunakan cukup besar.

4. Dynamic programming merupakan metode yang dapat menyelesaikan permasalahan dimana dalam karakteristik masalah tersebut terdapat tahap-tahap yang bergantung pada hasil-hasil dari tahap sebelumnya.

### REFERENSI

- [1] Anany Levitin, "Introduction to The Design and Analysis of Algorithms", Addison-Wesley, 2003.
- [2] Rinaldi Munir, "Diktat Kuliah IF2251 Strategi Algoritmik", Prodi Teknik Informatika, Sekolah Teknik Elektro dan Informatika ITB, 2006.