

# Implementasi Pemrograman Dinamis dalam Pencarian Solusi Permainan Menara Hanoi

Jonathan Ery Pradana / 13508007  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Bandung, Indonesia  
if18007@students.if.itb.ac.id

Makalah ini berisi tentang pemanfaatan algoritma program dinamis dalam pencarian solusi permainan menara hanoi. Algoritma Program dinamis merupakan algoritma yang digunakan untuk melakukan optimalisasi dengan membagi-bagi masalah menjadi bagian-bagian tertentu yang lebih kecil. Permainan Menara Hanoi merupakan sebuah permainan puzzle kuno yang solusinya dicapai lewat gerakan perpindahan berkali-kali dari kepingan sampai mencapai tujuan akhir yaitu memindahkan piramida di tiang kiri ke tiang kanan. Pada hasil percobaan dapat disimpulkan bahwa algoritma program dinamis memang dapat digunakan sebagai solusi algoritma untuk penyelesaian permainan Menara Hanoi, karena dalam implementasinya, permainan Menara Hanoi ini dapat dibagi menjadi beberapa langkah solusi.

**Kata kunci:** algoritma, program dinamis, langkah solusi, permainan menara hanoi.

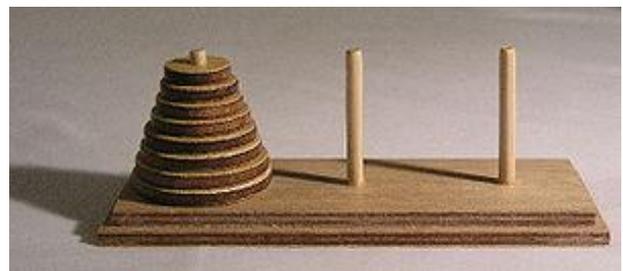
## I. PENDAHULUAN

Pemrograman dinamis (*dynamic programming*) adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian rupa sehingga solusi dari permasalahan ini dapat dipandang dari serangkaian keputusan-keputusan kecil yang saling berkaitan satu dengan yang lain. Penyelesaian persoalan dengan pemrograman dinamis ini akan menghasilkan sejumlah berhingga pilihan yang mungkin dipilih, lalu solusi pada setiap tahap-tahap yang dibangun dari solusi pada tahap sebelumnya, dan dengan metode ini kita menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Penyelesaian dengan metode program dinamis ini akan nampak sedikit mirip dengan metode *greedy*, karena metode *greedy* juga membentuk solusi yang ada secara bertahap. Pada metode *greedy* ini, kita membuat keputusan pada setiap tahap dengan cara mengambil pilihan-pilihan yang menurut kita paling menarik atau dengan kata lain paling menguntungkan untuk kita (memenuhi ukuran optimasi yang digunakan). Pengambilan keputusan yang diambil pada setiap tahap metode *greedy* hanya didasarkan pada informasi lokal,

dan pada setiap itu, memang, keputusan yang diambil tidak akan bisa dinilai benar salahnya, malahan selalu terlihat paling benar jika memenuhi optimasi yang ditentukan. Pada persoalan-persoalan tertentu, algoritma *greedy* bekerja dengan baik, namun pada kebanyakan persoalan yang lain, tidak demikian. Hal ini terjadi karena pengambilan keputusan pada setiap langkah *greedy* tidak pernah mempertimbangkan lebih jauh apakah pilihan tersebut pada langkah-langkah selanjutnya merupakan pilihan yang tepat. Dari sejumlah pilihan yang menarik, kita tidak akan pernah tahu pilihan mana yang terbaik sampai kita menuju proses yang lebih jauh kedepan.

Permainan Menara Hanoi (Tower of Hanoi), disebut juga Tower of Brahma atau Tower of Brahmas, adalah sebuah permainan matematika atau bisa disebut puzzle juga. Permainan ini terdiri dari papan yang memiliki tiga tiang dengan ada kepingan-kepingan lingkaran dengan diameter yang berbeda dan dapat berpindah ke tiang yang lain. Permainan ini dimulai dengan kepingan yang tersusun dengan susunan seperti gambar dibawah ini, dengan kepingan berdiameter terkecil berada di atas, dan membentuk piramida. Tujuan dari permainan ini adalah untuk memindahkan semua stack ke tiang yang berbeda dengan posisi yang sama seperti posisi semula.



Gambar 1 - Papan Menara Hanoi

Pada pemrograman dinamis, rangkaian keputusan optimal yang dibuat ini menggunakan prinsip optimalitas. Prinsip ini berbunyi: “Jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal.” Dengan prinsip optimalitas ini, akan dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya.

Dan dengan pemrograman dinamis ini, akan dipecahkan masalah optimasi solusi untuk permainan Menara Hanoi.

## II. LANDASAN TEORI

### A. Permainan Menara Hanoi

Permainan Permainan Menara Hanoi (Tower of Hanoi), atau disebut juga Tower of Brahma (Tower of Brahmas) merupakan sebuah permainan matematika yang kuno, dibuat oleh matematikawan dari Prancis, Edouard Lucas pada tahun 1883. Soal dari permainan ini dalam bahasa Inggris tertulis demikian:

“Legend has it that a group of Eastern monks are the keepers of three towers on which sit 64 golden rings. Originally all 64 rings were stacked on one tower with each ring smaller than the one beneath. The monks are to move the rings from this first tower to the third tower one at a time but never moving a larger ring on top of a smaller one. Once the 64 rings have all been moved, the world will come to an end”.

Sudah banyak tertulis di berbagai sumber bahwa solusi optimal untuk permasalahan ini adalah  $2^n - 1$  gerakan. Dengan asumsi satu gerakan akan memakan waktu 1 detik, maka dengan perhitungan singkat akan didapat bahwa untuk Menara Hanoi dengan ketinggian 64 akan memakan waktu sampai  $5.85 \times 10^{11}$  tahun.

Atau dapat ditawarkan lagi soal lain:

“Take only 5 rings and try all possible solutions. Then the Universe will come to an end”.

Jika kita mengambil asumsi bahwa setiap solusi (yang memiliki biaya yang berbeda) hanya satu detikpun, akan memakan waktu hingga  $8.967 \times 10^{21}$  tahun untuk mencoba semua kemungkinan solusi. Terlalu banyak waktu untuk membuat kemungkinan terjadi *Big Bang* berkali-kali.

Bahkan, untuk bahan lelucon, hingga ditawarkan soal seperti ini:

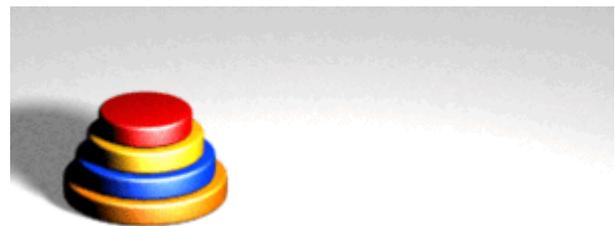
“Take only 4 rings and let each human on Earth solve the game in different way. When all different solutions are used the Universe will come to an end”.

Demikian sejarah yang membuat permainan ini menarik untuk orang-orang pada jaman itu. Dan dalam berjalannya waktu hingga sekarang banyak ahli yang mencoba menerapkan banyak algoritma untuk menyelesaikan permasalahan Menara Hanoi ini.

Permainan ini terdiri dari sebuah papan main yang di atasnya sudah tertancap tiga tiang dengan ketinggian yang sama. Selain itu, kakas lain dalam permainan ini adalah kepingan-kepingan lingkaran dengan diameter yang berbeda dan dapat berpindah ke tiang yang lain. Permainan ini dimulai dengan kepingan yang tersusun dengan susunan yang sudah ditentukan, yaitu kepingan berdiameter terkecil berada di atas, dan membentuk piramida. Tujuan dari permainan ini adalah untuk memindahkan semua tumpukan kepingan ke tiang yang berbeda, dengan posisi yang sama seperti posisi semula.

Ada beberapa aturan yang telah ditetapkan untuk menyelesaikan permainan ini, antara lain:

- Hanya satu kepingan puzzle yang boleh dipindahkan oleh pemain pada satu giliran.
- Setiap giliran gerakan terdiri dari memindahkan kepingan paling atas yang terdapat pada satu tiang, dan memasukkan kepingan tersebut ke tiang lain, sehingga kepingan tersebut menjadi kepingan paling atas di tiangnya yang baru, baik dibawahnya ada kepingan maupun tidak.
- Sebuah kepingan tidak dapat dipindahkan ke tiang yang lain apabila pada puncak tumpukan di tiang tersebut terdapat kepingan dengan ukuran diameter yang lebih kecil daripada kepingan yang ingin dipindahkan.



Gambar 2 - Tumpukan Kepingan Menara Hanoi

## B. Pemrograman Dinamis (Dynamic Programming)

Dynamic Programming (pemrograman dinamis) adalah salah satu metode pemecahan masalah yang dilakukan dengan cara menguraikan pemecahan masalah atau solution problem menjadi sekumpulan langkah (step) atau tahapan (stage) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan prinsip optimalitas. Prinsip Optimalitas. 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. Jika pada setiap tahap kita menghitung ongkos, maka dapat dirumuskan bahwa:

$$\text{Ongkos pada tahap } k + 1 = (\text{ongkos yang dihasilkan pada tahap } k) + (\text{ongkos dari tahap } k \text{ ke tahap } k + 1)$$

Dengan prinsip optimalitas ini akan dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya.

Nama dari Pemrograman Dinamis sendiri itu mengandung makna historis. Pemrograman Dinamis mengacu pada perhitungan yang menggunakan tabel. Seperti halnya dengan algoritma *greedy*, pemrograman dinamis digunakan untuk memecahkan masalah optimasi, yaitu masalah yang memiliki banyak kemungkinan solusi, dan kita diminta untuk mencari solusi yang memberikan nilai optimal.

Karakteristik Persoalan yang dimiliki oleh Program Dinamis adalah sebagai berikut :

1. Persoalan dapat dibagi menjadi beberapa tahap (stage), yang pada setiap tahap hanya dapat 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. Jumlahnya bisa berhingga atau tak berhingga.
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 mengidentifikasi 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.

Dalam implementasinya, terdapat dua pendekatan yang digunakan dalam Pemrograman Dinamis, yaitu 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  $1, 2, \dots, n$ . 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  $x_n, x_{n-1}, \dots, x_1$ .

Penyelesaian secara maju ataupun mundur akan menghasilkan hasil yang ekuivalen dan menghasilkan solusi optimum yang sama. Pengalaman menunjukkan bahwa penyelesaian dengan program dinamis mundur umumnya lebih mangkus. Kebanyakan literatur tentang program dinamis menyajikan penyelesaian masalah dengan pendekatan mundur.

Langkah-langkah Pengembangan Algoritma Program Dinamis sebagai berikut :

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

## III. DESKRIPSI MASALAH

Persoalan yang ada dalam permainan Menara Hanoi adalah bagaimana kita dapat menyelesaikan permainan ini dengan langkah sesedikit mungkin, dengan kata lain kita harus menentukan langkah yang tepat untuk memindahkan satu kepingan ke tiang yang lain tanpa adanya pengulangan yang tak perlu, sehingga langkah yang diambil di setiap tahap optimal dan permainan ini dapat diselesaikan dengan solusi yang terbaik.

Berikut ini adalah gambar susunan awal Menara Hanoi yang nantinya saat selesai dipindahkan bentuk Menara Hanoi juga harus seperti yang tergambar dibawah ini :



Gambar 3- Susunan awal Menara Hanoi

Untuk makalah ini, hanya akan dibahas untuk permainan menara Hanoi dengan tingkat ketinggian (n) sebesar 1 dan 2 tingkat sebagai contoh, dari situ, penulis akan mencoba menarik kesimpulan untuk kasus Menara Hanoi secara umum.

#### IV. PENYELESAIAN MASALAH

Dalam penyelesaian masalah Menara Hanoi ini, penulis akan memberikan 2 contoh, yaitu untuk Menara Hanoi dengan tingkat ketinggian (n) = 1 dan n = 2. Untuk kasus Menara Hanoi, pengambilan keputusan untuk setiap giliran cukup penting dan memberikan kemungkinan untuk giliran berikutnya, dimana kemungkinan itu dapat semakin mendekatkan kita dengan solusi, atau malah semakin menjauhkan kita dari solusi. Sehingga keputusan pada setiap tahap harus sesuai dengan jalur solusi. Karena state awal dan akhir solusi dari permainan Menara Hanoi ini sama hanya berbeda posisi saja, maka seharusnya tidak ada perbedaan antara melakukan Program Dinamis maju atau Program Dinamis mundur. Untuk solusi permasalahan ini, akan digunakan metode Program Dinamis maju, dimana langkah algoritma akan selalu dimulai dari awal, meskipun mencakup proses rekursif didalamnya.

Solusi untuk permasalahan ini dapat dimulai dengan mengkodekan beberapa hal berikut:

Tiang kiri : L (left)

Tiang tengah : C (center)

Tiang kanan : R (right)

n : total banyaknya kepingan, dengan aturan penomoran beri nomor 1 untuk kepingan terkecil (selalu berada di paling atas piramida), dan seperti itu seterusnya sampai n (nomor untuk kepingan terbesar, dan selalu berada di paling bawah).

Kunci dalam pembentukan solusi dari permasalahan Menara Hanoi ini adalah untuk mengetahui bahwa masalah ini dapat dipecahkan dengan membagi-bagi problem menjadi serangkaian masalah yang lebih kecil dan terus membagi-bagi masalah ini hingga solusi

tercapai. Sebenarnya kunci ini akan mengarahkan penulis untuk menggunakan algoritma Divide and Conquer, akan tetapi disini akan penulis coba penggunaan Pemrograman Dinamis untuk menyelesaikan permasalahan ini, diharapkan permasalahan Menara Hanoi ini dapat diselesaikan juga dengan penggunaan Pemrograman Dinamis.

Solusi dari permasalahan ini adalah membangkitkan tabel untuk mencari langkah terbaik yang harus dilakukan di setiap tahap dengan menetapkan algoritma rekursif ini terlebih dahulu.

Langkah-langkah untuk menyusun algoritma rekursif untuk memindahkan n kepingan dari tiang L ke tiang R adalah:

1. Pindahkan (n-1) kepingan-kepingan dari L ke C. Sehingga akan menyisakan kepingan dengan nomor n sendiri di L
2. Pindahkan kepingan dengan nomor n dari L ke R.
3. Pindahkan (n-1) kepingan-kepingan dari C ke R, sehingga (n-1) kepingan-kepingan itu akan berada di atas kepingan dengan nomor n di R.

Tiga langkah diatas merupakan langkah-langkah dalam algoritma rekursifnya, untuk mengerjakan langkah 1 dan 3, lakukan langkah yang sama lagi dan lagi untuk (n-1). Rangkaian prosedur solusi adalah sebuah langkah-langkah yang finite dimana dari langkah-langkah diatas akan dibutuhkan kondisi untuk n=1.

Untuk kondisi n=1, akan ada dua kemungkinan solusi langkah, yaitu pindahkan kepingan dari L ke R, atau pindahkan kepingan dari L ke C, lalu dari C ke R. Setelah itu, dengan asumsi n>=2, untuk mencapai final state dari solusi, kepingan terbesar (kepingan nomor n) harus dipindahkan dari L ke R.

Dalam penghitungan banyaknya solusi, kita akan menghitung F(n) yang dikomputasi dengan formula rekursif berikut:

$$F(n) = F(n-1)^2 + F(n-1)^3, n \geq 2$$

$$F(1) = 2$$

Setelah algoritma diatas siap digunakan, maka akan dibahas di contoh-contoh dibawah ini:

*Contoh 1: Menara Hanoi dengan n=1*

Untuk memindahkan Menara Hanoi dengan tingkat ketinggian (n) = 1, didapat tabel solusi yang berhasil digenerate sebagai berikut:

No	Langkah	Biaya
1	1LR	1
2	1LC 1CR	2

Tabel 1 - Tabel Langkah Solusi (n=1)

Dimana untuk tabel yang berhasil digenerate diatas, akan diambil langkah dengan biaya terkecil, yaitu langkah dengan nomor 1.

*Contoh 2: Menara Hanoi dengan n=2*

Untuk memindahkan Menara Hanoi dengan tingkat ketinggian (n) = 2, didapat tabel solusi yang berhasil

digenerate sebagai berikut:

No	Langkah	Biaya
1	1LC 2LR 1CR	3
2	1LC 2LR 1CL 1LR	4
3	1LR 1RC 2LR 1CR	4
4	1LR 1RC 2LR 1CL 1LR	5
5	1LR 2LC 1RL 2CR 1LR	5
6	1LR 2LC 1RL 2CR 1LC 1CR	6
7	1LR 2LC 1RC 1CL 2CR 1LR	6
8	1LR 2LC 1RC 1CL 2CR 1LC 1CR	7
9	1LC 1CR 2LC 1RL 2CR 1LR	6
10	1LC 1CR 2LC 1RL 2CR 1LC 1CR	7
11	1LC 1CR 2LC 1RC 1CL 2CR 1LR	7
12	1LC 1CR 2LC 1RC 1CL 2CR 1LC 1CR	8

Tabel 2 - Tabel Langkah Solusi ( $n=2$ )

Dimana untuk tabel yang berhasil digenerate diatas, akan diambil langkah dengan biaya terkecil, yaitu langkah dengan nomor 1.

Dari dua contoh diatas, telah terbukti bahwa penyusunan tabel untuk pemanfaatan Pemrograman Dinamis disini mampu menyelesaikan masalah Permainan Menara Hanoi.

## V. KESIMPULAN

Kesimpulan yang dapat diambil oleh penulis dalam pembuatan makalah ini adalah :

1. Pemrograman Dinamis (Dynamic Programming) dapat digunakan untuk pencarian solusi permainan Menara Hanoi.
2. Tidak semua masalah dapat diselesaikan dengan menggunakan Pemrograman Dinamis, terutama persoalan yang tidak bisa dibagi langkah keputusan bertahap.
3. Penggunaan algoritma program dinamis untuk permasalahan permainan Menara Hanoi dapat dikategorikan sebagai pemecahan masalah jalur terpendek.

## DAFTAR PUSTAKA

- [1] Munur, Rinaldi. 2005, "Diktat Kuliah IF2251 Strategi Algoritmik". Bandung : Institut Teknologi Bandung
- [2] <http://staff.um.edu.my/jskl1/dp/th.html>  
Tanggal akses : 7 Desember 2010
- [3] [http://en.wikipedia.org/wiki/Dynamic\\_programming](http://en.wikipedia.org/wiki/Dynamic_programming)  
Tanggal akses : 6 Desember 2010
- [4] [http://en.wikipedia.org/wiki/Tower\\_of\\_Hanoi](http://en.wikipedia.org/wiki/Tower_of_Hanoi)  
Tanggal akses : 6 Desember 2010

## PERNYATAAN

Dengan ini saya menyatakan bahwa Makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari Makalah orang lain, dan bukan plagiasi.

Bandung, 7 Desember 2010



Jonathan Ery Pradana / 13508007