# Penyelesaian Masalah Menara Hanoi dengan Pemrograman Dinamis

Candra Alim Sutanto (13508069)
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*if18069@students.if.itb.ac.id*

***Abstract*—Menara Hanoi yang juga disebut Menara Brahma merupakan sebuah permainan matematis atau teka-teki. Teka-teki ini terdiri dari tiga tiang dan sejumlah cakram dengan ukuran yang berbeda yang bisa dimasukkan ke tiang mana saja. Permainan dimulai dengan cakram-cakram yang tertumpuk rapi berurutan berdasarkan ukurannya dalam salah satu tiang, cakram terkecil diletakkan teratas, sehingga membentuk kerucut. Tujuan dari permainan matematis ini adalah memindahkan seluruh cakram dari satu tiang ke tiang yang lain dengan beberapa aturan. Dalam permasalahan menara hanoi ini, solusi berusaha didapatkan dengan algoritma 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.**

**Dalam penyelesaian Menara Hanoi terdapat beberapa algoritma yang bisa digunakan, termasuk salah satunya adalah Pemrograman Dinamis (Dynamic Programming). Langkah yang digunakan untuk menyelesaikan dengan fungsi rekursif dan dicari solusi optimal tiap langkah.**

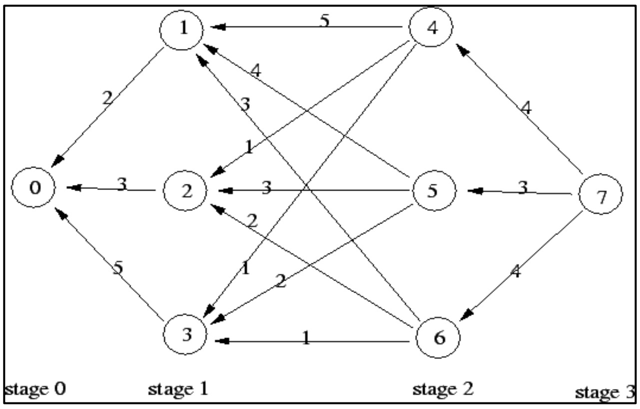
***Index Terms*—Dynamic Programming, Menara Hanoi, Rekursif.**

## I. PENDAHULUAN

### *A. Dynamic Programming*

Pemrograman Dinamis (dynamic programming) adalah sebuah metode untuk memecahkan 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.

Pemrograman dinamis (dynamic programming) istilah awalnya digunakan pada tahun 1940-an oleh Richard Bellman untuk menggambarkan proses pemecahan masalah dimana perlu menemukan keputusan terbaik satu demi satu. Pada 1953, Bellman kemdian menggunakan istilah pemrograman dinamis untuk proses penyelesaian masalah dimana keputusan yang lebih kecil berada dalam keputusan yang lebih besar.




Gambar 1. Ilustrasi Dynamic Programming

Pada penyelesaian persoalan dengan metode ini:

- Terdapat sejumlah berhingga pilihan yang mungkin.
- Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya.
- Digunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas. Dalam prinsip optimalitas jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal.

Dua pendekatan yang digunakan dalam Dynamic Progamming adalah maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*).

Misalkan x1, x2, …, xn menyatakan peubah (variable) keputusan yang harus dibuat masing-masing untuk tahap 1, 2, …, n. Maka,

- 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 x1, x2, …, xn.
- 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 xn, xn-1, …, x1.

Secara umum, ada empat langkah yang dilakukan dalam mengembangkana algoritma program dinamis:

- Karakteristikkan struktur solusi optimal.
- Definisikan secara rekursif nilai solusi optimal.
- Hitung nilai solusi optimal secara maju atau mundur.
- Konstruksi solusi optimal.

### *B. Menara Hanoi*

Menara Hanoi adalah sebuah permainan matematis atau teka-teki. Teka-teki ini ditemukan Eduard Lucas, ahli matematika Perancis di tahun 1883.

Gambar 2. Kondisi Awal Permainan Menara Hanoi

Ada sebuah legenda tentang candi Indian yang berisi ruang besar dengan tiga tiang yang dikelilingi 64 cakram emas. Pendeta Brahma, melaksanakan tugas dari peramal di masa lalu, sesuai dengan aturan teka-teki ini.

Jika legenda itu benar dan jika pendeta memindahkan satu cakram per detik, maka pendeta memerlukan $2^{64}$-1 detik untuk menyelesaikan masalah Menara Hanoi atau sekitar 585 milliar tahun dan pendeta akan memerlukan 18,446,744,073,709,551,615 langkah dalam menyelesaikannya.

Permainan ini terdiri dari tiga tiang dan sejumlah cakram dengan ukuran berbeda-beda yang bisa dimasukkan ke tiang mana saja.

Permainan Menara Hanoi dimulai dengan cakram-cakram yang tertumpuk rapi dari cakram paling besar sampai ke cakram paling terkecil dalam salah satu tiang, sehingga membentuk kerucut.

Objektif dari permainan Menara Hanoi adalah memindahkan tumpukan *n* buah cakram berlubang dari tiang asal ke tiang tujuan dengan memanfaatkan sebuah tiang perantara. Piringan berukuran tidak sama. Jumlah pemindahan dalam *n* buah cakram adalah sebanyak $2^n$-1 kali.

Permainan Menara Hanoi memiliki beberapa aturan yang harus dipatuhi untuk menyelesaikan teka-teki. Dalam melakukan pemindahan cakram harus mengikuti aturan berikut:

- Hanya satu cakram yang boleh dipindahkan dalam setiap kal perpindahan.
- Setiap perpindahan berupa pengambilan cakram teratas dari satu tiang dan memasukkannya ke tiang lain, di atas cakram lain yang mungkin sudah ada di tiang tersebut.
- Tidak boleh meletakkan cakram di atas cakram lain yang lebih kecil.

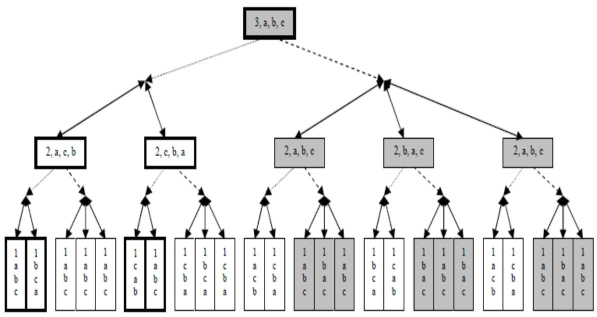
Tower of Hanoi sering digunakan dalam riset psikologi dalam pemecahan masalah. Terdapat pula variasi dari Tower of Hanoi, yaitu Tower of London untuk diagnose neuropsikologi dan pengerjaan fugsi eksklusif. Tower of Hanoi juga sering dipakai dalam skema Backup Rotation ketika membuat penggandaan data komputer dimana banyak tape/ media penyimpanan termasuk didalamnya.

Tower of Hanoi dapat dipakai untuk melatih kreativitas anak-anak dalam masa pertumbuhan. Selain itu, Tower of Hanoi juga sering diimplementasikan dalam proses pengajaran algoritma rekursif dasar. Tower of Hanoi juga digunakan untuk tes memory oleh para neuropsikolog untuk mengevaluasi amnesia.

## II. METODE PENYELESAIAN

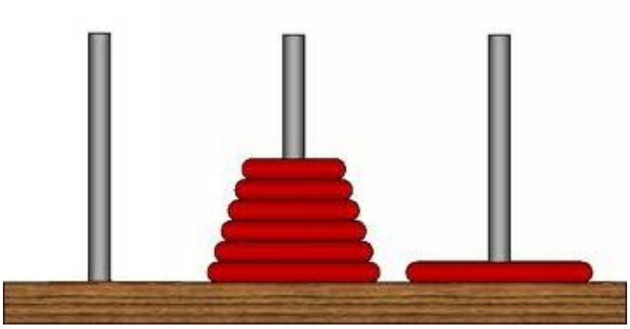
Pemrograman Dinamis memiliki strategi penyelesaian dengan cara mendapatkan solusi optimal dari sub-masalah untuk kemudian dibuat solusi dari masalah secara keseluruhan.

Permasalah Menara Hanoi memiliki empat buah parameter dalam penyelesaiannya, H(*n, a, b, c*). Sebanyak *n* buah cakram harus dipindahkan dari tiang *a* ke tiang *c* dengan memakai bantuan tiang *b*. Permasalahan utama adalah meminimalkan jumlah pemindahan yang harus dilakukan agar semua cakram bisa dipindahkan dari tiang a ke tiang b dengan tetap memenuhi aturan dari permainan atau teka-taki Menara Hanoi.




Gambar 3. Ruang Solusi Menara Hanoi

Ide utama dari penyelesaian permainan Menara Hanoi ini adalah memindahkan cakram terbesar dari tiang awal ke tiang tujuan. Cakram utama dipindahkan dari tiang a ke tiang b. Hal ini berarti (*n*-1) cakram sudah di pindahkan satu per satu dari tiang a ke tiang c. Setelah cakram terbesar dipindahkan ke tiang b, maka perlu dipindahkan (*n*-1) cakram dari tiang c ke tiang b secara satu per satu.



Gambar 4. Permainan Menara Hanoi

Dalam pemrograman dinamis, jika ingin mendapatkan solusi yang optimal dalam suatu masalah maka perlu didapatkan solusi yang optimal pula dalam sub-masalah yang ada.

Begitu pula dengan permainan Menara Hanoi ini. Untuk mendapatkan solusi optimal dalam permasalahan utama, maka perlu dicari solusi optimal untuk perpindahan (*n*-1) cakram dari tiang c ke tiang b. Maka, akan ada fungsi yang menentukan langkah untuk mendapatkan jumlah perpindahan minimum yang merupakan solusi optimal dari penyelesaian Menara Hanoi ini.

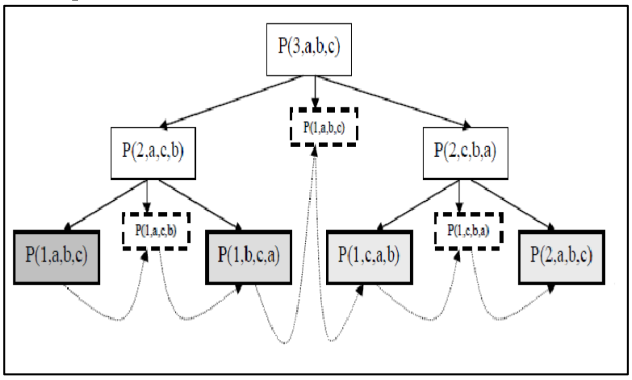
Fungsi rekursif ($f_{min}$) yang diperoleh dari penyelesaian Menara Hanoi

- if n = 1, then H(n, a, b, c) = (a dipindahkan ke b)
- if n > 1, then H (n, a, b, c) = H (n – 1, a, c, b)
  (a dipindahkan ke b)
  H (n - 1; c; b; a)

Fungsi rekursif ini akan memrlukan $2^n$-1 langkah untuk menyelesaikan permasalahan Menara Hanoi. Langkah tersebut merupakan solusi optimal dari penyelesaian Menara Hanoi.

## III. IMPLEMENTASI

Kesulitan utama dalam Pemrograman Dinamis adalah menentukan urutan dari penyelesaian sub-masalah untuk mendapatkan solusi dari keseluruhan masalah.




Gambar 5. Ilustrasi Dynamic Programming

Dalam Gambar 4 diatas diilustrasikan bagaimana proses Pemrograman Dinamis bottom-up dalam penyelesaian Menara Hanoi.

Solusi optimal dari permasalahan H(3, a, b, c) dapat dihasilkan dengan sub-solusi H(1, a, b, c), H(2, a, c, b) yang mana dihasilkan dari optimal solusi dari H(1, a, b, c) dan H(1, b, c, a) dan H(3, a, b, c) yang merupakan solusi optimal yang dihasilkan dari H(2, a, c, b) dan H(2, c, b, a)

Dalam pemrograman dinamis sering terjadi tumpukkan dalam pemecahan sub-masalah. Untuk menghindari hal itu, digunakan teknik *memoization*. Berdasarkan teknik ini, ketika suatu sub-masalah sudah memiliki solusi optimal, maka solusi tersebut disimpan (*memorized*). Ketika menemukan sub-masalah yang sama lagi, maka solusi optimal yang disimpan bisa diambil kembali (*retrieve*) dengan mudah.

Dari fungsi $f_{min}$ yang sudah dihasilkan diatas, maka bisa dihasilkan prosedur yang akan rekursif yang akan mencari sub-solusi optimal yang nantinya akan menghasilkan solusi optimal.

Prosedur yang dikembangkan juga akan memakai teknik *memoization*, sehingga apabila ditemukan sub-masalah dalam Menara Hanoi, seperti misalnya terdapat state yang sama, maka langsung bisa diketahui sub-solusi optimal dari sub-masalah tersebut.

Prosedur P_memoization dibawah ini menerapkan teknik *memoization*

```
void P_memozation (int k, char s, char d, char h, int n, int
*moves)
  {
    int i, p, pattern;
    pattern = ((n+k)&1);
    if(k==1)
    {
          moves[0]=pattern;
    }
    else
    {
        P memozation min(k-1, s, h, d, n, moves);
        p=(1<<(k-1))-1;  //jumlah perpindahan di dalam
                             sub-masalah
        moves[p]=pattern; //perpindahan cakram terbesar
        for(i=0;i<p;++i) //menghasilkan sub-solusi
        moves[p+1+i]=move_changing[pattern][moves[i]];
       }
  }
```

## IV. ANALISIS PEMBAHASAN

Dalam implementasi, fungsi rekursif yang dihasilkan dalam $f_{min}$ diatas kemudian diimplementasikan dalam sebuah prosedur, yaitu prosedur P_memoization yang menggunakan teknik *memoization* merupakan sudah menggunakan prinsip optimalisasi dalam menyelesaikan permasalahan Menara Hanoi.

Prosedur tersebut juga mengimplementasi sebuah proses bottom-up yang merupakan strategi implementasi dari Pemrograman Dinamis (Dynamic Programming).

Dalam implementasi prosedur diatas tidak hanya menghasilkan pemyelesaian yang lengkap, tetapi juga menghasilkan solusi terbaik yang mungkin dicapai dan faktanya solusi yang dapat ditemukan adalah :

```
Pindahkan keping 1 dari tower 1 ke tower 2.
Pindahkan keping 1 dari tower 2 ke tower 3.
Pindahkan keping 1 dari tower 3 ke tower 1.
```

Untuk menunjukan solusi yang tidak dapat direduksi perlu diperhatikan bahwa pada setiap tahap pekerjaan yang dilakukan dapat dirangkumkan sebagai perpindahan beberapa keping dari satu tower ke tower lain. Dan tidak ada cara lain untuk menyelesaikan pekerjaan ini terkecuali dengan memindahkan semua keping kecuali dengan diawali dengan memindahkan keping terbawah terlebih dahulu, kemudian mungkin membuat beberapa perpindahan yang berlebihan, kemudian memindahkan keping terbawah, lalu mungkin membuat perpindahan lagi dan pada akhirnya memindahkan keping teratas.

## V. KESIMPULAN

Dari pembahasan sebelumnya, kita dapatkan bahwa untuk mendapatkan solusi dari Tower of Hanoi kitadapat menerapkan berbagai macam teori. Salah satu diantaranya adalah memakai Pemrograman Dinamis atau Dynamic Programming.

Pemrograman Dinamis memiliki strategi penyelesaian dengan cara membagi masalal ke dalam beberapa sub-masalah. Dari masing-masing sub-masalah kemudian didapatkan solusi optimal. Dari masing-masing sub-solusi kemudian digabungkan untuk menjadi solusi optimal dari permasalahan utama dalam hal ini permainan atau teka-teki Menara Hanoi.

Namun lain halnya dengan versi lain Tower of Hanoi. Misalnya Puzzle Reve. Puzzle ini mirip seperti Tower of Hanoi namun memiliki jummlah tonggak yang berbeda. Solusi optimal dari Puzzle Reve ini belum dapat dipastikan sebagai solusi yang optimal karena belum ada pembuktian menggunakan teori lain.

Kembali pada versi umum Tower of Hanoi, puzzle ini dapat diaplikasikan dalam berbagai hal yang bermanfaat seperti penyimpanan Backup Data pada komputer maupun analisis psikologi.

## REFERENCES


http://en.wikipedia.org/wiki/Tower_of_Hanoi (Tanggal akses 6 Desember 2010)

http://en.wikipedia.org/wiki/Dynamic_Programming (Tanggal akses 6 Desember 2010)

http://www.ifors.ms.unimelb.edu.au/tutorial/hanoi/ (Tanggal akses 8 Desember 2010)

Munir, Rinaldi. 2009. *Strategi Algoritma*. Bandung : Penerbit Informatika.


## 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, 08 Desember 2010

Candra Alim Sutanto - 13508069