

# PENGGUNAAN ALGORITMA BRUTE FORCE DALAM Mencari SOLUSI MENARA HANOI

Khairul Fahmi / 13507125

Teknik Informatika, ITB  
Jalan Ganesha 10, Bandung  
e-mail: if17125@students.if.itb.ac.id

## ABSTRAK

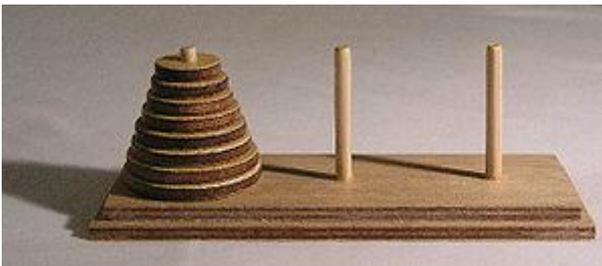
Menara Hanoi adalah sebuah puzzle matematika mengenai pemindahan cakram dari sebuah tiang ke tiang lain dengan bantuan satu tiang perantara. Pencarian solusi dilakukan dengan menggunakan algoritma *brute force* secara rekursif.

**Kata kunci:** puzzle, *brute force*.

## 1. PENDAHULUAN

### 1.1. Menara Hanoi

Menara Hanoi adalah sebuah permainan matematika. Ditemukan oleh seorang ahli matematika Perancis Édouard Lucas pada tahun 1883. Permainan ini terinspirasi dari legenda seorang biarawan yang memindahkan 64 monastery di Hanoi, Vietnam. Terdapat banyak versi tentang legenda tersebut.



Gambr 1 Perangkat permainan Menara Hanoi

Objektif dari permainan Menara Hanoi adalah memindahkan tumpukan  $n$  buah piringan berlubang dari tiang asal ke tiang tujuan dengan memanfaatkan sebuah tiang perantara. Piringan berukuran tidak sama.

Dalam melakukan pemindahan, harus mematuhi aturan-aturan :

- Tiap langkah hanya boleh memindahkan satu piringan
- tiap saat sebuah piringan yang lebih kecil harus selalu berada diatas piringan yang lebih besar.

- Taiap kali perpindahan piringan yang diambil adalah yang paling atas.

### 1.2. Algoritma Brute Force

*Brute force* adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan.

Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*).

Karakteristik Algoritma *Brute Force*

- Algoritma *brute force* umumnya tidak “cerdas” dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang-kadang algoritma *brute force* disebut juga algoritma naif (*naïve algorithm*).
- Algoritma *brute force* seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya itu, tetapi dengan mencari pola-pola yang mendasar, keteraturan , atau trik-trik khusus, biasanya akan membantu kita menemukan algoritma yang lebih cerdas dan lebih mangkus.
- Untuk masalah yang ukurannya kecil, kesederhanaan *brute force* biasanya lebih diperhitungkan daripada ketidakmangkusannya. Algoritma *brute force* sering digunakan sebagai basis bila membandingkan beberapa alternatif algoritma yang mangkus.
- Meskipun *brute force* bukan merupakan teknik pemecahan masalah yang mangkus, namun teknik *brute force* dapat diterapkan pada sebagian besar masalah. Agak sukar menunjukkan masalah yang tidak dapat dipecahkan dengan teknik *brute force*. Bahkan ada masalah yang hanya dapat dipecahkan secara *brute force*. Beberapa pekerjaan mendasar di dalam komputer dilakukan secara *brute force*, seperti

menghitung jumlah dari  $n$  buah bilangan, mencari elemen terbesar di dalam tabel, dan sebagainya.

- Selain itu, algoritma *brute force* seringkali lebih mudah diimplementasikan daripada algoritma yang lebih canggih, dan karena kesederhanaannya, kadang-kadang algoritma *brute force* dapat lebih mangkus (ditinjau dari segi implementasi).

## 2. METODE

### 2.1. Langkah penyelesaian

Seperti pada kebanyakan puzzle matematika, penemuan solusi akan lebih mudah dengan menyelesaikan sebuah pola yang generic. Permasalahan Menara Hanoi dapat diselesaikan dengan memecahkan solusi permasalahan “ bagaimana memindahkan sebuah tumpukan piringan tinggi  $h$  dari menara A ke menara tujuan C. B adalah menara bantu.  $A \neq C$ .

Permasalahan pemindahan piringan simetri dengan permutasi dari nama menara. Jika solusi dai masalah ini adalah memindahkan piringan dari A ke C maka dengan mengganti nama menara, solusi yang sama bias digunakan untuk pilihan menara asal dan tujuan.

Jika hanya ada satu piringan,  $h=1$  atau tidak ada menara maka solusi akan mudah sekali yaitu dengan memindahkan piringan dari A ke C secara langsung. Misalkan ada  $n$  buah piringan yang terurut dari kecil diatas hingga besar dibawah. Masing-masing pringan dinamai dari 1 ke  $n$  dengan 1 adalah piringan paling atas,  $n$  piringan paling bawah.

Jika  $n > 1$ , agar piringan ke  $n$  bias dipindahkan dari A ke C maka semua piringan dari 1 ke  $n-1$  harus berada di menara B. Setelah piringan ke- $n$  berada di C maka permasalahan cara yang sama bias digunakan untuk memindahkan piringan ke  $n-1$ . Cara yang sama dapat digunakan. Permasalahan sekarang adalah memindahkan  $n-1$  piringan dari menara B ke C. Permasalahan ini akan sama persis dengan permasalahan sebelumnya jika menara B diganti namanya jadi A, sementara menara A yang kosong diberi nama B. Jika sudah, lakukan langkah yang sama pada  $n-1$  disk. Jika disk-1 berhasil dipindahkan ke menara C maka pemasalahn berkurang jadi  $n-2$ , dan seterusnya hingga  $n = 1$ . Cara ini disebut rekursif.

Algoritma diatas dapat dirumuskan sebagai berikut :

- Urutkan piringan dari paling besar  $n$  dibawah hingga paling kecil diatas 1.
- Step 1 : Jika  $n > 1$  maka gunakan prosedur diatas untuk memindahkan  $n-1$  piringan yang lebih kecil dari A ke B.
- Step 2 : Pindahkan piringan paling besar dari A ke C.

- Jika  $n-1 > 1$  gunakan prosedur diatas untuk memindahkan piringan dari B ke C.

Dengan menggunakan induksi matematik dapat dibuktikan bahwa prosedur diatas menghabiskan jumlah langkah paling kecil yang mungkin dan solusi yang dihasilkan merupakan satu-satunya langkah minimal.

Dengan menggunakan korelasi rekurens, jumlah langkah yang dibutuhkan untuk memindahkan  $n$  buah piringan dari menara pada permainan Menara Hanoi adalah

$$f(n) = 2^n - 1 \quad (1)$$

### 2.2. Algoritma dalam Bahasa C

```
#include <stdio.h>
#include <string.h>
#include <sys/time.h>

void Tower(int n, char A, char B, char C){
    if(n == 1){
        printf("Move 1 from %c to %c\n", A, C);
    }
    else {
        Tower(n-1, A, C, B);
        printf("Move %d from %c to %c\n", n,A,B);
        Tower(n-1, B, A, C);
    }
}

int main(int argc, char *argv[]){
    int nDisks;
    nDisks = atoi(argv[1]);
    time_t awal = time(NULL);
    doTowers(nDisks, 'A', 'B', 'C');
    time_t akhir = time(NULL);
    printf("%d", (int) difftime(akhir, awal));
    return 0;
}
```

## 3. ANALISIS DAN KESIMPULAN

Dari testing program diperoleh bahwa langkah yang dibutuhkan untuk memindahkan  $n$  buah piringan adalah  $2^n-1$  langkah. Algoritma ini relative mangkus untuk menyelesaikan persoalan Menara Hanoi sejauh ini yang memberikan jumlah langkah minimal.

## REFERENSI

[1] Munir, Rinaldi. “Bahan Kuliah ke-1

IF2251 Strategi Algoritmik”, 2004

[2] [http://en.wikipedia.org/wiki/Tower\\_of\\_Hanoi](http://en.wikipedia.org/wiki/Tower_of_Hanoi)