

ALGORITMA BRUTE FORCE

UNTUK PENYELESAIAN GAME SUDOKU

Oleh :

A. Mu'azirudin (10112266)

Adhy Kartansa (10112268)

Yeni hendriyani (10112269)

Abstrak

Di era teknologi yang semakin berkembang pesat sekarang ini, banyak sekali permainan yang bermanfaat dan mengasah otak untuk berfikir. Permainan-permainan ini tak hanya ada di dalam komputer saja melainkan sudah merambak ke telepon pintar yang semakin canggih. Salah satu permainan yang mengasah otak adalah *Sudoku*. *Sudoku* adalah permainan angka pengasah otak dan logika ketelitian. Permainannya cukup mudah dipahami tapi belum tentu mudah untuk di selesaikan. Maka dari itu kami akan mencoba menyelesaikan *Sudoku* dengan *Algoritma Brute Force*.

Pendahuluan

Game *Sudoku* adalah salah satu game yang dapat mengasah otak pemainnya. Tidak banyak dari mereka yang memainkannya dapat menyelesaikan permainan tersebut sampai tuntas walaupun dalam waktu yang lama di level tertentu. Hal tersebut karena permainan ini lumayan rumit. Oleh karena itu kami mencoba untuk menyelesaikan permasalahan tersebut dengan *Algoritma Brute force* dan melihat ke efisiennya.

Algoritma Brute force

Brute force adalah sebuah serangan terhadap sistem dimana penyerang mencoba memasuki sebuah sistem dengan memasukan semua kata kunci yang memungkinkan. Lebih tepatnya penyerangan akan memasukan satu persatu kata kunci dari berbagai kemungkinan untuk mendapatkan suatu akses. Biasanya *brute force* dilakukan untuk memasuki

sebuah sistem dan juga digunakan untuk memecahkan suatu enkripsi satu arah (one way hash).

Tetapi *brute force* ini memiliki kelemahan, yaitu karena banyaknya kemungkinan yang digunakan, waktu pemrosesan membutuhkan waktu yang sangat lama. Maka dari itu kemungkinan itu diperkecil dengan adanya sebuah teknik bernama *dictionary attack*. Intinya, pembatasan yang dilakukan pada *dictionary attack* adalah seorang penyerang membatasi apa saja kata-kata yang akan digunakan sebagai sebuah kata kunci.

Implementasi Algoritma :

- Tempatkan angka “1” pada sel pertama. Periksa apakah penempatan “1” diperbolehkan (dengan memeriksa baris, kolom, dan kotak).
- Jika diperbolehkan atau tidak ada pelanggaran, maju ke sel berikutnya. Tempatkan “1” pada sel tersebut dan periksa apakah diperbolehkan atau tidak.
- Jika pada pemeriksaan ditemukan pelanggaran, yaitu penempatan “1” tidak diperbolehkan, maka coba dengan menempatkan “2”.
- Jika pada proses penempatan ditemukan bahwa tidak satupun dari sembilan (1/2/3/4/5/6/7/8/9) angka diperbolehkan, maka tinggalkan sel tersebut dalam keadaan kosong (0), lalu mundur satu langkah ke sel sebelumnya. Nilai di sel tersebut dinaikan 1 angka.
- Ulangi sampai 81 buah sel.

- Lakukan Penghitungan waktu awal, Proses pencarian solusi dan Penghitungan waktu akhir dimana proses berhenti.
- Hitung selisih waktu dan frekuensi *counter* per detik
- Mengeluarkan output dari program berupa solusi untuk sudoku, jumlah iterasi serta waktu eksekusinya.

Contoh Masukan :

9	0	0	1	0	4	0	0	2
0	8	0	0	6	0	0	7	0
0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1
0	7	0	0	0	0	0	3	0
3	0	0	0	0	0	0	0	7
0	0	0	0	2	0	0	0	0
0	3	0	0	7	0	0	8	0
1	0	0	0	0	9	0	0	0

Hasil :

```
Hasil sudoku :
9 6 7 1 8 4 3 5 2
5 8 4 3 6 2 1 7 9
2 1 3 9 5 7 6 4 8
4 9 6 7 3 8 5 2 1
8 7 1 2 9 5 4 3 6
3 5 2 4 1 6 8 9 7
7 4 8 6 2 3 9 1 5
6 3 9 5 7 1 2 8 4
1 2 5 8 4 9 7 6 3

Jumlah iterasi = 3897539
Waktu eksekusi = 20.1213
```

Alur Program :

- Inisiasi Sudoku dengan jumlah baris dan kolom masing-masing sembilan.
- Lakukan pengambilan input dari file eksternal
- Memberikan status “update-able” pada sel yang bernilai ‘0’
- Eksekusi Algoritma Utama

➤ Kesimpulan

Dari hasil percobaan di atas, dapat kita lihat hasil sudokunya benar. Namun, apabila kita melihat iterasi

yang harus dilakukan serta waktu yang dibutuhkan untuk mencari solusi sudoku, algoritma ini akan memakan waktu yang sangat lama. Apalagi bila menyelesaikan sudoku dengan tingkat level yang tinggi.

Referensi :

www.wikipedia.org

<http://www.slideshare.net/idsecconf/bruteforce-basic-paper-linx>

<http://id.wikipedia.org/wiki/Sudoku>