

Penggunaan Algoritma *Brute Force* Dalam Menyelesaikan Permainan *Go Figure!*

Mochamad Reza Akbar
(13507131)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Jl.Ganesha 10, Bandung
e-mail: if17131@students.if.itb.ac.id

ABSTRAK

Algoritma *brute force* masih dapat digunakan untuk menyelesaikan masalah lama ataupun baru. Algoritma *brute force* pasti dapat menyelesaikan masalah baik dalam skala kecil maupun besar, walaupun tidak semangkus algoritma-algoritma lainnya. Algoritma *brute force* merupakan pendekatan masalah secara *straightforward*. Masalah dipecahkan dengan sangat sederhana, jelas dan langsung. Sehingga masalah-masalah yang ada tidak dapat dipecahkan oleh algoritma lain dapat dipecahkan dengan algoritma ini. Dalam makalah ini, penulis membahas tentang aplikasi algoritma *brute force* dalam menyelesaikan permainan *go figure!*. *Go figure!* adalah sebuah permainan matematika yang dirilis pada tahun 1992 sebagai bagian dari *Microsoft Windows Entertainment Pack*. Tujuannya adalah untuk mengatur empat angka yang diberikan dan pengoperasian simbol - penambahan, pengurangan, perkalian, dan pembagian - untuk menghasilkan jawaban yang diinginkan.

Go figure! dibuat untuk mengajak orang untuk menikmati dunia angka. Permainan ini juga merupakan alat untuk membantu peningkatan diri dalam meningkatkan kemampuan dalam matematika. Karena bermain dengan matematika lebih menyenangkan dibandingkan di sekolah.

Dengan menggunakan algoritma *brute force*, akan dicoba setiap kemungkinan penempatan angka jika solusi belum ditemukan, maka akan dicoba setiap kemungkinan penempatan simbol operasi untuk setiap kemungkinan penempatan angka hingga ditemukan solusi atau solusi tidak ada. Algoritma *brute force* digunakan karena semua masalah dapat didekati dengan algoritma ini dan sangat cocok dalam aplikasi permainan *go figure!*.

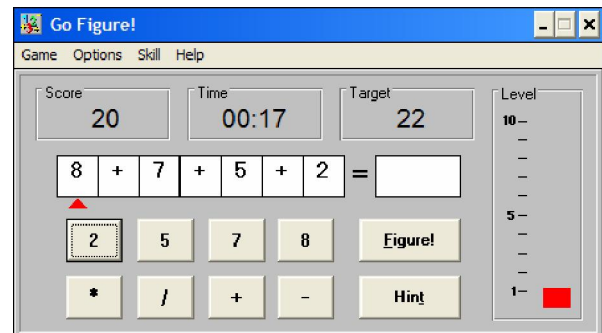
Kata kunci: *brute force*, *go figure!*.

1. PENDAHULUAN

Bermain *go figure!* sangat mudah, hanya menunjukkan dan meng-*click*. Tetapi didalamnya, komputer memilih angka-angka secara random dan melakukan operasi-operasi yang menghasilkan hasil operasi. Dengan banyaknya kemungkinan kombinasi, pemain tidak mungkin akan melihat teka-teki yang sama dua kali.

Inti dari permainan akan selalu sama: menyusun empat angka dan simbol-simbol operator yang mana saja untuk menghasilkan persamaan yang sama dengan jawaban yang ada di *target box*.

Permainan dapat dimainkan dengan tingkat permainan beginner(pemula), intermediate(menengah), atau expert(susah/ahli). Semua tingkat permainan akan memainkan sepuluh tingkat permainan. Pecahkan solusi dan pemain akan naik ke tingkat selanjutnya. Permainan akan berlanjut sampai tingkat sepuluh dan akan menghasilkan score(yang telah dikurangi untuk setiap jawaban yang salah, lama waktu, atau meminta bantuan)



Gambar 1. *go figure!*

1.1 Aturan Bermain

Setiap permainan dalam *go figure!* Terdapat 10 tingkat bermain. Setiap tingkat hanya terdapat satu persamaan teka-teki yang harus dipecahkan. Untuk tingkat permainan

beginner permainan terdapat 10 persamaan yang harus dipecahkan. *Intermediate* terdapat 20 persamaan dan untuk tingkat *expert* terdapat 30 persamaan.

Aturan untuk setiap tingkat permainan sama:

- Gunakan setiap empat angka yang diberikan hanya sekali untuk menghasilkan persamaan.
- Gunakan simbol operator yang mana saja sesukanya.
- Isi tujuh kotak yang berada di jendela persamaan.
- Isi dengan angka untuk kotak pertama, ketiga, kelima dan ketujuh.
- Isi dengan simbol operator untuk kotak kedua, keempat dan keenam.
- Jika melakukan kesalahan, pilih perintah 'Clear' untuk menghapus jendela persamaan
- Pemain hanya memiliki 3 bantuan dengan memakai perintah 'Hints' tetapi akan mengurangi nilai.

2. METODE

Metode yang dipakai dalam makalah ini adalah *brute force*.

2.1 Algoritma Brute Force

Brute force adalah sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah (problem statement) dan definisi konsep yang dilibatkan Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (obvious way) meskipun bukan merupakan solusi yang paling mangkus.

Dalam mencari solusi untuk mencari persamaan dari teka-teki yang diberikan. Akan dicoba kombinasi susunan angka dan simbol operator. Percobaan akan terus dilakukan sampai solusi ditemukan atau kombinasi yang ada sudah habis.

2.2 Penyelesaian dengan Algoritma Brute Force

Penggunaan brute force dalam mencari persamaan dapat dijelaskan secara umum sebagai berikut:

1. untuk kotak kedua, keempat dan keenam yang harus diisi dengan simbol operator, dicoba permutasi-permutasi yang mungkin dimulai dari simbol + + + sampai simbol ///.
2. agar memudah *iterasi*, operator tambah(+) ditandai 0, kurang(-) ditandai 1, kali(*) ditandai 2, bagi(/) ditandai 4. dan untuk angka yang tidak habis dibagi akan menghasilkan -999.

3. setiap permutasi simbol operator yang memungkinkan. Akan disusun kombinasi angka-angka yang memungkinkan pada kotak pertama, ketiga, kelima dan ketujuh.

4. untuk memudahkan *iterasi*, dilakukan permutasi hingga kombinasi yang dilakukan memakai semua angka yang sama.

5. jika solusi ditemukan, loop akan dihentikan dan menuliskan solusi persamaan.

6. jika solusi tidak ditemukan kombinasi angka akan dilakukan untuk kombinasi simbol selanjutnya.

Berikut *pseudocode* yang digunakan:

```
Function eksekusi (input operator, x1, x2 : integer) ->
integer
{ menghasilkan hasil operasi x1 dan x2 dengan
menggunakan operator }
```

Algoritma

depend on (operator)

0 : return (x1 + x2)

1 : return (x1 - x2)

2 : return (x1 * x2)

3 : return (x1 / x2)

```
Procedure figure (input x1,x2,x3,x4,hasil : integer)
{ mencari solusi persamaan }
```

Kamus

Solusi : boolean

i,j,k,l,m,n,o: integer

ii,jj : integer

op : array of integer [1..4]

Algoritma

Solusi = false

op[1] = x1

op[2] = x2

op[3] = x3

op[4] = x4

while (Solusi=false and ii<65) do

 i=ii div 16

 j=(ii mod 16) div 4

 k=(ii mod 16) mod 4

while (Solusi=false and jj<65) do

 l=jj div 16

 m=(jj mod 16) div 4

 n=(jj mod 16) mod 4

 o= 6-(l+m+n)

if l /= m and l /= n and m /= n

then

if eksekusi(k,eksekusi(j,eksekusi(i,op[1],

 op[m]),op[n]),op[o])

then Solusi=true

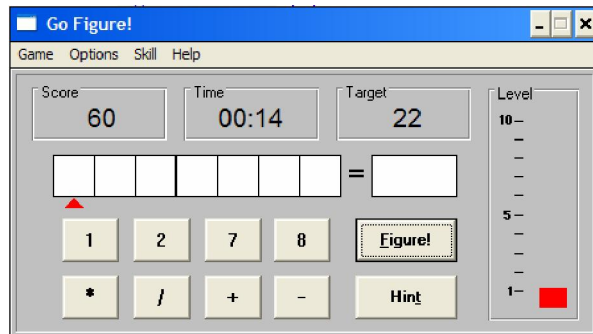
```

    jj=jj+1
    ii=ii+1
    jj=0
if Solusi=true
then
    output(i,j,k,l,m,n,o)
{i, j, k adalah operator sesuai urutan, dan l, m, n, o adalah
angka sesuai urutan}

```

2.3 Hasil Pengujian

Pengujian dilakukan dengan menggunakan bahasa C.



Gambar 2. contoh soal *go figure!*

Aplikasi diatas(*go figure!*) memilih angka-angka secara acak dan meminta persamaan untuk untuk menghasilkan angka sesuai kotak target.

```

7*1+2+8=17
7*1+8+2=17
7*2+1+8=23
7*2+8+1=23
7*8+1+2=59
7*8+2+1=59
8*1+2+7=17
8*1+7+2=17
8*2+1+7=24
8*2+7+1=24
8*7+1+2=59
8*7+2+1=59
1*2+7-8=1
1*2+8-7=3
1*7+2-8=1
1*7+8-2=13
1*8+2-7=3
1*8+7-2=13
2*1+7-8=1
2*1+8-7=3
2*7+1-8=7
2*7+8-1=21
2*8+1-7=10
2*8+7-1=22

```

Gambar 3. pencarian solusi persamaan(sebagian)

Gambar diatas memperlihatkan percobaan yang dilakukan sehingga didapat hasil persamaan sama dengan kotak target yang diinginkan.

```

masukkan angka :
1
2
7
8
masukkan hasil hitungan :
22
2*8+7-1=22
Press any key to continue . . .

```

Gambar 4. solusi soal *go figure!*

Input yang diperlukan dimasukkan, dimulai dari angka-angka dan diakhiri dengan target persamaan. Gambar diatas adalah hasil dari pencarian solusi persamaan yang diinginkan.

3. KESIMPULAN

Algoritma *brute force* dapat menyelesaikan masalah yang terdapat dalam permainan *go figure!*. Algoritma ini juga sangat cocok digunakan untuk mencari solusi persamaan dari permainan *go figure!* ini. Karena dalam permainan ini, diperlukan waktu yang cepat untuk mendapatkan nilai yang maksimum. Jika terdapat banyak solusi maka akan diambil persamaan yang lebih mengutamakan operasi tambah dan kurang dibandingkan kali dan bagi.

Pemakaian algoritma yang lain mungkin dapat mengurangi iterasi dan atau mengurangi jumlah operasi yang dilakukan tetapi waktu yang diperlukan relatif sama karena operasi yang dilakukan tidak terlalu banyak.

REFERENSI

[1] Munir, Rinaldi, "Diktat Kuliah IF2251 Strategi Algoritmik", Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung 2007