

Laporan
Tugas Kecil 1 IF2211 Strategi Algoritma
Penyelesaian *24 Game* dengan *Algoritma Brute Force*



oleh
Haris Salman Al-Ghifary / 13517052

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2019

BAB I

ALGORITMA *BRUTE FORCE*

1.1 Deskripsi Persoalan

24 game adalah sebuah permainan angka yang bertujuan untuk memanipulasi input yaitu 4 bilangan non negative dengan operator $+$, $-$, $*$, $/$, $()$ sehingga akan menghasilkan nilai 24. Sebagai contoh, misalkan dimasukkan 4 angka 4,7,8,28, salah satu solusi yang mungkin adalah $(4 * (7 - 8)) + 28 = 24$. Walaupun terdapat berbagai variasi permainan terkait operator, jumlah angka masukan, dan target hasil akhir, tugas kecil 1 ini akan menggunakan variasi dasar dengan 4 masukan, operator $+$ - $*$ / $()$, dan hasil akhir 24.

Sebagai contoh :

Welcome to 24 Game Solver

Enter your 4 numbers below, then click "Solve" to see every solution that equals 24.

4	7	8	28	Solve	Clear
---	---	---	----	-------	-------

36 solutions found | [Highlight similar solutions](#)

$(4 * (7 - 8)) + 28$	$(7 * 8) - (4 + 28)$	$8 + ((4 / 7) * 28)$	$((28 * 4) / 7) + 8$
$(4 / (7 - 8)) + 28$	$((7 * 8) - 4) - 28$	$8 + ((4 * 28) / 7)$	$(28 * (4 / 7)) + 8$
$((4 / 7) * 28) + 8$	$((7 - 8) * 4) + 28$	$(8 * 7) - (4 + 28)$	$(28 - 4) * (8 - 7)$
$(4 / (7 / 28)) + 8$	$(7 - 8) * (4 - 28)$	$((8 * 7) - 4) - 28$	$(28 - 4) / (8 - 7)$
$((4 * 28) / 7) + 8$	$(7 * 8) - (28 + 4)$	$(8 * 7) - (28 + 4)$	$((28 * 7) - 4) / 8$
$(4 * (28 / 7)) + 8$	$((7 * 8) - 28) - 4$	$((8 * 7) - 28) - 4$	$((28 / 7) * 4) + 8$
$(4 - 28) * (7 - 8)$	$((7 * 28) - 4) / 8$	$((8 - 7) * 28) - 4$	$(28 / (7 / 4)) + 8$
$(4 - 28) / (7 - 8)$		$(8 - 7) * (28 - 4)$	$28 + ((7 - 8) * 4)$
		$8 + ((28 * 4) / 7)$	$(28 * (8 - 7)) - 4$
		$8 + ((28 / 7) * 4)$	$(28 / (8 - 7)) - 4$
			$28 - ((8 - 7) * 4)$

Gambar 1 : Contoh 24Game solver . Dikutip dari Spesifikasi Tugas Kecil 1 Strategi Algoritma

1.2 Algoritma Brute Force

Algoritma Brute Force merupakan algoritma yang akan selalu menemukan sebuah solusi jika solusi tersebut ada tetapi waktu yang dibutuhkan juga akan lama jika masukkan terlalu besar. Algoritma ini digunakan untuk menyelesaikan sebuah permasalahan dengan dengan melakukan pendekatan *straightforward*. Dalam pengerjaan tugas kecil ini kita akan melakukan pendekatan dengan *exhaustive search*.

Pertama kita tentukan dahulu skema penyelesaian 24 game solver. Saya mulai dengan memasukkan input. Input bernilai 4 bilangan yaitu a,b,c,d dan pada program saya membuat inputan menjadi string agar dapat dicetak dan dihitung dengan fungsi eval. Karena algoritma brute force merupakan suatu cara penyelesaian dengan pendekatan semua enumerasi kemungkinan maka langkah kedua adalah dengan melakukan inisialisasi operator yaitu (*,/,+,-) serta melakukan permutasi pada operator tersebut agar dapat dicari semua kemungkinan yang ada. Kemudian langkah ketiga adalah melakukan inisialisai angka yaitu a,b,c,d dan kita lakukan permutasi sehingga ada 4! Kemungkinan kombinasi. Kemudian langkah paling krusial adalah permutasi gabungan antara operator,angka, dan parenthesis. Kemungkinan kombinasi gabungan adalah sebagai berikut : (ab) (cd), (a(bc))d, ((ab)c)d, a((bc)d), a(b(cd)). Ada 5 kemungkinan kombinasi, untuk Langkah terakhir adalah mengecek semua kemungkinan dengan fungsi eval yang mana hasil dari string tersebut adalah 24, jika hasilnya 24 maka akan dicetak dan variable count akan bertambah satu setiap kali true. Alhasil pada output dikeluarkan solusi yang mungkin, jumlah solusi, serta waktu eksekusi.

BAB II

SOURCE CODE PROGRAM

2.1 Source Code Python

```
3  #Tugas Kecil Stima "24 Game Solver"
4  #Author : Haris Salman AL-Ghifary
5
6  import time
7  def main_program():
8      print("\n")
9      print("-----")
10     print("-----Welcome to 24 Game Solver-----")
11     print("-----")
12     print("\n")
13
14     #Bagian input program
15     list_angka = a,b,c,d = [str(n) for n in input("Enter your 4 numbers in positive
number: ").split()]
16     while (int(a)<0 or int(b)<0 or int(c)<0 or int(d)<0):
17         print("Please insert a positive number")
18         list_angka = a,b,c,d = [str(n) for n in input("Enter your 4 numbers in positive
number: ").split()]
19
20     #inisialisasi kombinatorik operator
21     operator = ['*', '/', '+', '-']
22     kombinasi_opr = []
23     for opr1 in range(4):
24         for opr2 in range(4):
25             for opr3 in range(4):
26                 kombinasi_opr.append([operator[opr1], operator[opr2], operator[opr3]]
)
27
28     #inisialisasi untuk menghitung waktu eksekusi
29     start_time = time.time()
30
31     #inisialisasi kombinatorik angka
32     temp = []
33     for angka1 in range(4):
34         for angka2 in range(4):
35             for angka3 in range(4):
36                 for angka4 in range(4):
37                     if (angka1!=angka2 and angka1!=angka3 and angka1!=angka4 and angka2
!=angka3 and angka2!=angka4 and angka3!=angka4):
38                         temp.append([list_angka[angka1], list_angka[angka2], list_angka[a
angka3], list_angka[angka4]])
39
40
41     kombinasi_angka = []
42     for n in temp:
43         kombinasi_angka.append(tuple(n))
44     kombinasi_angka = list(set(kombinasi_angka))
```

```

45
46     #inisialisasi kombinatorik parenthesis,angka,opertor
47     kombinasi_all = []
48     for i in range(len(kombinasi_angka)):
49         for j in range(len(kombinasi_opr)):
50             kombinasi_all= kombinasi_all + [ '(' + kombinasi_angka[i][0] +
kombinasi_opr[j][0] + kombinasi_angka[i][1] + ')' + kombinasi_opr[j][1] + '(' +
kombinasi_angka[i][2] + kombinasi_opr[j][2] + kombinasi_angka[i][3] + ')',
51             '(' + kombinasi_angka[i][0] +
kombinasi_opr[j][0] + kombinasi_angka[i][1] + ')' + kombinasi_opr[j][1] +
kombinasi_angka[i][2] + ')' + kombinasi_opr[j][2] + kombinasi_angka[i][3],
52             '(' + kombinasi_angka[i][0] +
kombinasi_opr[j][0] + '(' + kombinasi_angka[i][1] + kombinasi_opr[j][1] +
kombinasi_angka[i][2] + ')' + kombinasi_opr[j][2] + kombinasi_angka[i][3],
53             kombinasi_angka[i][0] +
kombinasi_opr[j][0] + '(' + kombinasi_angka[i][1] + kombinasi_opr[j][1] +
kombinasi_angka[i][2] + ')' + kombinasi_opr[j][2] + kombinasi_angka[i][3] + ')',
54             kombinasi_angka[i][0] +
kombinasi_opr[j][0] + '(' + kombinasi_angka[i][1] + kombinasi_opr[j][1] + '(' +
kombinasi_angka[i][2] + kombinasi_opr[j][2] + kombinasi_angka[i][3] + ')')' ]
55
56     #Banyak solusi
57     count = 0
58
59     for n in range(len(kombinasi_all)):
60         try:
61             if (round(eval(kombinasi_all[n]),6)==24):
62                 print("==>", kombinasi_all[n])
63                 count = count+1
64             except ZeroDivisionError:
65                 pass
66         print("\n")
67         print("Oh", count , "solution found")
68
69     end = time.time()
70     print("Time excecution = ",end-start_time)
71     print("\n")
72     main_program()

```

BAB III

EKSPERIMEN

1

```
F:\>python Tucil.py
```

```
-----Welcome to 24 Game Solver-----
Enter your 4 numbers in positive number: 8 2 5 1
==> (5-2)*(1*8)
==> ((5-2)*1)*8
==> (5-(2*1))*8
==> ((5-2)/1)*8
==> (5-(2/1))*8
==> (5-2)/(1/8)
==> (8*(5-2))*1
==> 8*((5-2)*1)
==> 8*(5-(2*1))
==> (8*(5-2))/1
==> 8*((5-2)/1)
==> 8*(5-(2/1))
==> 8*((5*1)-2)
==> 8*((5/1)-2)
==> (8*(5+1))/2
==> 8*((5+1)/2)
==> 8*(5-(1*2))
==> ((8+5)-1)*2
==> (8+(5-1))*2
```

```
==> ((5/1)-2)*8
==> ((5+1)/2)*8
==> (5+1)/(2/8)
==> (5-(1*2))*8
==> (5+1)*(8/2)
==> ((5+1)*8)/2
==> (5-1)*(8-2)
==> ((5-1)+8)*2
==> (5-(1-8))*2
==> 2*((8-1)+5)
==> 2*(8-(1-5))
==> (2-8)*(1-5)
```

```
Oh 68 solution found
Time excecution = 0.15782880783081055
```

2

```

F:\>python Tucil.py
-----
-----Welcome to 24 Game Solver-----
-----

Enter your 4 numbers in positive number: 4 7 8 8
==> 4*(7-(8/8))
==> ((4+7)-8)*8
==> (4+(7-8))*8
==> ((7+4)-8)*8
==> (7+(4-8))*8
==> (7*8)-(4*8)
==> ((7-8)+4)*8
==> (7-(8-4))*8
==> 8*((4+7)-8)
==> 8*(4+(7-8))
==> 8*((4-8)+7)
==> 8*(4-(8-7))
==> 8*((7+4)-8)
==> 8*(7+(4-8))
==> (8*7)-(4*8)
==> (7*8)-(8*4)
==> (7-(8/8))*4
==> (8*7)-(8*4)
==> 8*((7-8)+4)
==> 8*(7-(8-4))
==> ((4-8)+7)*8
==> (4-(8-7))*8

Oh 22 solution found
Time excecution = 0.0608363151550293

```

3

```
-----Welcome to 24 Game Solver-----  
  
Enter your 4 numbers in positive number: 6 7 8 9  
==> (6*8)/(9-7)  
==> 6*(8/(9-7))  
==> (8/(9-7))*6  
==> 8/((9-7)/6)  
==> (8*6)/(9-7)  
==> 8*(6/(9-7))  
==> (6/(9-7))*8  
==> 6/((9-7)/8)
```

```
Oh 8 solution found  
Time excecution = 0.11219215393066406
```

4


```

-----
-----Welcome to 24 Game Solver-----
-----

Enter your 4 numbers in positive number: 7 5 3 4
==> (5*4)+(7-3)
==> ((5*4)+7)-3
==> ((5-4)+7)*3
==> (5-(4-7))*3
==> ((5*4)-3)+7
==> (5*4)-(3-7)
==> (5+(4*3))+7
==> 5+((4*3)+7)
==> (5+7)+(4*3)
==> 5+(7+(4*3))
==> ((5+7)-4)*3
==> (5+(7-4))*3
==> (7+5)+(3*4)
==> 7+(5+(3*4))
==> (7-5)*(3*4)
==> ((7-5)*3)*4
==> (3*4)*(7-5)
==> 3*(4*(7-5))
==> (3*4)+(7+5)
==> ((3*4)+7)+5
==> 4+((7-3)*5)
==> (4*3)*(7-5)
==> 4*(3*(7-5))
==> (4*3)+(7+5)
==> ((4*3)+7)+5
==> 4-((3-7)*5)
==> (7+(4*5))-3
==> 7+((4*5)-3)
==> (7-4)*(5+3)
==> ((7-4)+5)*3
==> (7-(4-5))*3
==> (7+(5*4))-3

```

```

==> 4-(5*(3-7))
==> ((7-3)*5)+4
==> (7-3)+(5*4)
==> 7-(3-(5*4))
==> 3*((7-4)+5)
==> 3*(7-(4-5))
==> (5*(7-3))+4
==> (5+7)+(3*4)
==> 5+(7+(3*4))
==> (5+3)*(7-4)
==> (4*(7-5))*3
==> 4*((7-5)*3)
==> (7+(3*4))+5
==> 7+((3*4)+5)
==> (7-3)+(4*5)
==> 7-(3-(4*5))
==> 3*((5+7)-4)
==> 3*(5+(7-4))
==> (3+5)*(7-4)
==> (4*3)+(5+7)
==> ((4*3)+5)+7
==> (3*4)+(5+7)
==> ((3*4)+5)+7
==> 3*((5-4)+7)
==> 3*(5-(4-7))
==> 3*((7+5)-4)
==> 3*(7+(5-4))
==> (3*(7-5))*4
==> 3*((7-5)*4)

```

0h 78 solution found

Time excecution = 0.3469855785369873

5

F:\>python Tucil.py

```
-----  
-----Welcome to 24 Game Solver-----  
-----
```

Enter your 4 numbers in positive number: 9 6 8 5

==> $(9+6)*(8/5)$

==> $((9+6)*8)/5$

==> $(8*(6+9))/5$

==> $8*((6+9)/5)$

==> $(8/5)*(6+9)$

==> $8/(5/(6+9))$

==> $((8-9)+5)*6$

==> $(8-(9-5))*6$

==> $((5-9)+8)*6$

==> $(5-(9-8))*6$

==> $((5+8)-9)*6$

==> $(5+(8-9))*6$

==> $(8/5)*(9+6)$

==> $8/(5/(9+6))$

==> $((8+5)-9)*6$

==> $(8+(5-9))*6$

==> $(6+9)*(8/5)$

==> $((6+9)*8)/5$

==> $((6+9)/5)*8$

==> $(6+9)/(5/8)$

==> $6*((8-9)+5)$

==> $6*(8-(9-5))$

==> $6*((8+5)-9)$

==> $6*(8+(5-9))$

==> $6*((5-9)+8)$

==> $6*(5-(9-8))$

==> $6*((5+8)-9)$

==> $6*(5+(8-9))$

==> $(8*(9+6))/5$

==> $8*((9+6)/5)$

==> $((9+6)/5)*8$

==> $(9+6)/(5/8)$

Oh 32 solution found

Time excecution = 0.16103911399841309

Keterangan Tambahan

Pada web <http://24solver.us-west-2.elasticbeanstalk.com/> hanya terdapat 4 kemungkinan kombinasi parenthesis yaitu $(ab)(cd)$, $(a(bc))d$, $((ab)c)d$, $a((bc)d)$, oleh karena itu hasil solusinya terdapat perbedaan karena saya menambahkan kombinasi $a(b(cd))$ sebagai kombinasi tambahan. Karena saya menganggap $2*(8+(5-1))$ berbeda dengan $(8+(5-1))*2$. Jadi pada program yang saya buat terdapat kombinasi yang lebih banyak.

BAB IV

REFERENSI

Sumber Tertulis:

<https://stackoverflow.com/questions/1557571/how-do-i-get-time-of-a-python-programs-execution>

<https://www.w3schools.com/python/default.asp>

<http://24solver.us-west-2.elasticbeanstalk.com/>

Sumber Tidak Tertulis:

Ardy Satrio yang telah memberi masukan saat debugging

Muhammad Hanif Azkiya yang telah memberi hidayah dalam penyelesaian permutasi

Serta tak lupa Ahmad Naufal Hakim yang telah memberi masukan tentang input dan fungsi eval pada python

Poin	Ya	Tidak
1. Program berhasil dikompilasi	ya	-
2. Program berhasil dieksekusi	ya	-
3. Solusi 24 Game benar untuk semua data tes. Bandingkanlah hasilnya dengan aplikasi 24solver yang sudah ada.	ya	-

Keterangan untuk poin 3 : Pada web <http://24solver.us-west-2.elasticbeanstalk.com/> hanya terdapat 4 kemungkinan kombinasi parenthesis yaitu (ab) (cd), (a(bc))d, ((ab)c)d, a((bc)d), oleh karena itu hasil solusi nya terdapat perbedaan karena saya menambahkan kombinasi a(b(cd)) sebagai kombinasi tambahan. Karena saya menganggap $2*(8+(5-1))$ berbeda dengan $(8+(5-1))*2$. Jadi pada program yang saya buat terdapat kombinasi yang lebih banyak.