

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



Oleh :
Aditya Putra Santosa / 13517013

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

BAB I

ALGORITMA BRUTE FORCE

1.1 Deskripsi Algoritma

Dalam program ini, algoritma yang digunakan adalah algoritma bruteforce / *exhaustive search*. Dinamakan seperti itu dikarenakan algoritma ini mencoba segala kemungkinan yang mungkin lalu mengevaluasinya dan hanya mengambil yang benar saja. Program ini dapat dibagi menjadi 4 tahapan yaitu pembangkitan semua kemungkinan urutan 4 angka hasil input pengguna, kemudian pembangkitan semua kemungkinan urutan 3 operator yang berada diantara angka-angka tersebut, kemudian pembangkitan 5 variasi peletakan kurung yang sekaligus juga merupakan pembangkitan ekspresi matematikanya, kemudian tahap terakhir adalah mengevaluasi semua kemungkinan ekspresi matematika tersebut.

Dalam pelaksanaannya, terdapat kemungkinan bahwa pengguna akan menginput beberapa angka yang sama. Jika pembangkitan angkanya dilakukan secara biasa maka akan ada beberapa hasil pembangkitan yang sama. Contohnya jika pengguna memasukkan angka 1,2,1,3 , pembangkitan biasa akan membangkitkan (1,2,1,3) 2 kali dikarenakan angka 1 ada 2 dan program menganggapnya berbeda. Agar hal tersebut tidak terjadi, perlulah dihapus hasil pembangkitan yang duplikat. Dalam program ini, cara yang digunakan adalah memanfaatkan sifat *set* / himpunan dari bahasa *python*. *Set* memiliki sifat setiap anggotanya *unique* atau tidak ada duplikat.

Selain itu adapula permasalahan presisi pada bahasa *python*, sehingga untuk mengecek apakah suatu ekspresi bernilai 24 kita harus mengevaluasi hasilnya dan mencari selisih mutlaknya dengan 24, jika berada dalam batas *epsilon* (dalam program ini *epsilon* didefinisikan sebagai $24 - (8/(3 - \frac{8}{3}))$, dalam *python* ekspresi ini bernilai $\epsilon \approx 1 \times 10^{-14}$, nilai *epsilon* tersebut sudah cukup bagus sehingga input 8,3,8,3 memiliki solusi. Sebelumnya, nilai 8,3,8,3 dianggap tidak ada solusi pada program sebelum menggunakan *epsilon*. Hal ini dikarenakan solusi 8,3,8,3 yaitu $8/(3 - \frac{8}{3})$ bernilai 23.9999..., mendekati 24 tapi bukan 24. Sementara seharusnya ekspresi tersebut jika dievaluasi secara matematis oleh manusia hasilnya adalah tepat 24.

Dalam program ini, banyak kemungkinan peletakan kurung dalam ekspresi ada 5 jenis yaitu ((a b) c) d, (a b) (c d), (a (b c)) d, a ((b c) d), dan a (b (c d)). Dalam web yang diberikan pada spesifikasi tugas, jenis kurung yang digunakan hanya 4 jenis, sesuai dengan 5 jenis diatas tapi tanpa yang terakhir. Hal tersebut menyebabkan perbedaan banyak solusi antara web dengan program yang dibuat. Menurut saya sendiri, yang benar adalah variasi kurung yang berjumlah 5 jenis karena tanpa itu program tidak akan pernah menemukan solusi dari input 8,3,8,3.

BAB II

SOURCE CODE PROGRAM

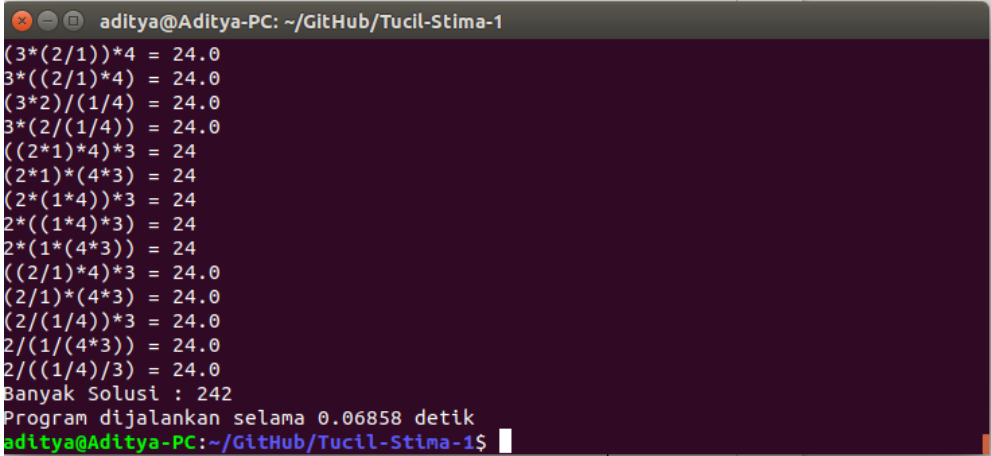
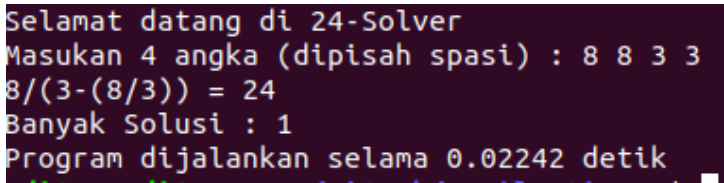
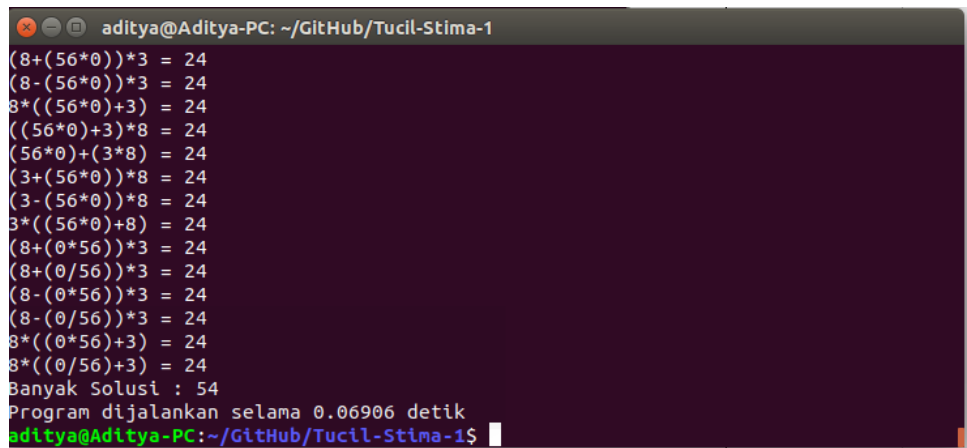
2.1 Source Code Python

import time

```
if(__name__ == "__main__"):
    print('Selamat datang di 24-Solver')
    inp = [int(i) for i in input('Masukan 4 angka (dipisah spasi) : ').split(' ')]
    start = time.time()
    permutasiAngka = [[inp[a], inp[b], inp[c], inp[d]] for a in range(
        4) for b in range(4) for c in range(4) for d in range(4)
        if a!=b!=c!=d!=a!=c!=b!=d]
    #Hapus duplikat dengan mengubahnya menjadi set/himpunan lalu menjadi list lagi
    permutasiAngka = set(tuple(tup) for tup in permutasiAngka)
    permutasiAngka = [list(li) for li in permutasiAngka]
    #Generate kemungkinan operator
    op = ['+', '-', '*', '/']
    permutasiOperator = [[op[a], op[b], op[c]] for a in range(4) for b in range(4)
        for c in range(4)]
    #Generate kemungkinan ekspresi beserta kurungnya
    cnt = 0
    epsilon = abs(24-(8/(3-(8/3))))
    for pA in permutasiAngka:
        a,b,c,d = pA
        for p0 in permutasiOperator:
            op1,op2,op3 = p0
            tupleFormatString = (a,op1,b,op2,c,op3,d)
            #Kemungkinan kurung
            '''
            ((a b) c) d
            (a b) (c d)
            (a (b c)) d
            a ((b c) d)
            a (b (c d))
            '''
            ekspresi = []
            ekspresi.append("(" + ("%d%c%d)%c%d" % tupleFormatString)
            ekspresi.append("(" + ("%d%c%d)%c(%d%c%d)" % tupleFormatString)
            ekspresi.append("(" + ("%d%c(%d%c%d))%c%d" % tupleFormatString)
            ekspresi.append("%d%c(" + ("%d%c%d)%c%d" % tupleFormatString)
            ekspresi.append("%d%c(%d%c(%d%c%d))" % tupleFormatString)
            for e in ekspresi:
                try:
                    hasil = eval(e)
                    if(abs(hasil-24)<=epsilon):
                        print(e+" = "+str(hasil))
                        cnt += 1
                except ZeroDivisionError:
                    pass
    print('Banyak Solusi : %d' % cnt)
    end = time.time()
    print('Program dijalankan selama %.5f detik' % (end-start))
```

BAB III

EKSPERIMEN

1	<p>Input : 1 2 3 4</p> <p>Output : 242 persamaan</p>  <pre>aditya@Aditya-PC: ~/GitHub/Tucil-Stima-1 (3*(2/1))*4 = 24.0 3*((2/1)*4) = 24.0 (3*2)/(1/4) = 24.0 3*(2/(1/4)) = 24.0 ((2*1)*4)*3 = 24 (2*1)*(4*3) = 24 (2*(1*4))*3 = 24 2*((1*4)*3) = 24 2*(1*(4*3)) = 24 ((2/1)*4)*3 = 24.0 (2/1)*(4*3) = 24.0 (2/(1/4))*3 = 24.0 2/(1/(4*3)) = 24.0 2/((1/4)/3) = 24.0 Banyak Solusi : 242 Program dijalankan selama 0.06858 detik aditya@Aditya-PC:~/GitHub/Tucil-Stima-1\$</pre>
2	<p>Input : 8 3 8 3</p> <p>Output : 1 persamaan</p>  <pre>Selamat datang di 24-Solver Masukan 4 angka (dipisah spasi) : 8 8 3 3 8/(3-(8/3)) = 24 Banyak Solusi : 1 Program dijalankan selama 0.02242 detik</pre>
3	<p>Input : 56 8 0 3</p> <p>Output : 54 persamaan</p>  <pre>aditya@Aditya-PC: ~/GitHub/Tucil-Stima-1 (8+(56*0))*3 = 24 (8-(56*0))*3 = 24 8*((56*0)+3) = 24 ((56*0)+3)*8 = 24 (56*0)+(3*8) = 24 (3+(56*0))*8 = 24 (3-(56*0))*8 = 24 3*((56*0)+8) = 24 (8+(0*56))*3 = 24 (8+(0/56))*3 = 24 (8-(0*56))*3 = 24 (8-(0/56))*3 = 24 8*((0*56)+3) = 24 8*((0/56)+3) = 24 Banyak Solusi : 54 Program dijalankan selama 0.06906 detik aditya@Aditya-PC:~/GitHub/Tucil-Stima-1\$</pre>

4	<p>Input : 4 4 4 4</p> <p>Output : 6 persamaan</p> <pre>Selamat datang di 24-Solver Masukan 4 angka (dipisah spasi) : 4 4 4 4 (4+4)+(4*4) = 24 4+(4+(4*4)) = 24 (4+(4*4))+4 = 24 4+((4*4)+4) = 24 ((4*4)+4)+4 = 24 (4*4)+(4+4) = 24 Banyak Solusi : 6 Program dijalankan selama 0.00532 detik</pre>
5	<p>Input : 123456 246912 21728388 987654</p> <p>Output : 2 persamaan</p> <pre>Selamat datang di 24-Solver Masukan 4 angka (dipisah spasi) : 123456 246912 21728388 987654 (21728388/987654)+(246912/123456) = 24 (246912/123456)+(21728388/987654) = 24 Banyak Solusi : 2 Program dijalankan selama 0.07112 detik</pre>

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

*Nb : Solusi benar, namun berbeda dengan aplikasi 24solver pada web spesifikasi tugas dikarenakan ada perbedaan variasi kurung