

LAPORAN TUGAS KECIL

Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

**Dibuat dalam rangka:
Tugas Kecil IF-2211 Strategi Algoritma**

**Oleh:
Fadel Ananda Dotty - 13519146**



**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2020**

I. Algoritma Brute Force

Persoalan yang akan diselesaikan dengan algoritma brute force adalah persoalan *cryptarithmic* yang merupakan puzzle penjumlahan dengan setiap huruf diganti angka yang berbeda. Dalam puzzle ini, diberikan penjumlahan huruf lalu tugas kita adalah mencari angka berbeda (dari 0-9) untuk merepresentasikan huruf tersebut sehingga penjumlahannya sesuai.

Contoh:

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Solusinya adalah:

$$\begin{array}{r} 9 5 6 7 \\ + 1 0 8 5 \\ \hline 1 0 6 5 2 \end{array}$$

Jadi, S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2

Langkah-langkah algoritma program :

1. Membaca file eksternal yang memiliki format .txt yang berisi soal cryptarithmic
2. Memisahkan antara operand dengan hasil dengan menandai indeks tanda operasi
3. Memasukkan operand kedalam bentuk array
4. Untuk array operand dan hasil (diakses menggunakan indeks operasi+1) kedalam array dan kemudian mengubahnya ke dalam bentuk set dengan modul bawaan python sehingga didapatkan karakter unik
5. Mengubah set yang didapat ke dalam bentuk array sehingga didapat array karakter unik
6. Apabila jumlah karakter unik lebih dari 10, program menampilkan pesan error dan program akan keluar
7. Apabila jumlah karakter unik kurang dari sama dengan 10, memasuki loop yang akan keluar apabila solusi sudah ketemu
 - a. Dalam loop, akan memulai timer program
 - b. Mengiterasi setiap baris dan karakter pada operand dan mengiterasi array karakter unik dan kemudian dicocokkan untuk mencari jumlah dari operand
 - c. Mengiterasi setiap karakter dari result dan mengiterasi array karakter unik dan kedua variabel ini dicocokkan untuk mencari jumlah result

- d. Mencocokkan jumlah operand dan jumlah result, apabila sama, maka melakukan iterasi untuk mengecek apakah awalnya tidak nol dan semua key value nya unik. Apabila keduanya terpenuhi, akan keluar dari loop while menuju proses selanjutnya. Apabila tidak, akan mengiterasi array key value dan menambah setiap angka di dalamnya sampai solusi ketemu.
- 8. Apabila solusi ketemu, maka akan menampilkan soal, solusi, karakter unik, jumlah testing, dan waktu testing

II. Source program

```
# Nama      : Fadel Ananda Dotty
# NIM       : 13519146

import time

name = input("Masukkan nama file: ")

tesCount = 0
f = open("../test/" + name, "r")

text=f.readlines()

stripIndex = 0 #mencari index untuk memisahkan operand

for i in range(len(text)): #mencari index berisi strip
    text[i]=text[i].strip()
    if (text[i][0]=="-"):
        stripIndex = i

a = []
for i in range(len(text)): #memasukkan semua operand ke dalam array
    if i<stripIndex:
        a.append(text[i])

a.append(text[stripIndex+1]) #memasukkan result ke dalam array

uniqueCharSet = set("".join(a)) #mengambil karakter unik di dalam array
dan merubahnya ke bentuk set, waktu eksekusi program bergantung urutan
set yang dibuat

uniqueChar = []
for letter in uniqueCharSet:
    uniqueChar.append(letter) #membuat array karakter unik

keyValue = []
for i in range(len(uniqueChar)):
    keyValue.append(0) #membuat array value yang berkorespondensi dengan
```

```

array karakter unik dan menginisialisasi dengan 0

result = text[stripIndex+1] #memasukkan result ke dalam variabel
intResult = []
intOperand = []

print("Counting...")
if (len(uniqueChar) > 10): #apabila jumlah karakter unik lebih dari 10,
    ditampilkan pesan error
    print("Jumlah karakter unik tidak boleh melebihi 10")
    exit()
else:
    isNotFound = True
    start = time.time() #start timer
    while isNotFound: #looping selama solusi belum ketemu
        sumOperand = 0
        sumRes = 0
        for i in range(len(text)): #proses untuk mengevaluasi jumlah
            nilai operand
            if i < stripIndex:
                for k in range(len(text[i])):
                    for j in range(len(uniqueChar)):
                        if(text[i][k] == uniqueChar[j]):
                            intOperand.append(keyValue[j])
                strOperand = [str(integer) for integer in intOperand]
                str1 = "".join(strOperand)
                sumO = int(str1)
                sumOperand = sumOperand + sumO
                intOperand = []

            for i in range(len(result)): #proses untuk mengevaluasi jumlah
                nilai result
                for j in range(len(uniqueChar)):
                    if(result[i] == uniqueChar[j]):
                        intResult.append(keyValue[j])

                strRes = [str(integer) for integer in intResult]
                str2 = "".join(strRes)
                sumRes = int(str2)

```

```

intResult = []

isUnique = True
isNotZero = True
if (sumOperand == sumRes): #apabila jumlah operand sama dengan
jumlah hasil
    for i in range(len(text)): #mengevaluasi apakah awalan huruf
berupa 0
        for j in range(len(uniqueChar)):
            if (text[i][0] == uniqueChar[j] and keyValue[j]==0):
                isNotZero = False

        for i in range(len(keyValue)): #mengevaluasi apakah key
value semuanya unik
            for j in range(i+1, len(keyValue)):
                if (keyValue[i] == keyValue[j]):
                    isUnique = False

        if (isUnique and isNotZero): #keluar dari loop apabila
solusi ketemu, unik, dan awalan tidak 0
            isNotFound = False
        else:
            for i in range(len(keyValue)): #proses mengubah key
value satu persatu
                if (keyValue[i] < 9 and keyValue[i] >= 0):
                    keyValue[i] += 1
                    break
                elif (keyValue[i] == 9):
                    keyValue[i] = 0
            else:
                for i in range(len(keyValue)): #proses mengubah key value
satu persatu
                    if (keyValue[i] < 9 and keyValue[i] >= 0):
                        keyValue[i] += 1
                        break
                    elif (keyValue[i] == 9):
                        keyValue[i] = 0
        tesCount += 1 #menambah tes count

```

```

end = time.time() #end time

#prosedur print hasil output
print("---SOAL---")
for i in range(len(text)):
    print(text[i].strip())
print("*****")
print("Karakter unik dalam soal \t: ")
print(uniqueChar)
print("*****")
print("Tes Count \t= ", end="")
print(tesCount)
print("*****")
print("Execution Time \t= ", end="")
print(end - start, end="")
print(" Detik")
print("*****")
print("Solusi \t:")
for i in range(len(keyValue)):
    print(uniqueChar[i], end=" ")
    print("=", end=" ")
    print(keyValue[i], end="")
    if (i != len(keyValue) - 1):
        print(",", end=" ")

print(".")
print("*****")
print("SOLUSI NUMERIK")
for i in range(len(text)):
    if i < stripIndex:
        for k in range(len(text[i])):
            for j in range(len(uniqueChar)):
                if (text[i][k] == uniqueChar[j]):
                    intOperand.append(keyValue[j])
strOperand = [str(integer) for integer in intOperand]
str1 = "".join(strOperand)
sumO = int(str1)
print(sumO)

```

```
intOperand = []  
  
print("----- +")  
print(sumRes)  
f.close()
```


III. Screenshot Input Output

```
(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
---SOAL---
NO
GUN
NO
----- +
HUNT
*****
Karakter unik dalam soal      :
['T', 'G', 'N', 'U', 'O', 'H']
*****
Tes Count      = 170893
*****
Execution Time = 7.8288304805755615 Detik
*****
Solusi :
T = 2, G = 9, N = 8, U = 0, O = 7, H = 1.
*****
SOLUSI NUMERIK
87
908
87
----- +
1082
```

Case 1

```
(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
USA
USSR
----- +
PEACE
*****
Karakter unik dalam soal      :
['S', 'A', 'C', 'E', 'R', 'U', 'P']
*****
Tes Count      = 1980724
*****
Execution Time = 119.13662075996399 Detik
*****
Solusi :
S = 3, A = 2, C = 7, E = 0, R = 8, U = 9, P = 1.
*****
SOLUSI NUMERIK
932
9338
----- +
10270
```

Case 2

```

Fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/Strategi Algoritma$ python3 tucil.py
---SOAL---
NUMBER
NUMBER
----- +
PUZZLE
*****
Karakter unik dalam soal      :
['P', 'E', 'B', 'L', 'U', 'M', 'R', 'Z', 'N']
*****
Tes Count      = 239107685
*****
Execution Time = 16241.5062084198 Detik
*****
Solusi :
P = 4, E = 8, B = 6, L = 7, U = 0, M = 1, R = 9, Z = 3, N = 2.
*****
SOLUSI NUMERIK
201689
201689
----- +
403378

```

Case 3

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
MEMO
FROM
----- +
HOMER
*****
Karakter unik dalam soal      :
['E', 'R', 'M', 'H', 'F', 'O']
*****
Tes Count      = 571835
*****
Execution Time = 23.640774965286255 Detik
*****
Solusi :
E = 4, R = 3, M = 8, H = 1, F = 7, O = 5.
*****
SOLUSI NUMERIK
8485
7358
----- +
15843

```

Case 4

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
COCA
COLA
----- +
OASIS
*****
Karakter unik dalam soal      :
['O', 'S', 'L', 'A', 'I', 'C']
*****
Tes Count      = 896022
*****
Execution Time = 41.32993292808533 Detik
*****
Solusi :
O = 1, S = 2, L = 0, A = 6, I = 9, C = 8.
*****
SOLUSI NUMERIK
8186
8106
----- +
16292

```

Case 5

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
LOGIC
LOGIC
----- +
PROLOG
*****
Karakter unik dalam soal      :
['R', 'C', 'I', 'L', 'P', 'G', 'O']
*****
Tes Count      = 419529
*****
Execution Time = 24.719988584518433 Detik
*****
Solusi :
R = 8, C = 2, I = 5, L = 9, P = 1, G = 4, O = 0.
*****
SOLUSI NUMERIK
90452
90452
----- +
180904

```

Case 6

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
MOST
MOST
----- +
TOKYO
*****
Karakter unik dalam soal      :
['O', 'S', 'K', 'Y', 'M', 'T']
*****
Tes Count      = 164573
*****
Execution Time = 8.95913577079773 Detik
*****

Solusi :
O = 2, S = 7, K = 5, Y = 4, M = 6, T = 1.
*****
SOLUSI NUMERIK
6271
6271
----- +
12542

```

Case 7

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
EAT
THAT
----- +
APPLE
*****
Karakter unik dalam soal      :
['E', 'H', 'A', 'P', 'T', 'L']
*****
Tes Count      = 390129
*****
Execution Time = 15.64909291267395 Detik
*****

Solusi :
E = 8, H = 2, A = 1, P = 0, T = 9, L = 3.
*****
SOLUSI NUMERIK
819
9219
----- +
10038

```

Case 8

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
HERE
SHE
----- +
COMES
*****
Karakter unik dalam soal      :
['O', 'M', 'C', 'R', 'H', 'S', 'E']
*****
Tes Count      = 4895131
*****
Execution Time = 267.41732382774353 Detik
*****
Solusi :
O = 0, M = 3, C = 1, R = 5, H = 9, S = 8, E = 4.
*****
SOLUSI NUMERIK
9454
894
----- +
10348

```

Case 9

```

(base) H:\ITB Semester 4\Strategi Algoritma>python tucil.py
Counting...
---SOAL---
CROSS
ROADS
----- +
DANGER
*****
Karakter unik dalam soal      :
['R', 'D', 'C', 'N', 'G', 'E', 'O', 'S', 'A']
*****
Tes Count      = 532478917
*****
Execution Time = 32219.083333730698 Detik
*****
Solusi :
R = 6, D = 1, C = 9, N = 8, G = 7, E = 4, O = 2, S = 3, A = 5.
*****
SOLUSI NUMERIK
96233
62513
----- +
158746

```

Case 10

```

fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/New folder/bin$ python3 tucil.py
Masukkan nama file: case1.txt
Counting...
---SOAL---
THREE
THREE
TWO
TWO
ONE
----- +
ELEVEN
*****
Karakter unik dalam soal      :
['W', 'E', 'R', 'V', 'H', 'T', 'L', 'N', 'O']
*****
Tes Count      = 397842611
*****
Execution Time = 21651.0383810997 Detik
*****
Solusi :
W = 0, E = 1, R = 6, V = 2, H = 4, T = 8, L = 7, N = 9, O = 3.
*****
SOLUSI NUMERIK
84611
84611
803
803
391
----- +
171219

```

Case 11

IV. Alamat github

<https://github.com/fadelananda/tucil1.git>

V. Checklist

| Poin | Ya | Tidak |
|--|----|-------|
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | V | |
| 2. Program berhasil running | V | |
| 3. Program dapat membaca file masukan dan menuliskan luaran. | V | |
| 4. Solusi cryptarithmic hanya benar untuk persoalan cryptarihtmetic dengan dua buah operand. | | V |
| 5. Solusi cryptarithmic benar untuk persoalan cryptarihtmetic untuk lebih dari dua buah operand. | V | |