

LAPORAN TUGAS KECIL 1 - CRYPTARITHMETIC

NAMA: Leonard Matheus

NIM : 13519215

Strategi Algoritma

Kelas : K-4

ALGORITMA BRUTE FORCE

1. Dari operan yang ada, setiap huruf yang ada disubstitusikan dengan angka dari 0 hingga 9
2. Setiap Test Case kurang dari 10 huruf, apabila lebih dari 10 huruf tidak akan diterima
3. Setiap 1 huruf disubstitusi dengan 1 angka, apabila salah, akan diulangi dengan dictionary kombinasi huruf-angka yang lain sampai ketemu.
4. Apabila huruf di depan operan atau hasil disubstitusi dengan angka 0, percobaan akan dilompati (skip)
5. Semua penjumlahan operan akan dicocokkan dengan substitusi hasilnya, apabila sama, akan muncul ke layar.

SOURCE CODE (Python)

```
#TUGAS KECIL 1 - CRYPTARITHMETIC
#NAMA: Leonard Matheus
#NIM : 13519215
#Strategi Algoritma
#Kelas : K-4

import sys
import time
import os
from pathlib import Path

path = os.path.dirname(Path(__file__).absolute().parent)

def find_letter(let, lst):
    #mengecek apakah ada huruf yang sama di dalam array sebelum dimasukkan
    return (lst or False) and \
        ((isinstance(lst[0], str) and let in lst[0]) or find_letter(let,
lst[1:]))

def masukkanInput(namafile):
    f = open(namafile, "r") #membuka file
    for x in f:
        x = x.replace("\n", "") #membersihkan tanda enter
        print(x)
```

```

def bacafile(namafile):
    data = [] #list kosong untuk input
    huruf = [] #list kosong untuk hurufnya
    f = open(namafile,"r")
    for x in f:
        x = x.replace("\n", "") #membersihkan tanda enter
        print(x)
        x = x.replace("-", "") #membersihkan tanda -----
        x = x.replace("+", "") #membersihkan tanda +
        x = x.replace(" ", "") #membersihkan tanda spasi
        data.append(x)
        for y in x:
            if (find_letter(str(y), huruf)==False) and (y.isalpha()==True):
                #input huruf unik saja ke list huruf
                huruf.append(y)
    data = list(filter(None, data)) #membuang elemen kosong
    f.close()
    return data,huruf

def permutasi(lst, n):
    #Cara Pakai: permutasi([x for x in <listnya>], n)
    if n == 0:
        return [[]] #bila 0, return list kosong
    l = []
    for i in range(0, len(lst)):
        m = lst[i]
        temp = lst[:i] + lst[i+1:]
        for p in permutasi(temp, n-1): #rekursif
            l.append([m]+p)
    li = [[int(j) for j in i] for i in l] #konversi tipe data non-type ke
integer
    return li

def katangka(kata, coba):
    #Mengubah kata ke angka dalam bentuk integer
    return int(''.join(str(coba[huruf]) for huruf in kata))

def solusi(data, huruf):
    angka = "0123456789" #digit angka yang mungkin
    data = [kata.upper() for kata in data] #membuat semua huruf jadi huruf
besar
    jawab = [] #untuk menampung jawaban angka yang cocok
    percobaan = 0 #untuk menghitung proses
    if len(huruf) > 10:

```

```

        print("Huruf kebanyakan, maksimal 10 ya!")
    else:
        for i in permutasi(angka,len(huruf)):
            coba = dict(zip(huruf, i)) #kombinasi dictionary huruf dan pasangan
            angka yang benar
            if all((coba[kata[0]] > 0) for kata in data): #membatasi huruf awal
            tidak boleh 0
                percobaan += 1 #menghitung jumlah percobaannya
                bilangan = [katangka(kata, coba) for kata in data] #memngubah
            operan -> kata jadi integer
                if sum(bilangan[:len(data)-1]) == bilangan[len(data)-1]: #bila
            hasilnya cocok
                    jawab.append(bilangan)
        return jawab, percobaan

def tampilkanLayar(arr1):
    if (len(arr1)==0): #Jika panjang arraynya 0, tidak ada solusi ya kan
        print("TIDAK ADA SOLUSI YANG MUNGKIN")
    else:
        print("SOLUSI YANG MUNGKIN ADALAH:")
        for i in range(len(arr1)): #panjang operan + hasil
            for j in range(len(arr1[i])): #jumlah digit
                if (j == (len(arr1[i])-1)): #bila ternyata i= angka iterasi
            sebelum hasil, tambahkan "-----"
                    for k in range (len(str(arr1[i][j]))+3):
                        print("-",end="")
                    print("")
                    print(str(arr1[i][j]))
                    if (j == (len(arr1[i]))-2): #tambahkan " "
                        for m in range
            (len(str(arr1[i][-1]))-(len(str(arr1[i][j])))):
                            print(" ",end="")
                            print (str(arr1[i][j]),"+")
                    elif ((j != (len(arr1[i])-1)) and (j != (len(arr1[i]))-2)) :
                        for m in range
            (len(str(arr1[i][-1]))-(len(str(arr1[i][j])))):
                            print(" ",end="")
                            print(str(arr1[i][j]))
            print("")

def converttxt(namafile, arr1, waktu, percobaan, path): #ubah ke file txt
    sys.stdout = open(os.path.join(path , "test" ,"hasil.txt"), "w")
    masukkanInput(namafile)
    print("")

```

```

    tampilkanLayar(arr1)
    print("Waktu yang dibutuhkan:", waktu, "detik")
    print("Total Percobaan Cryptarithhmetic:", percobaan, "kali")
    sys.stdout.close()

#Program Utama
nama = "input1.txt" #nama file <SILAHKAN UBAH NAMA FILE DI SINI>
namafile = os.path.join(path , "test" , nama)
data,huruf = bacafile(namafile) #membaca input file txt
print("")
awal = time.time() #waktu mulai
arr1, percobaan = solusi(data, huruf)
tampilkanLayar(arr1)
akhir = time.time() #waktu akhir
waktu = round(float(akhir-awal), 3)
print("Waktu yang dibutuhkan:", waktu, "detik")
print("Total Percobaan Cryptarithhmetic:", percobaan, "kali")
converttxt(namafile,arr1,waktu,percobaan,path) #menyalin ke file txt hasil

```

Penjelasan Fungsi

1. Boolean find_letter (input huruf dan array)
Fungsi ini berguna untuk mengecek apakah suatu huruf sudah terdaftar pada array atau tidak.
2. masukkanInput (input namafile)
Fungsi ini berguna untuk membaca input pada file terpisah sehingga pengguna bebas menaruh kombinasi
3. Bacafile (input namafile)
Fungsi ini terlihat sama dengan fungsi masukkanInput tetapi sesungguhnya berbeda. Fungsi ini berguna untuk mengolah input operan yang ada pada file kemudian mengolahnya sehingga menghasilkan output berupa array data dan huruf.
 - Array data adalah array yang berisi semua huruf dalam operan yang harus disubstitusi
 - Array huruf adalah set array yang berisi elemen huruf-huruf yang unik.
4. Permutasi (input list dan jumlah angka yang diacak)
Fungsi ini membangkitkan permutasi sehingga bisa menghasilkan array yang berisi permutasi elemen dari array yang ada di dalamnya.
5. Katangka (input array kata dan array coba)
Fungsi ini berguna untuk mengkonversi huruf menjadi angka supaya bisa dioperasikan, menghasilkan output berupa integer
6. Solusi(input array data dan data huruf)

Fungsi yang utama untuk menghasilkan jawaban berupa array bilangan (berupa operan dan hasil). Dengan menggunakan algoritma brute force, semua kombinasi angka digunakan untuk bisa mendapatkan hasil yang persis seperti susunan hurufnya.

Untuk jumlah jenis huruf yang lebih dari 10, program tidak akan memprosesnya

Program ini menggunakan dictionary sehingga dapat mendaftarkan setiap pasangan huruf dengan angka yang bersesuaian sehingga lebih efisien untuk dicoba semua pasangannya.

Program ini mengecek apakah huruf awal dari tiap operan atau hasil sehingga nantinya tidak merupakan angka 0.

7. Tampilkan layar(input berupa array jawaban yang akan ditampilkan)

Fungsi ini berguna untuk mengolah operan dan hasil sehingga dapat menampilkan jawaban ke layar dalam formasi yang sudah rapih.

8. converttxt(input nama file, array file, waktu, dan jumlah percobaan)

Fungsi ini berguna untuk menghasilkan jawaban dalam bentuk txt sehingga bisa disimpan dengan baik pada file text I/O

Saya menggunakan library time untuk mengukur waktu dan library sys untuk menyimpan file dalam bentuk I/O

HASIL KOMPILASI DAN TESTING

Hasil Kompilasi uji dapat dilihat pada gambar di bawah ini:

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
COCA
COLA+
-----
OASIS

SOLUSI YANG MUNGKIN ADALAH:
 8186
8106 +
-----
16292

Waktu yang dibutuhkan: 1.815 detik
Total Percobaan Cryptarithhmetic: 120960 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
NUMBER
NUMBER+
-----
PUZZLE

SOLUSI YANG MUNGKIN ADALAH:
201689
201689 +
-----
403378

Waktu yang dibutuhkan: 74.704 detik
Total Percobaan Cryptarithhmetic: 2903040 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
TILES
PUZZLES+
-----
PICTURE

SOLUSI YANG MUNGKIN ADALAH:
 91542
3077542 +
-----
3169084

Waktu yang dibutuhkan: 80.977 detik
Total Percobaan Cryptarithhmetic: 2903040 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
HERE
SHE+
-----
COMES

SOLUSI YANG MUNGKIN ADALAH:
 9454
 894 +
-----
10348

Waktu yang dibutuhkan: 8.639 detik
Total Percobaan Cryptarithhmetic: 423360 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
CLOCK
TICK
TOCK+
-----
PLANET

SOLUSI YANG MUNGKIN ADALAH:
90892
6592
6892 +
-----
104376

Waktu yang dibutuhkan: 76.742 detik
Total Percobaan Cryptarithhmetic: 2540160 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
DOUBLE
DOUBLE
TOIL+
-----
TROUBLE

SOLUSI YANG MUNGKIN ADALAH:
798064
798064
1936 +
-----
1598064

Waktu yang dibutuhkan: 80.345 detik
Total Percobaan Cryptarithhmetic: 2903040 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
NO
GUN
NO+
-----
HUNT

SOLUSI YANG MUNGKIN ADALAH:
87
908
87 +
-----
1082

Waktu yang dibutuhkan: 2.068 detik
Total Percobaan Cryptarithhmetic: 105840 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
SEND
MORE+
-----
MONEY

SOLUSI YANG MUNGKIN ADALAH:
9567
1085 +
-----
10652

Waktu yang dibutuhkan: 34.142 detik
Total Percobaan Cryptarithhmetic: 1451520 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
MEMO
FROM+
-----
HOMER

SOLUSI YANG MUNGKIN ADALAH:
8485
7358 +
-----
15843

Waktu yang dibutuhkan: 2.105 detik
Total Percobaan Cryptarithhmetic: 105840 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
FORTY
TEN
TEN+
-----
SIXTY

SOLUSI YANG MUNGKIN ADALAH:
29786
850
850 +
-----
31486

Waktu yang dibutuhkan: 227.684 detik
Total Percobaan Cryptarithhmetic: 2540160 kali
```

```
WHO
YOU+
-----
ARE

SOLUSI YANG MUNGKIN ADALAH:
103
539 +
-----
642

103
639 +
-----
742
```

```
842
125 +
-----
967

843
132 +
-----
975

852
124 +
-----
976

Waktu yang dibutuhkan: 78.285 detik
Total Percobaan Cryptarithhmetic: 1270080 kali
```

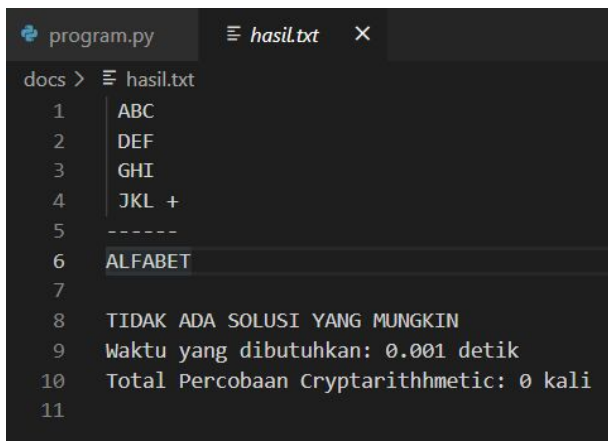
```
[Running] python -u "d:\git\Cryptarithmetic\src\program.py"
SOD
IS+
-----
SOLD

TIDAK ADA SOLUSI YANG MUNGKIN
Waktu yang dibutuhkan: 0.448 detik
Total Percobaan Cryptarithhmetic: 24192 kali
```

```
[Running] python -u "d:\git\Cryptarithmic\src\program.py"
ABC
DEF
GHI
JKL +
-----
ALFABET

Huruf kebanyakan, maksimal 10 ya!
TIDAK ADA SOLUSI YANG MUNGKIN
Waktu yang dibutuhkan: 0.001 detik
Total Percobaan Cryptarithhmetic: 0 kali
```

Output terminal juga berhasil disimpan di hasil.txt



```
docs > hasil.txt
1  ABC
2  DEF
3  GHI
4  JKL +
5  -----
6  ALFABET
7
8  TIDAK ADA SOLUSI YANG MUNGKIN
9  Waktu yang dibutuhkan: 0.001 detik
10 Total Percobaan Cryptarithhmetic: 0 kali
11
```

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 cryptarithmetic benar untuk persoalan cryptarihtmetic dengan dua buah operand saja.		V
5. Solusi cryptarithmetic benar untuk persoalan cryptarihtmetic untuk lebih dari dua buah operand.	V	

LETAK DRIVE UNTUK DIUJI COBA

<https://github.com/leomatt547/Cryptarithmic>