

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)

```
1  #TUGAS KECIL 1 - CRYPTARITHMETIC
2  #NAMA: Leonard Matheus
3  #NIM : 13519215
4  #Strategi Algoritma
5  #Kelas : K-4
6
7  import sys
8  import time
9  import os
10 from pathlib import Path
11
12 path = os.path.dirname(Path(__file__).absolute().parent)
13
14 def find_letter(let, lst):
15     #mengecek apakah ada huruf yang sama di dalam array sebelum dimasukkan
16     return (lst or False) and \
17         ((isinstance(lst[0], str) and let in lst[0]) or find_letter(let, lst[1:]))
18
19 def masukkanInput(namafile):
20     f = open(namafile, "r") #membuka file
21     for x in f:
22         x = x.replace("\n", "") #membersihkan tanda enter
23         print(x)
24
```

```

25 def bacafile(namafile):
26     data = [] #list kosong untuk input
27     huruf = [] #list kosong untuk hurufnya
28     f = open(namafile,"r")
29     for x in f:
30         x = x.replace("\n", "") #membersihkan tanda enter
31         print(x)
32         x = x.replace("-", "") #membersihkan tanda -----
33         x = x.replace("+", "") #membersihkan tanda +
34         x = x.replace(" ", "") #membersihkan tanda spasi
35         data.append(x)
36         for y in x:
37             if (find_letter(str(y), huruf)==False) and (y.isalpha()==True):
38                 #input huruf unik saja ke list huruf
39                 huruf.append(y)
40     data = list(filter(None, data)) #membuang elemen kosong
41     f.close()
42     return data,huruf
43
44 def permutasi(lst, n):
45     #Cara Pakai: permutasi([x for x in <listnya>], n)
46     if n == 0:
47         return [[]] #bila 0, return list kosong
48     l = []
49     for i in range(0, len(lst)):
50         m = lst[i]
51         temp = lst[:i] + lst[i+1:]
52         for p in permutasi(temp, n-1): #rekursif
53             l.append([m]+p)
54     li = [[int(j) for j in i] for i in l] #konversi tipe data non-type ke integer
55     return li
56
57 def katangka(kata, coba):
58     #Mengubah kata ke angka dalam bentuk integer
59     return int(''.join(str(coba[huruf]) for huruf in kata))

```

```

61 def solusi(data, huruf):
62     angka = "0123456789" #digit angka yang mungkin
63     data = [kata.upper() for kata in data] #membuat semua huruf jadi huruf besar
64     jawab = [] #untuk menampung jawaban angka yang cocok
65     percobaan = 0 #untuk menghitung proses
66     if len(huruf) > 10:
67         print("Huruf kebanyakan, maksimal 10 ya!")
68     else:
69         for i in permutasi(angka,len(huruf)):
70             coba = dict(zip(huruf, i)) #kombinasi dictionary huruf dan pasangan angka yang benar
71             if all((coba[kata[0]] > 0) for kata in data): #membatasi huruf awal tidak boleh 0
72                 percobaan += 1 #menghitung jumlah percobaannya
73                 bilangan = [katangka(kata, coba) for kata in data] #memngubah operan -> kata jadi integer
74                 if sum(bilangan[:len(data)-1]) == bilangan[len(data)-1]: #bila hasilnya cocok
75                     jawab.append(bilangan)
76     return jawab, percobaan

```

```

78 def tampilkanLayar(arr1):
79     if (len(arr1)==0): #jika panjang arraynya 0, tidak ada solusi ya kan
80         print("TIDAK ADA SOLUSI YANG MUNGKIN")
81     else:
82         print("SOLUSI YANG MUNGKIN ADALAH:")
83         for i in range(len(arr1)): #panjang operan + hasil
84             for j in range(len(arr1[i])): #jumlah digit
85                 if (j == (len(arr1[i])-1)): #bila ternyata i= angka iterasi sebelum hasil, tambahkan "-----"
86                     for k in range (len(str(arr1[i][j]))+3):
87                         print("-",end="")
88                     print("")
89                     print(str(arr1[i][j]))
90                 if (j == (len(arr1[i]))-2): #tambahkan " "
91                     for m in range (len(str(arr1[i][-1]))-(len(str(arr1[i][j])))):
92                         print(" ",end="")
93                     print (str(arr1[i][j]),"+")
94                 elif ((j != (len(arr1[i])-1)) and (j != (len(arr1[i]))-2)) :
95                     for m in range (len(str(arr1[i][-1]))-(len(str(arr1[i][j])))):
96                         print(" ",end="")
97                     print(str(arr1[i][j]))
98             print("")
99
100 def converttxt(namafile, arr1, waktu, percobaan, path): #ubah ke file txt
101     sys.stdout = open(os.path.join(path , "test" ,"hasil.txt"), "w")
102     masukkanInput(namafile)
103     print("")
104     tampilkanLayar(arr1)
105     print("Waktu yang dibutuhkan:", waktu, "detik")
106     print("Total Percobaan Cryptarithmetic:", percobaan, "kali")
107     sys.stdout.close()

```

```

109 #Program Utama
110 nama = "input1.txt" #nama file <SILAHKAN UBAH NAMA FILE DI SINI>
111 namafile = os.path.join(path , "test" , nama)
112 data,huruf = bacafile(namafile) #membaca input file txt
113 print("")
114 awal = time.time() #waktu mulai
115 arr1, percobaan = solusi(data, huruf)
116 tampilkanLayar(arr1)
117 akhir = time.time() #waktu akhir
118 waktu = round(float(akhir-awal), 3)
119 print("Waktu yang dibutuhkan:", waktu, "detik")
120 print("Total Percobaan Cryptarithhmetic:", percobaan, "kali")
121 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, library sys untuk menyimpan file dalam bentuk I/O, dan library os untuk merujuk pada alamat file berada.

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 Cryptarithmic: 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 Cryptarithmic: 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
```

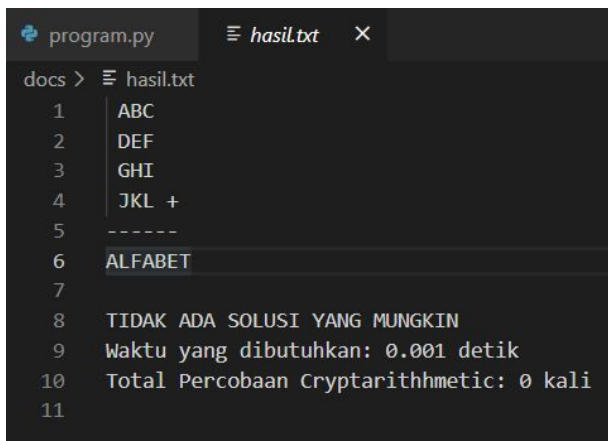
```
[Running] python -u "d:\git\Cryptarithmetic\src\program.py"
SOD
IS+
-----
SOLD

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 hanya 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>