

LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
PENYELESAIAN *CRYPTARITHMETIC* DENGAN ALGORITMA BRUTE FORCE

Disusun dalam rangka memenuhi tugas Strategi Algoritma (IF2211)



Disusun oleh :

Muhammad Rayhan Ravianda (13519201)

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2020

BAB I

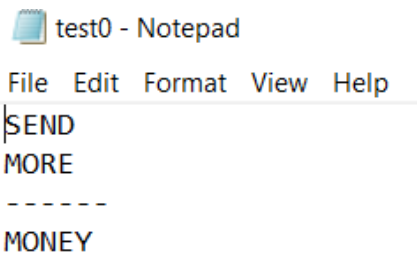
PENJELASAN ALGORITMA

Di dalam program ini digunakan algoritma *brute force* untuk menyelesaikan persoalan *cryptarithmic*. Urutan langkah-langkah yang dilakukan oleh program dalam menemukan solusi dari persoalan-persoalan yang diberikan secara sederhana dapat dijelaskan sebagai berikut,

1. Program menerima input berupa string operand dan hasil operasi dari file eksternal.

Contoh :

Didapatkan file eksternal sebagai berikut,



```
test0 - Notepad
File Edit Format View Help
SEND
MORE
-----
MONEY
```

Maka program akan menerima string ["SEND", "MORE", "MONEY"].

2. Setiap huruf pada string yang diterima akan dimasukkan ke dalam array of character berisi huruf-huruf unik.

Contoh :

Dari string yang program terima yaitu ["SEND", "MORE", "MONEY"], program akan membuat array of character ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y'].

3. Program akan menghitung jumlah elemen dari array of character dan membuat array of integer dengan jumlah elemen yang sama.

Contoh :

Jumlah elemen pada array of character ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y'] adalah 8, maka program akan membuat array of integer dengan ukuran 8 elemen.

4. Program akan melakukan permutasi pada array of integer dengan batas-batas angka yang mungkin adalah 0 sampai 9. Setiap kombinasi angka dari array of integer akan dipetakan ke array of character selama proses permutasi dijalankan.
5. Kemudian program akan mengkonversi elemen-elemen pada array of integer menjadi string agar dapat disusun menjadi sebuah bilangan. Bilangan tersebut lalu dikonversi

kembali menjadi integer dan dihitung apakah hasil operasinya sesuai dengan penjumlahan operand-operand.

Contoh :

Array of character ['S', 'E', 'N', 'D'] memiliki salah satu hasil kombinasi dan permutasi yang sudah dipetakan ke array of integer [9,5,6,7]. Kemudian elemen-elemen array tersebut dikonversikan dan disusun menjadi 9567.

6. Jika sesuai maka solusi tersebut akan ditampilkan ke layar dan menjadi output.

Contoh :

```
Solusi :
9567
1085
-----+
10652
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 1393690
```

7. Kemudian program akan mengulangi langkah ke-3 sampai ke-6 hingga proses permutasi selesai, dan menampilkan waktu program menjalankan semua proses yang dilakukan.

Contoh :

```
Total waktu yang dibutuhkan untuk menemukan solusi : 15.52 detik
```

BAB II

SOURCE CODE PROGRAM

Pada bab ini akan ditampilkan hasil tangkapan layar berisi source code program, yaitu sebagai berikut,

```
crypt.py
D: > githubStima > src > crypt.py > ...
1  # Muhammad Rayhan Ravianda
2  # 13519201
3  # K-04
4
5  # Tugas Kecil 1 IF2211 Strategi Algoritma
6  # Penyelesaian Cryptarithmic dengan Algoritma Brute Force
7
8  # File : crypt.py
9
10
11 # ----- FUNGSI TAMBAHAN ----- #
12
13
14
15 # Module yang digunakan
16 import re
17 import time
18
19 # Fungsi 'maks' untuk mencari jumlah digit dari operand yang akan dijadikan batas maksimal untuk digit hasil
20 def maks(A) :
21     maksA = len(A[0])
22     for i in range(1, len(A)) :
23         if maksA <= len(A[i]) :
24             maksA = len(A[i])
25     return maksA
26
27 # Fungsi 'printSoal' untuk mencetak Persoalan yang ingin dicari solusinya
28 def printSoal(A) :
29     empty = " "
30     diff = 0
31     for i in range(len(A)) :
32         diff = len(hasilList) - len(A[i])
33         space = (empty * diff) + A[i]
34         print(space)
35     div = (len(hasilList) * "-") + "+"
36     print(div)
37     print(hasilList)
38
```

crypt.py

D: > githubStima > src > crypt.py > ...

```
39 # Fungsi 'printSolusi' untuk mencetak Solusi yang sudah ditemukan
40 def printSolusi(A) :
41     empty = " "
42     diff = 0
43     for i in range(len(A)-1) :
44         diff = len(hasillist) - len(A[i])
45         space = (empty * diff) + A[i]
46         print(space)
47     div = (len(hasillist) * "-") + "+"
48     print(div)
49     print(A[len(A)-1])
50
51 # Fungsi 'solveCrypt' untuk menyelesaikan permasalahan cryptarithmic
52 def solveCrypt(letter, number, isVisit, word) :
53
54     # Mengambil variabel boolean 'isSolved'
55     global isSolved
56
57     # Basis
58     if len(hurufList) == len(number) :
59         map = {}
60         for n, m in zip(letter,number) :
61             map[n] = m
62         for i in range(len(word)) :
63             if map[word[i][0]] == 0 :
64                 return
65         solveCrypt.counter = solveCrypt.counter + 1
66
67         wordList = [0 for i in range(len(word))]
68         for i in range(len(word)) :
69             wordList[i] = ''
70         for i in range(len(word)) :
71             for j in word[i] :
72                 wordList[i] += str(map[j])
73         sum = 0
74         for i in range(len(wordList)-1) :
75             sum = sum + int(wordList[i])
76         if sum == int(wordList[len(wordList)-1]) :
```

crypt.py

D: > githubStima > src > crypt.py > ...

```
77         print("Solusi : ")
78         printSolusi(wordList)
79         print("Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : ",solveCrypt.counter,"\n \n")
80         isSolved = True
81         return
82
83     # Rekurens
84     for i in range(10) :
85         if not isVisit[i] :
86             isVisit[i] = True
87             number.append(i)
88             solveCrypt(letter,number,isVisit,word)
89             number.pop()
90             isVisit[i] = False
91
92     # ----- END of FUNGSI TAMBAHAN ----- #
93
94
95     # ----- PROGRAM UTAMA ----- #
96
97     # Input file
98     fileTest = input("Nama file : ")
99     folderTest = "../test/"
100     sourceTest = folderTest + fileTest
101     operandList = []
102
103     with open(sourceTest,'r') as files :
104         for line in files :
105             operandList += filter(None, re.split(r'\w|\d', line))
106
107     # Variabel boolean untuk mengetahui apakah percobaan sudah selesai
108     isSolved = False
109
110     # Memasukkan operand dan hasil kedalam list
111     hasilList = operandList[len(operandList)-1]
112     operandList.pop()
113     opList = operandList
114
```

crypt.py

D: > githubStima > src > crypt.py > ...

```
114
115 # Mengambil waktu saat memulai percobaan
116 start_time = time.perf_counter()
117
118 # Memasukan list per line kedalam list berisi setiap huruf
119 if len(hasillist) > (maks(opList)+1) :
120     print("\n")
121     print("Solusi tidak dapat ditemukan")
122 else :
123     hurufList = []
124     for i in range(len(opList)) :
125         for j in opList[i] :
126             if j not in hurufList :
127                 hurufList.append(j)
128     for n in hasillist :
129         if n not in hurufList :
130             hurufList.append(n)
131     if len(hasillist) > 10 :
132         print("\n")
133         print("Jumlah maksimal huruf dalam operand adalah 10 buah !")
134         exit()
135
136 # Memunculkan soal kelayar dengan fungsi 'printSoal'
137 print("\n")
138 print("Soal : ")
139 printSoal(opList)
140 print("\n")
141
142 # Memanggil fungsi 'solveCrypt' untuk menemukan solusi
143 opList.append(hasillist)
144 solveCrypt.counter = 0
145 solveCrypt(hurufList,[],[False for _ in range(10)],opList)
146
147 # Jika soal tidak dapat ditemukan solusinya
148 if not isSolved :
149     print("\n")
150     print("Solusi tidak dapat ditemukan")
151
```

crypt.py

D: > githubStima > src > crypt.py > ...

```
151
152 # Memunculkan total waktu yang digunakan untuk menyelesaikan percobaan
153 print("Total waktu yang dibutuhkan untuk menemukan solusi : ", "%.2f"%(time.perf_counter() - start_time), "detik")
154
155 # ----- END of PROGRAM UTAMA ----- #
```

BAB III

HASIL PERCOBAAN

Pada percobaan kali ini dibuat sebanyak 10 file eksternal untuk melakukan testing pada program. Hasil dari percobaan kesepuluh test adalah sebagai berikut,

1. test1.txt

Input :

```
NUMBER
NUMBER
-----
PUZZLE
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test1.txt

Soal :
NUMBER
NUMBER
-----+
PUZZLE

Solusi :
201689
201689
-----+
403378
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 325304

Total waktu yang dibutuhkan untuk menemukan solusi : 38.20 detik
```

2. test2.txt

Input :

```
TILES
PUZZLES
-----
PICTURE
```


Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test2.txt

Soal :
  TILES
  PUZZLES
-----+
  PICTURE

Solusi :
  91542
 3077542
-----+
 3169084
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 2640843

Total waktu yang dibutuhkan untuk menemukan solusi : 53.88 detik
```

3. test3.txt

Input :

```
CLOCK
TICK
TOCK
-----
PLANET
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test3.txt

Soal :
  CLOCK
  TICK
  TOCK
-----+
  PLANET

Solusi :
  90892
  6592
  6892
-----+
104376
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 2294475

Total waktu yang dibutuhkan untuk menemukan solusi : 40.05 detik
```

4. test4.txt

Input :

```
COCA
COLA
-----
OASIS
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test4.txt

Soal :
COCA
COLA
-----+
OASIS

Solusi :
8186
8106
-----+
16292
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 95135

Total waktu yang dibutuhkan untuk menemukan solusi : 1.33 detik
```

5. test5.txt

Input :

```
HERE

SHE

-----

COMES
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test5.txt

Soal :
HERE
SHE
-----+
COMES

Solusi :
9454
894
-----+
10348
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 401302

Total waktu yang dibutuhkan untuk menemukan solusi : 4.41 detik
```

6. test6.txt

Input :

```
DOUBLE

DOUBLE

TOIL

-----

TROUBLE
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test6.txt

Soal :
DOUBLE
DOUBLE
  TOIL
-----+
TROUBLE

Solusi :
798064
798064
  1936
-----+
1598064
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 2254276

Total waktu yang dibutuhkan untuk menemukan solusi : 48.45 detik
```

7. test7.txt

Input :

```
NO
GUN
NO
-----
HUNT
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test7.txt

Soal :
NO
GUN
NO
-----+
HUNT

Solusi :
87
908
  87
-----+
1082
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 92641

Total waktu yang dibutuhkan untuk menemukan solusi : 1.17 detik
```

8. test8.txt

Input :

```
THREE
THREE
TWO
TWO
ONE
-----
ELEVEN
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test8.txt

Soal :
THREE
THREE
TWO
TWO
ONE
-----+
ELEVEN

Solusi :
84611
84611
803
803
391
-----+
171219
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 2126207

Total waktu yang dibutuhkan untuk menemukan solusi : 45.98 detik
```

9. test9.txt

Input :

```
CROSS
ROADS
-----
DANGER
```

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test9.txt

Soal :
CROSS
ROADS
-----+
DANGER

Solusi :
96233
62513
-----+
158746
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 2445336

Total waktu yang dibutuhkan untuk menemukan solusi : 33.09 detik
```

10. test10.txt

Input :

MEMO
FROM

HOMER

Output :

```
===== RESTART: D:\githubStima\src\crypt.py =====
Nama file : test10.txt

Soal :
MEMO
FROM
-----+
HOMER

Solusi :
8485
7358
-----+
15843
Jumlah tes yang dilakukan untuk mendapat solusi ini sebanyak : 88554

Total waktu yang dibutuhkan untuk menemukan solusi : 1.25 detik
```

BAB IV

LAMPIRAN

A. Tautan Source Code

Semua file terkait Tugas Kecil 1 IF2211 Strategi Algoritma baik source code dalam bahasa python, file testing sebanyak 11 buah, laporan dalam format pdf, dan readme dapat ditemui di tautan https://github.com/ravmhmd/Tucil1_13519201

B. Tabel Laporan

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