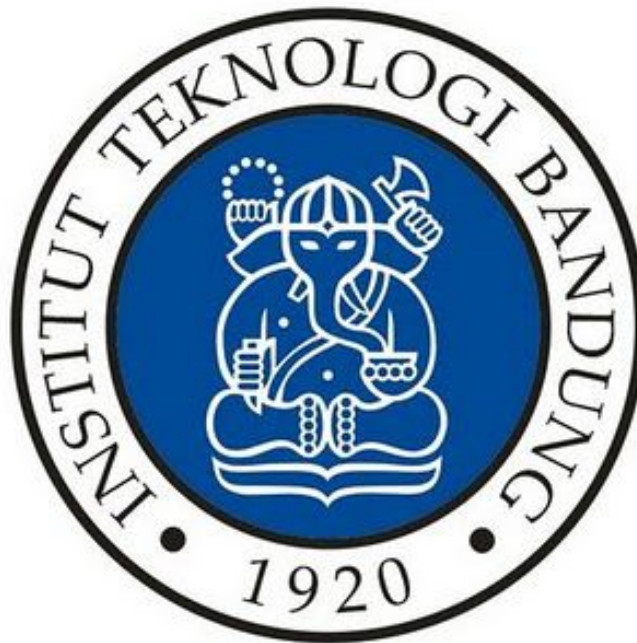


LAPORAN TUGAS KECIL 1

IF2211 - STRATEGI ALGORITMA

Penyelesaian *Cryptarithmic* dengan Algoritma Brute Force



Oleh :
Jesica 13519011

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

Deskripsi Masalah

Cryptarithmic (atau *cryptarithm*) adalah sebuah puzzle penjumlahan di dalam matematika dimana angka diganti dengan huruf. Setiap angka dipresentasikan dengan huruf yang berbeda. Deskripsi permainan ini adalah: diberikan sebuah penjumlahan huruf, carilah angka yang merepresentasikan huruf-huruf tersebut.

Soal	Solusi	Keterangan	
SEND MORE+ ----- MONEY	9567 1085+ ----- 10652	S = 9 N = 6 M = 1 R = 8	E = 5 D = 7 O = 0 Y = 2
JUNE JULY+ ----- APRIL	7924 7906+ ----- 15830	J = 7 U = 9 N = 2 E = 4 R = 8	L = 0 Y = 6 A = 1 P = 5 I = 3
FORTY TEN TEN+ ----- SIXTY	29786 850 850 -----+ 31486	F = 2 O = 9 R = 7 T = 8 Y = 6	E = 5 N = 0 S = 3 I = 1 X = 4

Tabel 1 Contoh permasalahan dan solusi *cryptarithmic*

Algoritma *Brute Force*

Deskripsi langkah-langkah algoritma *Brute Force*

1. Membaca *input* berupa *string* operand dan hasil operasi.
2. Setiap huruf dari *string* yang diperoleh akan dibuat menjadi *array of character* yang berisi huruf yang unik.
3. Bentuk *array of integer* dengan jumlah elemen yang sama dengan jumlah elemen dari *array of character*.
4. Lakukan permutasi pada *array of integer* tersebut dengan *range* angka yang mungkin adalah 0 sampai 9.
5. Selama proses permutasi, setiap kombinasi angka dari *array of integer* akan dipetakan ke *array of character*.
6. Elemen di dalam *array of integer* akan dikonversi menjadi *string* agar dapat dibentuk susunan angka yang merepresentasikan suatu nilai. Contohnya, untuk *array of character* $A = ['H', 'A', 'L', 'O']$ diperoleh salah satu hasil kombinasi dari permutasi *array of integer* $B = [1, 2, 3, 4]$. Kemudian dari B akan dibentuk *string* susunan angka 1234.
7. Kemudian susunan angka dalam bentuk *string* dikonversikan kembali menjadi *integer* dan dicek apakah hasil penjumlahannya sesuai dengan hasil operasinya. Jika sesuai maka jawaban tersebut akan menjadi *output* program.

8. Ulangi langkah ke-3 sampai ke-7 hingga proses permutasi selesai.

Source Code Program

```
#Nama      :   Jessica
#NIM       :   13519011
#Kelas    :   01
#Nama file :   cryp3.py
#Deskripsi :   Cryptarithmic Solver - Tucil 1 Strategi Algoritma IF2211

import re
import time
found = False

def readWords(x,y):
    with open(x,'r') as f:
        for line in f:
            y += filter(None, re.split(r'\W|\d', line))

def maxx(arrayofwords): #mencari jumlah karakter maksimal dari array of kata
    maxnya = len(arrayofwords[0]) # indeks dimulai dari 0
    for i in range (1,len(arrayofwords)):
        if(maxnya <= len(arrayofwords[i])):
            maxnya = len(arrayofwords[i])
    return maxnya

def cetak(arrayofwords): #array of words include result
    ...

    menghasilkan output
    SEND
    MORE
    +-----
    MONEY
    ...

    print("\n")
    blank = " "
    selisih = 0
```

```

    for i in range(len(arrayofwords)-1):
        selisih = len(result) - len(arrayofwords[i])
        preview = (blank * selisih) + arrayofwords[i]
        print(preview)
    divider = (len(result) * "-") + "+"
    print(divider)
    print(result)

def cetakangka(arrayofwords): #array of words include result
    '''
    menghasilkan output
    SEND
    MORE
    +-----
    MONEY
    ...
    print("\n")
    blank = " "
    selisih = 0
    for i in range(len(arrayofwords)-1):
        selisih = len(result) - len(arrayofwords[i])
        preview = (blank * selisih) + arrayofwords[i]
        print(preview)
    divider = (len(result) * "-") + "+"
    print(divider)
    print(arrayofwords[len(arrayofwords)-1])

def magic(huruf, angka, visited, words):

    global found
    ''' BASIS '''
    if len(set) == len(angka):

        map = {} #dictionary

        for letter, val in zip(huruf, angka):
            map[letter] = val

        for i in range(len(words)):
            if map[words[i][0]] == 0:
                # menangani kasus ketika huruf pertama setiap kata bernilai 0
                return
            magic.counter += 1

```

```

#inisialisasi
wordi = [0 for i in range (len(words))]
for i in range(len(words)):
    wordi[i] = ""

for i in range(len(words)):
    for c in words[i]:
        wordi[i] += str(map[c])

sum = 0
for i in range(len(wordi)-1):
    sum = sum + int(wordi[i])

if sum == int(wordi[len(wordi)-1]):
    cetak(words)
    cetakangka(wordi)
    print("\n")
    print(magic.counter,"total tes sampai ditemukan solusi ini")
    found = True
return

''' REKURENS '''
for i in range(10):
    if not visited[i]:
        visited[i] = True
        angka.append(i)
        magic(huruf, angka, visited, words)
        angka.pop()
        visited[i] = False

#DIASUMSIKAN INPUT DALAM BENTUK UPPERCASE SEMUA
##### MAIN #####
directory = '..\\test\\'
namafile = input("Masukkan nama file : ")
WordList = []

readWords(directory+namafile,WordList)

result = WordList[len(WordList)-1]
WordList.pop()
wordcomp = WordList

```

```
start_time = time.perf_counter()
if len(result) > (maxx(wordcomp)+1):
    print("\nTidak ada solusi")
else:
    set = []
    for i in range(len(wordcomp)):
        for char in wordcomp[i]:
            if char not in set:
                set.append(char)

    for c in result:
        if c not in set:
            set.append(c)

    if len(set) > 10:
        print("\nJumlah huruf melebihi 10 buah")
        exit()

    print("Solusi:")

    wordcomp.append(result)
    magic.counter = 0
    magic(set, [], [False for _ in range(10)], wordcomp)

    if not found:
        print("\nTidak ada solusi")

    print("Total waktu yang dibutuhkan", "%.2f"%(time.perf_counter() - start_time), "detik")
```

Input Output

Input (dalam bentuk file txt)	Output
<pre>SEND MORE+ ----- MONEY </pre>	<pre>Masukkan nama file : test1.txt Solusi: SEND MORE -----+ MONEY 9567 1085 -----+ 10652 1393690 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 23.47 detik</pre>
<pre>THREE THREE TWO TWO + ONE ----- ELEVEN</pre>	<pre>Masukkan nama file : test2.txt Solusi: THREE THREE TWO TWO ONE -----+ ELEVEN 84611 84611 803 803 391 -----+ 171219 2126207 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 66.09 detik</pre>

<pre> MEMO FROM +---- HOMER </pre>	<pre> Masukkan nama file : test3.txt Solusi: MEMO FROM -----+ HOMER 8485 7358 -----+ 15843 88554 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 1.53 detik </pre>
<pre> DOUBLE DOUBLE TOIL +----- TROUBLE </pre>	<pre> Masukkan nama file : test4.txt Solusi: DOUBLE DOUBLE TOIL -----+ TROUBLE 798064 798064 1936 -----+ 1598064 2254276 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 74.00 detik </pre>
<pre> TILES PUZZLES -----+ PICTURE </pre>	<pre> Masukkan nama file : test5.txt Solusi: TILES PUZZLES -----+ PICTURE 91542 3077542 -----+ 3169084 2640843 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 59.94 detik </pre>

<p>CROSS ROADS +----- DANGER</p>	<p>Masukkan nama file : test6.txt Solusi:</p> <pre> CROSS ROADS -----+ DANGER 96233 62513 -----+ 158746 2445336 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 49.15 detik </pre>
<p>CLOCK TICK TOCK +----- PLANET</p>	<p>Masukkan nama file : test7.txt Solusi:</p> <pre> CLOCK TICK TOCK -----+ PLANET 90892 6592 6892 -----+ 104376 2294475 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 64.17 detik </pre>
<p>COCA COLA +----- OASIS</p>	<p>Masukkan nama file : test8.txt Solusi:</p> <pre> COCA COLA -----+ OASIS 8186 8106 -----+ 16292 95135 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 1.93 detik </pre>

<pre> HERE SHE +----- COMES</pre>	<pre>Masukkan nama file : test9.txt Solusi: HERE SHE -----+ COMES 9454 894 -----+ 10348 401302 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 6.30 detik</pre>
<pre>NUMBER NUMBER +----- PUZZLE</pre>	<pre>Masukkan nama file : test10.txt Solusi: NUMBER NUMBER -----+ PUZZLE 201689 201689 -----+ 403378 325304 total tes sampai ditemukan solusi ini Total waktu yang dibutuhkan 60.23 detik</pre>

Tabel 2 Screenshot input output

Link Source Code

[jestsee/tucil-1-stima: Tugas Kecil 1 Strategi Algoritma IF2211 \(github.com\)](https://github.com/jestsee/tucil-1-stima)

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