LAPORAN TUGAS KECIL 1
IF2211 – STRATEGI ALGORITMA

Oleh :

Raihan Astrada Fathurrahman 13519113

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

1. Algoritma *brute force*

Pertama, huruf-huruf yang terdapat dalam operand akan disimpan ke dalam sebuah *list* bernama *chars_value* yang berisi huruf dan nilai angka untuk huruf tersebut (untuk setiap huruf akan diassign dengan nilai -1, yang dianggap sebagai huruf belom diassign dengan suatu nilai). Kemudian, program akan menjalankan fungsi *solve* yang akan mencari solusi dari cryptarithmetic dengan algoritma *brute force*. Fungsi solve akan mencari seluruh kemungkinan permutasi dari untuk setiap pasangan huruf dan angka dengan catatan bagian awal kata tidak boleh 0 dan setiap huruf harus memiliki angka yang berbeda-beda.

Pada fungsi *solve* program akan mengecek apakah seluruh huruf pada *chars_value* telah berpasangan dengan suatu angka. Jika belum semua huruf diassign dengan suatu angka, maka program akan menjalankan *looping* dari angka 0-9 dan jika memenuhi syarat maka akan mengassign ke huruf yang berada pada posisi ke i (paramater fungsi *solve*, saat awal i = 0). Kemudian, fungsi *solve* akan rekursif memanggil dirinya sendiri untuk meng-*assign* angka pada huruf yang berada di posisi selanjutnya. Jika semua huruf telah ter-*assign* maka fungsi *solve* akan mengecek apakah dengan pasangan huruf dan angka yang ada dapat menyelesaikan persoalan. Jika huruf pada posisi terakhir telah selesai melakukan *looping* maka akan melakukan *backtracking* untuk melanjutkan *looping* pada huruf yang berada pada posisi sebelumnya. Secara garis besar, program akan melakukan *assign* angka dan mengeceknya pada persoalan awal terus menerus untuk setiap hasil permutasi.

2. Source Code Program

Fungsi-fungsi

```
def get_files(filename):
    # Mengembalikan list berisi kata-kata yang berada dalam file
    list_of_words = []
    cur_path = os.path.dirname(__file__)
    fpath = os.path.join(cur_path, '..\\test\\'+filename)
    try:
        f = open(fpath, "r")
        EOF = False
        while (not EOF):
            word = ""
            char = f.read(1)
            # Skip Blank or Line
            while (char == " ") or (char == "-"):
                char = f.read(1)
            # { EOP = (char != " ") or (char != "-") }
            # Copy Word
            while (char) and (char != "\n"):
                word += char
```

```
                char = f.read(1)
                if (char == "+"):
                    char = f.read(1)
            # { EOP = (not char) or (char = "\n") }
            if (word != ""):
                list_of_words.append(word)
            if not char:
                EOF = True
        # { EOP = EOF }
        f.close()

    except:
        print("Tidak ditemukan file dengan nama tersebut\n")

    return list_of_words

def isFirst(words, char):
    # Mengembalikan True jika char merupakan huruf pertama suatu kata
    first = False
    i = 0
    while not(first) and (i < len(words)):
        if (char == words[i][0]):
            first = True
        else:
            i += 1
    # { EOP : first or i >= len(words) }
    return first

def isAllAssigned(chars_value):
    # Mengembalikan True jika seluruh char sudah diassign suatu angka
    allAssigned = True
    i = 0
    while (allAssigned) and (i < len(chars_value)):
        if (chars_value[i][1] == -1):
            allAssigned = False
        else :
            i += 1
    # { EOP : allAssigned or i >= len(chars_value) }
    return allAssigned

def getValue(char, chars_value):
    # Mengembalikan angka yang telah diassign ke char
```

```
    i = 0
    while (i < len(chars_value)):
        if (char == chars_value[i][0]):
            return (chars_value[i][1])
        else:
            i += 1

def isSolved(words, chars_value):
    # Mengembalikan True jika nilai chars_value menyelesaikan Cryptarithmetic
    numbers = []
    for i in range(len(words)):
        number = ""
        for char in words[i]:
            number += str(getValue(char, chars_value))
        numbers.append(int(number))

    operands = len(words)-1
    sum = 0
    for i in range(operands):
        sum += numbers[i]
    return (sum == numbers[-1])

def longestWord(words):
    # Mengembalikan panjang kata yang terbesar
    longest = 0
    for i in range(len(words)):
        if (len(words[i]) > longest):
            longest = len(words[i])
    return longest

def printSolution(words, chars_value):
    # Mencetak solusi sesuai format
    n = longestWord(words)
    for i in range(len(words)+1):
        # Print Soal
        if (i == len(words)-2):
            print(" + ",end = "")
        else:
            print("   ",end = "")

        if (i == len(words)-1):
            for k in range(n):
```

```
            print("—",end = "")

        else:
            if (i == len(words)):
                # Cetak spasi
                for j in range(n-len(words[i-1])):
                    print(" ", end = "")
                # Cetak huruf
                for j in range(len(words[i-1])):
                    print(words[i-1][j], end = "")
            else:
                # Cetak spasi
                for j in range(n-len(words[i])):
                    print(" ", end = "")
                # Cetak huruf
                for j in range(len(words[i])):
                    print(words[i][j], end = "")

        # Print Jarak
        print("          ", end = "")

        # Print Jawaban
        if (i == len(words)-2):
            print(" + ",end = "")
        else:
            print("   ",end = "")

        if (i == len(words)-1):
            for k in range(n):
               print("—",end = "")
               if (k == n-1):
                   print("")

        else:
            if (i == len(words)):
                # Cetak spasi
                for j in range(n-len(words[i-1])):
                    print(" ", end = "")
                # Cetak angka
                for j in range(len(words[i-1])):
                    print(getValue(words[i-1][j], chars_value), end = "")
                    if (j == len(words[i-1])-1):
```

```
                        print("")
            else:
                # Cetak spasi
                for j in range(n-len(words[i])):
                    print(" ", end = "")
                # Cetak angka
                for j in range(len(words[i])):
                    print(getValue(words[i][j], chars_value), end = "")
                    if (j == len(words[i])-1):
                        print("")
    print("")

def solve(words, chars_value, i, used_num):
    if not(isAllAssigned(chars_value)): # Jika ada char yang belom diassign
        for j in range(10):
            if (j not in used_num): # Cek apakah bilangan j sudah digunakan
                if (isFirst(words,chars_value[i][0])) and (j!=0):
                    chars_value[i][1] = j
                    used_num.append(j)
                    solve(words, chars_value, i+1, used_num)
                    used_num.remove(j)
                    chars_value[i][1] = -1

                elif not(isFirst(words,chars_value[i][0])):
                    chars_value[i][1] = j
                    used_num.append(j)
                    solve(words, chars_value, i+1, used_num)
                    used_num.remove(j)
                    chars_value[i][1] = -1

    else: # allAssigned
        global tries
        tries += 1
        if (isSolved(words, chars_value)): # Cek pada cryptarithm
            # Print Solusi
            printSolution(words, chars_value)
            # Print jumlah tes yang dilakukan
            print("Total tes: ", end = "")
            print(tries)
            # Print waktu eksekusi
            print("Estimated time: ", end = "")
            end_time = time.time()
```

```
            global start_time
            print(str(end_time-start_time) + "s")
            print("————————————————————————————————————————")

def menu():
    print("╔═══════════════════════╗")
    print("║        MENU           ║")
    print("╠═══════════════════════╣")
    print("║ 1. RUN PROGRAM        ║")
    print("║ 0. EXIT               ║")
    print("╚═══════════════════════╝")
    print("")
```

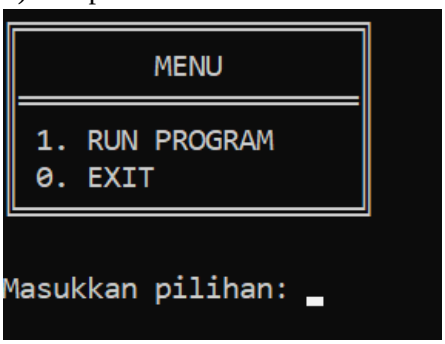
Main Program

```
# Modul yang digunakan
import time
import os

running = True
while (running):
    menu()
    print("Masukkan pilihan: ", end = "")
    choice = int(input())
    if (choice == 1):
        filename = str(input("Masukkan nama file: "))
        start_time = time.time()
        words = get_files(filename)
        if len(words) != 0: # Jika isi file tidak kosong
            # Tiap huruf dalam kata dimasukkan ke dalam list
            chars_value = []
            for i in range(len(words)):
                for char in words[i]:
                    if ([char,-1]) not in chars_value:
                        chars_value.append([char,-1]) # Assign dgn nilai -1

            used_num = [] # Inisialisasi list angka yang terpakai
            i = 0 # Index chars_value
            tries = 0 # Jumlah percobaan
            print("Mencari solusi...")
            print("————————————————————————————————————————")
            solve(words, chars_value, i, used_num)
            print("")
```

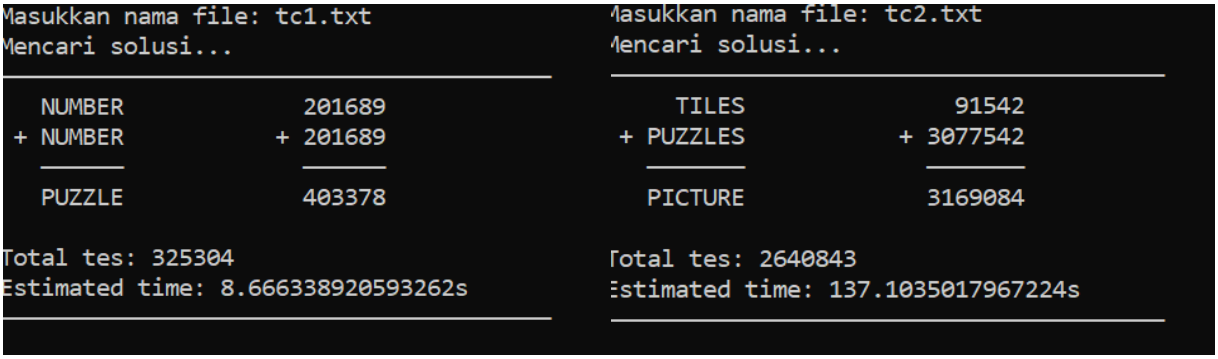
```
    elif (choice == 0):
        running = False
    else:
        print("Pilihan tidak tersedia")
    # { EOP: running = False }
```

3. Screenshot Program

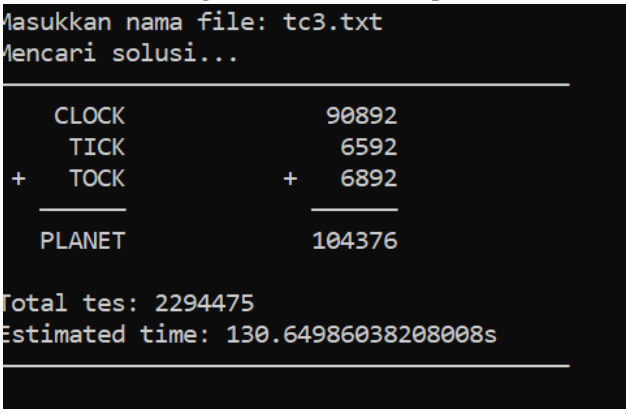
1) Tampilan awal



2) Persoalan dengan dua buah operand



3) Persoalan dengan lebih dari dua operand



```
Masukkan pilihan: 1
Masukkan nama file: tc7.txt
Mencari solusi...
--------------------------------------------------

    NO              87
   GUN             908
+   NO          +   87
  ____            ____
  HUNT            1082

Total tes: 92641
Estimated time: 1.466346025466919s
--------------------------------------------------
```

```
Masukkan nama file: tc8.txt
Mencari solusi...
--------------------------------------------------

   THREE            84611
   THREE            84611
     TWO              803
     TWO              803
+    ONE          +   391
  _______          ______
  ELEVEN           171219

Total tes: 2126207
Estimated time: 76.88174986839294s
--------------------------------------------------
```

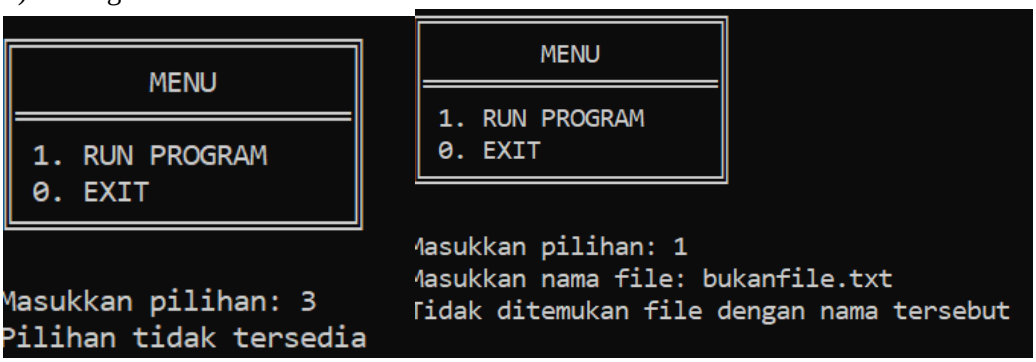
4) Persoalan dengan solusi banyak

```
Masukkan nama file: multi.txt
Mencari solusi...
--------------------------------------------------

   SUN             123
+  FUN          +  923
  ____            ____
  SWIM            1046

Total tes: 8162
Estimated time: 0.36040711402893066s
--------------------------------------------------

   SUN             127
+  FUN          +  927
  ____            ____
  SWIM            1054

Total tes: 11050
Estimated time: 0.4147708415985107 4s
--------------------------------------------------

   SUN             128
+  FUN          +  928
  ____            ____
  SWIM            1056

Total tes: 11771
Estimated time: 0.43703460693359375s
--------------------------------------------------

   SUN             132
+  FUN          +  932
  ____            ____
  SWIM            1064

Total tes: 14049
Estimated time: 0.4857418537139892 6s
--------------------------------------------------

   SUN             134
+  FUN          +  934
  ____            ____
  SWIM            1068
```

```
--------------------------------------------------

   SUN             136
+  FUN          +  936
  ____            ____
  SWIM            1072

Total tes: 16213
Estimated time: 0.5729880332946777s
--------------------------------------------------

   SUN             138
+  FUN          +  938
  ____            ____
  SWIM            1076

Total tes: 17660
Estimated time: 0.6300146579742432s
--------------------------------------------------

   SUN             143
+  FUN          +  943
  ____            ____
  SWIM            1086

Total tes: 20659
Estimated time: 0.7359178066253662s
--------------------------------------------------

   SUN             167
+  FUN          +  867
  ____            ____
  SWIM            1034

Total tes: 34446
Estimated time: 0.9785404205322266s
--------------------------------------------------

   SUN             173
+  FUN          +  873
  ____            ____
  SWIM            1046
```

```
--------------------------------------------------

   SUN             176
+  FUN          +  876
  ____            ____
  SWIM            1052

Total tes: 40333
Estimated time: 1.1524975299835205s
--------------------------------------------------

        MENU
1. RUN PROGRAM
0. EXIT

Masukkan pilihan:
```

5) Penanganan khusus



4. Alamat Kode Program

Link Github : https://github.com/raihanastrada/Cryptarithmetic-Solver

5. Checklist Program

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* hanya benar untuk persoalan *cryotarithmetic* dengan dua buah *operand*		V
5. Solusi *cryptarithmetic* benar untuk persoalan *cryptarithmetic* lebih dari dua buah operand	V	