

Penyelesaian Cryptarithmic dengan Algoritma Brute Force

Laporan Tugas Kecil 1 IF2123 Strategi Algoritma

Disusun sebagai syarat tugas kecil 1 mata kuliah IF2211 Strategi Algoritma

Semester II tahun 2020/2021

Disusun oleh:

Ferdy Irawan Firdaus (13519030)



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

2021

BAB I

ALGORITMA BRUTE FORCE

1.1.Algoritma Brute Force dalam fungsi string_unik()

Dalam fungsi `string_unik()` terdapat dua parameter yaitu `mylist` yaitu list string dan `string_operan` yaitu string input (terdiri dari huruf alfabet, spasi ' ', strip '-', dan tambah '+'). Fungsi ini bertujuan untuk menyalin input menjadi string unik yang akan disimpan di dalam `list_unik`. Iterasi *for* yang pertama diulangi sepanjang string input kemudian dicari di dalam `list_unik` apakah karakter dari input sudah ada di `list_unik` atau belum jika belum ada dan karakter tersebut bukan spasi ' ', strip '-', dan tambah '+' (hanya huruf alfabet) maka karakter/huruf tersebut akan ditambahkan atau konkat ke `list_unik`. Misalnya pada penjumlahan SEND + MORE = MONEY, maka fungsi ini akan mengembalikan `list_unik` = [S, E, N, D, M, O, R, Y] sehingga tidak ada lagi karakter yang muncul dua kali, karakter spasi ' ', tambah '+', dan strip '-'. Algoritma *brute force* yang terdapat di dalam fungsi ini yaitu mencari secara satu persatu dari depan sampai belakang sepanjang karakter input dan membandingkan dengan karakter yang sudah terdapat di `list_unik`, jika belum ada baru ditambahkan karakternya ke dalam `string_unik` tersebut.

```
def string_unik(mylist, string_operan):
    list_unik = []
    for i in range(0, panjang_string(string_operan)):
        j = 0
        huruf_sama = False
        while (j < panjang_list(list_unik) and huruf_sama == False):
            if (list_unik[j] == mylist[i]):
                huruf_sama = True
            j = j + 1
        if (huruf_sama == False and mylist[i] != '-' and mylist[i] != ' ' and mylist[i] != '+'):
            list_unik = list_unik + [mylist[i]]
    return list_unik
```

Gambar 1.1 Fungsi string_unik

1.2.Algoritma Brute Force dalam fungsi ubah_desimal()

Dalam fungsi `ubah_desimal()` ini bertujuan mengembalikan desimal bertipe *integer* yaitu mengubah *list string* angka menjadi sebuah bilangan desimal, misalnya sudah ditemukan solusi untuk operan [S,E,N,D] = [9,5,6,7] maka fungsi ini mengembalikan desimal = $9 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 7 \times 10^0 = 9567$. Sehingga nantinya angka ini akan disubstitusi untuk mencari solusi yang benar dari penjumlahan operan dan hasil dari penjumlahan operan ini akan disamakan dengan *input* jawaban. Algoritma *brute force* pada

fungsi ini yaitu ditandai dengan menjumlahkan satu persatu *integer* mulai digit paling depan hingga satuan.

```
def ubah_desimal(list_string):  
    desimal = 0  
    for i in range(0, panjang_list(list_string)):  
        desimal = desimal + list_string[i]*(10**((panjang_list(list_string)-i-1)))  
    return desimal
```

Gambar 1.2 Fungsi ubah_desimal

1.3. Algoritma Brute Force dalam fungsi digit_maksimum()

Fungsi ini bertujuan mencari jumlah digit terpanjang dari semua operan serta jawaban input, hasil keluaran dari fungsi ini akan digunakan dalam proses mencetak ke layar sehingga dapat dihitung berapa kali menambahkan karakter spasi di depan operan sehingga terbentuk output yang rata kanan sesuai penjumlahan bersusun. Algoritma *brute force* dalam fungsi ini ditunjukkan oleh pencarian sekuensial dari depan hingga ke belakang dan membandingkan satu persatu jika ada yang lebih besar atau panjang maka nilai *variable max* akan diganti.

```
def digit_maksimum(mylist):  
    max = 0  
    for i in range(0, panjang_list(mylist)):  
        if max < panjang_string(mylist[i]):  
            max = panjang_string(mylist[i])  
    return max
```

Gambar 1.3 Fungsi digit_maksimum

1.4. Algoritma Brute Force dalam proses substitusi mencari solusi

Algoritma *brute force* banyak diimplementasikan pada bagian ini, pada iterasi *for* yang pertama dilakukan pengulangan sepanjang atau sebanyak hasil dari permutasi kemudian `list_unik = [S,E,N,D,M,O,R,Y]` akan dicocokkan dengan satu persatu hasil permutasi misalnya `[1,0,2,3,4,5,6,7]` kemudian dari hasil tersebut akan disalin ke `list_copy` yaitu awalnya `[[S,E,N,D], [M,O,R,E], [], [M,O,N,E,Y]]` (ada salah satu list kosong itu akan diisi dengan output '-----'), lalu di ganti dengan `[[1,0,2,3], [4,5,6,0], [], [4,5,2,0,7]]`. Kemudian akan diseleksi agar tidak ada angka nol pada digit pertama dan akan diubah menjadi `list_desimal = [[1023], [4560], [], [45207]]`, jika setelah dicocokkan hasilnya sama atau $1023 + 4560 = 45207$ benar maka solusi hasil permutasinya akan dimasukkan ke dalam

list_solusi sedangkan bentuk decimal tersebut akan dimasukkan ke dalam list_solusi2 sehingga mempermudah proses output ke layar

```
list_solusi = []
list_solusi2 = []
for i in range(0, panjang_list(list_permutasi)):
    for j in range(0, panjang_list(list_string_unik)):
        for k in range(0, panjang_list(list_copy)):
            for l in range(0, panjang_list(list_copy[k])):
                if list_string_unik[j] == list_operan_hasil[k][l]:
                    list_copy[k][l] = list_permutasi[i][j]

list_desimal = []
for m in range(0, panjang_list(list_copy)):
    list_desimal = list_desimal + [ubah_desimal(list_copy[m])]
hasil = 0
for n in range(0, panjang_list(list_desimal)-2):
    hasil = hasil + list_desimal[n]
temu = False
for o in range(0, panjang_list(list_copy)-2):
    if list_copy[o][0] == 0:
        temu = True
if hasil == list_desimal[panjang_list(list_desimal) - 1] and temu == False and list_copy[panjang_list(list_copy)-1][0] != 0:
    list_solusi = list_solusi + [list_permutasi[i]]
    list_solusi2 = list_solusi2 + [list_desimal]
```

Gambar 1.4 Proses substitusi

BAB II

SOURCE CODE PROGRAM

```
import time

def buka_file():
    with open('coba.txt', 'r') as file:
        data = file.read().replace('\n', '')
    return data

def buka_file2():
    with open('coba.txt', 'r') as file2:
        my_list = list(file2)
    return my_list

def hitung_baris():
    file = open('coba.txt', 'r')
    jumlah_baris = 0
    for line in file:
        if line != "\n":
            jumlah_baris += 1
    return jumlah_baris

def panjang_string(string):
    panjang = 0
    for char in string:
        panjang = panjang + 1
    return panjang

def panjang_list(mylist):
    panjang = 0
    for x in mylist:
        panjang = panjang + 1
    return panjang

def string_unik(mylist, string_operan):
    list_unik = []
    for i in range(0, panjang_string(string_operan)):
        j = 0
        huruf_sama = False
        while (j < panjang_list(list_unik) and huruf_sama == False):
            if (list_unik[j] == mylist[i]):
                huruf_sama = True
            j = j + 1
        if (huruf_sama == False and mylist[i] != '-' and mylist[i] != ' '
and mylist[i] != '+'):
            list_unik = list_unik + [mylist[i]]
    return list_unik

def permutasi(angka, panjang):
    list_hasil = []
    list_angka = list(angka)
    n = panjang_list(list_angka)
    putaran = list(range(n, n-panjang, -1))
    indeks = list(range(n))
    list_hasil += [list(list_angka[i] for i in indeks[:panjang])]
    while n:
        for i in range(panjang-1, -1, -1):
```

```

        putaran[i] = putaran[i] - 1
        if putaran[i] != 0:
            j = putaran[i]
            indeks[i], indeks[-j] = indeks[-j], indeks[i]
            list_hasil += [list(list_angka[i] for i in
indeks[:panjang])]
            break
        else:
            indeks[i:] = indeks[i + 1:] + indeks[i:i + 1]
            putaran[i] = n - i
    else:
        return list_hasil

def ubah_desimal(list_string):
    desimal = 0
    for i in range(0, panjang_list(list_string)):
        desimal = desimal + list_string[i] * (10 ** (panjang_list(list_string) -
i - 1))
    return desimal

def operan(mylist):
    list_operan = []
    for i in range(0, panjang_list(mylist)):
        list_copy = [x for x in operan_hasil[i]]
        list_temp = []
        for j in range(0, panjang_list(list_copy)):
            if (list_copy[j] != '-' and list_copy[j] != '+' and
list_copy[j] != '\n' and list_copy[j] != ' '):
                list_temp = list_temp + [list_copy[j]]
        list_operan = list_operan + [list_temp]
    return list_operan

def digit_maksimum(mylist):
    max = 0
    for i in range(0, panjang_list(mylist)):
        if max < panjang_string(mylist[i]):
            max = panjang_string(mylist[i])
    return max

def print_soal():
    print("Soal")
    with open('coba.txt', 'r') as file:
        data = file.read()
    print(data, '\n')

def print_jawaban(mylist, mylist2):
    print("Jawaban")
    if panjang_list(mylist) == 0:
        print("Tidak ada solusi yang memenuhi")
    else:
        print("Ada", panjang_list(mylist), "solusi yaitu:")
        maksimum = digit_maksimum(list_operan_hasil)
        for i in range(0, panjang_list(mylist)):
            print("Solusi ke-" + str(i + 1))
            for j in range(0, panjang_list(mylist2[i]) - 3):
                selisih = maksimum - panjang_list(list_operan_hasil[j])
                if panjang_list(list_operan_hasil[j]) < maksimum:
                    for k in range(0, selisih + 1):
                        print(' ', end='')
                    print(str(mylist2[i][j]))
            else:

```

```

        print(' ' + str(mylist2[i][j]))

    print("+", end='')
    selisih2 = maksimum -
panjang_list(list_operan_hasil[panjang_list(mylist2[i])-3])
    if panjang_list(list_operan_hasil[panjang_list(mylist2[i])-3])
< maksimum:
        for l in range(0, selisih2):
            print(' ', end='')
            print(str(mylist2[i][panjang_list(mylist2[i])-3]))
        else:
            print(str(mylist2[i][panjang_list(mylist2[i])-3]))

        for m in range(0,maksimum+1):
            print('-', end='')
            if m == maksimum:
                print('\n', end='')

        selisih3 = maksimum -
panjang_list(list_operan_hasil[panjang_list(mylist2[i]) - 1])
        if panjang_list(list_operan_hasil[panjang_list(mylist2[i]) -
1]) < maksimum:
            for l in range(0, selisih3+1):
                print(' ', end='')
                print(str(mylist2[i][panjang_list(mylist2[i]) - 1]))
            else:
                print(' ' + str(mylist2[i][panjang_list(mylist2[i]) - 1]))

# Main Program
input_sebaris = buka_file()
operan_hasil = buka_file2()

start = time.time()
list_input_sebaris = [x for x in input_sebaris]
list_string_unik = string_unik(list_input_sebaris,input_sebaris)

list_permutasi =
permutasi([0,1,2,3,4,5,6,7,8,9],panjang_list(list_string_unik))

list_operan_hasil = operan(operan_hasil)

list_copy = []
for i in range(0,panjang_list(list_operan_hasil)):
    dummy = []
    for j in range(0,panjang_list(list_operan_hasil[i])):
        dummy = dummy + [list_operan_hasil[i][j]]
    list_copy = list_copy + [dummy]

list_solusi = []
list_solusi2 = []
for i in range(0,panjang_list(list_permutasi)):
    for j in range(0,panjang_list(list_string_unik)):
        for k in range(0,panjang_list(list_copy)):
            for l in range(0,panjang_list(list_copy[k])):
                if list_string_unik[j] == list_operan_hasil[k][l]:
                    list_copy[k][l] = list_permutasi[i][j]

    list_desimal = []
    for m in range(0, panjang_list(list_copy)):
        list_desimal = list_desimal + [ubah_desimal(list_copy[m])]
    hasil = 0

```

```

    for n in range(0, panjang_list(list_desimal)-2):
        hasil = hasil + list_desimal[n]
    temu = False
    for o in range(0,panjang_list(list_copy)-2):
        if list_copy[o][0] == 0:
            temu = True
        if hasil == list_desimal[panjang_list(list_desimal) - 1] and temu ==
False and list_copy[panjang_list(list_copy)-1][0] != 0:
            list_solusi = list_solusi + [list_permutasi[i]]
            list_solusi2 = list_solusi2 + [list_desimal]

print_soal()
print_jawaban(list_solusi,list_solusi2)
print("\nJumlah total tes adalah", panjang_list(list_permutasi), "kali
substitusi")
end = time.time()
print("\nWaktu yang dibutuhkan adalah", end - start, "detik")

```


BAB III

SCREENSHOT INPUT DAN OUTPUT

3.1. Input ke-1

The screenshot shows a Python script in a PyCharm editor and its output in a terminal window. The script, named `Tucil1_13519030.py`, imports `time` and `os`. It defines two functions: `buka_file()` which reads a file and replaces newlines with spaces, and `buka_file2()` which reads a file and returns its content as a list. The script then calls `buka_file2()` and prints the result. The terminal output shows the execution of the script, which reads from `input1.txt` and prints the following text:

```
Soal
SEND
+ MORE
-----
MONEY

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
9567
+ 1085
-----
10652

Jumlah total tes adalah 1814400 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 63.63144612312317 detik

Process finished with exit code 0
```

3.2 Input ke-2

The screenshot shows a Python script in a PyCharm editor and its output in a terminal window. The script, named `Tucil1_13519030.py`, imports `time` and `os`. It defines two functions: `buka_file()` which reads a file and replaces newlines with spaces, and `buka_file2()` which reads a file and returns its content as a list. The script then calls `buka_file2()` and prints the result. The terminal output shows the execution of the script, which reads from `input2.txt` and prints the following text:

```
Soal
JUNE
+ JULY
-----
APRIL

Jawaban
Ada 12 solusi yaitu:
Solusi ke-1
5486
+ 5437
-----
10923
Solusi ke-2
5487
+ 5436
-----
10923
Solusi ke-3
7924
+ 7906
-----
15830
Solusi ke-4
7926
```

```
Tucil1_13519030 <
Solusi ke-4
7926
+ 7904
-----
15830
Solusi ke-5
8432
+ 8475
-----
16907
Solusi ke-6
8435
+ 8472
-----
16907
Solusi ke-7
8524
+ 8539
-----
17063
Solusi ke-8
8529
+ 8534
-----
17063
Solusi ke-9
-----
17063
Solusi ke-10
8534
+ 8562
-----
17096
Solusi ke-11
9253
+ 9207
-----
18460
Solusi ke-12
9257
+ 9203
-----
18460

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 207.97621488571167 detik

Process finished with exit code 0

Tucil1_13519030.py <
1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input2.txt', curent_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input2 - Notepad
File Edit Format View Help
+ JUNE
+ JULY
-----
APRIL

Tucil1_13519030.py <
1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input2.txt', curent_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input2 - Notepad
File Edit Format View Help
+ JUNE
+ JULY
-----
APRIL
```

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/
Soal
    FORTY
    TEN
+   TEN
-----
    SIXTY

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
29786
    850
+   850
-----
    31486

Jumlah total tes adalah 3628800 Kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 182.63288187980652 detik

Process finished with exit code 0
```

3.4. Input ke-4

Tucil1_13519030

C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
NUMBER
+ NUMBER

PUZZLE

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
201689
+201689

403378

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 173.09325075149536 detik

Process finished with exit code 0

Tucil1_13519030.py

1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('..\test\input4.txt', current_path)
7
8 def buka_file():
9 with open(new_path, 'r') as file:
10 data = file.read().replace('\n', '')
11 return data
12
13 def buka_file2():
14 with open(new_path, 'r') as file2:
15 my_list = list(file2)
16 return my_list

input4 - Notepad
File Edit Format View Help
NUMBER
+ NUMBER

PUZZLE

3.5. Input ke-5

Tucil1_13519030

C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
TILES
+ PUZZLES

PICTURE

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
91542
+3077542

3169084

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 192.7997326850891 detik

Process finished with exit code 0

Tucil1_13519030.py

1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('..\test\input5.txt', current_path)
7
8 def buka_file():
9 with open(new_path, 'r') as file:
10 data = file.read().replace('\n', '')
11 return data
12
13 def buka_file2():
14 with open(new_path, 'r') as file2:
15 my_list = list(file2)
16 return my_list

input5 - Notepad
File Edit Format View Help
TILES
+ PUZZLES

PICTURE

3.6. Input ke-6

Tucil1_13519030

C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
CLOCK
TICK
+ TOCK

PLANET

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
90892
6592
+ 6892

104376

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 194.47522592544556 detik

Process finished with exit code 0

Tucil1_13519030.py

1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('..\test\input6.txt', current_path)
7
8 def buka_file():
9 with open(new_path, 'r') as file:
10 data = file.read().replace('\n', '')
11 return data
12
13 def buka_file2():
14 with open(new_path, 'r') as file2:
15 my_list = list(file2)
16 return my_list

input6 - Notepad
File Edit Format View Help
CLOCK
TICK
+ TOCK

PLANET

3.7. Input ke-7

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
COCA
+ COLA
-----
OASIS

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
8186
+ 8106
-----
16292

Jumlah total tes adalah 151200 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 4.4738805294036865 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input7.txt', current_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input7 - Notepad
File Edit Format View Help
| COCA
+ COLA
-----
OASIS
```

3.8. Input ke-8

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
HERE
+ SHE
-----
COMES

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
9454
+ 894
-----
10348

Jumlah total tes adalah 604800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 17.769282341003418 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input8.txt', current_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input8 - Notepad
File Edit Format View Help
| HERE
+ SHE
-----
COMES
```

3.9. Input ke-9

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
DOUBLE
DOUBLE
+ TOIL
-----
TROUBLE

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
798064
798064
+ 1936
-----
1598064

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 205.17800498008728 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 current_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input9.txt', current_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input9 - Notepad
File Edit Format View Help
| DOUBLE
DOUBLE
+ TOIL
-----
TROUBLE
```

3.10. Input ke-10

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/
Soal
NO
GUN
+ NO
-----
HUNT

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
87
908
+ 87
-----
1082

Jumlah total tes adalah 151200 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 4.302511930465698 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input10.txt', curent_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input10 - Notepad
File Edit Format View Help
| NO
| GUN
+ NO
-----
HUNT
```

3.11. Input ke-11

```
Tucil1_13519030 x
Soal
THREE
THREE
TWO
TWO
+ ONE
-----
ELEVEN

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
84611
84611
803
803
+ 391
-----
171219

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 228.71931648254395 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input11.txt', curent_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input11 - Notepad
File Edit Format View Help
| THREE
| THREE
| TWO
| TWO
+ ONE
-----
ELEVEN
```

3.12. Input ke-12

```
Tucil1_13519030 x
C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/
Soal
CROSS
+ ROADS
-----
DANGER

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
96233
+ 62513
-----
158746

Jumlah total tes adalah 3628800 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 147.7535800933838 detik

Process finished with exit code 0

Tucil1_13519030.py x
1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input12.txt', curent_path)
7
8 def buka_file():
9     with open(new_path, 'r') as file:
10         data = file.read().replace('\n', '')
11     return data
12
13 def buka_file2():
14     with open(new_path, 'r') as file2:
15         my_list = list(file2)
16     return my_list

input12 - Notepad
File Edit Format View Help
| CROSS
+ ROADS
-----
DANGER
```

3.13. Input ke-13

Tucil1_13519030 x

C:\Users\ferdy\PycharmProjects\Tucil1\venv\Scripts\python.exe C:/U
Soal
MEMO
+ FROM

HOMER

Jawaban
Ada 1 solusi yaitu:
Solusi ke-1
8485
+ 7358

15843

Jumlah total tes adalah 151200 kali substitusi atau permutasi

Waktu yang dibutuhkan adalah 4.555529832839966 detik

Process finished with exit code 0

Tucil1_13519030.py x

1 import time
2 import os
3
4 curent_path = os.path.dirname(__file__)
5
6 new_path = os.path.relpath('../test\\input13.txt', curent_path)
7
8 def buka_file():
9 with open(new_path, 'r') as file:
10 data = file.read().replace('\n', '')
11 return data
12
13 def buka_file2():
14 with open(new_path, 'r') as file2:
15 my_list = list(file2)
16 return my_list

input13 - Notepad
File Edit Format View Help
MEMO
+ FROM

HOMER

BAB IV
ALAMAT REPOSITORY

https://github.com/ferdyirawanf/Tucil1_IF2211_13519030.git