

LAPORAN TUGAS KECIL 1

Mata Kuliah Strategi Algoritma IF 2211

Dosen Pengampu : Dr. Nur Ulfa Maulidevi, S.T., M.Sc.

Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

$$\begin{array}{r} \text{IPK} : 152 \\ \text{ALEX} : 6893 \\ \hline \text{MNTP} : 7045 \end{array}$$

Disusun Oleh :

Alexander - 13519090

13519090@std.stei.itb.ac.id

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

2021

A. Algoritma *Brute Force*

Algoritma *brute force* merupakan algoritma untuk mencari sesuatu dengan cara yang brutal, brutal sebagaimana semua kemungkinan solusi yang ada akan diuji satu per-satu sehingga akan ditemukan solusi yang tepat dan benar, akan tetapi kelemahan dari algoritma *brute force* adalah proses komputasinya yang memakan waktu

Pada kode dengan bahasa Python yang terlampir pada bagian B, dapat dilihat bahwa algoritma brute force yang saya gunakan bekerja dengan cara mencari permutasi dari 10 angka terlebih dahulu, yaitu 0,1,2,3,4,5,6,7,8,9 sehingga didapat beberapa hasil sebagai berikut :

[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]

[1, 2, 3, 4, 5, 6, 7, 8, 0, 9]

....

[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

Namun karena tidak semua soal yang diuji memiliki jumlah huruf unik sebanyak 10, maka list permutasi berikut akan digunakan sepanjang dengan jumlah huruf unik.

Selanjutnya setelah List permutasi didapatkan, maka akan dilakukan pembuatan dictionary untuk membantu proses konversi huruf menjadi angka, yang nantinya akan diuji untuk menguji kesesuaian antara soal dan jawaban.

Setelah Dictionary selesai dibentuk, maka proses konversi soal (huruf) menjadi angka sesuai dengan dictionary yang ada, selanjutnya angka angka tersebut akan dijumlahkan dan dicocokkan pada jawaban, jika solusi merupakan solusi yang tepat maka solusi akan dicetak pada layar.

B. Source Code

```
#!/usr/bin/python3

import datetime

konstanta = ['1','0','2','3','4','5','6','7','8','9']
def clean_string(pisang):
    clean=''
    for string in pisang:
        for i in range(len(string)):
            if (string[i] not in clean) and (string[i] != "+") and
(string[i] != " "):
                clean += string[i]
    return clean

def open_file(namafile):
    char = ""
    cek = True
    with open(namafile) as f:
        lines = f.readlines()

    for line in lines :
        line = line.rstrip("\n")
        if (line.startswith('-')) :
            char += " = "
            cek = False
        elif (line == ""):
            char += "\n"
            cek = True
        else:
            if "+" in line :
                line = line.replace("+","")
                char += line
            else :
                char += line
                if cek :
                    char += " + "

    return char
```

```

def sum_list(lista):
    jumlah = 0
    for i in range(len(lista)-1):
        jumlah += int(lista[i])
    return jumlah

def permutasi(apel): # Kode permutasi untuk mencari semua
kemungkinan dari suatu list apel
    if len(apel) == 0:
        yield []
    elif len(apel) == 1:
        yield apel
    else:
        for i in range(len(apel)):
            # Rekursi dilakukan tanpa elemen ke i, sehingga hasil
permutasi didapatkan seluruhnya
            for p in permutasi(apel[:i] + apel[i+1:]):
                # Mengeluarkan salah satu hasil permutasi, tanpa
keluar dari fungsi
                yield([apel[i]] + p)

# Program Utama
char = open_file('../test/soal.txt')
count = 0
countsoal = 1
for soal in char.strip().splitlines():
    # loop untuk setiap soal pada soal.txt
    AdaHasil = False
    apel,hasil = soal.split(" = ")
    jeruk = apel.split(" + ")
    jeruk.append(hasil)
    count = 0
    begin_time = datetime.datetime.now()
    print("Soal ke ",countsoal)
    countsoal += 1
    if (len(clean_string(jeruk)) <= 10 ):
        for perm in permutasi(konstanta):
            # Melakukan loop terhadap setiap permutasi
            jambu =
dict(zip(clean_string(jeruk),perm[:len(clean_string(jeruk))]))

```

```

        # Membuat kamus untuk setiap huruf unik, dan
        menggabungkannya dengan hasil permutasi
        listbaru = []
        cek = True
        for string in jeruk:
            # Proses konversi huruf -> angka
            angka = ""
            for i in string:
                angka += str(jambu[i])
            listbaru.append(angka)
        jumlah = sum_list(listbaru)
        count += 1
        if ((jumlah == int(listbaru[len(listbaru)-1]))) :
            # Pengecekan apakah kemungkinan solusi benar
            for i in range(len(listbaru)):
                if (int(listbaru[i][0]) == 0):
                    # Mengecek jika huruf pertama pada soal bernilai
0
                    cek = False
            if cek:
                # Mencetak hasilnya pada layar
                for j in range(len(listbaru)):
                    if (j == len(listbaru)-1):
                        print("-----+")
                        print(jeruk[j], " : ",end="")
                        print(listbaru[j])
                print("Jumlah Kasus yang diuji : ",count)
                print("Waktu yang digunakan :
",datetime.datetime.now() - begin_time)
                print()
                AdaHasil = True
                break
        if (not AdaHasil):
            if (int(len(clean_string(jeruk)) > 10)):
                print("Maaf Jumlah Huruf unik lebih dari 10")
                print()
            else:
                print("Tidak ada solusi untuk soal ini")
                print()

```

C. Screenshoot Keluaran kode

1. Soal dengan Jumlah huruf unik lebih dari 10

```
Soal ke 1
Maaf Jumlah Huruf unik lebih dari 10
```

2. Soal tanpa Solusi

```
Soal ke 2
Tidak ada solusi untuk soal ini
```

3. Soal dengan Jumlah Huruf dibawah 10

```
Soal ke 3
IPK : 152
ALEX : 6893
-----+
MNTP : 7045
Jumlah Kasus yang diuji : 169067
Waktu yang digunakan : 0:00:01.629125
```

```
Soal ke 4
SEND : 9567
MORE : 1085
-----+
MONEY : 10652
Jumlah Kasus yang diuji : 3496339
Waktu yang digunakan : 0:00:34.410516
```

```
Soal ke 5
TILES : 91542
PUZZLES : 3077542
-----+
PICTURE : 3169084
Jumlah Kasus yang diuji : 3288387
Waktu yang digunakan : 0:00:41.819315
```

```
Soal ke 6
CLOCK : 90892
TICK : 6592
TOCK : 6892
-----+
PLANET : 104376
Jumlah Kasus yang diuji : 3342795
Waktu yang digunakan : 0:00:43.274895
```

```
Soal ke 12
CROSS : 96233
ROADS : 62513
-----+
DANGER : 158746
Jumlah Kasus yang diuji : 3519744
Waktu yang digunakan : 0:00:47.419969
```

```
Soal ke 7
COCA : 8186
COLA : 8106
-----+
OASIS : 16292
Jumlah Kasus yang diuji : 2928337
Waktu yang digunakan : 0:00:31.882528
```

```
Soal ke 8
HERE : 9454
SHE : 894
-----+
COMES : 10348
Jumlah Kasus yang diuji : 3451687
Waktu yang digunakan : 0:00:35.217527
```

```
Soal ke 9
DOUBLE : 798064
DOUBLE : 798064
TOIL : 1936
-----+
TROUBLE : 1598064
Jumlah Kasus yang diuji : 2899396
Waktu yang digunakan : 0:00:48.895579
```

```
Soal ke 10
NO : 87
GUN : 908
NO : 87
-----+
HUNT : 1082
Jumlah Kasus yang diuji : 3221281
Waktu yang digunakan : 0:00:40.913770
```

```
Soal ke 13
MEMO : 8485
FROM : 7358
-----+
HOMER : 15843
Jumlah Kasus yang diuji : 3088441
Waktu yang digunakan : 0:00:39.918805
```

D. Alamat Github

https://github.com/Zenovore/Tucil1_13519090

E. Tabel Penilaian

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.	✓	