

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



**Disusun oleh:**

**Dzaki Muhammad – 13519049 – K1**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2020**

# BAB 1

## ALGORITMA PROGRAM

### 1.1 Langkah-langkah Program

- 1) Membaca input file
- 2) Inisialisasi array untuk menyalin kata, mengisi huruf, dan menyimpan permutasi bilangan
- 3) Menyalin setiap kata dari pembacaan file ke dalam suatu array
- 4) Menuliskan tiap huruf yang ada pada persoalan ke dalam suatu array dengan tambahan elemen *dummy* jika jumlah huruf kurang dari 10
- 5) Menyalin tiap kemungkinan permutasi dari bilangan 0-9 ke dalam suatu array
- 6) Mengubah kata-kata pada persoalan dengan bilangan sesuai indeks dari array huruf dan array hasil permutasi hingga ditemukan array permutasi yang menghasilkan penyesuaian persoalan dan solusi yang benar
- 7) Jika ditemukan array permutasi yang benar program akan menampilkan persoalan *cryptarithmic* yang dibaca beserta solusinya, waktu eksekusi program dihitung setelah membaca input hingga menampilkan luaran, dan jumlah tes yang dilakukan hingga ditemukan permutasi yang benar.

## BAB 2

### SOURCE CODE PROGRAM

Keseluruhan program direalisasikan dalam bahasa python pada file main.py, source code program adalah sebagai berikut :

```
# NAMA : DZAKI MUHAMMAD
# NIM : 13519049
# KELAS : K-1
# DESKRIPSI : TUGAS KECIL 1 STRATEGI ALGORITMA

# PENYELESAIAN CRYPTARITHMETIC DENGAN ALGORITMA BRUTE FORCE

# FUNGSI DAN PROSEDUR

# fungsi permutasi dari array bilangan 0-9
def permutate(num_array):
    perm_result = []
    length=len(num_array)
    if length <= 1:
        return [num_array]
    else:
        for i in range(length):
            for p in permutate(num_array[:i] + num_array[i+1:]):
                perm_result.append([num_array[i]] + p)
        return perm_result

# fungsi konversi kata menjadi bilangan
def word_to_int(num_array, let_array, word_array):
    int_array = []
    for word in word_array:
        int_string = ""
        for char in word :
            i = 0
            while let_array[i] != "?":
                if char == let_array[i] :
                    int_string += str(num_array[i])
                    i += 1
                if i ==10:
                    break
            int_array.append(int(int_string))
    return int_array

# fungsi memeriksa apakah kondisi huruf pertama terpenuhi
def firstlet_check(firstlet_array, num_array, let_array):
    i=0
    while let_array[i] != "?":
        if let_array[i] in firstlet_array and num_array[i] == 0:
            return False
        i+=1
    if i==10:
        break
    return True
```

```

def solve(num_array, let_array, word_array, firstlet_array):
    valid = False
    test_sum = 0
    perm_num = permutate(num_array)
    for arr in perm_num:
        test_sum += 1
        if firstlet_check(firstlet_array, arr, let_array):
            int_arr = word_to_int(arr, let_array, word_array)
            length = len(int_arr)
            sums = 0
            result = int_arr[length-1]
            for i in range(length-1):
                sums += int_arr[i]
            if sums == result :
                valid = True
                break

    if valid :
        print("\nPERSOALAN :")
        i = 0
        while(i!=length-2):
            print(word_array[i])
            i+=1
        print(word_array[i] +"")
        print("-----")
        print(word_array[length-1])

        print("\n")
        print("SOLUSI :")
        i = 0
        while(i!=length-2):
            print(int_arr[i])
            i+=1
        print(str(int_arr[i]) +"")
        print("-----")
        print(int_arr[length-1])

    else:
        print("persoalan tidak dapat diselesaikan")

    print("\n")
    print("waktu eksekusi program : %s detik" % (time.time() -
start_time))
    print('total tes yang dilakukan : %s tes' %(test_sum))

# PROGRAM UTAMA
import time

nama_file = input("Masukkan nama file yang akan dibaca : ")
#input nama file
text = open(nama_file,'r')      #membuka file dan merekamnya dalam
variabel arsip
lines = text.readlines()
start_time = time.time()

# inisialisasi array
arr_word = []
arr_firstlet = []
arr_letter = ['?' for i in range(10)]

```

```

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

# menyimpan kata-kata yang dibaca ke dalam sebuah array
for line in lines :
    line = line[:-1]      #untuk menghilangkan newline di akhir string
    if line != "-----" :
        length = len(line)
        if line[length-1] == "+":
            line = line[:-1]      #menghilangkan tanda "+"
            arr_word.append(line)

# mengisi array huruf
list_of_char = []
for word in arr_word:
    i=0
    for char in word:
        if i==0:
            arr_firstlet.append(char)
            if char not in list_of_char :
                list_of_char.append(char)
            i+=1

for i in range (len(list_of_char)) :
    arr_letter[i] = list_of_char[i]

# menyelesaikan persoalan
solve(arr_num, arr_letter, arr_word, arr_firstlet)

```

## BAB 3

### HASIL UJI

#### 3.1 Hasil Uji 1

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes1.txt

PERSOALAN :
SEND
MORE+
-----
MONEY

SOLUSI :
9567
1085+
-----
10652

waktu eksekusi program : 64.66715145111084 detik
total tes yang dilakukan : 3087475 tes
```

Gambar 3.1 Luaran hasil uji 1

#### 3.2 Hasil Uji 2

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes2.txt

PERSOALAN :
JUNE
JULY+
-----
APRIL

SOLUSI :
5486
5437+
-----
10923

waktu eksekusi program : 46.48444581031799 detik
total tes yang dilakukan : 1600134 tes
```

Gambar 3.2 Luaran hasil uji 2

### 3.3 Hasil Uji 3

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes3.txt

PERSOALAN :
FORTY
TEN
TEN+
-----
SIXTY

SOLUSI :
29786
850
850+
-----
31486

waktu eksekusi program : 31.754525899887085 detik
total tes yang dilakukan : 674493 tes
```

Gambar 3.3 Luaran hasil uji 3

### 3.4 Hasil Uji 4

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes4.txt

PERSOALAN :
NUMBER
NUMBER+
-----
PUZZLE

SOLUSI :
201689
201689+
-----
403378

waktu eksekusi program : 34.268821001052856 detik
total tes yang dilakukan : 688184 tes
```

Gambar 3.4 Luaran hasil uji 4

### 3.5 Hasil Uji 5

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes5.txt

PERSOALAN :
CLOCK
TICK
TOCK+
-----
PLANET

SOLUSI :
90892
6592
6892+
-----
104376

waktu eksekusi program : 89.84331274032593 detik
total tes yang dilakukan : 3262155 tes
```

Gambar 3.5 Luaran hasil uji 5

### 3.6 Hasil Uji 6

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes6.txt

PERSOALAN :
COCA
COLA+
-----
OASIS

SOLUSI :
8186
8106+
-----
16292

waktu eksekusi program : 49.581340312957764 detik
total tes yang dilakukan : 2564737 tes
```

Gambar 3.6 Luaran hasil uji 6

### 3.7 Hasil Uji 7

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes7.txt

PERSOALAN :
THREE
THREE
TWO
TWO
ONE+
-----
ELEVEN

SOLUSI :
84611
84611
803
803
391+
-----
171219

waktu eksekusi program : 90.59988951683044 detik
total tes yang dilakukan : 2681927 tes
```

Gambar 3.7 Luaran hasil uji 7



### 3.8 Hasil Uji 8

```
C:\Users\dzaki\Documents\ITB\Tingkat 2\Semester 4\Strategi Algoritma\Tucil1_13519049\src>python main.py
Masukkan nama file yang akan dibaca : ../test/tes8.txt

PERSOALAN :
DOUBLE
DOUBLE
TOIL+
-----
TROUBLE

SOLUSI :
798064
798064
1936+
-----
1598064

waktu eksekusi program : 84.22360277175903 detik
total tes yang dilakukan : 2494756 tes
```

Gambar 3.8 Luaran hasil uji 8

## LAMPIRAN

### Lampiran 1

*Checklist* penilaian :

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> .	✓	

### Lampiran 2

Alamat link repositori *github* program : <https://github.com/dzakimuhammad/Tucil1-13519049>