

# **Criptarithmeic Solver**

**Laporan Tugas Besar  
Mata kuliah IF2211 Strategi Algoritma**



**Disusun Oleh :**

**Juan Louis Rombetasik (13519075)**

**Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Semester 2 Tahun 2020/2021**

## ALGORITMA BRUTEFORCE

```
# catatan ide: bruteforce: mencari cara untuk menjumlahkan string dan
# mencocokkannya dari semua kemungkinan solusi Permutasi jmlh huruf yang ada
# kemungkinan terburuk = (10!)
```

Pada algoritma bruteforce file ".txt" akan di baca, lalu program akan menghitung jumlah baris pada file tersebut. Isi file disimpan kedalam list/larik perbarisnya (list of strings), character "+" pada line kedua sebelum terakhir akan dihapus character-nya. Baris yang berisi yang duplikasi dari character dash ("-----") akan dihapus dari list.

Dari list of strings ini dicari character-character unik, setelah itu dibuatlah sebuah kamus (dictionary) yang mempunyai key yaitu character-character dan valuenya adalah 0-9.

Dilakukan pencocokkan nilai kata-kata pada list of strings dengan acuan dictionary yang telah dibuat. Jika pencocokan salah, maka dibuat kamus baru lagi dengan cara permutasi nPr dimana n adalah jumlah kemungkinan angka dan r adalah jumlah huruf yang berbeda

## KODE PROGRAM

```
from time import time

namaFile = "cryptarithmic.txt"

def split(word):
    return [char for char in word]

def pemecehCryptarithmic>NamaFile):
    f = open>NamaFile, "r")
    content = f.read()
    listContent = content.split("\n")
    f.close()

    menghitungWaktu = time()
    listString = []
    jmlhBaris = 0
    for i in listContent:
        if i:
            listString.append(i.upper()) # ASUMSIKAN HURUF BESAR SEMUA
            jmlhBaris += 1

    # listString[jmlhBaris-3] = listString[jmlhBaris-3][:-1]
```

```

listString[jmlhBaris - 3] = listString[jmlhBaris - 3].replace("+", "") # Menghilangkan character '+'
listString.pop(jmlhBaris - 2) # Menghapus "-----"

listHuruf = []
contentLagi = []
for i in range(jmlhBaris - 1):
    # listString[i].split()
    listString[i] = split(listString[i])
    listHuruf += listString[i]
    contentLagi.append(listString[i])

listHuruf = list(set(listHuruf)) # MENGHILANGKAN DUPLIKASI
tipeAngka = list(range(10)) # Angka yang bisa digunakan adalah 0 - 9
coba = 0 # Menghitung jumlah percobaan

for percobaan in solusiPermutasi(tipeAngka):
    kamusKata = dict(zip(listHuruf, percobaan[::-1])) # {char:tipeAngka}

    nol = 0
    for i in range(len(contentLagi)):
        if (kamusKata[contentLagi[i][0]] == 0):
            nol += 1
    if nol != 0:
        continue

    else:
        hasil = 0
        jawaban = []

        for i in range(len(contentLagi) - 1):
            hasil += nilaiKata(contentLagi[i], kamusKata)
            jawaban.append(nilaiKata(contentLagi[i], kamusKata))

        if (hasil == nilaiKata(contentLagi[-1], kamusKata)):
            jawaban.append(nilaiKata(contentLagi[-1], kamusKata))
            end = time()
            print(content)
            print()
            n = len(jawaban)

            for i in range(n - 2):
                print(jawaban[i])
                print(f"{jawaban[n - 2]}")
                print("-----")

```

```

        print(jawaban[n - 1])

    print()
    print("Jumlah percobaan :", coba)
    print("Waktu yang dibutuhkan :", end - menghitungkWaktu, "detik")
    break
    coba += 1

# catatan ide: bruteforce: mencari cara untuk memjumlahkan string dan
# mencocokkannya dari semua kemungkinan solusiPermutasi jmlh huruf yang ada
# kemungkinan terburuk = (10!)

# Menghitung nilai kata
def nilaiKata(word, dict):
    total = 0
    k = 1
    dibalik = word[::-1]
    for x in range(len(dibalik)):
        total += dict[dibalik[x]] * k
        k *= 10
    return total

def solusiPermutasi(array):
    if len(array) == 0:
        return []
    if len(array) == 1:
        return [array]
    temp = []
    for i in range(len(array)):
        m = array[i]
        remarr = array[:i] + array[i + 1:]
        for p in solusiPermutasi(remarr):
            temp.append([m] + p)
    return temp

pemecehCryptarithmic(namaFile)

```

## SKRINSHUT OUTPUT PROGRAM

```
PS D:\Python Projects\Latihan> python -u tucil1.py
NO
GUN
NO+
-----
HUNT
87
988
87
-----
1082

Jumlah percobaan : 949895
Waktu yang dibutuhkan : 26.005616188049316 detik
```

```
PS D:\Python Projects\Latihan> & "C:/Users/Juan Louis/Py"
NUMBER
NUMBER+
-----
PUZZLE
201689
201689
-----
403378

Jumlah percobaan : 1514383
Waktu yang dibutuhkan : 34.824063301086426 detik
```

```
PS D:\Python Projects\Latihan> & "C:/Users/Juan Louis/Py"
THREE
THREE
TWO
TWO
ONE
-----
ELEVEN
84611
84611
803
803
391
-----
171219

Jumlah percobaan : 1413535
Waktu yang dibutuhkan : 41.30304002761841 detik
```

```
PS D:\Python Projects\Latihan> & "C:/Users/Juan Louis/Py"
DOUBLE
DOUBLE
+TOIL
-----
TROUBLE
798064
798064
1936
-----
1598064

Jumlah percobaan : 662245
Waktu yang dibutuhkan : 25.576043128967285 detik
```

```
PS D:\Python Projects\Latihan> python -u tucil1.py
CROSS
ROADS+
-----
DANGER
96233
62513
-----
158746

Jumlah percobaan : 186623
Waktu yang dibutuhkan : 17.987378358840942 detik
```

```
PS D:\Python Projects\Latihan> & "C:/Users/Juan Louis/Py"
CLICK
TICK
+TOCK
-----
PLANET
90892
6892
6592
-----
104376

Jumlah percobaan : 1013984
Waktu yang dibutuhkan : 31.4791898727417 detik
```

```

PS D:\Python Projects\Latihan> python -u "d:\Python Projects\Latihan\cryptarithmic.py"
COCA
COLA+
-----
OASIS

8186
8106
-----
16292

Jumlah percobaan : 1042752
Waktu yang dibutuhkan : 24.581261157989502 detik

```

```

PS D:\Python Projects\Latihan> & "C:/Users/Juan Louis/Desktop/Python Projects/Latihan/cryptarithmic.py"
HERE
SHE+
-----
COMES

9454
894
-----
10348

Jumlah percobaan : 765287

```

## CONTOH-CONTOH INPUT

```

≡ cryptarithmic.txt
1  DOUBLE
2  DOUBLE
3  +TOIL
4  -----
5  TROUBLE|

```

```

≡ cryptarithmic.txt
1  CLICK
2  TICK
3  +TOCK
4  -----
5  PLANET|

```

```

≡ cryptarithmic.txt
1  COCA
2  COLA+
3  -----
4  OASIS|

```

```

≡ cryptarithmic.txt
1  HERE
2  SHE+|
3  -----
4  COMES

```

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> .	✓	