

**LAPORAN**  
**Tugas Kecil 1 IF2121 Strategi Algoritma**

**DISUSUN OLEH**

M. Reyhanullah Budiaman      13519045

**INSTITUT TEKNOLOGI BANDUNG**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**TEKNIK INFORMATIKA**  
**2020/2021**

## BAB I

### Deskripsi Masalah

Cryptarithmic merupakan sebuah permainan teka-teki penjumlahan di dalam matematika dimana angka diganti dengan huruf alfabet atau suatu simbol lain. Jika huruf yang sama muncul lebih dari sekali, huruf itu harus diberikan digit yang sama. Tidak ada dua huruf atau lebih yang memiliki nilai yang sama. Masalah yang harus diselesaikan dari teka-teki ini adalah mencari angka unik yang sesuai dengan huruf unik.

Berikut merupakan salah satu contoh yang populer dari Cryptarithmic,

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Solusi dari puzzle diatas adalah sebagai berikut.

$$\begin{array}{r} \phantom{+} \phantom{1} \phantom{0} \phantom{6} \phantom{5} \phantom{2} \\ \phantom{+} \phantom{1} \phantom{0} \phantom{6} \phantom{5} \phantom{2} \\ + \phantom{1} \phantom{0} \phantom{6} \phantom{5} \phantom{2} \\ \hline 1 \phantom{0} \phantom{6} \phantom{5} \phantom{2} \phantom{0} \phantom{0} \phantom{2} \end{array}$$

Solusinya adalah  $S = 9$ ,  $M = 1$ ,  $E = 5$ ,  $N = 6$ ,  $D = 7$ ,  $O = 0$ , dan  $Y = 2$ .

Beberapa aturan yang ada pada Cryptarithmic adalah sebagai berikut.

- Setiap huruf direpresentasikan dengan digit yang unik
- Angka tidak boleh diawali dengan 0
- Solusinya harus unik (kecuali ada keterangan)

## BAB II

### IMPLEMENTASI PROGRAM

Bahasa pemrograman yang digunakan adalah python dengan library yang dibutuhkan adalah time dan itertools. Library time digunakan untuk mengukur waktu komputasi pencarian solusi secara brute-force. Library itertools digunakan hanya untuk membuat permutasi kejadian yang mungkin dari pasangan digit dan huruf yang unik. Algoritma yang digunakan adalah algoritma brute-force, yaitu semua kemungkinan akan dicoba dan dites apakah ada kemungkinan yang sesuai dengan puzzle.

Gambaran dari Langkah – Langkah algoritma adalah sebagai berikut.

1. Program pertama akan meminta masukan berupa nama file yang berisi puzzle *cryptarithmic*. File harus dipastikan sudah terdapat di dalam folder test.
2. Program mulai membaca kata – kata yang ada pada puzzle dan menambahkan semua ke sebuah list dan memisahkan hasil akhir pada suatu variabel string
3. Program membuat sebuah list yang menampung semua huruf yang muncul pada puzzle.
4. Program mulai mencatat waktu.
5. Dari 10 digit angka dari 0 – 9, dibuat permutasinya dengan panjang yang sama dengan jumlah huruf pada puzzle, kemudian program membuat *dictionary* dengan key berupa huruf dari list huruf dan valuenya berasal dari permutasi digit.
6. Program kemudian menghitung berdasarkan permutasi yang telah dibuat dan menentukan apakah solusi permutasinya sudah benar atau tidak. Program akan mencatat juga jumlah percobaan yang dilakukan. Jumlah tersebut bertambah seiring dengan pengecekan dilakukan. Program akan keluar dari *loop* jika sudah ditemukan satu solusi yang benar.
7. Jika solusi permutasinya tidak benar, maka Langkah 5-6 akan diulangi sampai ditemukan solusi.
8. Setelah selesai maka solusi puzzle-nya akan ditampilkan kelayar dengan format yang sama dengan puzzle-nya. Waktu akan selesai dihitung ketika program telah menuliskan luaran jumlah percobaan yang dilakukan.

Berikut merupakan Source code dari program yang telah dibuat dalam bahasa python.

```
import time
import itertools

def read_code(filename):
    cd_file = open(filename, 'r')
    wordlist = cd_file.readlines()
    mod_wl = []
    hasil = wordlist[len(wordlist)-1]
    for i in range(len(wordlist)):
        mod_wl.append(wordlist[i].strip().replace("\n", "").replace("-", "").replace("+", ""))

    kamus = []
    for teks in mod_wl:
        for kata in teks:
            if kata not in kamus:
                kamus.append(kata)

    mod_wl.pop()
    mod_wl.pop()
    return kamus, mod_wl, hasil

def str_to_int(word, word_dict):
    tot = 0
    f = 1
    for huruf in reversed(word):
        tot += f*word_dict[huruf]
        f *= 10
    return tot

def solvecrypt(filename):
    count = 0
    kamus, wordlist, result = read_code(filename)
    solusi = []

    angka = range(10)
    for perm in itertools.permutations(angka, len(kamus)):
        kms2 = dict(zip(kamus, perm))

        if sum(str_to_int(kata, kms2) for kata in wordlist) == str_to_int(result, kms2):
            if kms2[result[0]] != 0:
                solusi = kms2
                count += 1
                break

    count += 1

    if len(solusi) != 0:
        return count, solusi, result, wordlist
    else:
        print("Tidak ditemukan solusi")
```

```
def main_prog(file_name):
    t1_start = time.process_time()

    count, solusi, result, wordlist = solvecrypt(file_name)
    for word in wordlist:
        print()
        print(" "*((len(result)-len(word)), end=""))
        print(word, end="")
        print("+")
        print("-"*((len(result)+1)))
        print(result)

    for word in wordlist:
        print()
        print(" "*((len(result)-len(word)), end=""))
        for letter in word:
            print(solusi[letter], end="")
        print("+")
        print("-"*((len(result)+1)))
        for letter in result:
            print(solusi[letter], end="")

    print()
    print()
    print("Jumlah Percobaan : " + str(count))
    t1_stop = time.process_time()
    print("Waktu program berjalan : " + str(t1_stop - t1_start) + " s")

#----- MAIN PROGRAM -----
exit = 0
while exit == 0:
    file_name = input("Masukkan nama file : ")
    try:
        main_prog("../test/" + str(file_name))
    except:
        print("File tidak ditemukan")

    print()
    print("Ingin mencoba puzzle lain? (Y/N)")
    xy = input()

    if (xy == "N") or (xy == "n"):
        exit = 1
```

Fungsi – fungsi yang ada :

- `read_code(file_name)` : fungsi ini membaca puzzle yang ada didalam file dan mengisi list yang berisikan operand, list yang memuat semua huruf yang ada pada puzzle. Kemudian fungsi mengembalikan list-list tersebut dan juga hasil (Kata yang sebagai hasil penjumlahan) yang ada pada puzzle tersebut.
- `str_to_int(word, word_dict)` : fungsi ini mengubah kata pada puzzle menjadi sebuah integer berdasarkan *dictionary* yang berisikan solusi.
- `solvecryp(filename)` : mencari solusi dari puzzle dan menuliskannya dalam bentuk *dictionary*
- `main_prog(file_name)` : menjalankan program utama dan menghitung waktu pencarian solusi dan menuliskan solusi yang didapatkan

## BAB III

### EKSPERIMEN

Contoh 1 (contoh 1.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptametric.py
Masukkan nama file : contoh 1.txt

SEND
MORE+
-----
MONEY

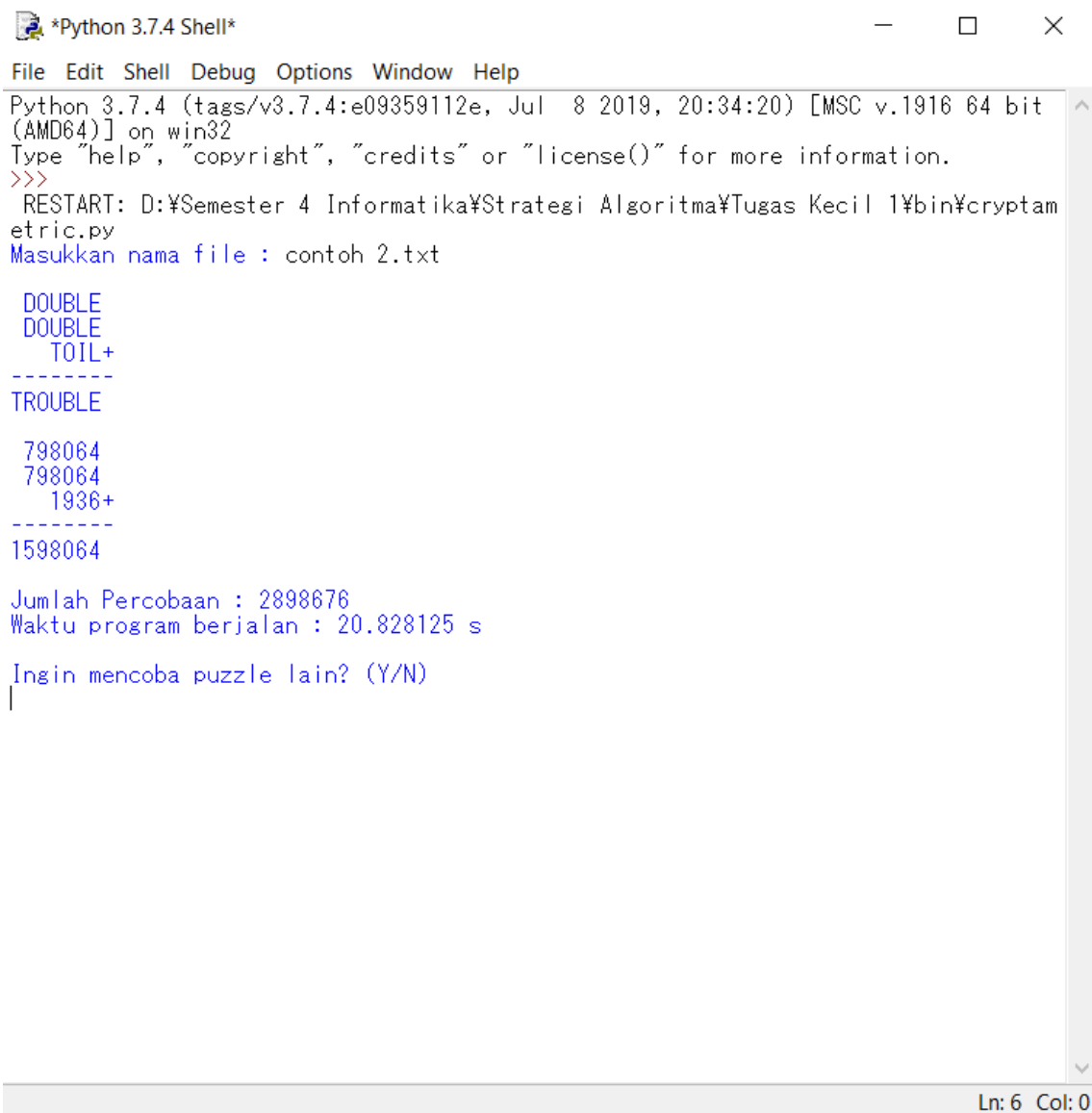
9567
1085+
-----
10652

Jumlah Percobaan : 1748230
Waktu program berjalan : 9.625 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0

Contoh 2 (contoh 2.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptametric.py
Masukkan nama file : contoh 2.txt

DOUBLE
DOUBLE
TOIL+
-----
TROUBLE

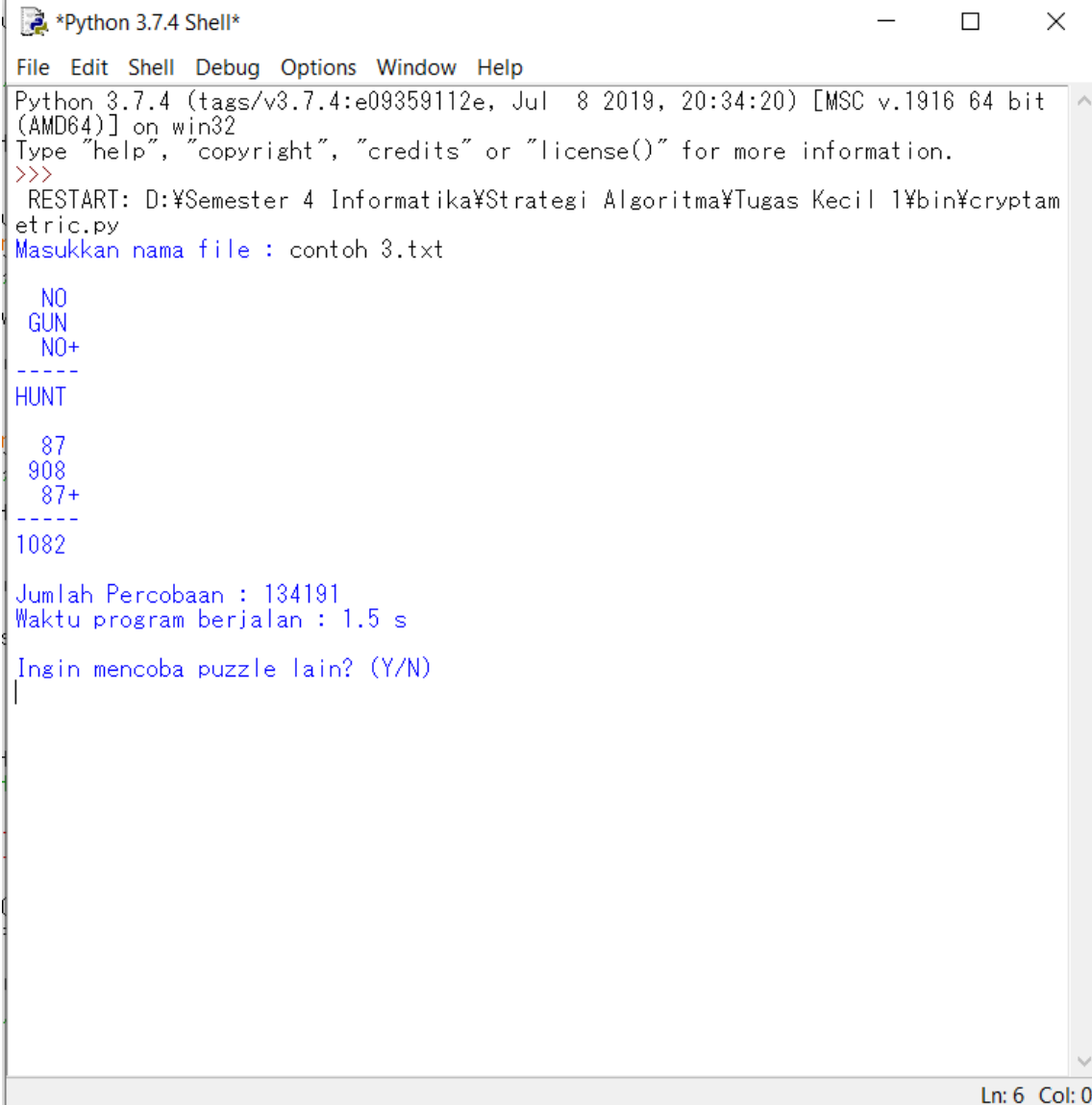
798064
798064
1936+
-----
1598064

Jumlah Percobaan : 2898676
Waktu program berjalan : 20.828125 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0

Contoh 3 (contoh 3.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptam
etric.py
Masukkan nama file : contoh 3.txt

NO
GUN
NO+
-----
HUNT

87
908
87+
-----
1082

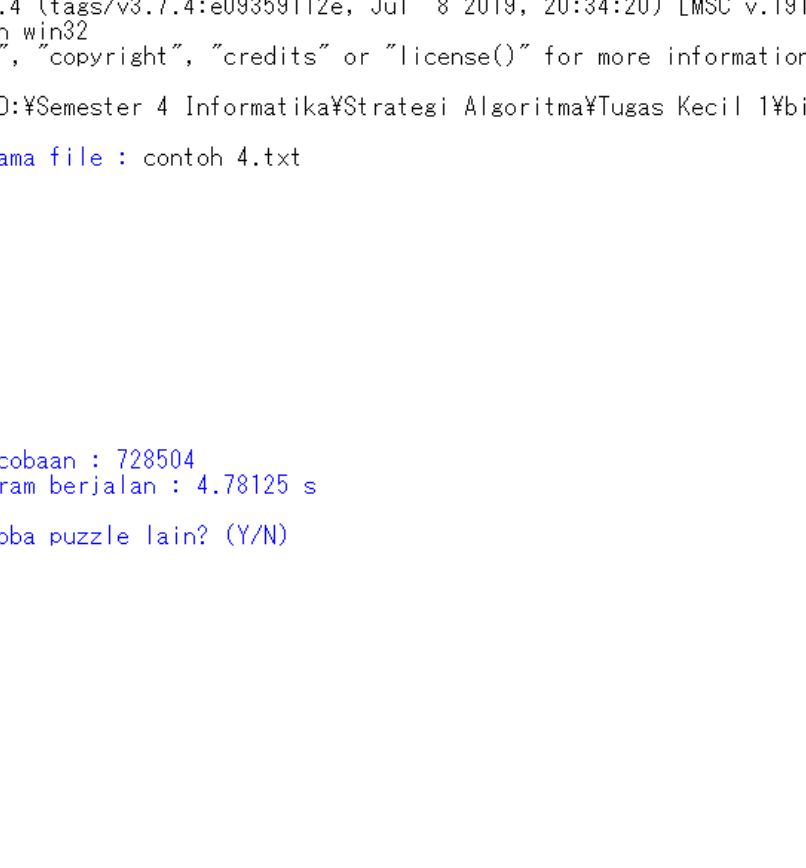
Jumlah Percobaan : 134191
Waktu program berjalan : 1.5 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0



Contoh 4 (contoh 4.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptam
etric.py
Masukkan nama file : contoh 4.txt

NUMBER
NUMBER+
-----
PUZZLE

201689
201689+
-----
403378

Jumlah Percobaan : 728504
Waktu program berjalan : 4.78125 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Contoh 5 (contoh 5.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptam
etric.py
Masukkan nama file : contoh 5.txt

TILES
PUZZLES+
-----
PICTURE

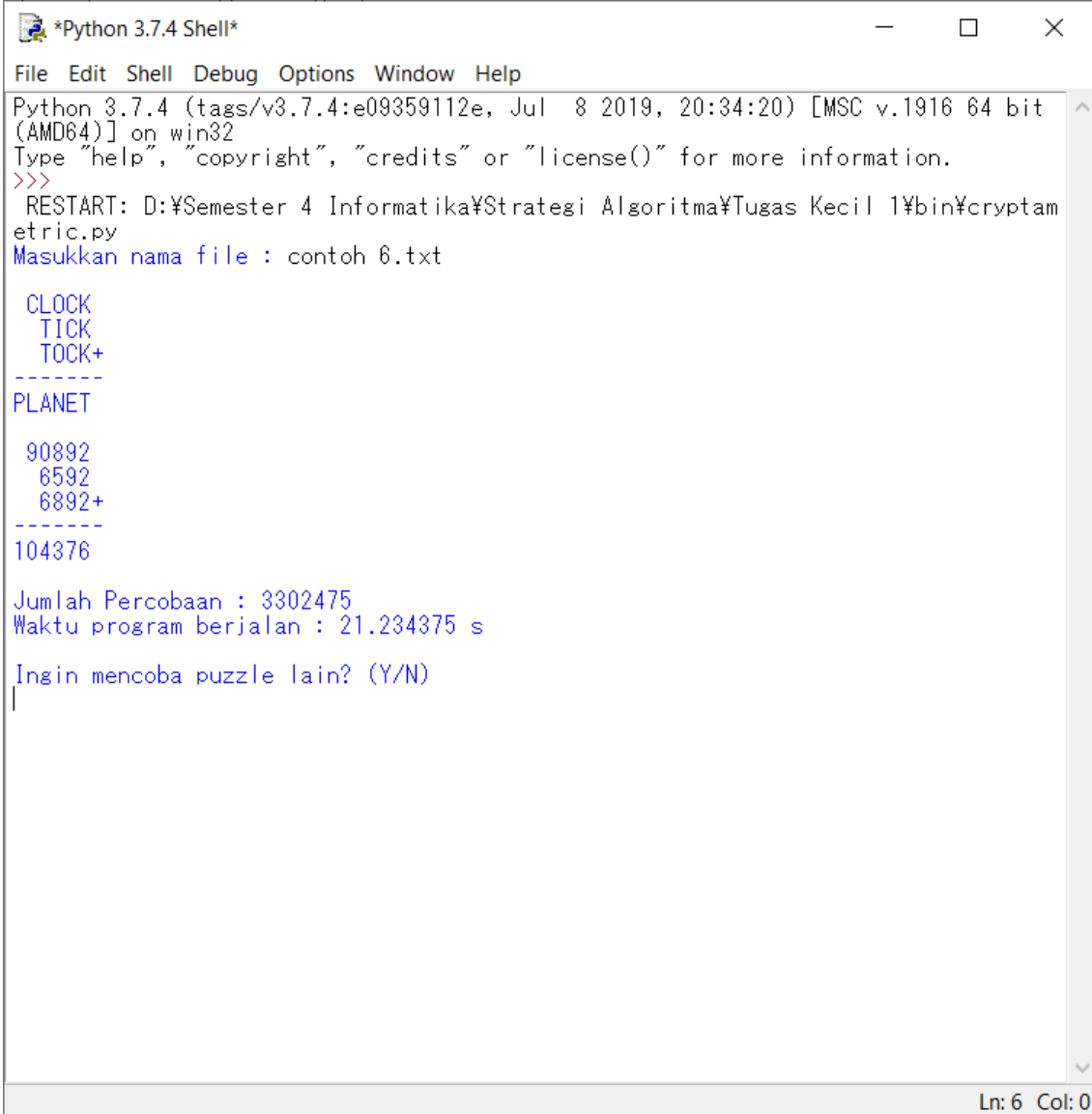
91542
3077542+
-----
3169084

Jumlah Percobaan : 3328707
Waktu program berjalan : 20.328125 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0

Contoh 6 (contoh 6.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptametric.py
Masukkan nama file : contoh 6.txt

CLOCK
TICK
TOCK+
-----
PLANET

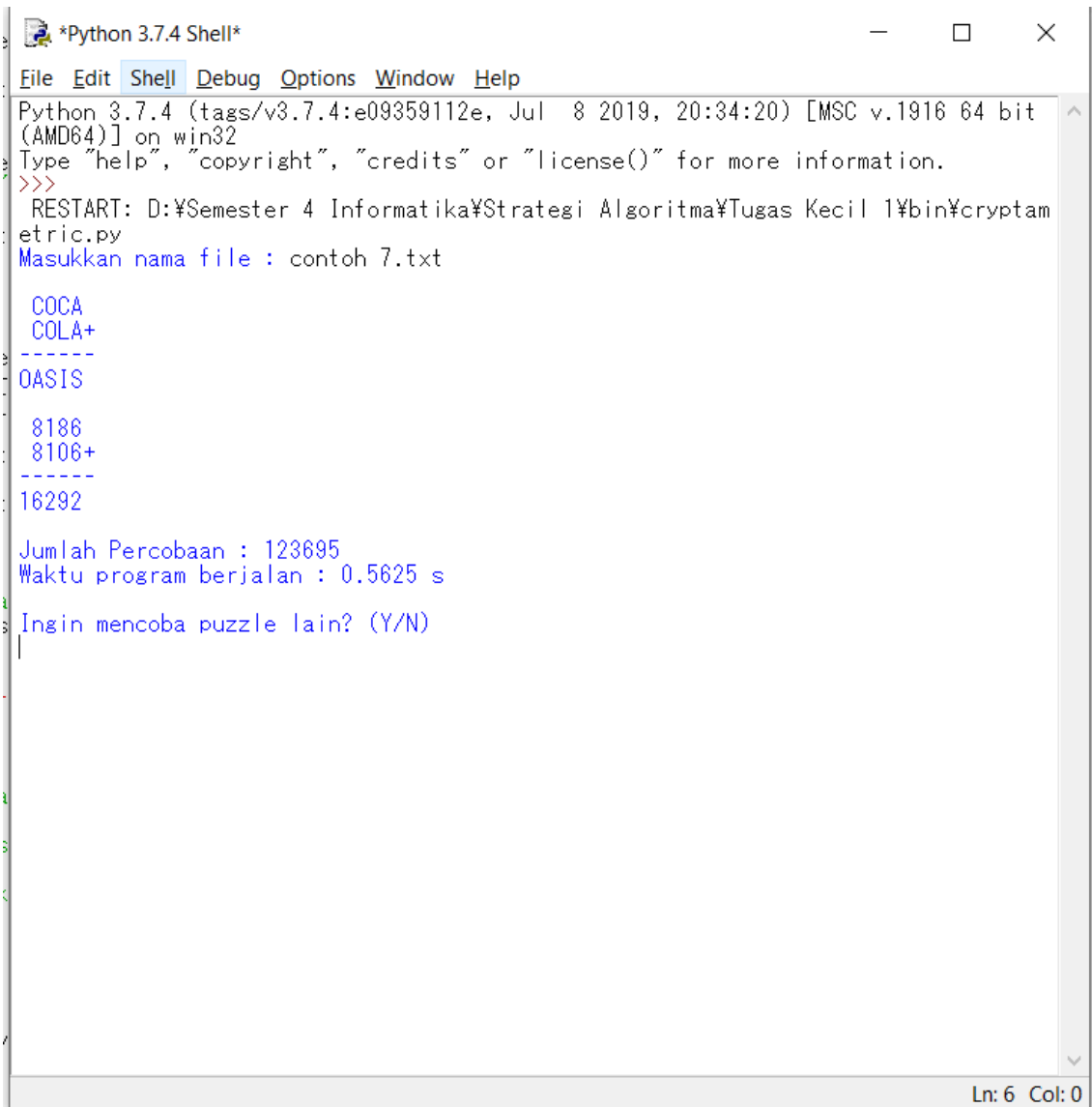
90892
6592
6892+
-----
104376

Jumlah Percobaan : 3302475
Waktu program berjalan : 21.234375 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0

Contoh 7 (contoh 7.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptam
etric.py
Masukkan nama file : contoh 7.txt

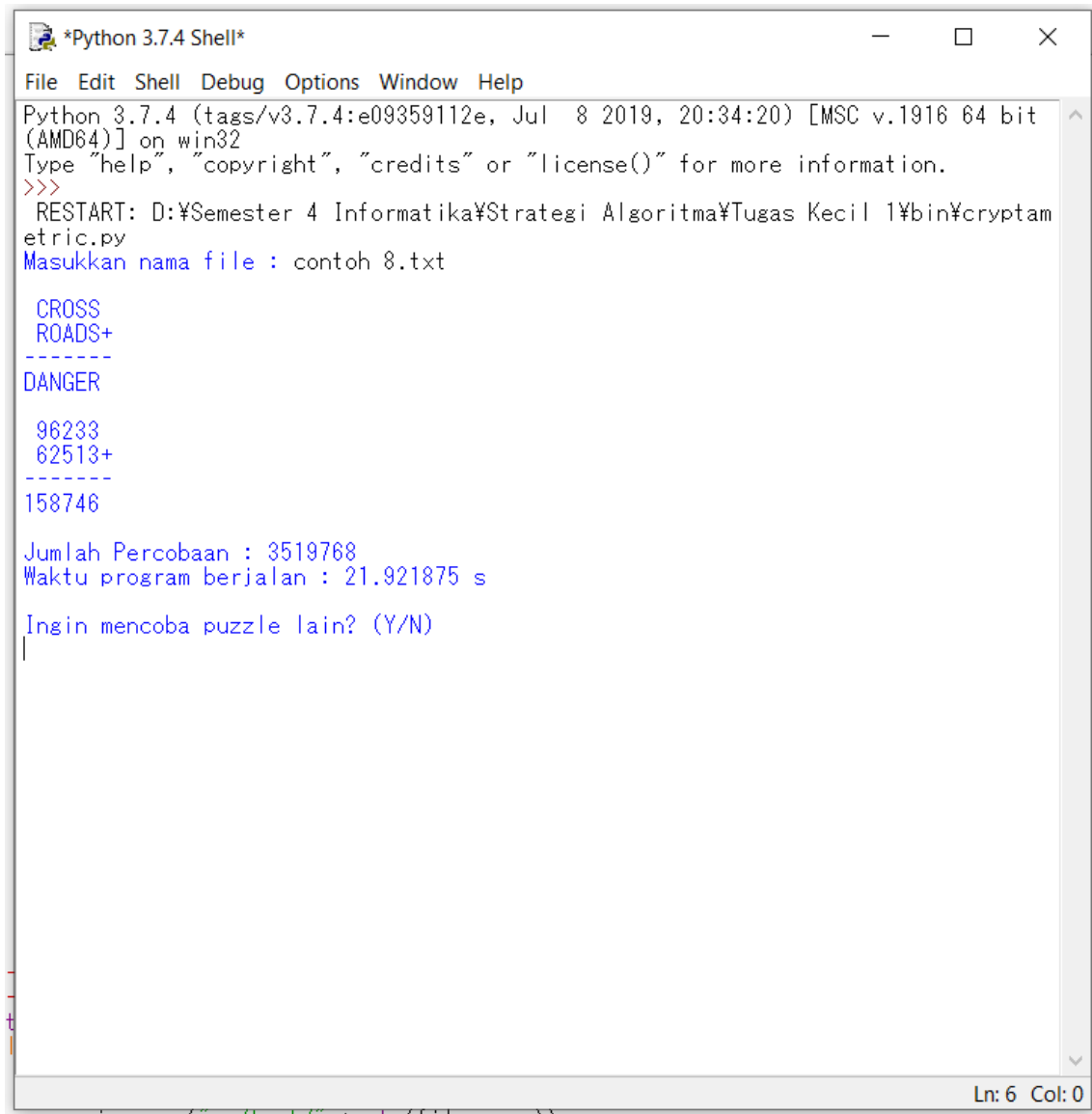
COCA
COLA+
-----
OASIS
8186
8106+
-----
16292

Jumlah Percobaan : 123695
Waktu program berjalan : 0.5625 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Ln: 6 Col: 0

Contoh 8 (contoh 8.txt):



```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Semester 4 Informatika\Strategi Algoritma\Tugas Kecil 1\bin\cryptam
etric.py
Masukkan nama file : contoh 8.txt

CROSS
ROADS+
-----
DANGER

96233
62513+
-----
158746

Jumlah Percobaan : 3519768
Waktu program berjalan : 21.921875 s

Ingin mencoba puzzle lain? (Y/N)
|
```

Analisis :

Program yang dibuat memiliki variasi dalam waktu penyelesaiannya. Dapat dilihat bahwa pada puzzle contoh 7 waktu komputasinya tercepat dengan waktu 0,5625 s dan waktu komputasi terlamanya ada pada contoh 8 dengan waktu komputasi 21,921875 s. Algoritma brute-force ini memakan waktu yang lama dan *time complexity*-nya adalah  $O(n!)$  dengan  $n$  adalah jumlah huruf unik pada puzzle. Walaupun terdapat batasan bahwa  $n \leq 10$ , worst case scenarionya adalah program harus mengetes 3,628,800 kejadian.

Waktu komputasi yang tertera pada gambar juga bisa jadi berbeda saat dilakukan pada waktu yang lain, karena waktu komputasi tersebut juga bergantung dari kemampuan multitasking hardware yang digunakan. Pada saat program dijalankan dan banyak aplikasi lain yang sedang digunakan, waktu komputasi yang terekam pun akan menurun.

## BAB IV

### LAMPIRAN

#### 5.1 Alamat Repository

<https://github.com/reynull20/Tucil1---Cryparithmetic-Solver>

#### 5.2 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> benar untuk persoalan <i>cryptarithmic</i> dengan dua buah operand		✓
5. Solusi <i>cryptarithmic</i> benar untuk persoalan <i>cryptarithmic</i> dengan lebih dari dua buah operand	✓	