

**LAPORAN TUGAS KECIL**  
**IF2211 Strategi Algoritma**

**Cyberpunk 2077 Hacking Minigame Solver**

Dipersiapkan oleh:  
Pradipta Rafa Mahesa (13522162)

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung 40132

## Daftar Isi

1	Algoritma Brute Force	2
2	Source Program	3
3	Data Test	4
	3.1 File Input	4
	3.2 Randomize Input	5
4	Repository	5
<b>5</b>	<b>Lampiran</b>	<b>6</b>

# 1 Algoritma Brute Force

Algoritma ini mencoba-coba setiap kemungkinan yang dapat terjadi. Setiap kali ada buffer yang dapat memperoleh reward yang lebih besar, variabel yang menyimpan reward maximum, isi buffernya, dan koordinat akan diubah.

Pertama program akan meloop sebanyak kolom untuk menentukan titik awal dari pergerakan. Didalam loop tersebut akan diperoleh isi buffer dan koordinatnya berdasarkan movement\_sequence. Apabila reward yang didapat dari isi buffer tersebut lebih besar dari yang sebelumnya maka akan disimpan nilai, isi buffer, dan koordinatnya. Setelah itu pada titik yang sama, akan digunakan movement\_sequence selanjutnya. Movement sequence akan berubah terus hingga sama seperti last\_movement\_sequence yang diperoleh dari fungsi move\_seq (source program, function.py, nomor 4). Saat movement\_sequence sudah sama seperti last\_movement\_sequence, titik awal akan bergeser ke kanan.

Hal ini dilakukan sampai semua kemungkinan titik awal, dan movement\_sequence terpenuhi atau sudah didapatnya sebuah isi buffer dengan nilai maksimum, yang mana nilai maksimum didapat dari menambahkan setiap reward sequence yang positif.

# 2 Source Program

1). baca\_input.py

berisi program untuk membaca file .txt sebagai input

No	Code	Deskripsi
1	<pre>def baca_file(x):     file = open(x)     buffer_size = int(file.readline())     r2 = file.readline().split()     row = int(r2[0])     col = int(r2[1])     matrix = [" " for i in range(row)]     for i in range(row):         matrix[i] = file.readline().split()     number_of_seq = int(file.readline())     seq = [" " for i in range(number_of_seq)]     seq_reward = [0 for i in range(number_of_seq)]     for i in range(number_of_seq):         seq[i] = file.readline().split()         seq_reward[i] =</pre>	membaca inputan dari file x .txt

	<pre> int(file.readline())     return buffer_size,matrix,row,col,seq,seq_reward </pre>	
--	--	--

2). util.py  
berisi kode-kode utility, untuk membantu kode lain

No	Code	Deskripsi
1	<pre> def isSubset(seq,array): # mencari apakah seq adalah subset dari array     found = False     i = 0     k = 0     while (found == False and i &lt; len(array)):         if (seq[k] == array[i] and found == False):             k+=1             i+=1         else:             i+=1             k = 0         if (k == len(seq)):             found = True     return found </pre>	Mencari apakah seq merupakan subset berderet dari array
2	<pre> def isOdd(x):     if x%2 == 1:         return True     else:         return False </pre>	Mencari apakah x ganjil
3	<pre> def is_unique(array):     a = np.unique(array)     if len(array) == len(a):         return True     else:         return False </pre>	Mencari apakah suatu array diisi dengan elemen unik semua (memanfaatkan numpy)

4	<pre>def idx_to_digit(col,a,b):     d = a*col + b     return d</pre>	mengubah index (row,col) menjadi digit
5	<pre>def dig_to_idx(d,col):     y = d//col     x = d%col     return x,y</pre>	mengubah digit menjadi koordinat (x,y)
6	<pre>def clear_file(file):     f = open(file,"w")     f.write("")     f.close()</pre>	Menghapus isi suatu file .txt

3). function.py  
berisi fungsi-fungsi utama dari algoritma (mengimport util.py)

No	Code	Deskripsi
1	<pre>def get_cords_token(matrix,Initial_point,movement_sequence):     current_row = 0     current_col = Initial_point     cord = idx_to_digit(len(matrix[0]),current_row,current_col)     cords = [cord]     token = str(matrix[current_row][current_col])     for i in range (len(movement_sequence)):         if isOdd(i):             current_col = (current_col + movement_sequence[i])%len(matrix[0])         else:             current_row = (current_row + movement_sequence[i])%len(matrix)         token += " " + str(matrix[current_row][current_col])         cord = idx_to_digit(len(matrix[0]),current_row,current_col)         cords.append(cord)     return cords,token</pre>	Menghasilkan array koordinat dan buffer berdasarkan titik awal dan array movement_sequence (pergerakan titik)
2	<pre>def get_max_reward(seq_reward):     total = 0     for i in range(0, len(seq_reward)):         if seq_reward[i] &gt; 0:             total += seq_reward[i]     return total</pre>	Menghasilkan reward maximum yang dapat diperoleh

3	<pre>def next_move_seq(mov_seq,row,col,id):     if isOdd(id):         max = col-1     else:         max = row-1      if mov_seq[id] == max and id != len(mov_seq)-1:         next_move_seq(mov_seq,row,col,id+1)         mov_seq[id] = 1     else:         mov_seq[id] +=1     return mov_seq</pre>	Menghasilkan array move sequence selanjutnya
4	<pre>def mov_seq(buffer_size,row,col):     ms = [1 for i in range (buffer_size-1)]     lms = [0 for i in range(buffer_size-1)]     for i in range(len(lms)):         if isOdd(i):             lms[i] = col-1         else:             lms[i] = row -1     return ms,lms</pre>	menghasilkan move sequence awal dan move sequence akhir
5	<pre>def rewarding(seq,seq_reward,token):     reward = 0     for i in range(len(seq)):         if isSubset(seq[i],token):             reward += seq_reward[i]     return reward</pre>	Menghasilkan reward yang diperoleh suatu isi buffer
6	<pre>def output(value,buffer,cords,col):     print("")     print("Solusi : ")</pre>	Menampilkan solusi

	<pre> print(value) if value &gt; 0:     print(buffer)     print_cords(cords,col) </pre>	
7	<pre> def print_cords(cords,col):     for i in range(len(cords)):         a,b = dig_to_idx(cords[i],col)         print(str(a+1)+",",b+1) </pre>	Menampilkan koordinat dari elemen dalam buffer
8	<pre> def save(value,buffer,cords,col,time):     print("")     print("")     Simpan = input("Apakah ingin menyimpan solusi?(y/n)")     while Simpan != "y" and Simpan != "n":         print("maaf tolong jawab dengan y atau n")         Simpan = input("Apakah ingin menyimpan solusi?(y/n)")     if Simpan == "y":         file = input("Nama file: ")         clear_file(file)         f = open(file,"a")         f.write(str(value)+"\n")         f.write(str(buffer)+"\n")         for i in range (len(cords)):             x,y = dig_to_idx(cords[i],col)             f.write(str(x+1)+", "+str(y+1)+"\n")         f.write(str(time)+" ms") </pre>	fungsi untuk menyimpan solusi

#### 4). main.py

Berisi program utama yang menggunakan semua fungsi diatas

```

from function import *
from baca_input import *
from matrix_generator import*
import time

print("Cyberpunk 2077 Breach Protocol Solver")
print("Pilih cara menginput")
print("1). File.txt")
print("2). Randomize")
cara_input = int(input("(1/2) : "))
while cara_input != 1 and cara_input != 2:

```

```

        print("maaf tidak ada pilihan tersebut, mohon untuk memilih dari
pilihan dibawah ini")
        print("1). File.txt")
        print("2). Randomize")
        cara_input = input("(1/2) : ")

if cara_input == 1:
    file = input("Nama file input : ")
    buffer_size,matrix,row,col,seq,seq_reward = baca_file(file)
else:
    buffer_size,matrix,row,col,seq,seq_reward = generate()

max = 0
max_cords = []
max_token = ""
max_possible = get_max_reward(seq_reward)

start = time.time()

for i in range(len(matrix[0])):
    movement_sequence,last_mov_seq = mov_seq(buffer_size,row,col)
    while movement_sequence != last_mov_seq and max != max_possible:
        cords,token = get_cords_token(matrix,i,movement_sequence)
        if is_unique(cords):
            tk = token.split()
            reward = rewarding(seq,seq_reward,tk)
            if reward > max:
                max = reward
                max_token = token
                max_cords = cords
        movement_sequence = next_move_seq(movement_sequence,row,col,0)

output(max,max_token,max_cords,col)

end = time.time()
timer = (end-start)*1000
print("")
print("")
print(timer,"ms")

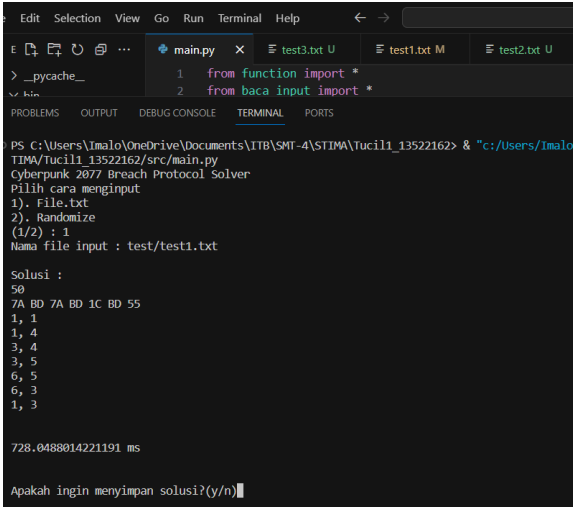
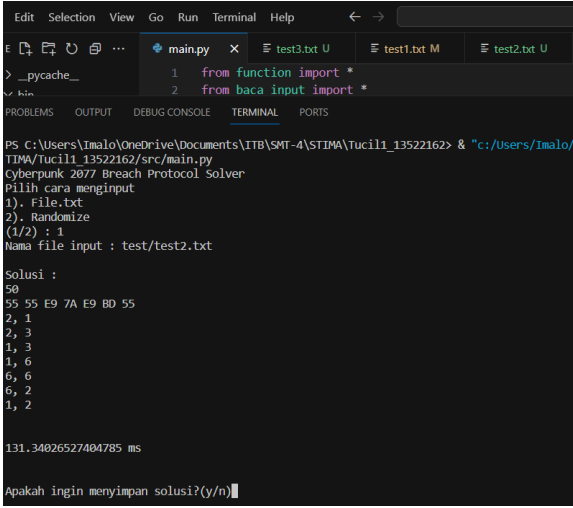
save(max,max_token,max_cords,col,timer)

```



## 3 Data Test

### 3.1 File Input

No	Input	Output
1	<pre>1 7 2 6 6 3 7A 55 E9 E9 1C 55 4 7A 7A 1C 7A E9 55 5 55 1C 1C 55 E9 BD 6 BD 1C 7A 1C 55 BD 7 BD 55 BD 7A 1C 1C 8 1C 55 55 7A 55 7A 9 3 10 BD E9 1C 11 15 12 BD 7A BD 13 20 14 BD 1C BD 55 15 30</pre>	 <pre>PS C:\Users\Imalo\OneDrive\Documents\ITB\SMI-4\STIMA\Tucil1_13522162&gt; &amp; "c:/Users/Imalo/ TUMA/Tucil1_13522162/src/main.py Cyberpunk 2077 Breach Protocol Solver Pilih cara menginput 1). File.txt 2). Randomize (1/2) : 1 Nama file input : test/test1.txt  Solusi : 50 7A BD 7A BD 1C BD 55 1, 1 1, 4 3, 4 3, 5 6, 5 6, 3 1, 3  728.0488014221191 ms  Apakah ingin menyimpan solusi?(y/n)</pre>
2	<pre>1 7 2 6 6 3 BD 55 7A 7A 7A 1C 4 55 1C 55 7A 7A BD 5 E9 55 1C 55 55 55 6 1C 55 7A 7A 55 1C 7 1C 55 55 1C 55 BD 8 7A 1C 7A BD BD E9 9 3 10 E9 7A E9 11 15 12 E9 BD 13 15 14 BD 55 15 20</pre>	 <pre>PS C:\Users\Imalo\OneDrive\Documents\ITB\SMI-4\STIMA\Tucil1_13522162&gt; &amp; "c:/Users/Imalo/ TUMA/Tucil1_13522162/src/main.py Cyberpunk 2077 Breach Protocol Solver Pilih cara menginput 1). File.txt 2). Randomize (1/2) : 1 Nama file input : test/test2.txt  Solusi : 50 55 55 E9 7A E9 BD 55 2, 1 2, 3 1, 3 1, 6 6, 6 6, 2 1, 2  131.34026527404785 ms  Apakah ingin menyimpan solusi?(y/n)</pre>

3

```

Go Run Terminal Help
main.py test3.txt U test1.txt M test2.txt U output.txt

1 10
2 6 6
3 GG 78 45 78 GG 45
4 78 78 45 GG GG GG
5 78 78 45 45 45 78
6 45 45 45 GG 78 45
7 45 45 78 45 78 45
8 GG GG 78 GG 45 45
9 7
10 45 GG 45
11 40
12 78 GG GG
13 -50
14 45 45 GG
15 -15
16 78 45 45
17 -25
18 GG GG
19 35
20 78 78 GG
21 -40
22 GG GG 45
23 40
24
25

```

```

File Edit Selection View Go Run Terminal Help
main.py test3.txt U test1.txt M test2.txt U output.txt
> _pycache_
1 from function import *
2 from baca_input import *

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Cyberpunk 2077 Breach Protocol Solver
Pilih cara menginput
1). File.txt
2). Randomize
(1/2) : 1
Nama file input : test\test3.txt

Solusi :
115
GG GG 45 GG 45 GG 78 78 45
1, 1
1, 6
5, 6
5, 1
6, 1
6, 2
1, 2
1, 3
2, 3
2, 4

1.0287761688232422 ms

Apakah ingin menyimpan solusi?(y/n)y
C:\Users\Imalo\OneDrive\Documents\ITB\SMF-4\STIPA\tucil1_13522162>

```

### 3.2 Randomize Input

No	Input	Output
1	<pre> PS C:\Users\Imalo\OneDrive\Documents\ITB\SMF-4\STIPA&gt; cd ..\Tucil1_13522162\src\main.py Cyberpunk 2077 Breach Protocol Solver Pilih cara menginput 1). File.txt 2). Randomize (1/2) : 2 5 BD 1C 7A 55 E9 7 6 6 3 4 </pre>	<pre> Matrix : 55 BD 55 E9 1C 55 7A 7A 55 7A 7A 1C E9 BD 7A E9 E9 BD 7A E9 7A E9 E9 E9 55 7A BD 7A 7A 7A 7A 55 BD 1C E9 55  Sequence : 1C BD 55 55 40 1C 1C BD 20 7A BD BD -10  Solusi : 40 55 55 7A 1C BD 55 55 1, 1 1, 5 4, 5 4, 6 3, 6 3, 1 6, 1  706.9578170776367 ms  Apakah ingin menyimpan solusi?(y/n) </pre>

```
PS C:\Users\Imalo\OneDrive
TIMA/Tucil1_13522162/src/m
Cyberpunk 2077 Breach Prot
Pilih cara menginput
1). File.txt
2). Randomize
(1/2) : 2
5
BD 1C 7A 55 E9
4
6 9
5
3
```

```
Matrix :
E9 BD 55 E9 E9 1C BD 1C E9
E9 7A BD 1C E9 E9 1C 1C 7A
7A 55 55 1C 55 E9 E9 E9 BD
BD BD 1C BD 55 1C BD 1C 55
BD 1C BD 55 1C 55 1C E9 BD
7A 55 BD 1C 55 55 7A 55 7A

Sequence :

1C E9
10
55 7A
5
E9 1C 55
10
7A E9
25
BD E9 BD
15

Solusi :
35
1C E9 7A E9
6, 1
6, 3
1, 3
1, 1

36.71693801879883 ms

Apakah ingin menyimpan solusi?(y/n)
```

```
PS C:\Users\lmallo\OneDrive\Documents\ITB\
TIMA\Tucil1_13522162\src\main.py
Cyberpunk 2077 Breach Protocol Solver
Pilih cara menginput
1). File.txt
2). Randomize
(1/2) : 2
10
AA GG JK PP 69 42 17 4F 6G H3
10
2 9
8
3
```

```

Matrix :
H3 JK AA 4F JK JK 42 4F 17
69 AA 4F PP 4F PP H3 JK H3

Sequence :

17 AA
0
42 AA GG
-40
H3 GG
-25
PP 69
-5
PP 4F
30
JK H3 AA
-5
6G 17 JK
40
6G JK H3
-5

Solusi :
30
H3 69 AA JK AA 4F PP 4F JK 4F
1, 1
1, 2
2, 2
2, 1
3, 1
3, 2
4, 2
4, 1
5, 1
5, 2

390.40350914001465 ms

```

## 4 Repository

[https://github.com/Rapa285/Tucil1\\_13522162.git](https://github.com/Rapa285/Tucil1_13522162.git)

## 5 Lampiran

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2. Program berhasil dijalankan	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3. Program dapat membaca masukan berkas .txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4. Program dapat menghasilkan masukan secara acak	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. Solusi yang diberikan program optimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6. Program dapat menyimpan solusi dalam berkas .txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7. Program memiliki GUI	<input type="checkbox"/>	<input checked="" type="checkbox"/>