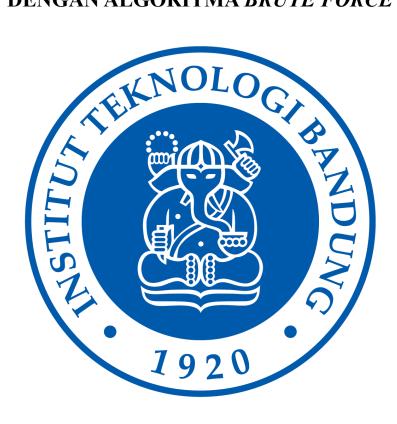
# LAPORAN TUGAS KECIL 1 IF 2211 STRATEGI ALGORITMA PENYELESAIAN "CYBERPUNK 2077 BREACH PROTOCOL" DENGAN ALGORITMA BRUTE FORCE



# DISUSUN OLEH:

**Aland Mulia Pratama** 

13522124

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG 2024

# **DAFTAR ISI**

BAB	I	2
DESI	KRIPSI PROGRAM	2
BAB II		
ALGORITMA PROGRAM  BAB III	3	
BAB III		
		8
		A.
В.	Masukan dari file.txt	9
C.	Masukan secara acak dengan ketentuan sesuai spesifikasi	11

# BAB I DESKRIPSI PROGRAM



Gambar 1. Permainan Breach Protocol pada Cyberpunk 2077

Breach Protocol dalam Cyberpunk 2077 adalah sebuah minigame yang mensimulasikan proses meretas jaringan lokal yang dilindungi oleh ICE (Intrusion Countermeasures Electronics). Ini merupakan bagian dari gameplay di Cyberpunk 2077 di mana pemain harus memecahkan kode dan mengelola protokol keamanan untuk mengakses sistem yang terproteksi. Komponen yang terdapat pada permainan ini antara lain adalah:

- 1. Token terdiri dari dua karakter alfanumerik seperti E9, BD, dan 55.
- 2. Matriks terdiri atas token-token yang akan dipilih untuk menyusun urutan kode.
- 3. Sekuens sebuah rangkaian token (dua atau lebih) yang harus dicocokkan.
- 4. Buffer jumlah maksimal token yang dapat disusun secara sekuensial.

#### Aturan permainan Breach Protocol antara lain:

- 1. Pemain bergerak dengan pola horizontal, vertikal, horizontal, vertikal (bergantian) hingga semua sekuens berhasil dicocokkan atau buffer penuh.
- 2. Pemain memulai dengan memilih satu token pada posisi baris paling atas dari matriks.
- 3. Sekuens dicocokkan pada token-token yang berada di buffer.
- 4. Satu token pada buffer dapat digunakan pada lebih dari satu sekuens.
- 5. Setiap sekuens memiliki bobot hadiah atau reward yang variatif.
- 6. Sekuens memiliki panjang minimal berupa dua token.

Deskripsi program *Breach Protocol Solver* dalam sebuah minigame di video game Cyberpunk 2077 disusun berdasarkan <u>spesifikasi tugas</u> berikut.

IF - 2211 S	Strategi Algoritma	2
-------------	--------------------	---

#### BAB II ALGORITMA PROGRAM

Program *Breach Protocol Solver* untuk minigame dalam *video game* Cyberpunk 2077 menggunakan algoritma brute-force untuk menemukan solusi dengan reward yang paling optimal sesuai dengan *buffer size* yang dimiliki. Berikut adalah Langkah-langkah yang digunakan pada program yaitu:

- 1. Program menerima masukan dari *user*. User dapat memberikan dua tipe masukan yaitu input dari file.txt (folder config) atau input secara acak sesuai dengan ketentuan pada spesifikasi tugas. Pada masukan secara acak, program akan memiliki kecenderungan untuk menghasilkan reward yang lebih tinggi untuk sekuens dengan kombinasi lebih banyak.
- 2. Mencari segala kemungkinan lintasan dan melakukan verifikasi terhadap koordinat supaya memenuhi syarat seperti [(a, 1), (a, b), (c, b), (c, d), ...].
- 3. Ketika urutan cocok dengan node yang ditargetkan, tambah skor sementara sebesar 1. Jika nilai yang salah dicocokkan dengan urutan teratas saat ini, skor sementara dikurangi 1 (Skor tidak boleh sampai 0).
- 4. Mengembalikan (return) baris/kolom berikutnya yang tersedia untuk belokan dan koordinat tertentu. Jika giliran pertama indeksnya adalah 0 maka next\_line akan mengembalikan baris pertama. Untuk putaran kedua, kolom ke-n akan dikembalikan, dengan n menjadi baris koordinat.
- 5. Dari semua kemungkinan lintasan yang dilalui pada langkah 2, lewati koordinat jika sudah pernah dikunjungi dan jalur lengkap yang memenuhi syarat ditambahkan ke list bernama 'completed\_paths'. Proses ini memerlukan penanganan khusus untuk matriks persegi dan yang bukan matriks persegi dikarenakan proses penentuan kandidat koordinat yang berbeda.
- 6. Mencari nilai maksimum dari segala kemungkinan lintasan yang dilalui dengan koordinat yang telah diverifikasi menggunakan fungsi bawaan yaitu max().
- 7. Sekuens yang diketahui berdasarkan masukan dari *user* akan dicek apakah sekuens tersebut merupakan sublist dari lintasan yang memenuhi pada langkah ke 6. Apabila sekuens merupakan sublist dari lintasan tersebut maka reward pada sekuens tersebut ditambahkan ke total hadiah yang akan diperoleh nantinya.
- 8. Program akan memperbaiki format keluaran total hadiah maksimal, koordinat sumbuXY, urutan token lintasan maksimal, beserta waktu eksekusi program agar sesuai dengan ketentuan spesifikasi tugas.
- 9. Masukan *user* yang memiliki reward maksimal (tidak bernilai 0) dapat menyimpan keluaran file dengan format.txt yang akan disimpan pada folder test.

Dengan algoritma ini, *user* dapat mendapatkan solusi optimal dari permainan Breach Protocol minigame dalam *video game* Cyberpunk 2077 secara cepat dan tepat. Segala kemungkinan lintasan yang mungkin dilalui dalam matriks telah diverifikasi sehingga mengurangi jumlah lintasan yang harus ditangani oleh program.

Pranala Repository: <a href="https://github.com/alandmprtma/Tucil1\_13522124">https://github.com/alandmprtma/Tucil1\_13522124</a>

IF - 2211 Strategi Algoritma	3
------------------------------	---

## BAB III SOURCE CODE PROGRAM

4

```
def __init__(self, coords=[]):
    self.coords = coords
       if any(map(lambda coord: coord in self.coords, other.coords)):
    raise DuplicatedCoordinateException()
       self.reward_level = reward_level
self.buffer_size = buffer_size
```

## BAB IV EKSPERIMEN

Berikut adalah beberapa contoh keluaran program Cyberpunk 2077 Breach Protocol Solver:

- A. Masukan yang tidak memiliki Solusi
  - 1) Percobaan Pertama

```
3
5 5
55 1C 1C 1C 55
BD 7A 55 1C 7A
7A BD 1C BD 1C
E9 1C 55 BD 1C
1C 1C BD E9 BD
1
55 BD 7A BD
160
```

Gambar 2. Masukan file.txt yang tidak memiliki solusi percobaan pertama

```
The protocol has been successfully loaded!
buffer size: 3

MATRIX OF TOKENS
55 1C 1C 1C 55
BD 7A 55 1C 7A
7A BD 1C BD 1C
E9 1C 55 BD 1C
1C 1C BD E9 BD

SEQUENCES | REWARD LEVELS
['55', 'BD', '7A', 'BD'] | 160

Please wait for a moment, the program is currently solving the breach protocol...

There is no sequence fulfilled with every possible step!
1 ms
```

Gambar 3. Keluaran yang tidak memiliki solusi percobaan pertama

## 2) Percobaan Kedua

```
2
4 4
BD BD 7A 55
1C 55 55 55
7A 55 BD 55
BD 7A E9 55
3
BD E9
60
55 BD
100
55 55 55
```

Gambar 4. Masukan file.txt yang tidak memiliki solusi percobaan kedua

```
The protocol has been successfully loaded!
buffer size: 2

MATRIX OF TOKENS
BD BD 7A 55
1C 55 55 55
7A 55 BD 55
BD 7A E9 55

SEQUENCES | REWARD LEVELS
['BD', 'E9'] | 60
['55', 'BD'] | 100
['55', '55', '55'] | 60

Please wait for a moment, the program is currently solving the breach protocol...

There is no sequence fulfilled with every possible step!
0 ms
```

Gambar 5. Keluaran yang tidak memiliki solusi percobaan kedua

#### B. Masukan dari file.txt

#### 1) Percobaan Pertama

```
4
4 3
1C BD BD BD
E9 7A 55 55
1C 7A 55 E9
3
7A 1C BD
150
BD 55
100
E9 E9 7A
60
```

Gambar 6. Masukan file.txt percobaan pertama

```
The protocol has been successfully loaded!
buffer size: 4

MATRIX OF TOKENS
1C BD BD BD
1E9 7A 55 55
1C 7A 55 E9

SEQUENCES | REWARD LEVELS
['7A', '1C', 'BD'] | 150
['8D', '55'] | 100
['E9', 'E9', '7A'] | 60

Please wait for a moment, the program is currently solving the breach protocol...

100
BD BD BD 55
2 1
3 1
4 1
4 2
1 ms
```

Gambar 7. Keluaran Program berdasarkan masukan file.txt percobaan pertama

#### 2) Percobaan Kedua

```
7
3 3
E9 7A E9
55 7A 55
BD 7A BD
3
BD BD E9 7A
160
55 7A BD 55
200
BD 1C E9
120
```

Gambar 8. Masukan file.txt percobaan kedua

```
The protocol has been successfully loaded!
buffer size: 7

MATRIX OF TOKENS
E9 7A E9
55 7A 55
BD 7A BD

SEQUENCES | REWARD LEVELS
['BD', 'BD', 'E9', '7A'] | 169
['55', '7A', 'BD', '55'] | 200
['BD', '1C', 'E9'] | 120

Please wait for a moment, the program is currently solving the breach protocol...
160
69
BD BD E9 7A 7A 55
1
1
3
3
3
1
1
2
1
2
1
1
8
```

Gambar 9. Keluaran Program berdasarkan masukan file.txt percobaan kedua

# 3) Percobaan Ketiga

IF - 2211 Strategi Algoritma	10
------------------------------	----

```
55 1C BD BD BD 55
BD BD 7A 7A 1C E9
1C 55 7A 7A BD 1C
1C BD 55 1C
120
80
```

Gambar 10. Masukan file.txt percobaan ketiga

```
The protocol has been successfully loaded!
MATRIX OF TOKENS
55 1C BD BD BD 55
55 1C 1C BD 55 BD
1C 55 7A 7A BD 1C
SEQUENCES | REWARD LEVELS
['1C', 'BD', '55', '1C'] |
['55', '55'] | 80
['E9', 'E9', '55', '7A'] |
Please wait for a moment, the program is currently solving the breach protocol...
```

Gambar 11. Keluaran Program berdasarkan masukan file.txt percobaan ketiga

- C. Masukan secara acak dengan ketentuan sesuai spesifikasi
  - 1) Percobaan Pertama

```
The protocol has been successfully loaded!
buffer size: 7

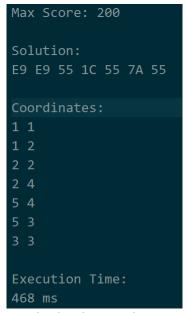
MATRIX OF TOKENS
E9 7A 7A BD BD 7A
E9 55 7A BD E9 1C
E9 7A 55 1C 7A 55
7A 1C 7A E9 55 E9
7A 55 7A 55 55 BD
1C E9 1C 1C E9 BD

SEQUENCES | REWARD LEVELS
['1C', '55', '7A', '55'] | 200

Please wait for a moment, the program is currently solving the breach protocol...
200
E9 E9 55 1C 55 7A 55
1 1
2 2
2 2
4 5
4 5
3 3
3 3

468 ms
```

Gambar 12. Masukan secara acak percobaan pertama



Gambar 13. Keluaran program berdasarkan masukan secara acak percobaan pertama

#### 2) Percobaan Kedua

IF - 2211 Strategi Algoritma	12
------------------------------	----

```
The protocol has been successfully loaded!
buffer size: 4

MATRIX OF TOKENS
55 7A BD 7A
1C BD 55 7A
55 55 1C BD
E9 55 7A 55
55 BD 1C 7A
E9 E9 7A 55

SEQUENCES | REWARD LEVELS
['BD', '55'] | 20
['7A', '7A'] | 40

Please wait for a moment, the program is currently solving the breach protocol...

40
7A BD 7A 7A
2 1
3 1
4 1
4 2
1 ms
```

Gambar 14. Masukan secara acak percobaan kedua

```
Max Score: 40

Solution:
7A BD 7A 7A

Coordinates:
2 1
3 1
4 1
4 2

Execution Time:
1 ms
```

Gambar 15. Keluaran program berdasarkan masukan secara acak percobaan kedua

## 3) Percobaan Ketiga

IF - 2211 Strategi Algoritma	13
------------------------------	----

```
The protocol has been successfully loaded!
buffer size: 4

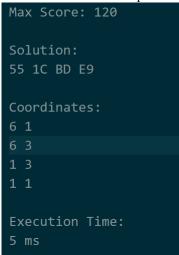
MATRIX OF TOKENS
E9 E9 E9 55 7A 55
55 7A E9 E9 1C 1C
BD 55 55 55 1C 1C
55 BD BD E9 1C E9
1C 55 7A BD 1C 7A
7A BD 7A E9 E9 BD

SEQUENCES | REWARD LEVELS
['1C', 'E9', 'BD'] | 150
['1C', '7A', '55'] | 60
['55', '1C', 'BD', 'E9'] | 120

Please wait for a moment, the program is currently solving the breach protocol...

120
55 1C BD E9
6 1
6 3
1 3
1 1
5 ms
```

Gambar 16. Masukan secara acak percobaan ketiga



Gambar 17. Keluaran program berdasarkan masukan secara acak percobaan ketiga