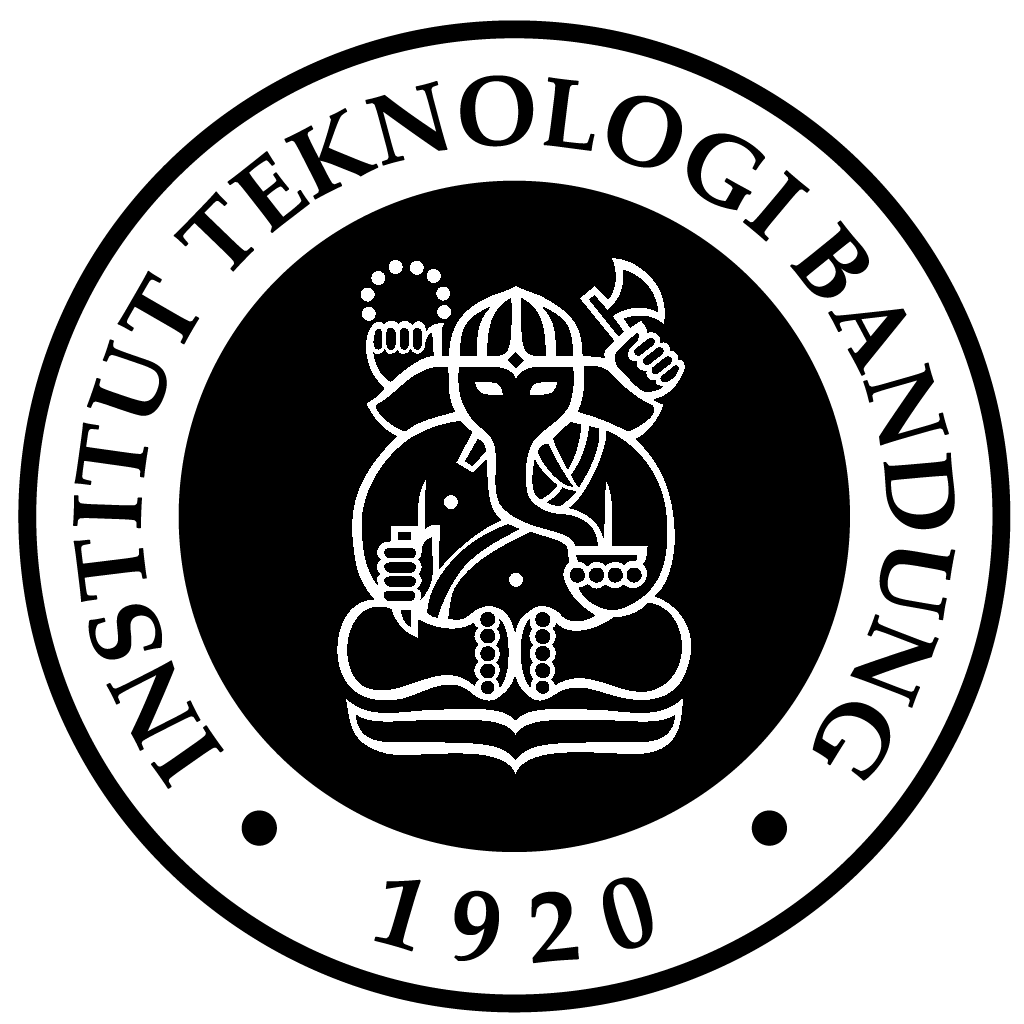
**Laporan Tugas Kecil 1  
IF2211 Strategi Algoritma**

Aplikasi Brute Force dalam Pencarian Sequence Optimal pada Minigame Breach Protocol dari Game Cyberpunk 2077



Disusun oleh:

Akbar Al Fattah 13522036

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

# **Bab 1: Deskripsi Masalah**



Gambar 1 Permainan Breach Protocol

(Sumber: <https://cyberpunk.fandom.com/wiki/Quickhacking>)

**Cyberpunk 2077 Breach Protocol** adalah *minigame* meretas pada permainan video *Cyberpunk 2077*. *Minigame* ini merupakan simulasi peretasan jaringan local dari *ICE (Intrusion Countermeasures Electronics)* pada permainan *Cyberpunk 2077*. Komponen 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.

**Ilustrasi kasus** :

Diberikan matriks sebagai berikut dan ukuran buffernya adalah tujuh

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7A | 55 | E9 | E9 | 1C | 55 |
| 55 | 7A | 1C | 7A | E9 | 55 |
| 55 | 1C | 1C | 55 | E9 | BD |
| BD | 1C | 7A | 1C | 55 | BD |
| BD | 55 | BD | 7A | 1C | 1C |
| 1C | 55 | 55 | 7A | 55 | 7A |

Dengan sekuens sebagai berikut:

1. BD E9 1C dengan hadiah berbobot 15.
2. BD 7A BD dengan hadiah berbobot 20.
3. BD 1C BD 55 dengan hadiah berbobot 30.

Maka solusi yang optimal untuk matriks dan sekuens yang diberikan adalah sebagai berikut:

* Total bobot hadiah : 50 poin
* Total langkah : 6 langkah

A computer screen shot of a diagram

Description automatically generated

Gambar 2 Contoh Solusi

(Sumber: <https://cyberpunk-hacker.com/>)

# **Bab 2: Pemecahan Masalah dengan Brute Force**

Misalkan matriks yang diberikan adalah sebagai berikut:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7A | 55 | E9 | E9 | 1C | 55 |
| 55 | 7A | 1C | 7A | E9 | 55 |
| 55 | 1C | 1C | 55 | E9 | BD |
| BD | 1C | 7A | 1C | 55 | BD |
| BD | 55 | BD | 7A | 1C | 1C |
| 1C | 55 | 55 | 7A | 55 | 7A |

Misalkan panjang buffer maksimal yang diinput adalah 7, banyak sequence adalah 3 dengan daftar sequence dan poin rewardnya adalah sebagai berikut:

1. BD E9 1C. Poin: 15
2. BD 7A BD. Poin: 20
3. BD 1C BD 55 Poin: 30

Algoritma brute force yang digunakan oleh penulis untuk memecahkan masalah ini adalah:

1. Mulai dari sel di baris 1 dan kolom 1 dalam state Horizontal, tandai semua sel yang berada di dalam garis horizontal yang sama untuk dikunjungi nanti.

State: Horizontal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7A (sudah dikunjungi) | 55 | E9 | E9 | 1C | 55 |
| 55 | 7A | 1C | 7A | E9 | 55 |
| 55 | 1C | 1C | 55 | E9 | BD |
| BD | 1C | 7A | 1C | 55 | BD |
| BD | 55 | BD | 7A | 1C | 1C |
| 1C | 55 | 55 | 7A | 55 | 7A |

1. Masukkan token tersebut ke dalam buffer dan tandai sel tersebut sudah dikunjungi.
2. Cek untuk semua sequence, jika sequence adalah substring/subarray dari buffer, tambahkan poin sementara untuk buffer tersebut.
3. Perbarui jawaban yang optimal (jawaban yang optimal adalah jawaban buffer dengan poin terbanyak dan panjang buffer terpendek).
4. Ubah state menjadi “Vertical” dan lakukan hal yang sama pada langkah 1 namun yang ditandai hanyalah sel yang berada di dalam satu garis vertikal

State: Vertical

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7A (sudah dikunjungi) | 55 | E9 | E9 | 1C | 55 |
| 55 | 7A | 1C | 7A | E9 | 55 |
| 55 | 1C | 1C | 55 | E9 | BD |
| BD | 1C | 7A | 1C | 55 | BD |
| BD | 55 | BD | 7A | 1C | 1C |
| 1C | 55 | 55 | 7A | 55 | 7A |

1. Jika sel yang diperiksa sudah pernah dikunjungi sebelumnya, lewati pengulangan langkah 2-4 dan iterasi ke sel selanjutnya sesuai state yang sedang aktif. (artinya, jika state = “Horizontal” iterasi sel selanjutnya hanya dilakukan untuk
2. Jika sel yang diperiksa belum pernah dikunjungi sebelumnya, ulangi langkah 2 sampai 4.
3. Ulangi terus prosedur ini dengan perubahan state secara selang-seling dengan pola Horizontal – Vertical -Horizontal-Vertical….
4. Setelah tiap kali langkah rekursif dilakukan, batalkan semua perubahan

Untuk memudahkan pemahaman, berikut simulasi beberapa langkah di dalam matriks 3 x 3 dengan panjang buffer maksimal 3

Sequence : a. 7A 55, Poin: 10 b. 7A 55 1C, Poin: 25

* 1. State: Horizontal

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 | 1C | 1C |

Buffer: 7A Poin: 0

* 1. State: Vertical (Dilewati karena 1,1 sudah dikunjungi)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 | 1C | 1C |

* 1. State: Vertical

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 (Sudah dikunjungi) | 7A | 1C |
| 55 | 1C | 1C |

Buffer: 7A 55 Poin: 10

* 1. State: Horizontal (sel 1,2 dilewati karena sudah dikunjungi)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 (Sudah dikunjungi) | 7A (Sudah Dikunjungi) | 1C |
| 55 | 1C | 1C |

Buffer: 7A 55 7A Poin:10

* 1. State: Horizontal (karena panjang buffer sudah maksimum)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 (Sudah dikunjungi) | 7A | 1C (sudah dikunjungi) |
| 55 | 1C | 1C |

Buffer: 7A 55 1C Poin: 35

* 1. State: Vertical (backtrack ke state sebelumnya dan terjadi iterasi)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 (Sudah dikunjungi) | 1C | 1C |

Buffer: 7A 55 Poin: 10

* 1. State: Horizontal (sel 1,3 dilewati)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 (Sudah dikunjungi) | 1C (Sudah Dikunjungi) | 1C |

Buffer: 7A 55 1C Poin: 35

* 1. State: Horizontal

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 (Sudah dikunjungi) | 1C | 1C(Sudah dikunjungi) |

Buffer: 7A 55 1C Poin: 35

* 1. State: Vertical (backtrack)

|  |  |  |
| --- | --- | --- |
| 7A (Sudah dikunjungi) | 55 | E9 |
| 55 | 7A | 1C |
| 55 (Sudah dikunjungi) | 1C | 1C |

* 1. State: Horizontal (backtrack dan iterasi selanjutnya)

|  |  |  |
| --- | --- | --- |
| 7A | 55 (Sudah dikunjungi) | E9 |
| 55 | 7A | 1C |
| 55 | 1C | 1C |

Buffer: 55 Poin: 0

* 1. State: Vertical

|  |  |  |
| --- | --- | --- |
| 7A | 55 (Sudah dikunjungi) | E9 |
| 55 | 7A (Sudah dikunjungi) | 1C |
| 55 | 1C | 1C |

Buffer: 55 7A Poin: 0

* 1. State: Horizontal

|  |  |  |
| --- | --- | --- |
| 7A | 55 (Sudah dikunjungi) | E9 |
| 55 (Sudah dikunjungi) | 7A (Sudah dikunjungi) | 1C |
| 55 | 1C | 1C |

Buffer: 55 7A 55 Poin: 10

Dan seterusnya sampai semua kemungkinan buffer tercipta secara brute force dan pasti akan didapatkan jawaban yang paling optimal.

# **Bab 3: Source Code Program**

Semua source code pada program ini ditulis menggunakan bahasa C++.

1. gamematrix.h dan gamematrix.cpp (objek matrix breach protocol)

gamematrix.h

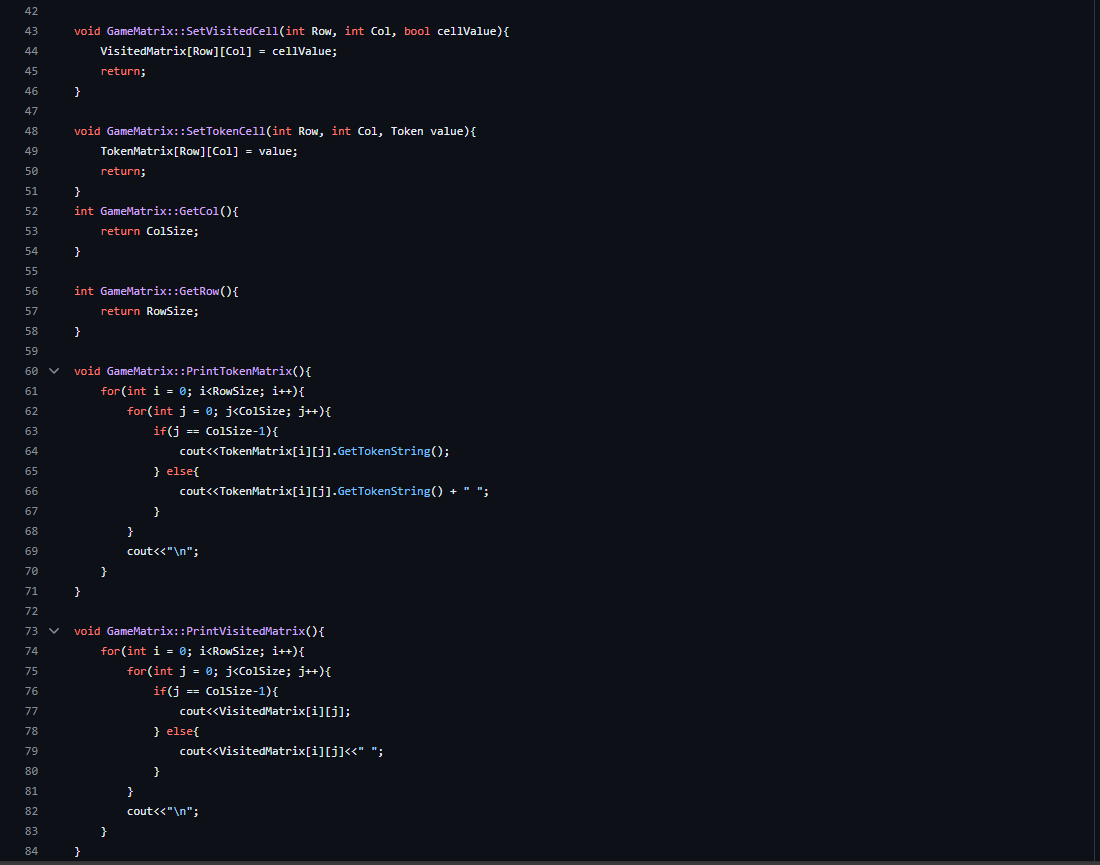
A screenshot of a computer code

Description automatically generated

gamematrix.cpp

A screen shot of a computer program

Description automatically generated



A screen shot of a computer program

Description automatically generated

1. helper.h dan helper.cpp (subprogram pembantu tambahan)

helper.h

A black background with white text

Description automatically generated

helper.cpp

A computer screen shot of white text

Description automatically generated

A screen shot of a computer

Description automatically generated

1. sequence.h dan sequence.cpp (objek sequence)

sequence.h

A computer screen shot of text

Description automatically generated

sequence.cpp

A screenshot of a computer program

Description automatically generated

1. token.h dan token.cpp (objek token)

token.h

A screenshot of a computer

Description automatically generated

token.cpp

A screenshot of a computer program

Description automatically generated

1. main.cpp (Program utama CLI)

A computer screen shot of a computer error

Description automatically generated with medium confidence

A computer screen with text on it

Description automatically generated

A screen shot of a computer error

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer code on a black background

Description automatically generated

# **Bab 4: Uji Coba Program**

Tes 1: Input File

Isi file input (test1.txt):

A screenshot of a computer

Description automatically generated

Hasil ujicoba (output file testout1.txt):

A blue screen with white text

Description automatically generated

Tes 2: Input File

Isi file input (test2.txt):

A screenshot of a computer

Description automatically generated

Hasil ujicoba: (output file testout2.txt):

A screenshot of a computer

Description automatically generated

Tes 3: Input File

Isi file input (test3.txt):

A screenshot of a computer code

Description automatically generated

Hasil ujicoba (output file testout2.txt):

A screenshot of a computer

Description automatically generated

Tes 4: Random Generation

Input CLI:

A screen shot of a computer

Description automatically generated

Hasil ujicoba (output file testrandom1.txt):

A screenshot of a computer

Description automatically generated

Tes 5: Random Generation

Input CLI:

A screenshot of a computer program

Description automatically generated

Hasil ujicoba (output file testrandom2.txt):

A screenshot of a computer

Description automatically generated

Tes 6: Random Generation

Input CLI:

A screenshot of a computer

Description automatically generated

Hasil ujicoba (output file testrandom3.txt):

A screenshot of a computer

Description automatically generated

# **Bab 5: Pranala Repositori dan Lampiran**

Repositori github dari program ini dapat dilihat di pranala berikut: <https://github.com/DeltDev/Tucil1_13522036>

Check list program:

|  |  |  |
| --- | --- | --- |
| **Poin** | **Ya** | **Tidak** |
| 1. Program berhasil dikompilasi tanpa kesalahan | V |  |
| 1. Program berhasil dijalankan | V |  |
| 1. Program dapat membaca masukan berkas .txt | V |  |
| 1. Program dapat menghasilkan masukan secara acak | V |  |
| 1. Solusi yang diberikan program optimal | V |  |
| 1. Program dapat menyimpan solusi dalam berkas .txt | V |  |
| 1. Program memiliki GUI |  | V |