

Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Brute Force



Rafiki Prawhira Harianto

13522065

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2024

Daftar Isi

Daftar Isi	2
Deskripsi Persoalan	3
Algoritma Brute Force	4
Lampiran	5

Deskripsi Persoalan

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 di cocokkan.
4. Buffer – jumlah maksimal token yang dapat disusun secara sekuensial.



Gambar 1 Permainan Breach Protocol

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

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.

Algoritma Brute Force

Dalam pemecahan Cyberpunk 2077 menggunakan pemrograman, digunakan algoritma brute force dengan traversal semua kemungkinan Buffer yang mungkin. Cara traversal semua kemungkinan menggunakan konsep fungsi rekursif, dan mirip dengan konsep traversal tree. Fungsi tersebut, yang diberi nama findAllBuffer, bekerja dengan cara berikut:

1. Basis pertama: Jika buffer sudah penuh, append buffer ke matrix berisi semua buffer
2. Basis kedua: Jika koordinat matrix (yang ada di parameter fungsi) sudah berada di dalam buffer, langkah tidak valid sehingga keluar dari fungsi
3. Rekurens: Menambahkan ukuran buffer, membuat variabel baru yang berisi buffer dengan tambahkan koordinat matrix. Kemudian, jika saatnya bergerak vertikal, traversalkan i dari 1 sampai baris terakhir dan panggil findAllBuffer dengan i baru, atau jika saatnya bergerak horizontal, traversalkan j dari 1 sampai kolom terakhir dan panggil findAllBuffer dengan j baru.

Setelah mencari semua kemungkinan buffer, diperlukan pencarian buffer dengan poin maksimum. Pencariannya menggunakan 3 fungsi, yaitu isSekuensInBuffer, poinInBuffer, dan solver. isSekuensInBuffer menentukan jika sekuens berada di dalam buffer, dengan cara mencocokkan setiap elemen pada sekuens dengan setiap kemungkinan subset berukuran sekuens pada buffer. poinInBuffer menentukan nilai poin pada buffer untuk setiap sekuens yang berada pada buffer. Terakhir, solver menentukan buffer dengan poin maksimum beserta poinnya, dengan traversal setiap buffer pada matrix semua buffer, menentukan elemen pertama sebagai buffer maksimum sementara, dan jika poinInBuffer buffer selanjutnya lebih tinggi akan menggantikan buffer baru menjadi buffer maksimum.

Selain itu, program diperlukan untuk membuat matrix beserta sekuens dan poin-poinnya dengan spesifikasi pengguna secara acak. Menggunakan library random python, matrix didapatkan dengan membuat elemen yang dipilih secara acak dari token spesifikasi untuk setiap baris dan kolom, sekuens didapatkan dengan membuat elemen yang dipilih secara acak dari token spesifikasi seukuran antara 1 sampai sekuens maksimum dikurangi 1 secara acak sebanyak jumlah sekuens, dan poin sekuens dibuat dengan memilih angka antara 0 sampai 100 secara acak untuk setiap sekuens.

Lampiran

A. Tabel checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil dijalankan	✓	
3. Program dapat membaca masukan berkas .txt	✓	
4. Program dapat menghasilkan masukan secara acak	✓	
5. Solusi yang diberikan program optimal	✓	
6. Program dapat menyimpan solusi dalam berkas .txt	✓	
7. Program memiliki GUI		✓

B. Source program python

```
import time
import os
import random

def findAllBuffer(iTemp: int, jTemp: int, currentBufferSize: int, currentBuffer: list[list],
isMoveVertikal: bool):
    if currentBufferSize == bufferSize: #Jika buffer penuh
        if currentBuffer not in allBuffer:
            allBuffer.append(currentBuffer)
    else:
        if ([iTemp, jTemp]) in currentBuffer: #Jika koordinat sudah diakses, move tidak valid
            pass
        else:
            currentBufferSize += 1
            nextBuffer = currentBuffer.copy()
            nextBuffer.append([iTemp, jTemp])
            if isMoveVertikal:
                isMoveVertikal = not isMoveVertikal
                for i in range(0, mBar):
                    findAllBuffer(i, jTemp, currentBufferSize, nextBuffer, isMoveVertikal)
            else:
                isMoveVertikal = not isMoveVertikal
                for j in range(0, mKol):
                    findAllBuffer(iTemp, j, currentBufferSize, nextBuffer, isMoveVertikal)

def isSekuensInBuffer(buffer, sekuens):
    n = len(buffer)
    m = len(sekuens)
```

```

    for i in range(n - m + 1):
        for j in range(m):
            if matrix[buffer[i + j][0]][buffer[i + j][1]] != sekuens[j]:
                break
            else:
                return True
        return False

def poinInBuffer(buffer):
    poin = 0
    for case in range(len(sekuensMatrix)):
        if (isSekuensInBuffer(buffer, sekuensMatrix[case])):
            poin += poinArray[case]
    return poin

def solver(allBuffer):
    for i in range(len(allBuffer)):
        if i == 0:
            maxBuffer = allBuffer[0]
            maxPoin = poinInBuffer(maxBuffer)
        else:
            if poinInBuffer(allBuffer[i]) > maxPoin:
                maxBuffer = allBuffer[i]
                maxPoin = poinInBuffer(maxBuffer)

    return (maxBuffer, maxPoin)

def randomElementFromArr(array):
    i = random.randrange(len(array))
    return array[i]

print("Selamat datang ke program solver minigame Cyberpunk2077!")
print("1. Melalui Text File")
print("2. Melalui CLI")
while True:
    cara = input("Mau pakai cara mana?: ")
    if cara.isdigit():
        cara = int(cara)
        if cara == 1 or cara == 2:
            break
    print("Cara tidak valid.")

#Input cara text
if cara == 1:
    filename = input("Masukkan nama file (Taruh di dalam folder yang sama): ")
    try:
        f = open(filename, "r")
    except FileNotFoundError:
        print("File tidak ditemukan. ")
        time.sleep(3)

```

```

        exit()
    print()

    bufferSize = int(f.readline())
    mBarKol = f.readline().split(" ")
    mKol = int(mBarKol[0])
    mBar = int(mBarKol[1])
    matrix = [f.readline().rstrip().split(" ") for _ in range(mBar)]

    nSekuens = int(f.readline())
    sekuensMatrix = [[0] for i in range(nSekuens)]
    poinArray = [0 for i in range(nSekuens)]
    for i in range(nSekuens):
        sekuensMatrix[i] = f.readline().rstrip().split(" ")
        poinArray[i] = int(f.readline())

#input cara CLI
else:
    nToken = int(input("Masukkan jumlah token unik: "))
    token = [str(x) for x in input("Masukkan token: ").split()]
    bufferSize = int(input("Masukkan ukuran buffer: "))
    mKol, mBar = map(int, input("Masukkan ukuran matriks: ").split())
    nSekuens = int(input("Masukkan jumlah sekuens: "))
    sekuensSize = int(input("Masukkan ukuran maksimal sekuens: "))
    print()

    matrix = [[randomElementFromArr(token) for j in range(mKol)] for i in range(mBar)]
    sekuensMatrix = [[randomElementFromArr(token) for j in
range(random.randint(1,sekuensSize-1))] for i in range(nSekuens)]
    poinArray = [random.randint(0,100) for i in range(nSekuens)]

    print("Matrix: ")
    for bar in matrix:
        print(*bar)
    print()

    print("Sekuens: ")
    for i in range(nSekuens):
        print(*sekuensMatrix[i])
        print(poinArray[i])
    print()

start = time.time()

allBuffer = []
findAllBuffer(0, 0, 0, [], True)
(maxBuffer, maxPoin) = solver(allBuffer)

end = time.time()

```

```

waktu = str(round((end - start)*1000)) + " ms"

print(maxPoin)

for i in range(bufferSize-1):
    print(matrix[maxBuffer[i][0]][maxBuffer[i][1]], end= " ")
print(matrix[maxBuffer[bufferSize-1][0]][maxBuffer[bufferSize-1][1]])

for i in range(bufferSize):
    print(maxBuffer[i][1]+1, end=", ")
    print(maxBuffer[i][0]+1)

print()
print(waktu)
print()

konfirmasi = input("Apakah ingin menyimpan solusi? (y/n) ")
if konfirmasi == "y": #Jika input n atau bukan y, tidak melakukan penyimpanan
    f2 = open("SolusiCyberpunk2077.txt", "w")
    if cara == 2: #Format output text matrix dan sekuens untuk cara 2
        f2.write("Matrix: \n")
        for i in range(mBar):
            for j in range(mKol):
                f2.write(str(matrix[i][j]))
                f2.write(" ")
            f2.write("\n")
        f2.write("\n")

        f2.write("Sekuens: \n")
        for i in range(nSekuens):
            for j in range(len(sekuensMatrix[i])):
                f2.write(str(sekuensMatrix[i][j]))
                f2.write(" ")
            f2.write("\n")
            f2.write(str(poinArray[i]))
            f2.write("\n")
        f2.write("\n")

        f2.write(str(maxPoin))
        f2.write("\n")

    l = ""
    for i in range(bufferSize-1):
        l += str(matrix[maxBuffer[i][0]][maxBuffer[i][1]]) + " "
    l += str(matrix[maxBuffer[bufferSize-1][0]][maxBuffer[bufferSize-1][1]])
    f2.write(l)
    f2.write("\n")

    for i in range(bufferSize):

```



```

1 = ""
1 += str(maxBuffer[i][1]+1) + ", "
1 += str(maxBuffer[i][0]+1)
f2.write(1)
f2.write("\n")

f2.write("\n")
f2.write(str(waktu))
f2.write("\n")
f2.close()

if cara == 1:
    f.close()

```

C. Test Case (1-3 menggunakan input Text, 4-6 menggunakan input CLI)

1. Case 1

Isi case1.txt:

```

6 6
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
3
BD E9 1C
15
BD 7A BD
20
BD 1C BD 55
30

```

```

D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe
Selamat datang ke program solver minigame Cyberpunk2077!
1. Melalui Text File
2. Melalui CLI
Mau pakai cara mana?: 1
Masukkan nama file (Taruh di dalam folder yang sama): case1.txt

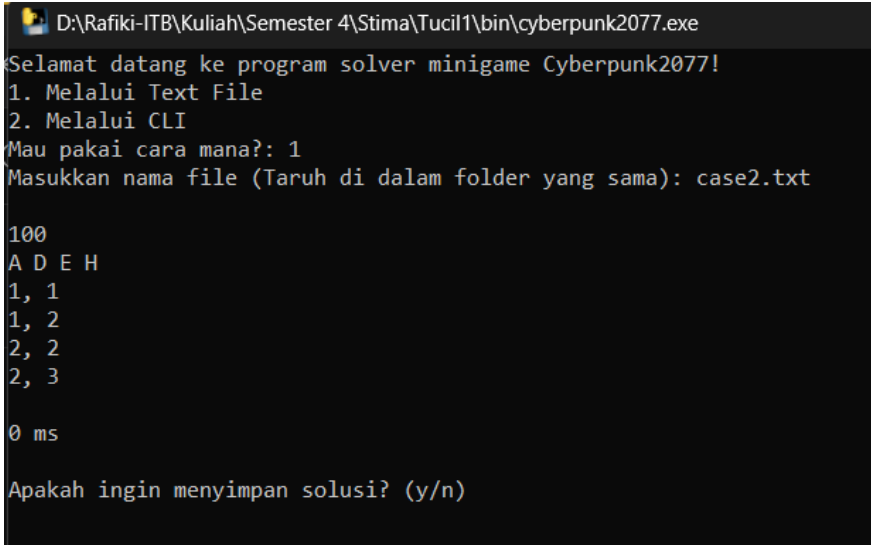
50
7A BD 7A BD 1C BD 55
1, 1
1, 4
3, 4
3, 5
6, 5
6, 3
1, 3

10386 ms

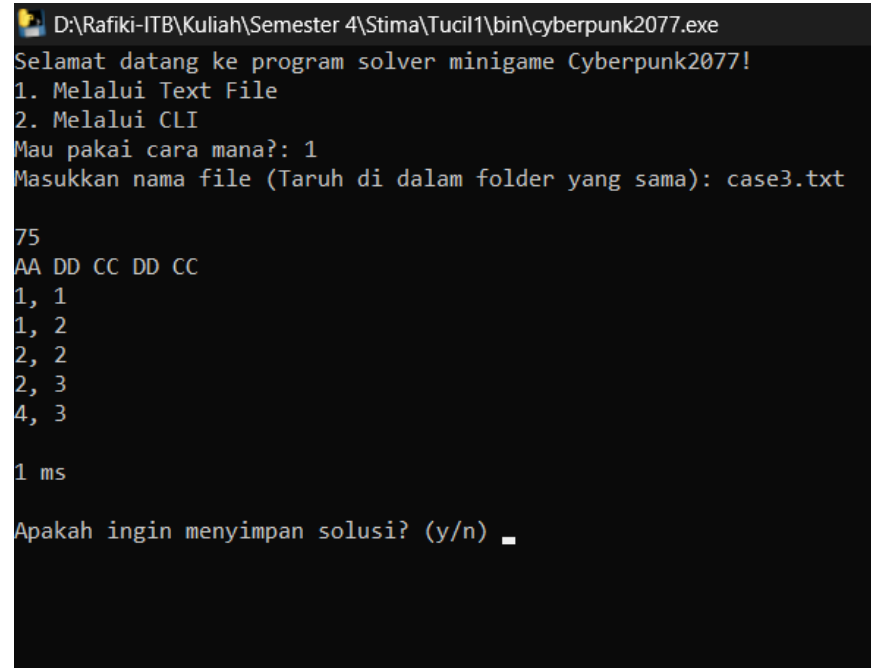
Apakah ingin menyimpan solusi? (y/n)

```

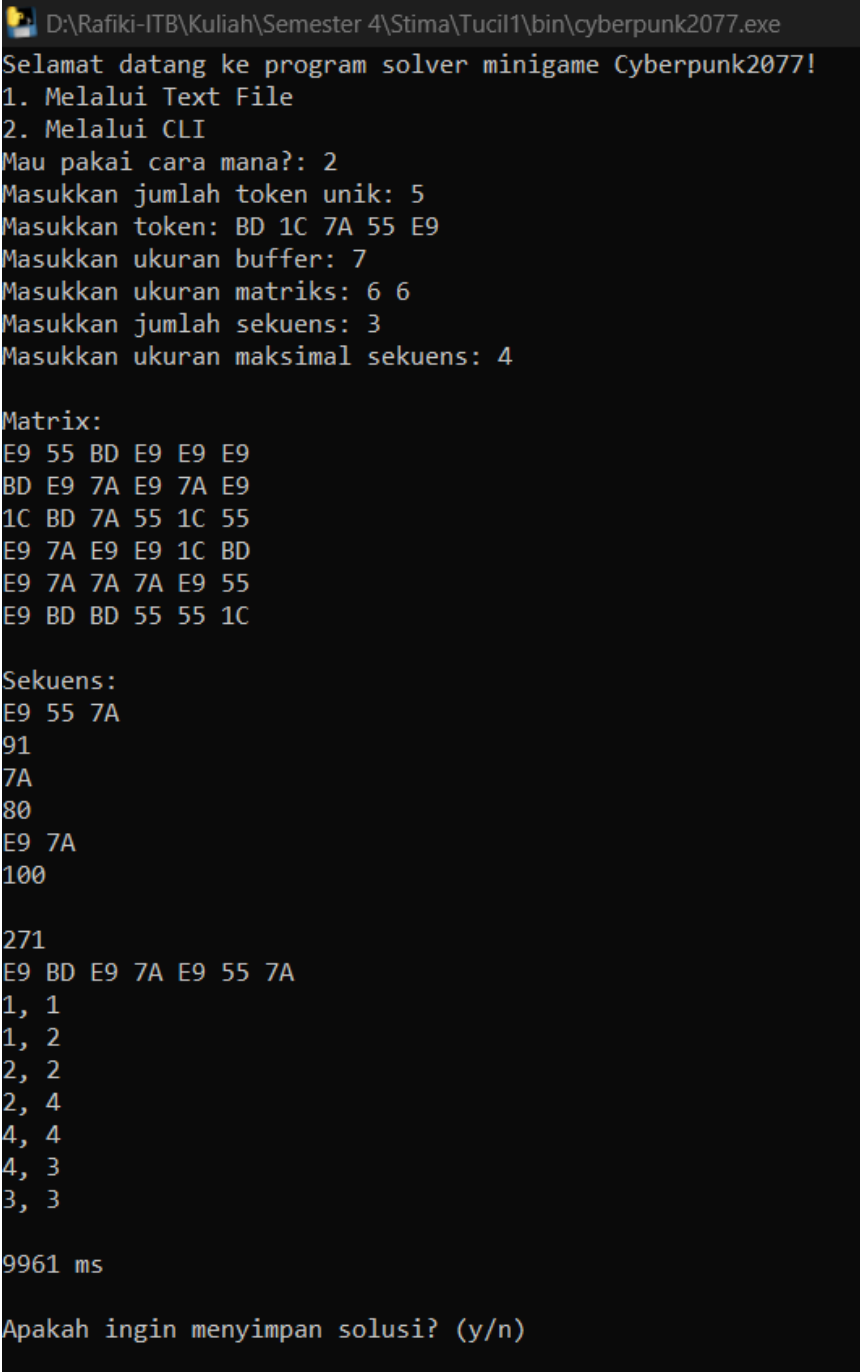
2. Case 2

Isi case2.txt: 4 2 3 A B D E G H 2 A D E B 50 A D E H 100	 <pre> D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe Selamat datang ke program solver minigame Cyberpunk2077! 1. Melalui Text File 2. Melalui CLI Mau pakai cara mana?: 1 Masukkan nama file (Taruh di dalam folder yang sama): case2.txt 100 A D E H 1, 1 1, 2 2, 2 2, 3 0 ms Apakah ingin menyimpan solusi? (y/n) </pre>
---	--

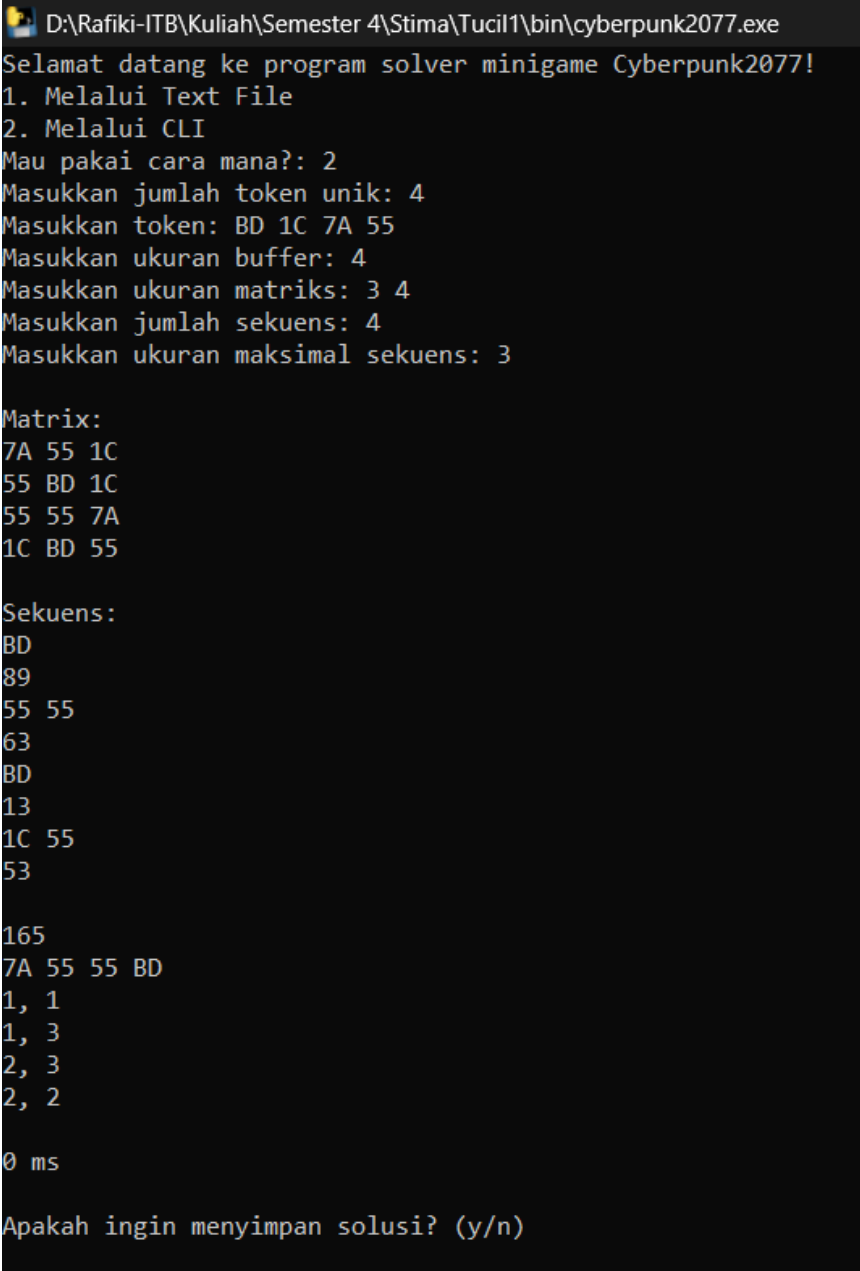
3. Case 3

Isi case3.txt: 5 4 3 AA BB CC DD DD CC BB AA BB DD AA CC 3 AA BB DD 50 AA AA 100 CC DD CC 75	 <pre> D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe Selamat datang ke program solver minigame Cyberpunk2077! 1. Melalui Text File 2. Melalui CLI Mau pakai cara mana?: 1 Masukkan nama file (Taruh di dalam folder yang sama): case3.txt 75 AA DD CC DD CC 1, 1 1, 2 2, 2 2, 3 4, 3 1 ms Apakah ingin menyimpan solusi? (y/n) </pre>
--	--

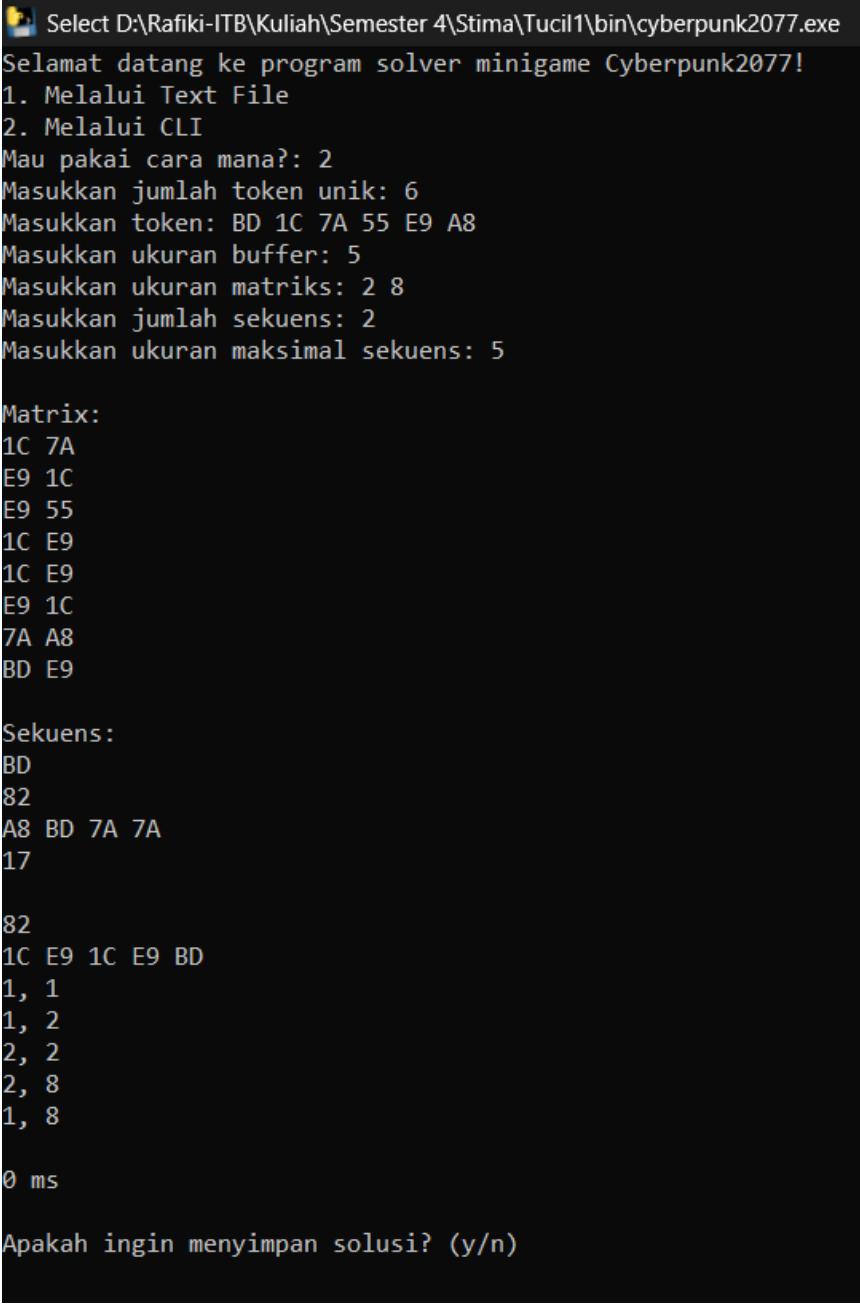
4. Case 4

<p>Input CLI:</p> <p>5</p> <p>BD 1C 7A 55</p> <p>E9</p> <p>7</p> <p>6 6</p> <p>3</p> <p>4</p>	 <pre> D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe Selamat datang ke program solver minigame Cyberpunk2077! 1. Melalui Text File 2. Melalui CLI Mau pakai cara mana?: 2 Masukkan jumlah token unik: 5 Masukkan token: BD 1C 7A 55 E9 Masukkan ukuran buffer: 7 Masukkan ukuran matriks: 6 6 Masukkan jumlah sekuens: 3 Masukkan ukuran maksimal sekuens: 4 Matrix: E9 55 BD E9 E9 E9 BD E9 7A E9 7A E9 1C BD 7A 55 1C 55 E9 7A E9 E9 1C BD E9 7A 7A 7A E9 55 E9 BD BD 55 55 1C Sekuens: E9 55 7A 91 7A 80 E9 7A 100 271 E9 BD E9 7A E9 55 7A 1, 1 1, 2 2, 2 2, 4 4, 4 4, 3 3, 3 9961 ms Apakah ingin menyimpan solusi? (y/n) </pre>
---	--

5. Case 5

Input CLI: 4 BD 1C 7A 55 4 3 4 4 3	 <pre> D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe Selamat datang ke program solver minigame Cyberpunk2077! 1. Melalui Text File 2. Melalui CLI Mau pakai cara mana?: 2 Masukkan jumlah token unik: 4 Masukkan token: BD 1C 7A 55 Masukkan ukuran buffer: 4 Masukkan ukuran matriks: 3 4 Masukkan jumlah sekuens: 4 Masukkan ukuran maksimal sekuens: 3 Matrix: 7A 55 1C 55 BD 1C 55 55 7A 1C BD 55 Sekuens: BD 89 55 55 63 BD 13 1C 55 53 165 7A 55 55 BD 1, 1 1, 3 2, 3 2, 2 0 ms Apakah ingin menyimpan solusi? (y/n) </pre>
--	--

6. Case 6

Input CLI: 6 BD 1C 7A 55 E9 A8 5 2 8 2 5	 <pre> Select D:\Rafiki-ITB\Kuliah\Semester 4\Stima\Tucil1\bin\cyberpunk2077.exe Selamat datang ke program solver minigame Cyberpunk2077! 1. Melalui Text File 2. Melalui CLI Mau pakai cara mana?: 2 Masukkan jumlah token unik: 6 Masukkan token: BD 1C 7A 55 E9 A8 Masukkan ukuran buffer: 5 Masukkan ukuran matriks: 2 8 Masukkan jumlah sekuens: 2 Masukkan ukuran maksimal sekuens: 5 Matrix: 1C 7A E9 1C E9 55 1C E9 1C E9 E9 1C 7A A8 BD E9 Sekuens: BD 82 A8 BD 7A 7A 17 82 1C E9 1C E9 BD 1, 1 1, 2 2, 2 2, 8 1, 8 0 ms Apakah ingin menyimpan solusi? (y/n) </pre>
---	---

D. Link Repository: https://github.com/Intermaze/Tucil1_13522065