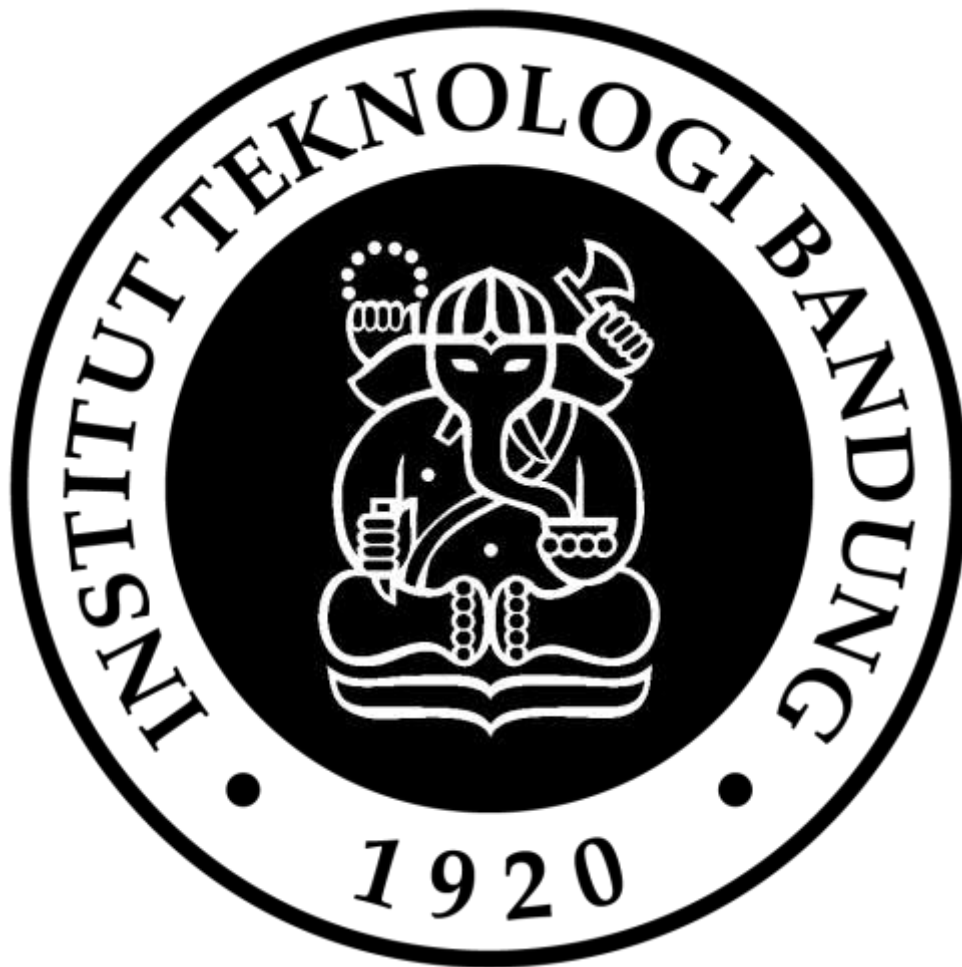


Laporan Tugas Kecil 1

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

Tugas Kecil 1 IF2211 Strategi Algoritma

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

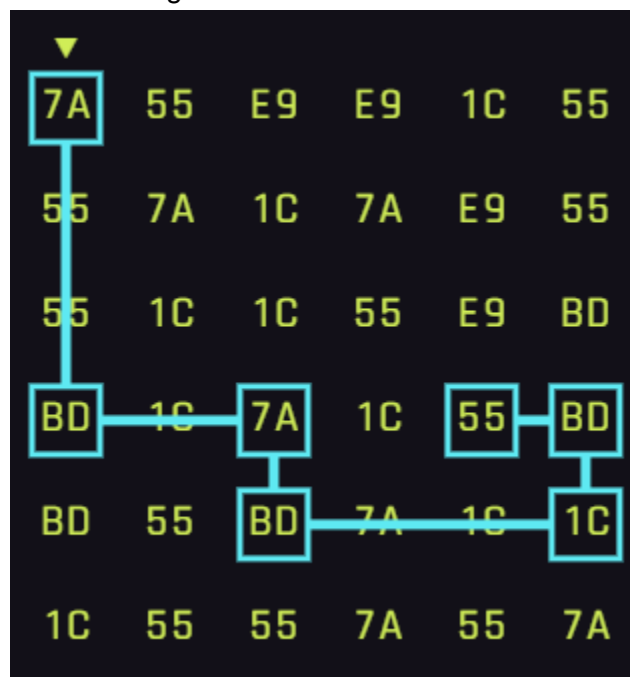
7A	55	E9	E9	1C	55
55	7A	1C	7A	E9	55
55	1C	1C	55	E9	BD
BD	1C	7A	1C	55	BD
BD	55	BD	7A	1C	1C
1C	55	55	7A	55	7A

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



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

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

Tugas Kecil 1 IF2211 Strategi Algoritma

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

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

- 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
 - Jika sel yang diperiksa belum pernah dikunjungi sebelumnya, ulangi langkah 2 sampai 4.
 - Ulangi terus prosedur ini dengan perubahan state secara selang-seling dengan pola Horizontal – Vertical -Horizontal-Vertical....
 - 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

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

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

3. State: Vertical

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

Buffer: 7A 55 Poin: 10

Tugas Kecil 1 IF2211 Strategi Algoritma

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

4. 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

5. 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

6. 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

7. 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

8. State: Horizontal

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

Buffer: 7A 55 1C Poin: 35

Tugas Kecil 1 IF2211 Strategi Algoritma

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

9. State: Vertical (backtrack)

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

10. State: Horizontal (backtrack dan iterasi selanjutnya)

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

Buffer: 55 Poin: 0

11. State: Vertical

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

Buffer: 55 7A Poin: 0

12. 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

```
1  #include <bits/stdc++.h>
2  #include "token.h"
3  using namespace std;
4  //GameMatrix adalah matriks yang berisi token dan matriks yang berisi boolean untuk menentukan apakah cell sudah dikunjungi
5  class GameMatrix{
6  private:
7      int ColSize, RowSize;
8      vector<vector<token>> > TokenMatrix; //matriks yang hanya berisi token
9      vector<vector<bool>> > VisitedMatrix; //matriks yang menunjukkan apakah cell sudah dikunjungi atau belum
10     bool isGameMatrixValid; //atribut yang menyatakan apakah semua token pada matriks valid atau tidak.
11     //GameMatrix yang valid adalah GameMatrix yang tidak mengandung token yang tidak valid
12     void InitiateTokenMatrix(set<string> ValidTokenStrings); //membuat matriks token baru
13     void InitiateVisitedMatrix(); //membuat matriks visited baru
14     void CheckGameMatrixValidity(); //mengecek apakah GameMatrix valid
15 public:
16     GameMatrix(int Row, int Col, set<string> ValidTokenStrings); //konstruktor GameMatrix
17     token GetTokenCell(int Row, int Col); //mendapatkan token pada baris dan kolom yang dicari
18     bool GetVisitedCell(int Row, int Col); //mendapatkan nilai apakah cell sudah dikunjungi atau belum
19     void InputGameMatrix(vector<vector<string>> > StringMatrix, set<string> ValidTokenStrings); //BELUM DIIMPLEMENTASI menginput gameMatrix
20     void SetVisitedCell(int Row, int Col, bool cellValue); //mengubah nilai cell visited matrix (untuk backtracking)
21     int GetRow(); //mendapatkan banyak baris matriks
22     int GetCol(); //mendapatkan banyak kolom matriks
23     void PrintTokenMatrix(); //mencetak matriks token
24     void PrintVisitedMatrix(); //mencetak matriks visited
25     bool GetGameMatrixValidity(); //mendapatkan nilai isGameMatrixValid
26     void SetTokenCell(int Row, int Col, Token value);
27 }
```


Tugas Kecil 1 IF2211 Strategi Algoritma

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

gamematrix.cpp

```
1  #include <bits/stdc++.h>
2  #include "token.h"
3  #include "gamematrix.h"
4
5  using namespace std;
6
7  GameMatrix::GameMatrix(int Row, int Col, set<string> ValidTokenStrings){
8      RowSize = Row;
9      ColSize = Col;
10     isGameMatrixValid = false;
11     InitiateTokenMatrix(ValidTokenStrings);
12     InitiateVisitedMatrix();
13 }
14
15 void GameMatrix::InitiateTokenMatrix(set<string> ValidTokenStrings){
16     for(int i = 0; i<RowSize; i++){
17         vector<Token> rowInit;
18
19         for(int j = 0; j<ColSize; j++){
20             Token TokenInit = Token("-", ValidTokenStrings);
21             rowInit.push_back(TokenInit);
22         }
23         TokenMatrix.push_back(rowInit);
24         rowInit.clear();
25     }
26 }
27
28 void GameMatrix::InitiateVisitedMatrix(){
29     for(int i = 0; i<RowSize; i++){
30         vector<bool> rowInit;
31         for(int j = 0; j<ColSize; j++){
32             rowInit.push_back(false);
33         }
34         VisitedMatrix.push_back(rowInit);
35         rowInit.clear();
36     }
37 }
38
39 Token GameMatrix::GetTokenCell(int Row, int Col){
40     return TokenMatrix[Row][Col];
41 }
42
43 bool GameMatrix::GetVisitedCell(int Row, int Col){
44     return VisitedMatrix[Row][Col];
45 }
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

```
42
43 void GameMatrix::SetVisitedCell(int Row, int Col, bool cellValue){
44     VisitedMatrix[Row][Col] = cellValue;
45     return;
46 }
47
48 void GameMatrix::SetTokenCell(int Row, int Col, Token value){
49     TokenMatrix[Row][Col] = value;
50     return;
51 }
52 int GameMatrix::GetCol(){
53     return ColSize;
54 }
55
56 int GameMatrix::GetRow(){
57     return RowSize;
58 }
59
60 void GameMatrix::PrintTokenMatrix(){
61     for(int i = 0; i<RowSize; i++){
62         for(int j = 0; j<ColSize; j++){
63             if(j == ColSize-1){
64                 cout<<TokenMatrix[i][j].GetTokenString();
65             } else{
66                 cout<<TokenMatrix[i][j].GetTokenString() + " ";
67             }
68         }
69         cout<<"\n";
70     }
71 }
72
73 void GameMatrix::PrintVisitedMatrix(){
74     for(int i = 0; i<RowSize; i++){
75         for(int j = 0; j<ColSize; j++){
76             if(j == ColSize-1){
77                 cout<<VisitedMatrix[i][j];
78             } else{
79                 cout<<VisitedMatrix[i][j]<<" ";
80             }
81         }
82         cout<<"\n";
83     }
84 }
85
86 bool GameMatrix::GetGameMatrixValidity(){
87     return isGameMatrixValid;
88 }
89
90 void GameMatrix::CheckGameMatrixValidity(){
91     isGameMatrixValid = true;
92     for(int i = 0; i<RowSize; i++){
93         for(int j = 0; j<ColSize; j++){
94             if(!TokenMatrix[i][j].GetTokenValidity()){
95                 isGameMatrixValid = false;
96                 return;
97             }
98         }
99     }
100     return;
101 }
102
103 void GameMatrix::InputGameMatrix(vector<vector<string>> StringMatrix, set<string> ValidTokenStrings){
104     for(int i = 0; i<RowSize; i++){
105         for(int j = 0; j<ColSize; j++){
106             TokenMatrix[i][j] = Token(StringMatrix[i][j], ValidTokenStrings);
107         }
108     }
109     CheckGameMatrixValidity();
110 }
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

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

helper.h

```
1 #include <bits/stdc++.h>
2 #include "token.h"
3 using namespace std;
4 vector<string> StringToStringList(string inputString); //untuk melakukan stripping token dari string yang mengandung spasi
5 vector<Token> StringListToTokenList(vector<string> StringList, set<string> ValidTokens); //mengubah string list jadi token list
6 //debugger
7 void PrintTokenList(vector<Token> TokenList);
8 int isSubstring(string s1, string s2); //mengecek apakah s2 adalah substring dari s1
```

helper.cpp

```
1 #include <bits/stdc++.h>
2 #include "token.h"
3 #include "helper.h"
4 using namespace std;
5
6 vector<string> StringToStringList(string inputString){
7     vector<string> StringList;
8     string tempString;
9     tempString = "";
10
11     for(int i = 0; i<inputString.length(); i++){
12
13         if(i == 0){
14             if(inputString[i] != ' '){
15                 tempString = tempString + inputString[i];
16                 if(inputString.length() == 1){
17                     StringList.push_back(tempString);
18                 }
19             }
20
21         } else {
22             if(i == inputString.length()-1){
23                 if(inputString[i] != ' '){
24                     tempString = tempString + inputString[i];
25                 }
26                 StringList.push_back(tempString);
27                 tempString = "";
28             } else {
29                 if(inputString[i] == ' '){
30                     if(inputString[i-1] != ' '){
31                         StringList.push_back(tempString);
32                         tempString = "";
33                     }
34                 } else {
35                     tempString = tempString + inputString[i];
36                 }
37             }
38         }
39     }
40 }
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

```
40     return StringList;
41 }
42 vector<Token> StringListToTokenList(vector<string> StringList, set<string> ValidTokens){
43     vector<Token> ListOut;
44     for(int i = 0; i<StringList.size(); i++){
45         Token temp = Token(StringList[i],ValidTokens);
46         ListOut.push_back(temp);
47     }
48     return ListOut;
49 }
50 void PrintTokenList(vector<Token> TokenList){
51     for(int i = 0; i<TokenList.size(); i++){
52         cout<<TokenList[i].GetTokenString()<<"\n";
53     }
54 }
55
56 int isSubstring(string s1, string s2)
57 {
58
59     if (s1.find(s2) != string::npos)
60         return s1.find(s2);
61     return -1;
62 }
```

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

```
1  #include <bits/stdc++.h>
2  #include "token.h"
3  using namespace std;
4  //TokenSequence adalah senarai/list yang terdiri atas token yang berurutan
5  class TokenSequence{
6  private:
7      vector<Token> Sequence; //Sequence dalam bentuk list
8      bool ValidSequence; //menentukan apakah sequence valid
9      int sequencePoints; //poin dari sequence
10     //sequence yang valid adalah sequence yang tidak mengandung token yang tidak valid sama sekali
11     string SequenceString; //sequence dalam bentuk string
12     //alasan ada string ini adalah untuk mengurangi kompleksitas waktu dibandingkan dengan membandingkan array
13     string ConvertSequenceToString(); //mengubah sequence menjadi string tanpa spasi
14 public:
15     TokenSequence(vector<Token> InputSequence,int points); //konstruktur TokenSequence
16     vector<Token> GetSequence(); //mendapatkan atribut Sequence
17     bool GetSequenceValidity(); //mendapatkan atribut ValidSequence
18     string GetSequenceString(); //mendapatkan atribut SequenceString
19     int GetSequencePoints(); //mendapatkan nilai sequencePoints
20 }
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

sequence.cpp

```
1  #include <bits/stdc++.h>
2  #include "sequence.h"
3
4  TokenSequence::TokenSequence(vector<Token> InputSequence, int points){
5      Sequence = InputSequence;
6      sequencePoints = points;
7      ValidSequence = true;
8
9      for(int i = 0; i<InputSequence.size(); i++){
10         if(!InputSequence[i].GetTokenValidity()){
11             ValidSequence = false;
12             break;
13         }
14     }
15
16     if(ValidSequence){
17         SequenceString = ConvertSequenceToString();
18     }
19 }
20
21 vector<Token> TokenSequence::GetSequence(){
22     return Sequence;
23 }
24 bool TokenSequence::GetSequenceValidity(){
25     return ValidSequence;
26 }
27 string TokenSequence::GetSequenceString(){
28     return SequenceString;
29 }
30 string TokenSequence::ConvertSequenceToString(){
31     string out = "";
32
33     for(int i = 0; i<Sequence.size(); i++){
34         out = out + Sequence[i].GetTokenString();
35     }
36
37     return out;
38 }
39 int TokenSequence::GetSequencePoints(){
40     return sequencePoints;
41 }
```

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

```
Code Blame 18 lines (17 loc) - 777 Bytes Code 55% faster with GitHub Copilot Raw
1  #ifndef TOKEN_H
2  #define TOKEN_H
3  #include <bits/stdc++.h>
4
5  using namespace std;
6  //Token adalah objek yang berupa string (sai individu pada matrix dan sequence)
7  class Token{
8  private:
9      string TokenString; //string token
10     bool isValidToken; //tak sbnlah token valid
11     //token yang valid adalah token yang memiliki nilai tokenString berupa salah satu dari string berikut: "IC", "SS", "ZA", "BD", "ES", "FF"
12     bool isValidToken(astring InputString, set<string> ValidTokenStrings); //periksa apakah token valid
13 public:
14     Token(string InputString, set<string> ValidTokenStrings); //konstruktor token
15     string GetTokenString(); //mengembalikan nilai TokenString
16     bool GetTokenValidity(); //mengembalikan nilai isValidToken
17 };
18
19 #endif
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

token.cpp

```
1  #include <bits/stdc++.h>
2  #include "token.h"
3
4  using namespace std;
5
6  Token::Token(string InputString, set<string> ValidTokenStrings){
7      TokenString = InputString;
8      isValidToken = isTokenValid(InputString, ValidTokenStrings);
9  }
10
11 bool Token::isTokenValid(string InputString, set<string> ValidTokenStrings){
12     set<string> ValidTokens;
13     ValidTokens = ValidTokenStrings;
14     bool check = false;
15     for(int i = 0; i<ValidTokens.size(); i++){
16         if(ValidTokens.find(InputString) != ValidTokens.end()){
17             check = true;
18             break;
19         }
20     }
21     return check;
22 }
23
24 string Token::GetTokenString(){
25     return TokenString;
26 }
27
28 bool Token::GetTokenValidity(){
29     return isValidToken;
30 }
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

5. main.cpp (Program utama CLI)

```
1 #include <iostream>
2 #include "helper.h"
3 #include "token.h"
4 #include "sequence.h"
5 #include "gameMatrix.h"
6 using namespace std;
7 typedef chrono::milliseconds MSC;
8 typedef chrono::high_resolution_clock HRC;
9 //Membuat objek koordinat awal
10 //Membuat sequence awal
11 bool method; //metode input
12 string OpCoordinateAnswer; //string jawaban
13 int NewPoint; //Jumlah titik koordinat
14 int NewBuffer; //Jumlah buffer awal
15 vector<Token> TokenListFinalAnswer; //Membuat token yang diinput sebagai jawaban akhir
16 vector<pair<int,int>> CoordinateListFinalAnswer; //Membuat koordinat yang diinput sebagai jawaban akhir
17
18 int MatrixRow,MatrixCol; //menjaga baris dan kolom pada matrix di atas
19 int BufferLeft; //panjang buffer
20 int SequenceIndex; //posisi sequence
21 vector<pair<int,int>> CurrentPoints; //list yang berisi koordinat input
22 vector<TokenSequence> SequenceList; //Membuat sequence yang ada
23 vector<bool> SequenceCheckList; //list yang berfungsi untuk mengecek apakah sequence sudah benar atau tidak
24
25 set<string> validToken; //Jumlah token yang valid
26 vector<vector<string>> InputStringMatrix;
27 = void mainMethodAnswer(int currentPoints, int currentBuffer, vector<Token> currentTokenList, vector<pair<int,int>> currentCoordinateList){
28     //Membuat objek Token First dengan Token
29     //Membuat currentPoints //Membuat titik awal saat ini = currentPoints
30     NewPoints = currentPoints; //Membuat titik akhir dengan currentPoints
31     NewBuffer = currentBuffer; //Membuat buffer akhir dengan currentBuffer
32     TokenListFinalAnswer = currentTokenList; //Membuat token list akhir dengan currentTokenList
33     CoordinateListFinalAnswer = currentCoordinateList; //Membuat koordinat list akhir dengan currentCoordinateList
34 } else if(NewPoints = currentPoints){ //Jika titik akhir saat ini = currentPoints
35     if(NewBuffer = currentBuffer){ //Jika buffer akhir saat ini = currentBuffer
36         //Membuat objek string akhir saat ini seperti dengan koordinat jika titik akhir = currentPoints
37         NewPoints = currentPoints;
38         NewBuffer = currentBuffer;
39         TokenListFinalAnswer = currentTokenList;
40         CoordinateListFinalAnswer = currentCoordinateList;
41     }
42     //Jika titik akhir saat ini = currentPoints, dengan koordinat jawaban
43     return;
44 }
45
46 //Membuat
47 // string test, test;
48 // state string state1, state2;
49 void SolveMainMethodMatrix gameMatrix, string state, string currentAnswer, int BufferLeft, int currentPoints,
50 vector<Token> currentTokenList, vector<pair<int,int>> currentCoordinateList, int currentRow, int currentCol,
51 int currentBuffer, vector<TokenSequence> currentSequenceList){
52     //Membuat objek Token dengan "BufferLeft" dan "BufferLeft"
53     if(BufferLeft=0){ //Membuat jika BufferLeft=0, dengan jawaban prosedur ini karena buffer sudah penuh
54         return;
55     }
56     string newAnswer;
57     newAnswer = currentAnswer;
58     vector<Token> newTokenList;
59     newTokenList = currentTokenList;
60     vector<pair<int,int>> newCoordinateList = currentCoordinateList;
61     int newBufferLeft, newPoint, newBufferCount;
62     newBufferLeft = BufferLeft;
63     newBufferCount = currentBuffer;
64
65     pair<int,int> currentCoordinate;
66     int subStringCheck;
67     if(state == "Horizontal"){
68         //Membuat objek state Horizontal "H"
69         for(int i = 0; i<MatrixCol; i++){
70             if(gameMatrix.GetTokenCell(currentRow,i)){ //Jika token di sel(currentRow[i]) tidak kosong
71                 newPoint = i;
72                 newAnswer = newAnswer + gameMatrix.GetTokenCell(currentRow,i).GetTokenString(); //Membuat string token yang diinput ke string answer
73                 newTokenList.push_back(gameMatrix.GetTokenCell(currentRow,i)); //Membuat token ke TokenList dari jawaban
74                 currentCoordinate = make_pair(i,i,currentRow);
75                 newCoordinateList.push_back(currentCoordinate); //Membuat koordinat saat ini ke CoordinateList dari jawaban
76                 gameMatrix.SetTokenCell(currentRow,i,true); //Membuat setTokenCell(i) sudah diinput
77             }
78
79             newBufferLeft--; //Membuat BufferLeft dengan 1
80             newBufferCount++; //Membuat currentBuffer dengan 1
81
82             for(int i = 0; i<currentSequenceList.size(); i++){ //L. Cek semua sequence di SequenceList
83                 subStringCheck = isSubString(newAnswer,currentSequenceList[i].GetSequenceString());
84
85                 if(subStringCheck != -1){ //Jika string dari jawaban tersebut adalah subString dari sequence, maka
86                     newPoint=currentSequenceList[i].GetSequencePoints(); //Membuat poin awal dengan yang diberikan dari sequence
87                 }
88             }
89 }
```


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

```

40         UpdateCoordinates(newPoints, newBufferCount, newPointsList, newCoordinatesList); //Update location
41
42         Substring(generatePoints, "Vertical", newBuffer, newBufferLeft, newPoints, newPointsList, newCoordinatesList, currentRow, newBufferCount, currentSequenceList);
43         //Pengisi fungsi ini lagi secara otomatis dengan parameter yang sudah diupdate di atas dimana dengan nilai currentCol = 1, nilai currentRow sama, dan state = "Vertical"
44         //Batalan semua parameter parameter untuk thread berikutnya
45         newBuffer.Reset(newBuffer.Count) - generatePoints.GetRowCol(currentRow, 1).GetLength(1).Length(1); //1. Maka string baru yang diinput dari string awal (thru token terlewat yang diabaikan)
46         newPointsList.Clear(); //2. dan newPointsList dari belakang
47         newCoordinatesList.Clear(); //3. dan CoordinatesList dari belakang
48         generatePoints.GetRowCol(currentRow, 1).GetLength(1).Length(1); //1. Maka nilai currentRow(1) akan diinput
49         newBufferLeft -= 1; //Maka BufferLeft dengan 1
50         newBufferCount -= 1; //Maka newBufferCount dengan 1
51
52     }
53
54     } else if (state == "Vertical") {
55
56         for (int i = 0; i < GenerateRow; i++) {
57             if (generatePoints.GetRowCol(i, currentCol)) { //1. Maka token di nilai currentRow(1) akan diinput
58                 newPoints += 1;
59                 newBuffer = newBuffer + generatePoints.GetRowCol(i, currentCol).GetLength(1); //Maka string baru yang diinput ke string awal
60                 newPointsList.Add(generatePoints.GetRowCol(i, currentCol)); //2. Maka token ke TokenList dari belakang
61                 currentCoordinates = new Pair(currentCol + 1, i + 1);
62                 newCoordinatesList.Add(currentCoordinates); //3. Maka koordinat dari nilai ke CoordinatesList dari belakang
63                 generatePoints.GetRowCol(i, currentCol, true); //Maka nilai currentRow(1) akan diinput
64
65                 newBufferLeft -= 1; //Maka BufferLeft dengan 1
66                 newBufferCount += 1; //Maka newBufferCount dengan 1
67
68             }
69
70             for (int i = 0; i < currentSequenceList.Count; i++) { //1. dan semua elemen di SequenceList
71                 substringCheck = substring(newBuffer, currentSequenceList[i].GetSequenceString());
72
73                 if (substringCheck != -1) { //1. Maka string dari sequence terlewat dalam substring dari newBuffer, maka
74                     newPoints += currentSequenceList[i].GetSequencePoints(); //Maka token baru akan dengan yang diberikan oleh sequence
75                 }
76             }
77
78         }
79
80         //Maka Token yang akan diabaikan "vertical"
81         //Maka Token yang akan diabaikan dari sequence ini "vertical"
82         UpdateCoordinates(newPoints, newBufferCount, newPointsList, newCoordinatesList); //Update location
83
84     }
85
86     Substring(generatePoints, "Horizontal", newBuffer, newBufferLeft, newPoints, newPointsList, newCoordinatesList, currentCol, newBufferCount, currentSequenceList);
87     //Pengisi fungsi ini lagi secara otomatis dengan parameter yang sudah diupdate di atas dimana dengan nilai currentCol = 1, nilai currentRow sama, dan state = "Horizontal"
88     //Batalan semua parameter parameter untuk thread berikutnya
89     newBuffer.Reset(newBuffer.Count) - generatePoints.GetRowCol(1, currentCol).GetLength(1).Length(1); //1. Maka string baru yang diinput dari string awal (thru token terlewat yang diabaikan)
90     newPointsList.Clear(); //2. dan newPointsList dari belakang
91     newCoordinatesList.Clear(); //3. dan CoordinatesList dari belakang
92     generatePoints.GetRowCol(1, currentCol, false); //1. Maka nilai currentRow(1) akan diinput
93     newBufferLeft -= 1; //Maka BufferLeft dengan 1
94     newBufferCount -= 1; //Maka newBufferCount dengan 1
95
96 }
97
98 }
99
100 }
101
102 }
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
86
```


Tugas Kecil 1 IF2211 Strategi Algoritma

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

```
170 OutputFile<<"\n";
171
172 for(int i = 0; i<sequenceList[i].GetSequence().size(); i++){
173     string sequenceOut = "";
174     for(int j = 0; j<sequenceList[i].GetSequence().size(); j++){
175         sequenceOut = sequenceOut + sequenceList[i].GetSequence()[j].GetTimeString();
176         if(j != sequenceList[i].GetSequence().size()-1){
177             sequenceOut = sequenceOut + " ";
178         }
179     }
180     OutputFile<<sequenceOut;
181     OutputFile<<"\n";
182     OutputFile<<string(sequenceList[i].GetSequenceOut());
183     OutputFile<<"\n";
184 }
185 OutputFile<<"\n";
186 }
187
188 OutputFile<<string(NumberOfPoints);
189 OutputFile<<"\n";
190 string sequenceFinalAnswer = "";
191 for(int i = 0; i<tokenListFinalAnswer.size(); i++){
192     sequenceFinalAnswer = sequenceFinalAnswer + tokenListFinalAnswer[i].GetTimeString();
193     if(i != tokenListFinalAnswer.size()-1){
194         sequenceFinalAnswer = sequenceFinalAnswer + " ";
195     }
196 }
197 OutputFile<<sequenceFinalAnswer;
198 OutputFile<<"\n";
199 for(int i = 0; i<CoordinateListFinalAnswer.size(); i++){
200     string CoordinateOut = to_string(CoordinateListFinalAnswer[i].first) + " ";
201     OutputFile<<CoordinateOut;
202     OutputFile<<"\n";
203 }
204 OutputFile<<sequenceFinalAnswer;
205 OutputFile<<"\n";
206 OutputFile<<choice;
207 cout<<OutputFile<<" sudah berakhir di folder test1\n";
208 }
209
210 void OutputAnswer(into esc_time){
211     cout<<"New anwser: ";<<NumberOfPoints<<"\n";
212     cout<<"Final Answer: ";
213     for(int i = 0; i<tokenListFinalAnswer.size(); i++){
214         cout<<tokenListFinalAnswer[i].GetTimeString()<<" ";
215     }
216     cout<<"\n";
217     cout<<"Final Coordinates: ";
218     for(int i = 0; i<CoordinateListFinalAnswer.size(); i++){
219         cout<<CoordinateListFinalAnswer[i].first<<" ";<<CoordinateListFinalAnswer[i].second<<"\n";
220     }
221     cout<<"\n";
222     cout<<esc_time<<endl<<"\n";
223 }
224
225 void OutputFileInput(GameMatrix gameMatrix, bool choice, into esc_time){
226     cout<<"Masuk input banyak nilai sebagai file .txt (Isi dengan cara ketik 0000 dengan 'y' atau 'n' untuk YA atau tidak selain kode input tersebut untuk TIME): ";
227
228     string answer;
229     cin>>answer;
230     if(answer == "y" || answer == "Y"){
231         OutputFile(gameMatrix,choice,esc_time);
232     }
233 }
234
235 int main(){
236     cout<<"Masukkan metode input Breach Protocol yang diinginkan dengan mengklik: 0: Input File 1: Random Matrix and Sequence generator\n";
237     cin>>method;
238     if(method){
239         for (validTokenAnswer;
240             cout<<"Masukkan banyak token yang valid: ";
241             cin>>validTokenAnswer;
242             while(validTokenAnswer==0){ //handling input token valid
243                 cout<<"Banyak token yang valid yang diinput tidak valid silahkan input ulang: ";
244                 cin>>validTokenAnswer;
245             }
246             cout<<"Masukkan daftar token yang valid: ";
247             string ValidTokenListString;
248             cin.ignore();
249             getline(cin,ValidTokenListString);
250             vector<string> TokenStringList;
251             TokenStringList = stringToStringList(ValidTokenListString);
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

```

217     for(int i = 0; i<tokensStringList.size(); i++){
218         validToken.insert(tokensStringList[i]);
219     }
220     while(!validToken.empty()) { //Handling input token
221         cout<<"Masuk token ada yang diluar token sesuai dengan banyak token valid: Silahkan input ulang: ";
222         validToken.clear();
223         cin.ignore();
224         getline(cin,validTokenListString);
225         tokensStringList = stringToStringList(validTokenListString);
226         for(int i = 0; i<tokensStringList.size(); i++){
227             validToken.insert(tokensStringList[i]);
228         }
229     }
230     enterToken validTokenList;
231     validTokenList = stringListToTokenList(tokensStringList,validToken);
232     cout<<"Masukan ukuran buffer: ";
233     cin>>bufferSize;
234     while(bufferSize<0) { //Handling ukuran buffer
235         cout<<"ukuran buffer tidak valid: Silahkan input ulang: ";
236         cin>>bufferSize;
237     }
238     cout<<"Masukan angka baris dan kolom matriks: ";
239     cin>>matrixRow>>matrixCol;
240
241     while(matrixCol <= 0 || matrixRow <= 0){
242         cout<<"ukuran matriks tidak valid: Silahkan input ulang: ";
243         cin>>matrixRow>>matrixCol;
244     }
245     srand(time(0));
246     gameMatrix gameMatrix = GameMatrix(matrixRow,matrixCol,validToken);
247     for(int i = 0; i<matrixRow; i++){
248         for(int j = 0; j<matrixCol; j++){
249             int chosenToken = (rand() % validToken.size());
250             gameMatrix.SetTokenCell(i,j,validTokenList[chosenToken]);
251         }
252     }
253
254     cout<<"Masukan banyak sequence: ";
255     cin>>sequenceSize;
256     while(sequenceSize<0) { //Handling ukuran buffer
257         cout<<"banyak sequence tidak valid: Silahkan input ulang: ";
258         cin>>sequenceSize;
259     }
260     int MaxSequenceLength;
261     cout<<"Masukan panjang sequence maksimum: ";
262     cin>>MaxSequenceLength;
263     while(MaxSequenceLength<0) { //Handling ukuran buffer
264         cout<<"panjang sequence maksimum tidak valid: Silahkan input ulang: ";
265         cin>>MaxSequenceLength;
266     }
267     gameMatrix.PrintMatrix();
268     for(int i = 0; i<sequenceSize; i++){
269         int sequenceLength = (rand() % MaxSequenceLength) + 1;
270         int chosenPoints = (rand() % 100);
271         vector<Token> GeneratedSequence;
272         for(int j = 0; j<sequenceLength; j++){
273             int chosenToken = (rand() % validToken.size());
274             GeneratedSequence.push_back(validTokenList[chosenToken]);
275         }
276
277         TokenSequence newSequence = TokenSequence(GeneratedSequence,chosenPoints);
278         SequenceList.push_back(newSequence);
279         GeneratedSequence.clear();
280     }
281
282     for(int i = 0; i<sequenceSize; i++){
283         for(int j = 0; j<sequenceList[i].GetSequence().size(); j++){
284             cout<<SequenceList[i].GetSequence()[j].GetTokenString()+" ";
285         }
286     }
287     cout<<"Poin = " <<SequenceList[i].GetSequencePoints()<<"\n";
288 }
289
290 auto start = chrono::now();
291 SolveOptimal(gameMatrix,"horizontal","",bufferSize,0,tokensListFinalAnswer,CoordinateListFinalAnswer,0,0,0,SequenceList);
292 auto end = chrono::now();
293 OutputAnswer(chrono::duration_cast<chrono::milliseconds>(end-start));
294 OutputFileOptimal(gameMatrix,endTime,chrono::duration_cast<chrono::milliseconds>(end-start));
295 } else {
296     mapStart;
297     ifstream inputFile;
298     cout<<"Masukan nama file (GANTI DIAMERI DENGAN '.txt' dan AKHIR ADA TO PAKSA test): ";
299     string filename;

```

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

```

100 string InputPath = "...\\test\\";
101 cin<<filenam;
102 InputFile.open(InputPath + filenam);
103
104 while(!InputFile.is_open()){
105     cout<<"File tidak ditemukan! Silakan input ulang.\n";
106     cout<<"Masukkan nama file (Maksimal 30 karakter): ";
107     cin<<filenam;
108     InputFile.open(InputPath);
109 }
110
111 // if(!OpenFile.is_open()){
112 //     cout<<"File tidak ditemukan! Program akan berhenti. \n";
113 //     return 0;
114 // }
115
116 int linecount = 0;
117 string RowFile;
118 vector<string> RowFileStripped;
119 while(!InputFile.eof()){
120     linecount++;
121     getline(InputFile,RowFile);
122
123     RowFileStripped = StringToStringList(RowFile);
124
125     if(linecount==1){
126         if(linecount == 1){
127             bufferSize = stoi(RowFileStripped[0]);
128         }
129         MatrixCol = stoi(RowFileStripped[0]);
130         MatrixRow =stoi(RowFileStripped[1]);
131     }
132 } else if(linecount% 2 != linecount-MatrixRow+1){
133
134     InputStringMatrix.row_size(RowFileStripped);
135     for(int i = 0; i<MatrixCol; i++){
136         validIndex.insert(RowFileStripped[i]);
137     }
138 } else if(linecount == MatrixRow){
139     sequenceSize = stoi(RowFileStripped[0]);
140 } else{
141     if(InputFile.eof()){
142         break;
143     }
144
145     for(int i = 0; i<RowFileStripped.size(); i++){
146         validIndex.insert(RowFileStripped[i]);
147     }
148
149     vector<Token> TokenedLines;
150     TokenedSeq = StringListToTokenList(RowFileStripped,validIndex);
151
152     linecount++;
153     getline(InputFile,RowFile);
154
155     RowFileStripped = StringToStringList(RowFile);
156     int tempPoints = stoi(RowFileStripped[0]);
157
158     TokenSequence TempSequence = TokenSequence(TokenedSeq,tempPoints);
159     SequenceList.push_back(TempSequence);
160 }
161
162 if(MatrixRow==0 || MatrixCol==0 || MatrixRow==0 || sequenceSize==0 || MatrixCol != InputStringMatrix[0].size() || MatrixRow != InputStringMatrix.size()) || readingError
163     cout<<"File tidak valid. Silakan input ulang. \n";
164     goto loopStart;
165     return 0;
166 }
167
168 GasMatrix gasMatrix = GasMatrix(MatrixRow,MatrixCol,validIndex);
169 gasMatrix.setInputMatrix(InputStringMatrix,validIndex);
170
171 auto start = chrono::now();
172 InputString(gasMatrix,"horizontal","",bufferSize,4,isValidFinalAnswer,CoordinateListFinalAnswer,0,0,0,sequenceList);
173 auto end = chrono::now();
174 OutputAnswer(chrono::duration_cast<chrono::seconds>(end-start));
175 OutputListAnswer(gasMatrix,method,chrono::duration_cast<chrono::seconds>(end-start));
176
177 }
178
179 return 0;
180

```

Bab 4: Uji Coba Program

Tes 1: Input File

Isi file input (test1.txt):

```
7
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
```

Hasil ujicoba (output file testout1.txt):

```
test > testout1.txt
You, 6 seconds ago | 1 author (You)
1 50
2 7A BD 7A BD 1C BD 55
3 1,1
4 1,4
5 3,4
6 3,5
7 6,5
8 6,3
9 1,3
10 1200 ms You, 5 seconds ago • Uncommitted changes
```

Tes 2: Input File

Isi file input (test2.txt):

```
2
1 1
AF
2
AF
5
BD BC CG HE
10000000 You, 16
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

Hasil ujicoba: (output file testout2.txt):

```
test > testout2.txt
You, 4 seconds ago | 1 author (You)
1 5 You, 3 seconds ago • Uncommitted changes
2 AF
3 1,1
4 0 ms
```

Tes 3: Input File

Isi file input (test3.txt):

```
5
2 2
BD HE
BC CG
3
CG HE
10
BD BC CG HE
-10000000
BC CG
5 You, 16 hours ago
```

Hasil ujicoba (output file testout2.txt):

```
1 You, 12 seconds ago | 1 author (You)
1 5 You, 11 seconds ago • Uncommitted changes
2 BD BC CG
3 1,1
4 1,2
5 2,2
6 0 ms
```

Tes 4: Random Generation

Input CLI:

```
Masukkan metode input Breach Protocol yang diinginkan dengan mengetik:
0: input file
1: Random Matrix and Sequence generation
1
Masukkan banyak token yang valid: 5
Masukan daftar token yang valid: AA BB CC DD EE
Masukkan ukuran buffer: 6
Masukkan banyak baris dan kolom matriks: 7 8
Masukkan banyak sequence: 4
Masukkan panjang sequence maksimum: 5
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

Hasil ujicoba (output file testrandom1.txt):

```
1  CC CC AA AA BB EE BB DD
2  BB AA DD AA CC CC EE CC
3  DD BB BB AA AA BB EE AA
4  DD BB DD AA EE AA DD BB
5  BB BB EE EE CC CC CC AA
6  DD DD CC EE DD BB EE AA
7  CC AA CC AA CC DD DD CC
8
9  BB EE DD DD
10 57
11 BB
12 43
13 CC DD
14 68
15 AA CC
16 5
17
18 168
19 BB EE DD DD CC DD
20 5,1
21 5,4
22 1,4
23 1,6
24 3,6
25 3,2
26 1410 ms
```

Tes 5: Random Generation

Input CLI:

```
Masukkan metode input Breach Protocol yang diinginkan dengan mengetik:
0: input file
1: Random Matrix and Sequence generation
1
Masukkan banyak token yang valid: 8
Masukkan daftar token yang valid: AA BB CC DD EE FF GG HH
Masukkan ukuran buffer: 5
Masukkan banyak baris dan kolom matriks: 8 10
Masukkan banyak sequence: 5
Masukkan panjang sequence maksimum: 3
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

Hasil ujicoba (output file testrandom2.txt):

```
1  FF DD EE GG CC EE BB AA EE HH
2  DD GG GG CC DD EE FF BB DD AA
3  AA EE HH EE BB AA CC EE DD AA
4  GG AA DD GG AA DD EE EE DD FF
5  GG AA EE GG GG AA GG FF EE CC
6  HH AA BB HH CC EE AA AA HH AA
7  HH BB AA FF FF GG FF AA GG EE
8  CC DD FF BB HH AA BB GG AA AA
9
10 GG EE HH
11 19
12 BB BB DD
13 64
14 CC GG FF
15 51
16 DD BB
17 17
18 BB
19 42
20
21 123
22 BB BB DD BB
23 7,1
24 7,8
25 2,8
26 2,7
27 789 ms
```

Tes 6: Random Generation

Input CLI:

```
Masukkan metode input Breach Protocol yang diinginkan dengan mengetik:
0: input file
1: Random Matrix and Sequence generation
1
Masukkan banyak token yang valid: 4
Masukan daftar token yang valid: A6 HE 7U L0
Masukkan ukuran buffer: 5
Masukkan banyak baris dan kolom matriks: 6 7
Masukkan banyak sequence: 4
Masukkan panjang sequence maksimum: 5
```

Tugas Kecil 1 IF2211 Strategi Algoritma

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

Hasil ujicoba (output file testrandom3.txt):

```
1  L0 7U L0 HE HE 7U L0
2  A6 A6 7U L0 7U 7U L0
3  A6 HE HE L0 L0 7U L0
4  HE 7U HE HE A6 A6 A6
5  7U 7U HE A6 L0 L0 7U
6  L0 A6 7U 7U A6 HE L0
7
8  7U 7U L0 L0 A6
9  49
10 7U HE
11 12
12 7U 7U
13 95
14 A6 L0 A6
15 19
16
17 144
18 7U 7U L0 L0 A6
19 2,1
20 2,5
21 5,5
22 5,3
23 1,3
24 99 ms
```


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	
2. Program berhasil dijalankan	V	
3. Program dapat membaca masukan berkas .txt	V	
4. Program dapat menghasilkan masukan secara acak	V	
5. Solusi yang diberikan program optimal	V	
6. Program dapat menyimpan solusi dalam berkas .txt	V	
7. Program memiliki GUI		V