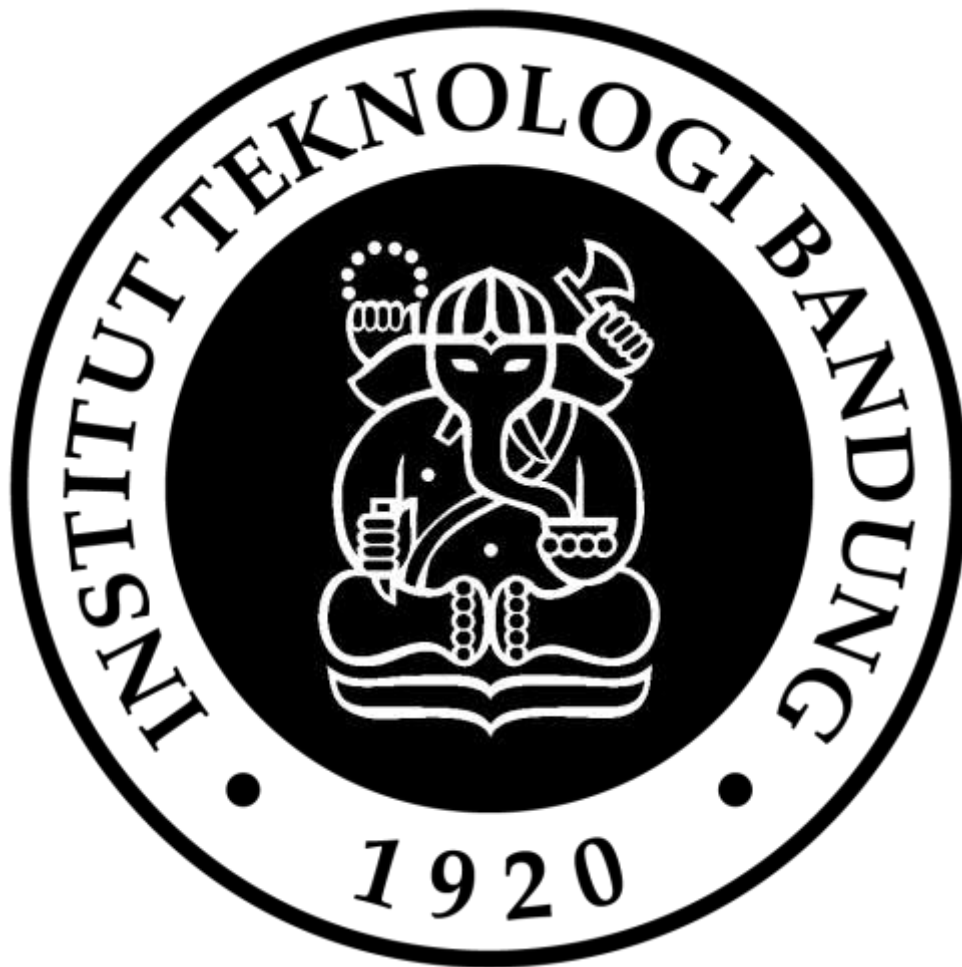


# **Laporan Tugas Kecil 1**

## **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

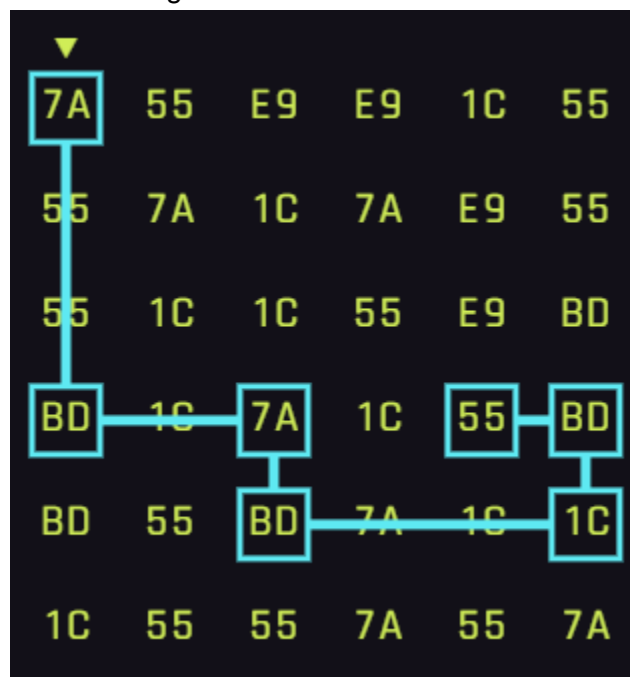
7A	55	E9	E9	1C	55
55	7A	1C	7A	E9	55
55	1C	1C	55	E9	BD
BD	1C	7A	1C	55	BD
BD	55	BD	7A	1C	1C
1C	55	55	7A	55	7A

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 Download Edit
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 }
```

### 5. main.cpp (Program utama CLI)

```
1 #include <bits/stdc++.h>
2 #include "header.h"
3 #include "token.h"
4 #include "sequence.h"
5 #include "gameutils.h"
6 using namespace std;
7
8 typedef chrono::milliseconds MSF;
9 typedef chrono::high_resolution_clock HRC;
10 //Matriks untuk menyimpan hasil
11 //Matriks jawaban akhir
12 bool isFinal; //status final
13 string GetInitialAnswer; //string jawaban
14 int MaxPoints; //jumlah poin maksimal
15 int MaxBuffer; //jumlah buffer maksimal
16 vector<Token> TokenListFinalAnswer; //daftar token yang dipilih sebagai jawaban akhir
17 vector<pair<int,int>> CoordinateListFinalAnswer; //daftar koordinat yang dipilih sebagai jawaban akhir
18
19 int MaxRow, MaxCol; //panjang baris dan kolom pada matriks di atas
20 int bufferSize; //panjang buffer
21 int sequenceSize; //banyak sequence
22 vector<pair<int,int>> CoordinateList; //list yang berisi koordinat tempat
23 vector<Sequence> SequenceList; //daftar sequence yang ada
24 vector<bool> SequenceCheckList; //list yang berfungsi untuk mengecek apakah sequence sudah pernah ada atau belum
25
26 set<string> validToken; //daftar token yang valid
27 vector<vector<string>> > InputStringMatrix;
28 void isTokenValidAnswer(int currentPoints, int currentBuffer, vector<Token> currentTokenList, vector<pair<int,int>> > currentCoordinateList){
29     //Matriks untuk menyimpan hasil
30     if(MaxPoints < currentPoints){
31         MaxPoints = currentPoints; //update poin akhir dengan currentPoints
32         MaxBuffer = currentBuffer; //update buffer akhir dengan currentBuffer
33         TokenListFinalAnswer = currentTokenList; //update token list akhir dengan currentTokenList
34         CoordinateListFinalAnswer = currentCoordinateList; //update koordinat list akhir dengan currentCoordinateList
35     } else if(MaxPoints == currentPoints){
36         //jika poin akhir sama saja
37         if(MaxBuffer > currentBuffer){
38             //jika buffer akhir lebih besar dari currentBuffer
39             //update semua string akhir yang sesuai dengan kondisi jika poin akhir < currentPoints
40             MaxPoints = currentPoints;
41             MaxBuffer = currentBuffer;
42             TokenListFinalAnswer = currentTokenList;
43             CoordinateListFinalAnswer = currentCoordinateList;
44         }
45     }
46     //jika poin akhir sama saja, jangan update jawaban
47     return;
48 }
```



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

[illegible]

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

[illegible]



## Tugas Kecil 1 IF2211 Strategi Algoritma

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

```
119 }
120 cout<<"\n";
121 cout<<"Final Coordinates: " << "\n";
122 for(int i = 0; i < finalAnswerList[0].size(); i++){
123     cout<<CoordinateListFinalAnswer[i].first<<" ",<<CoordinateListFinalAnswer[i].second<<"\n";
124 }
125 cout<<"\n";
126 cout<<"Time: " << runTime<<" sec";
127 }
128
129 // with OutputFormat(gamematrix, gamematrix, max choice, max runTime){
130     cout<<"Input logis menyajikan nilai sebagai file .txt? (Jawab dengan cara ketik HMMH dengan 'y' atau 'N' untuk YA atau ketik selain y untuk tidak) ";
131
132     string answer;
133     cin>>answer;
134     if(answer == "y" || answer == "Y"){
135         OutputFile(gamematrix, choice, maxTime);
136     }
137 }
138
139 // int main(){
140     cout<<"Masukkan metode input Breach Protocol yang diinginkan dengan mengetik: 0: Input file, 1: Random Matrix and Sequence generation\n";
141     cin>>method;
142     if(method){
143         int validTokenAmount;
144         cout<<"Masukkan banyak token yang valid: ";
145         cin>>validTokenAmount;
146         cout<<"Masukkan daftar token yang valid: ";
147         string ValidTokenListString;
148         cin>>ValidTokenListString;
149         getline(cin, ValidTokenListString);
150         vector<string> TokenStringList;
151         TokenStringList = StringToList(ValidTokenListString);
152
153         for(int i = 0; i < validTokenAmount; i++){
154             validTokenList.insert(TokenStringList[i]);
155         }
156
157         vector<int> ValidTokenList;
158         validTokenList = StringToIntList(TokenStringList, validTokenList);
159         cout<<"Masukkan ukuran buffer: ";
160         cin>>bufferSize;
161
162         cout<<"Masukkan banyak token dan jumlah matriks: ";
163         cin>>MatrixRowMatrixCol;
164         int rowMatrixCol;
165         gamematrix gamematrix = gamematrix(MatrixRowMatrixCol, validTokenList);
166
167         for(int i = 0; i < MatrixRow; i++){
168             for(int j = 0; j < MatrixCol; j++){
169                 int chosenToken = (rand() % validTokenAmount);
170                 gamematrix.setToken(i, j, ValidTokenList[chosenToken]);
171             }
172         }
173
174         cout<<"Masukkan banyak sequence: ";
175         cin>>sequenceLength;
176         int MaxSequenceLength;
177         cout<<"Masukkan panjang sequence maksimal: ";
178         cin>>MaxSequenceLength;
179
180         gamematrix.PrintMatrix();
181         for(int i = 0; i < sequenceLength; i++){
182             int SequenceLength = (rand() % MaxSequenceLength) + 1;
183             int chosenPuloks = (rand() % 100);
184             vector<int> GeneratedSequence;
185             for(int j = 0; j < SequenceLength; j++){
186                 int chosenToken = (rand() % validTokenAmount);
187                 GeneratedSequence.push_back(ValidTokenList[chosenToken]);
188             }
189
190             int sequenceIndex = findSequence(GeneratedSequence, chosenPuloks);
191             SequenceList.push_back(GeneratedSequence);
192             GeneratedSequence.clear();
193         }
194
195         for(int i = 0; i < sequenceLength; i++){
196             for(int j = 0; j < SequenceList[i].GetSequence().size(); j++){
197                 cout<<SequenceList[i].GetSequence()[j].GetTokenString()<<" ";
198             }
199             cout<<"Puls = " << SequenceList[i].GetSequencePuls()<<"\n";
200         }
201         auto start = chrono::now();
202         OutputFile(gamematrix, "poli.txt", "", bufferSize, TokenListFinalAnswer, CoordinateListFinalAnswer, MatrixRowMatrixCol, SequenceList);
203         auto end = chrono::now();
204         OutputTime(chrono::duration_cast<chrono::microseconds>(end - start));
205     }
```

## Tugas Kecil 1 IF2211 Strategi Algoritma

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

```
239 OutputFile<out>(gameMatrix,method,chrono::duration_cast<SEC>(end-start));
240 } else {
241     ifstream InputFile;
242     const char* MethodName = "File (MALLI STARKS DENGAN '.txt' dan WAJIB ADA DI FOLDER test): ";
243     string FileName;
244     string InputPath = "...Res/";
245     char* FilePath;
246     InputFile.open(InputPath + FileName);
247
248     while(!InputFile.is_open()){
249         const char* File = "Silahkan masukkan Input along.txt";
250         const char* FileName = "MALLI STARKS DENGAN '.txt' dan ADA DI FOLDER repository";
251         char* FilePath;
252         InputFile.open(FileName);
253     }
254     if(!InputFile.is_open()){
255         const char* File = "Silahkan masukkan Program akan berjalan.txt";
256         return 0;
257     }
258     int linecount = 0;
259     string RowFile;
260     vector<string> RowFileStripped;
261     while(!InputFile.eof()){
262         linecount++;
263         getline(InputFile,RowFile);
264
265         RowFileStripped = StringToVector<string>(RowFile);
266
267         if(linecount==0){
268             if(linecount == 1){
269                 bufferSize = stoi(RowFileStripped[0]);
270                 while{
271                     MatrixRow = stoi(RowFileStripped[0]);
272                     MatrixRow = stoi(RowFileStripped[1]);
273                 }
274             } else if(linecount==2 && linecount==MatrixRow){
275
276                 InputStringMatrixRow = stoi(RowFileStripped);
277                 for(int i = 0; i<MatrixRow; i++){
278                     validateToken.insert(RowFileStripped[i]);
279                 }
280             } else if(linecount == MatrixRow){
281
282                 sequenceSize = stoi(RowFileStripped[0]);
283             } else{
284                 if(!InputFile.eof()){
285                     break;
286                 }
287                 for(int i = 0; i<RowFileStripped.size(); i++){
288                     validateToken.insert(RowFileStripped[i]);
289                 }
290                 vector<Token> TokenSeqTemp;
291                 TokenSeqTemp = StringToIntToMatrix<int>(RowFileStripped,validateToken);
292
293                 linecount++;
294                 getline(InputFile,RowFile);
295
296                 RowFileStripped = StringToVector<string>(RowFile);
297                 int tempPoints = stoi(RowFileStripped[0]);
298
299                 TokenSequence TempSequence = TokenSequence(TokenSeqTemp,tempPoints);
300                 TempSequence.insert(TempSequence);
301             }
302         }
303     }
304     gameMatrix gameMatrix = gameMatrix(MatrixRow,MatrixRow,validateToken);
305     gameMatrix.InputMatrix(InputStringMatrix,validateToken);
306
307     auto start = chrono::now();
308     SolveOptimal(gameMatrix,"Horizontal","",bufferSize,0,TokenSeqTemp,CoordinateOfFinalRow,0,0,0,sequenceList);
309     auto end = chrono::now();
310     OutputAnswer(chrono::duration_cast<SEC>(end-start));
311     OutputFile<out>(gameMatrix,method,chrono::duration_cast<SEC>(end-start));
312 }
313 return 0;
314 }
```

## 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
Masukkan 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
Masukan 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](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