

**Laporan Tugas Kecil 1**  
**Strategi Algoritma**  
**IF 2211**



Disusun Oleh :  
Rayhan Fadhlan Azka  
13522095

## Algoritma Program (Brute Force)

Program ini dibuat dengan menggunakan algoritma brute force, yaitu dengan menelusuri seluruh kemungkinan dengan beberapa optimasi. Program ini menggunakan rekursi untuk menentukan jumlah nested loop yang digunakan, nested loop yang digunakan sama dengan jumlah maksimal ukuran buffer. Cara kerja program ini adalah :

1. Pertama program akan menelusuri seluruh titik di baris pertama dan disimpan pada currentBuffer, dan currentCoordinate.
2. Setelah mendapat titik di baris pertama, program akan menelusuri seluruh titik dengan kolom yang sama tetapi dengan baris berbeda, hal ini dilakukan sambil mengecek apakah titik yang dituju sudah dicapai sebelumnya, jika sudah, maka titik tersebut akan di skip, selain itu setiap melalui titik juga akan mengupdate currentBuffer dan currentCoordinate, dan currentBuffer akan dicek dengan masing-masing sekuens dan akan ditentukan jumlah poin yang didapat, jika poin yang didapat lebih besar daripada maxPoint, maka resultBuffer dan resultCoordinate akan diupdate seperti currentBuffer dan currentCoordinate, dan maxPoint akan diubah seperti poin yang didapat saat ini.
3. Selanjutnya , akan ditelusuri titik di baris yang sama dan dilakukan prosedur yang sama seperti tahap 2. Hal ini dilakukan bergantian, yaitu titik yang di baris sama, lalu kolom yang sama, secara bergantian.
4. Jika ukuran currentBuffer sudah sama dengan ukuran maksimum buffer, maka currentBuffer dan currentCoordinate akan di reset dan dilakukan penelusuran pada titik awal yang selanjutnya.
5. Dalam program ini saya melakukan optimasi yaitu jika pada iterasi titik kedua, program akan mengecek apakah titik pertama atau titik kedua sudah mencapai starting sequence, contoh titik : AA BB , sequence : CC DD dan EE FF, maka program akan mengecek apakah  $AA = CC$  ,  $AA = EE$ ,  $BB = CC$ , dan  $BB = EE$ . Dalam contoh ini tidak ditemukan kecocokan, maka titik kedua ini akan di skip. Hal ini dapat dilakukan karena seluruh titik di matriks dapat diraih dalam maksimal 2 movement, dan jika tidak ada titik di matriks yang mencapai sequence di 2 movement pertama maka tidak perlu dicek movement-movement selanjutnya, karena 2 movement pertamanya sudah terjadi pemborosan movement.

## Source Code Program

Program ini saya buat menggunakan bahasa C++, hal ini saya lakukan karena bahasa C++ terkenal dengan kecepatan eksekusinya.

1. Struktur buatan dan library yang digunakan serta program utama.

```
1. #include <iostream>
2. #include <random>
3. #include <vector>
4. #include <string.h>
5. #include <sstream>
6. #include <fstream>
7. #include <chrono>
8. using namespace std;
9.
10. struct InputData {
11.     int bufferSize;
12.     int matrixWidth, matrixHeight;
13.     vector<vector<string>> matrix;
14.     int numSequence;
15.     vector<vector<string>> sequences;
16.     vector<int> points;
17. };
18.
19. struct ProcessNest {
20.     int maxPoint;
21.     vector<int> indices;
22.     vector<string> currentBuffer;
23.     vector<string> resultBuffer;
24.     vector<pair<int,int>> currentCoordinates;
25.     vector<pair<int,int>> resultCoordinates;
26. };
27.
28. int main(){
29.     cout << "WELCOME DI PERMAIN MERETAS ALA PUNK SIBER (versi CLI) !\n";
30.     InputData inputData;
31.     cout << "read dari file (1) atau generate random (2) ???\n";
32.     string inp;
33.     cin >> inp;
34.     while(inp!="1" && inp!="2"){
35.         cout << "masukkan antara 1 dan 2 !!!!\n";
36.         cin >> inp;
37.     }
38.     if(inp=="1"){
39.         readFromTxt(inputData);
40.     }
41.     else if(inp=="2"){
42.         generateRandom(inputData);
43.     }
44.     cout << endl;
45.     printInputData(inputData);
46.
47.
48.     ProcessNest processNest;
49.     auto start = chrono::high_resolution_clock::now();
50.
51.
52.     int depth = inputData.bufferSize;
53.     int checker = 1;
54.     processNest.maxPoint = 0;
55.
56.     nestedLoop(depth,1,inputData,processNest);
57.     auto end = chrono::high_resolution_clock::now();
58.     double duration = chrono::duration<double, std::milli>(end - start).count();
59.     cout << endl << "result : " << endl;
60.     writeToCLI(processNest,duration);
61.
```

```

62.     cout << "Apakah ingin menyimpan solusi? (y/n)\n";
63.     string check;
64.     cin >> check;
65.     if(check == "y" || check == "Y"){
66.         writeToTxt(processNest,duration);
67.     }
68.
69. }
70.

```

## 2. Algoritma utama yang melakukan looping dengan rukursi

```

1. bool isSubString(vector<string> & stringBesar, vector<string>& stringKecil){
2.     int sizeKecil = stringKecil.size();
3.     int sizeBesar = stringBesar.size();
4.
5.     for(int i = 0; i<=sizeBesar-sizeKecil;i++){
6.         int j;
7.         for(j = 0; j<sizeKecil; j++){
8.             if(stringBesar[i+j] != stringKecil[j]){
9.                 break;
10.            }
11.        }
12.        if(j == sizeKecil){
13.            return true;
14.        }
15.    }
16.    return false;
17. }
18.
19. int checkPointOfBuffer(vector<string> & buffer, vector<vector<string>> &
sequences,vector<int> &points){
20.
21.     int point = 0;
22.     for(int i = 0;i<sequences.size();i++){
23.         if(isSubString(buffer,sequences[i])){
24.             point+=points[i];
25.         }
26.     }
27.     return point;
28. }
29.
30. void nestedLoop(int depth, int checker, InputData & inputData, ProcessNest & processNest){
31.     // rekursi untuk specify berapa nest loopnya
32.     if(depth>0){
33.         int i;
34.         int iterateCount;
35.         if(checker == -1){
36.             iterateCount = inputData.matrixHeight;
37.         }
38.         else{
39.             iterateCount = inputData.matrixWidth;
40.         }
41.         checker*= -1;
42.         for(i=0;i<iterateCount;i++){
43.
44.
45.             int row,col;
46.             int indicesSize = processNest.indices.size();
47.
48.
49.             if(indicesSize!=0){
50.                 if(indicesSize%2==1){
51.                     col = processNest.indices[indicesSize-1];
52.                     row = i;
53.                 }
54.                 else{

```

```

55.         col = i;
56.         row = processNest.indices[indicesSize-1];
57.     }
58. }
59. else{
60.     row = 0;
61.     col = i;
62. }
63.
64. // mengecek apakah titik tersebut sudah disinggahi sebelumnya
65. bool flagCoordinate = false;
66. for(int j = 0; j<processNest.currentCoordinates.size();j++){
67.     if(processNest.currentCoordinates[j].first== col+1 &&
processNest.currentCoordinates[j].second== row + 1){
68.         flagCoordinate = true;
69.         break;
70.     }
71. }
72. if(flagCoordinate){
73.     continue;
74. }
75.
76. processNest.currentBuffer.push_back(inputData.matrix[row][col]);
77.
78. // mengecek apakah titik pertama dan kedua merupakan starting sequence
(optimasi)
79. if(indicesSize==1){
80.     bool flag = true;
81.     for(int j = 0; j<inputData.sequences.size();j++){
82.         if((processNest.currentBuffer[0]==inputData.sequences[j][0] &&
processNest.currentBuffer[1]==inputData.sequences[j][1]) ||
processNest.currentBuffer[1]==inputData.sequences[j][0]){
83.             flag = false;
84.             break;
85.         }
86.     }
87.     if(flag){
88.         processNest.currentBuffer.pop_back();
89.         continue;
90.     }
91. }
92.
93. int currentPoint = 0;
94.
95. processNest.currentCoordinates.push_back(make_pair(col+1,row+1));
96.
97. currentPoint =
checkPointOfBuffer(processNest.currentBuffer,inputData.sequences,inputData.points);
98.
99.
100. if(currentPoint<processNest.maxPoint){
101.     // do nothing
102. }
103.
104. else if(currentPoint>processNest.maxPoint || currentPoint ==
processNest.maxPoint && (processNest.currentBuffer.size() < processNest.resultBuffer.size())){
105.     processNest.maxPoint = currentPoint;
106.     processNest.resultBuffer = processNest.currentBuffer;
107.     processNest.resultCoordinates = processNest.currentCoordinates;
108. }
109.
110.
111. processNest.indices.push_back(i);
112.
113. nestedLoop(depth-1,checker,inputData,processNest);
114. processNest.indices.pop_back();
115. processNest.currentBuffer.pop_back();
116. processNest.currentCoordinates.pop_back();
117.
118. }

```

```

119.     }
120.     else{
121.
122.     }
123. }
124.

```

### 3. Input dan Output ke file

```

1. void readFromTxt(InputData & inputData){
2.     fstream file;
3.     string line;
4.     string basepath = "../test/input/";
5.     string filename;
6.     cout << "Enter file name: \n";
7.     cin >> filename;
8.     file.open(basepath + filename,ios::in);
9.     while(!file.is_open()){
10.        cout << "\nFile not found!\nMasukin nama yang bener !!!!\nMake sure file is in the
folder test/input\n";
11.        cout << "Enter file name: \n";
12.        cin >> filename;
13.        file.open(basepath + filename,ios::in);
14.    }
15.    if(file.is_open()){
16.        getline(file,line);
17.        inputData.bufferSize = stoi(line);
18.
19.        getline(file,line);
20.        istringstream iss;
21.        iss.str(line);
22.        iss >> inputData.matrixWidth >> inputData.matrixHeight;
23.        iss.clear();
24.        for(int i = 0 ; i < inputData.matrixHeight; i++){
25.            getline(file,line);
26.            iss.str(line);
27.            vector<string> row;
28.            for(int j = 0 ; j < inputData.matrixWidth; j++){
29.                string value;
30.                iss >> value;
31.                row.push_back(value);
32.            }
33.            inputData.matrix.push_back(row);
34.            iss.clear();
35.        }
36.
37.        getline(file,line);
38.        iss.str(line);
39.        iss >> inputData.numSequence;
40.        iss.clear();
41.
42.        for(int i = 0 ; i < inputData.numSequence ; i++){
43.            getline(file, line);
44.            iss.str(line);
45.            string word;
46.            vector<string> row;
47.            while(iss >> word){
48.                row.push_back(word);
49.            }
50.            inputData.sequences.push_back(row);
51.            iss.clear();
52.
53.            getline(file,line);
54.            iss.str(line);
55.            int val;
56.            iss >> val;
57.            inputData.points.push_back(val);

```

```

58.         iss.clear();
59.     }
60. }
61.
62.     file.close();
63. }

1. void writeToTxt(ProcessNest & processNest, double time){
2.     string basepath = "../test/output/";
3.     string filename;
4.     cout << "Enter file name: \n";
5.     cin >> filename;
6.     ofstream file(basepath+filename);
7.     if(file.is_open()){
8.         file << processNest.maxPoint << endl;
9.         for(int i = 0; i < processNest.resultBuffer.size(); i++){
10.             file << processNest.resultBuffer[i] << " ";
11.         }
12.         file << endl;
13.
14.         for(int i = 0; i < processNest.resultCoordinates.size(); i++){
15.             file << processNest.resultCoordinates[i].first << ", " <<
processNest.resultCoordinates[i].second << endl;
16.         }
17.         file << endl;
18.         file << time << " ms" << endl;
19.         file.close();
20.     }
21. }
22.
23. void writeToCLI(ProcessNest & processNest, double time){
24.     cout << processNest.maxPoint << endl;
25.
26.     for(int i = 0; i < processNest.resultBuffer.size(); i++){
27.         cout << processNest.resultBuffer[i] << " ";
28.     }
29.     cout << endl;
30.
31.     for(int i = 0; i < processNest.resultCoordinates.size(); i++){
32.         cout << (processNest.resultCoordinates[i].first) << ", " <<
(processNest.resultCoordinates[i].second) << endl;
33.     }
34.     cout << endl;
35.     cout << time << " ms" << endl;
36. }
37.

```

#### 4. Algoritma generate random matriks dan sekuens

```

1. vector<string> generateSequence(int len, vector<string> & uniqueToken){
2.     vector<string> res;
3.     for(int i = 0; i < len; i++){
4.         int RandNum = rand()%uniqueToken.size();
5.         res.push_back(uniqueToken[RandNum]);
6.     }
7.     return res;
8. }
9. bool isSeqExist(vector<vector<string>> & sequences, vector<string> & sequence){
10.    for(int i = 0; i < sequences.size(); i++){
11.        if(sequences[i] == sequence){
12.            return true;
13.        }
14.    }
15.    return false;
16. }
17.
18. void generateSequences(int numSequence, int ukuranMaxSeq, vector<vector<string>> &
sequences, vector<string> & uniqueToken){

```

```

19.     for(int i = 0 ; i<numSequence;i++){
20.         vector <string> temp;
21.         int randUkuran = rand()%(ukuranMaxSeq)+1;
22.         temp = generateSequence(randUkuran,uniqueToken);
23.         while(isSeqExist(sequences,temp)){
24.             int randUkuran = rand()%(ukuranMaxSeq)+1;
25.             temp = generateSequence(randUkuran,uniqueToken);
26.         }
27.
28.         sequences.push_back(temp);
29.     }
30. }
31.
32. void generateMatrix(int width,int height, vector<vector<string>> & matrix, int numTokenUnik,
vector<string> & uniqueToken){
33.
34.     for(int i = 0; i<height;i++){
35.         vector <string> row;
36.         for(int j = 0; j < width;j++){
37.
38.             int randNum = rand()%(numTokenUnik);
39.             row.push_back(uniqueToken[randNum]);
40.
41.         }
42.         matrix.push_back(row);
43.     }
44. }
45.
46. void generatePoints(vector<int> & points,int len){
47.     for(int i = 0 ; i < len ; i++){
48.         int randNum = rand()%(100-(-100)+1) + (-100);
49.         // random between -100 and 100
50.         points.push_back(randNum);
51.     }
52. }
53.
54. void generateRandom(InputData & inputData){
55.     int numTokenUnik;
56.     int numBuffer;
57.     int width,height;
58.     int numSequence;
59.     int ukuranMaxSeq;
60.     istringstream iss;
61.     cout << "masukkan jumlah token unik: \n";
62.     cin >> numTokenUnik;
63.     cout << "Masukkan token unik: \n";
64.     string inp;
65.     vector<string> uniqueToken;
66.     for(int i = 0 ; i < numTokenUnik;i++){
67.         cin >> inp;
68.         uniqueToken.push_back(inp);
69.     }
70.
71.     cout << "Masukkan size buffer: \n";
72.     cin >> numBuffer;
73.     inputData.bufferSize = numBuffer;
74.     cout << "Masukkan width dan height matrix: \n";
75.     cin >> width >> height;
76.
77.     inputData.matrixWidth = width;
78.     inputData.matrixHeight = height;
79.
80.
81.
82.     generateMatrix(width,height,inputData.matrix,numTokenUnik,uniqueToken);
83.     cout << "Masukkan jumlah sequence: \n";
84.     cin >> numSequence;
85.     inputData.numSequence = numSequence;
86.     cout << "Masukkan ukuran maksimal per sequence: \n";
87.     cin >> ukuranMaxSeq;

```



```
88.     while(ukuranMaxSeq<1){
89.         cout << "Ukuran tidak valid!\nMasukkan ulang: ";
90.         cin >> ukuranMaxSeq;
91.     }
92.     generateSequences(numSequence,ukuranMaxSeq,inputData.sequences,uniqueToken);
93.
94.     generatePoints(inputData.points,numSequence);
95.     cout << endl;
96.
97. }
98.
```

Generate random pada random points disini menggunakan range diantara -100 sampai 100

# Tangkapan Layar Program

## Contoh program

```
WELCOME DI PERMAIN MERETAS ALA PUNK SIBER (versi CLI) !
read dari file (1) atau generate random (2) ???
1
Enter file name:
tes1.txt

Buffer Size: 7
Matrix Width: 6
Matrix Height: 6
Matrix:
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
Number of Sequences: 3
Sequences:
BD E9 1C
15
BD 7A BD
20
BD 1C BD 55
30
```

```
WELCOME DI PERMAIN MERETAS ALA PUNK SIBER (versi CLI) !
read dari file (1) atau generate random (2) ???
2
masukkan jumlah token unik:
5
Masukkan token unik:
AA BB CC DD EE
Masukkan size buffer:
7
Masukkan width dan height matrix:
6 8
Masukkan jumlah sequence:
4
Masukkan ukuran maksimal per sequence:
4

Buffer Size: 7
Matrix Width: 6
Matrix Height: 8
Matrix:
BB CC EE AA EE EE
DD DD CC EE AA AA
BB CC BB BB AA CC
CC BB BB EE CC DD
CC CC BB BB DD AA
CC BB BB DD EE CC
CC EE AA EE DD BB
CC DD DD EE BB BB
Number of Sequences: 4
Sequences:
DD CC
-80
CC
-20
CC EE
-67
BB EE DD BB
-13
```

## Uji Testcase

(titik pada waktu ms menandakan koma, bukan ribuan)

TC 1:

Input :

```
1 4
2 6 4
3 aa bb cc dd ee ff
4 bb cc dd ee ff aa
5 cc dd ee ff ff ff
6 aa bb dd ee ee ee
7 3
8 aa
9 -15
10 bb dd ff
11 20
12 aa bb
13 25
```

Output :

```
1 20
2 bb dd ff
3 2, 1
4 2, 3
5 4, 3
6
7 0 ms
8
```

TC 2:

Input :

```

1  9
2  5 6
3  bb cc ee aa ee
4  ee dd dd cc ee
5  aa aa bb cc bb
6  bb aa cc cc bb
7  bb ee cc dd cc
8  cc bb bb dd aa
9  5
10 bb bb dd ee cc cc
11 -7
12 aa ee
13 93
14 bb cc dd dd ee bb
15 75
16 dd dd
17 -45
18 ee cc cc cc ee dd
19 -80
20

```

Output :

```

result :
123
bb cc dd dd ee bb aa ee
1, 1
1, 6
4, 6
4, 5
2, 5
2, 6
5, 6
5, 1

82.851 ms
Apakah ingin menyimpan solusi? (y/n)

```

TC 3 :

Input :

```

1  6
2  6 6
3  bb dd cc aa bb aa
4  cc cc cc aa bb bb
5  bb dd bb dd dd cc
6  dd aa dd aa cc bb
7  aa cc bb aa cc dd
8  dd cc dd cc bb aa
9  6
10 dd dd cc dd
11 -20
12 cc bb bb aa
13 -7
14 dd bb
15 93
16 dd
17 75
18 cc
19 -45
20 bb dd
21 -80

```

Output :

```

test 7 Output 7 test.txt
1  168
2  dd dd bb
3  2, 1
4  2, 3
5  1, 3
6
7  2.156 ms
8

```

TC 4 :

Input :

```

1      4
2      5 5
3      30 30 B2 B2 30
4      B2 A5 A5 B2 30
5      30 30 B2 A5 B2
6      30 B2 30 A5 A5
7      A5 A5 30 A5 B2
8      3
9      A5 30
10     40
11     30 30 A5
12     69
13     A5 A5 B2
14     51

```

Output :

```

1      109
2      30 30 A5 30
3      1, 1
4      1, 4
5      5, 4
6      5, 1
7
8      0 ms
9

```

TC 5 :

Input :

```

1      10
2      8 8
3      FF FF EE EE FF EE AA AA
4      EE CC FF FF BB DD BB FF
5      BB CC DD AA DD AA CC DD
6      EE EE DD CC CC FF FF AA
7      FF AA DD EE FF BB BB AA
8      FF DD CC DD DD CC DD BB
9      FF EE FF CC EE DD DD BB
10     FF DD BB EE EE FF CC AA
11     8
12     EE
13     61
14     CC EE EE CC DD
15     -84
16     DD EE CC
17     11
18     DD
19     92
20     CC DD FF AA
21     -19
22     AA CC EE CC FF
23     8
24     AA DD CC BB FF
25     -5
26     CC AA DD FF DD
27     16

```

Output :

```
test / Output / test.txt
1  180
2  FF CC AA DD FF DD EE CC
3  2, 1
4  2, 3
5  6, 3
6  6, 2
7  3, 2
8  3, 4
9  2, 4
10 2, 2
11
12 33253.9 ms
13
```

TC 6 :

Input :

```
1  7
2  6 6
3  AA AA AA AA AA AA
4  AA AA AA AA AA AA
5  AA AA AA AA AA AA
6  AA AA AA AA AA AA
7  AA AA AA AA AA AA
8  AA AA AA AA AA BB
9  3
10 AA AA AA AA
11 -100
12 AA AA AA
13 15
14 AA AA AA AA AA
15 70
16
```

Output :

```
test / Output / test.txt
1  15
2  AA AA AA
3  1, 1
4  1, 2
5  2, 2
6
7  7 ms
8
```

## LAMPIRAN

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan2	X	
2. Program berhasil dijalankan	X	
3. Program dapat membaca masukan berkas .txt	X	
4. Program dapat menghasilkan masukan secara acak	X	
5. Solusi yang diberikan program optimal	X	
6. Program dapat menyimpan solusi dalam berkas .txt	X	
7. Program memiliki GUI		X

Link repositori github : [https://github.com/RayhanFadhlan/Tucil1\\_13522095](https://github.com/RayhanFadhlan/Tucil1_13522095)