

LAPORAN TUGAS KECIL 1

IF2211/Strategi Algoritma

MAKE IT 24

Dipersiapkan oleh:

Christophorus Dharma Winata / 13521009

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

1 Ringkasan

Tugas kecil 1 mata kuliah IF2211 Strategi Algoritma semester genap diminta untuk membuat program untuk mencari solusi dari game 24 yang dimainkan dengan kartu *bridge*. Pada dek kartu ini, nilai-nilai yang masuk ke perhitungan mulai dari A, bilangan 2-10, J, Q, dan K. Dari dek tersebut bilangan bulat dianggap sebagai bilangan bulat seperti biasa, tetapi dengan tambahan A bernilai 1, J bernilai 11, Q bernilai 12, dan K bernilai 13.

Peraturan bermain game ini adalah setiap sekali permainan terdapat 4 nilai dimana dari 4 nilai harus dicari suatu persamaan aritmatik yang memiliki hasil 24. Operasi aritmatik terdiri dari penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan pengelompokkan dengan tanda kurung (). Tiap 1 nilai harus digunakan 1 kali namun dengan pengurutan yang bebas.

Program yang dibuat mengimplementasikan algoritma brute force untuk mencari semua kemungkinan yang ada pada permainan 24 tersebut dengan bahasa C++.

2 Penjelasan Program

Pada program yang dibuat urutan pekerjaan programnya adalah sebagai berikut:

- 2.1 Program utama dijalankan pada file main di folder src, dengan awalnya mendeklarasikan 4 nilai kartu sebagai suatu array 4 elemen *angka* yang bernilai 0. Dideklarasikan juga vector kosong *solutions* sebagai wadah penulisan solusi-solusi yang ditemukan, nilai integer *count* sebagai penyimpan banyak solusi. Dideklarasikan juga array of string *cards* yang berisi nilai *angka* dalam format kartu bridge yang berkorespondensi.
- 2.2 Pada awal program setelah menunjukkan display awal, dipanggil prosedur input() yang bekerja sebagai berikut:
 - Menerima suatu nilai integer 1 atau 2, nilai 1 berarti melakukan input game 24 secara manual via keyboard, nilai 2 berarti melakukan input acak yang dikeluarkan oleh program
 - Jika memilih 1, program akan meminta input nilai yang juga divalidasi dengan syarat harus bernilai A, J, Q, K, atau bilangan 2-10. Jika masukan tidak sesuai, program akan looping hingga didapat nilai yang sesuai. Pada akhir tahap ini, dibuat juga terjemahan nilai masukan dalam integer rentang [1,13] dalam bentuk array *angka*.
 - Jika memilih 2, program langsung mengeluarkan sekumpulan nilai integer acak yang dikonversi menjadi nilai dalam kartu yang berkorespondensi untuk dicetak kepada user. Nilai integer dimasukkan dalam array *angka*, dan koresponden nilai kartu dimasukkan ke array *cards*.
 - Setelah proses input sudah dilakukan baik opsi 1 atau 2, nilai-nilai array *angka* diteruskan dalam bentuk integer 4 elemen dengan rentang [1,13].
- 2.3 Setelah input sudah selesai, program melanjutkan dengan menyalakan timer program dan memanggil prosedur bruteforce() yang menjalankan algoritma untuk mencari solusi-solusi permainan. Prosedur bruteforce() sendiri bekerja sebagai berikut:
 - Prosedur bruteforce memiliki parameter input/output array *angka*, vektor *solutions*, dan integer *count*.

- Pada awal prosedur, dilakukan permutasi dari setiap kemungkinan urutan array *angka*. Karena input yang pasti hanya ada 4 buah. Maka kombinasi urutan angka dilakukan manual.
- Pada setiap kejadian, dilakukan permutasi operator aritmatika dengan pengelompokan suku menjadi sebagai berikut:

Misalkan X_i adalah elemen array *angka* dengan i adalah indeks elemen dari 1, dan opi adalah operator $+$, $-$, $*$, atau $/$ dengan i adalah bilangan cacah.

$X_1 \text{ op1 } (X_2 \text{ op2 } (X_3 \text{ op3 } X_4))$
 $X_1 \text{ op1 } ((X_2 \text{ op2 } X_3) \text{ op3 } X_4)$
 $(X_1 \text{ op1 } (X_2 \text{ op2 } X_3)) \text{ op3 } X_4$
 $((X_1 \text{ op1 } X_2) \text{ op2 } X_3) \text{ op3 } X_4$
 $(X_1 \text{ op1 } X_2) \text{ op2 } (X_3 \text{ op3 } X_4)$

- Permutasi dilakukan untuk semua opi dari kelima kombinasi tanda kurung dengan nested for i , j , dan k . k menjadi iterator $op1$, j menjadi iterator $op2$, dan i jadi iterator $op1$
 - Iterasi dilakukan dengan iterator 0 sampai 3 dimana 0 adalah operasi penjumlahan, 1 adalah operasi pengurangan, 2 adalah operasi perkalian, dan 3 adalah operasi pembagian. Terdapat beberapa constraint yang diterapkan pada operasi pembagian yaitu pada suatu operasi pembagian, penyebutnya tidak boleh bernilai 0 dan hasil operasi pembagian tidak diperbolehkan bilangan yang tak bulat.
 - Ketika permutasi perhitungan mendapatkan nilai 24, nilai integer *count* dan semua nilai serta operasi pada kejadian tersebut diubah dan digabungkan menjadi sebuah string lalu dilakukan pushback pada vektor solutions.
- 2.4 Setelah prosedur dilakukan, yang selanjutnya dikerjakan adalah menghentikan timer program, serta mencetak semua hasil yang sudah didapat dari algoritma bruteforce.
- 2.5 Program selanjutnya meminta masukan untuk menyimpan hasil solusi yang ditemukan atau tidak. Jika ya, maka file text pada folder test bernama solutions.txt akan dibuat. Jika tidak, pembuatan file akan dilewati.
- 2.6 Setelah melakukan penyimpanan, program menunjukkan lama runtime program dalam menjalankan program bruteforce dalam milisekon.

3 Source Code

Berikut screenshot dari source code program yang dibuat dalam bahasa C++:

- main.cpp

```
File Edit Selection View Go Run ... main.cpp - Tucil1_13521009 - Visua...
main.cpp x
src > main.cpp > input()
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  #include <ctime>
5  #include <vector>
6  #include <bits/stdc++.h>
7  #include <chrono>
8  #include "bruteforce.cpp"
9  using namespace std::chrono;
10 using namespace std;
11
12 //KAMUS
13 int mode;
14 int angka[4];
15 string cards[4];
16
17 void input();
18 void translate_toAngka();
19 void translate_toCards();
20
21 void input(){
22     string valid_input = "A23456789JQK";
23     string sepuluh = "10";
24     do
25     {
26         cin >> mode;
27     } while (mode != 1 && mode != 2);
28
29     if (mode == 1){
30         cout << "KEYBOARD INPUT:" << endl;
31         for (int i = 0; i < 4; i++)
32         {
33             do
34             {
35                 cin >> cards[i];
36                 if (valid_input.find(cards[i]) == -1 && cards[i] != sepuluh)
37                 {
38                     cout << "Masukan tidak sesuai (input harus antara A, 2-10, J, Q, atau K)" << endl;
39                 }
40             } while (valid_input.find(cards[i]) == -1 && cards[i] != sepuluh);
41         }
42         translate_toAngka();
43     }
44     else if (mode == 2){
45         cout << "RANDOM INPUT:" << endl;
46         for (int i = 0; i < 4; i++)
47         {
48             angka[i] = rand() % 13+1;
49         }
50         translate_toCards();
51     }
52     if (mode == 1){
53         cout << "INPUT YANG DITERIMA: " << endl;
54     }
55     for (int i = 0; i < 4; i++)
56     {
57         cout << cards[i] << " ";
58     }
59 }
60
```

Ln 26, Col 21 Spaces: 4 UTF-8 CRLF C++ Win32

```
File Edit Selection View Go Run ... main.cpp - Tucil1_13521009 - Visua...
main.cpp x
src > main.cpp > input()
61 void translate_toAngka(){
62     for (int i = 0; i < 4; i++)
63     {
64         if (cards[i] == "A")
65         {
66             angka[i] = 1;
67         }
68         else if (cards[i] == "J")
69         {
70             angka[i] = 11;
71         }
72         else if (cards[i] == "Q")
73         {
74             angka[i] = 12;
75         }
76         else if (cards[i] == "K")
77         {
78             angka[i] = 13;
79         }
80         else
81         {
82             angka[i] = stoi(cards[i]);
83         }
84     }
85 }
86
87 void translate_toCards(){
88     for (int i = 0; i < 4; i++)
89     {
90         if (angka[i] == 1)
91         {
92             cards[i] = "A";
93         }
94         else if (angka[i] == 11)
95         {
96             cards[i] = "J";
97         }
98         else if (angka[i] == 12)
99         {
100             cards[i] = "Q";
101         }
102         else if (angka[i] == 13)
103         {
104             cards[i] = "K";
105         }
106         else
107         {
108             cards[i] = to_string(angka[i]);
109         }
110     }
111 }
112
113 int main() {
114     char save;
115     int count = 0;
116     vector<string> solutions;
117     //inisialisasi random untuk random input
118     srand(time(NULL));
119     //deklarasi kumpulan angka sebagai 0
120     int angka[4];
121     string cards[4];
122     int i;
123     while (count < 4)
124     {
125         for (i = 0; i < 4; i++)
126         {
127             cards[i] = "";
128             angka[i] = 0;
129         }
130         translate_toCards();
131         translate_toAngka();
132         count++;
133     }
134 }
```

Ln 26, Col 21 Spaces: 4 UTF-8 CRLF C++ Win32

```
File Edit Selection View Go Run ... main.cpp - Tucil1_13521009 - Visual ...
src > main.cpp > input()
112
113 int main() {
114     char save;
115     int count = 0;
116     vector<string> solutions;
117     //inisialisasi random untuk random input
118     srand(time(NULL));
119     //deklarasi kumpulan angka sebagai 0
120     for (int i = 0; i < 4; i++)
121     {
122         angka[i] = 0;
123     }
124     cout << "===== " << endl; \
125     cout << "PROGRAM MENCARI SOLUSI DARI PERMAINAN KARTU 24" << endl;
126     cout << "===== " << endl;
127     cout << "by Christophorus D.W. / NIM 13521009" << endl << endl;
128     cout << "PILIH MODE INPUT:" << endl;
129     cout << "1. KEYBOARD INPUT" << endl;
130     cout << "2. RANDOM INPUT" << endl;
131     cout << "PILIHAN: ";
132     input();
133     auto start = high_resolution_clock::now();
134     bruteforce(angka, solutions, count);
135     auto stop = high_resolution_clock::now();
136     cout << endl;
137     cout << count << " solutions found" << endl;
138     for (vector<string>::const_iterator i = solutions.begin(); i != solutions.end(); ++i){
139         cout << *i << endl;
140     }
141     cout << "Apakah ingin menyimpan solusi ke file? (Y/N): ";
142     do
143     {
144         cin >> save;
145         if (save == 'Y' || save == 'y'){
146             ofstream myfile("test/solutions.txt");
147             myfile << count << " solutions found" << endl;
148             for (vector<string>::const_iterator i = solutions.begin(); i != solutions.end(); ++i){
149                 myfile << *i << endl;
150             }
151             cout << "Solusi disimpan ke file solutions.txt" << endl;
152         } else if (save == 'N' || save == 'n'){
153             cout << "Solusi tidak disimpan" << endl;
154         }
155         else {
156             cout << "Masukan tidak sesuai (input harus Y atau N)" << endl;
157         }
158     } while (save != 'Y' && save != 'y' && save != 'N' && save != 'n');
159     auto duration = stop - start;
160     cout << "Runtime program : " << chrono::duration<double, milli>(duration).count() << " ms" << endl;
161     cout << "Program selesai! Terima kasih! Press enter to close" << endl;
162     cin.ignore();
163     cin.get();
164     return 0;
165 }
```

- bruteforce.cpp

```
File Edit Selection View Go Run Terminal Help
bruteforce.cpp - Tuci1_13521009 - Visual Studio Code

src> bruteforce.cpp > ...
1 #include <iostream>
2 #include <cstdlib>
3 #include <vector>
4 using namespace std;
5
6 //fungsi operasi 2 bilangan dengan operator tertentu
7 //operator diassign ke suatu nilai integer untuk mempermudah dalam looping
8 double operate(double a, double b, int o){
9     if (o == 0) {return a + b;}
10    else if (o == 1) {return a - b;}
11    else if (o == 2) {return a * b;}
12    else if (o == 3) {
13        if (b != 0 && a/b == (int)(a/b)) {return a/b;}
14        else {return 0;}
15    }
16 }
17
18 char translate_toOperator(int o){
19     if (o == 0) {return '+';}
20     else if (o == 1) {return '-'};
21     else if (o == 2) {return '*'};
22     else if (o == 3) {return '/'};
23 }
24
25 // sudah ada 4 bilangan, dicari nilai 24 dengan permutasi operator
26 void op_permutation(int numbers[4], vector<string> &result, int &count){
27     double temp;
28     //X1 op1 (X2 op2 (X3 op3 X4))
29     for (int i = 0; i < 4; i++)
30     {
31         for (int j = 0; j < 4; j++)
32         {
33             for (int k = 0; k < 4; k++)
34             {
35                 temp = operate(numbers[0], operate(numbers[1], operate(numbers[2], numbers[3], k), j), i);
36                 if (temp == 24) {
37                     count++;
38                     result.push_back(to_string(numbers[0]) + translate_toOperator(i) + "(" + to_string(numbers[1]) + translate_toOperator(j) + "(" + to_string(numbers[2]) + translate_toOperator(k) + to_string(numbers[3]) + ")")");
39                 }
40             }
41         }
42     }
43     //X1 op1 ((X2 op2 X3) op3 X4)
44     for (int i = 0; i < 4; i++)
45     {
46         for (int j = 0; j < 4; j++)
47         {
48             for (int k = 0; k < 4; k++)
49             {

```

```
File Edit Selection View Go Run Terminal Help
bruteforce.cpp - Tuci1_13521009 - Visual Studio Code

src> bruteforce.cpp > op_permutation(int & numbers, int & count)
43 //X1 op1 ((X2 op2 X3) op3 X4)
44 for (int i = 0; i < 4; i++)
45 {
46     for (int j = 0; j < 4; j++)
47     {
48         for (int k = 0; k < 4; k++)
49         {
50             temp = operate(numbers[0], operate(operate(numbers[1], numbers[2], k), numbers[3], j), i);
51             if (temp == 24) {
52                 count++;
53                 result.push_back(to_string(numbers[0]) + translate_toOperator(i) + "(" + to_string(numbers[1]) + translate_toOperator(k) + to_string(numbers[2]) + ")" + translate_toOperator(j) + to_string(numbers[3]) + ")");
54             }
55         }
56     }
57 }
58 //X1 op1 (X2 op2 X3) op3 X4
59 for (int i = 0; i < 4; i++)
60 {
61     for (int j = 0; j < 4; j++)
62     {
63         for (int k = 0; k < 4; k++)
64         {
65             temp = operate(operate(numbers[0], operate(numbers[1], numbers[2], k), j), numbers[3], i);
66             if (temp == 24) {
67                 count++;
68                 result.push_back("(" + to_string(numbers[0]) + translate_toOperator(j) + "(" + to_string(numbers[1]) + translate_toOperator(k) + to_string(numbers[2]) + ")") + translate_toOperator(i) + to_string(numbers[3]));
69             }
70         }
71     }
72 }
73 //((X1 op1 X2) op2 X3) op3 X4
74 for (int i = 0; i < 4; i++)
75 {
76     for (int j = 0; j < 4; j++)
77     {
78         for (int k = 0; k < 4; k++)
79         {
80             temp = operate(operate(operate(numbers[0], numbers[1], k), numbers[2], j), numbers[3], i);
81             if (temp == 24) {
82                 count++;
83                 result.push_back("(" + to_string(numbers[0]) + translate_toOperator(k) + to_string(numbers[1]) + ")" + translate_toOperator(j) + to_string(numbers[2]) + ")" + translate_toOperator(i) + to_string(numbers[3]));
84             }
85         }
86     }
87 }
88 //X1 op1 X2) op2 (X3 op3 X4)
89 for (int i = 0; i < 4; i++)
90 {
91     for (int j = 0; j < 4; j++)
92     {

```



```
File Edit Selection View Go Run Terminal Help
bruteforce.cpp - Tux1, 13/2/2020 - Visual Studio Code

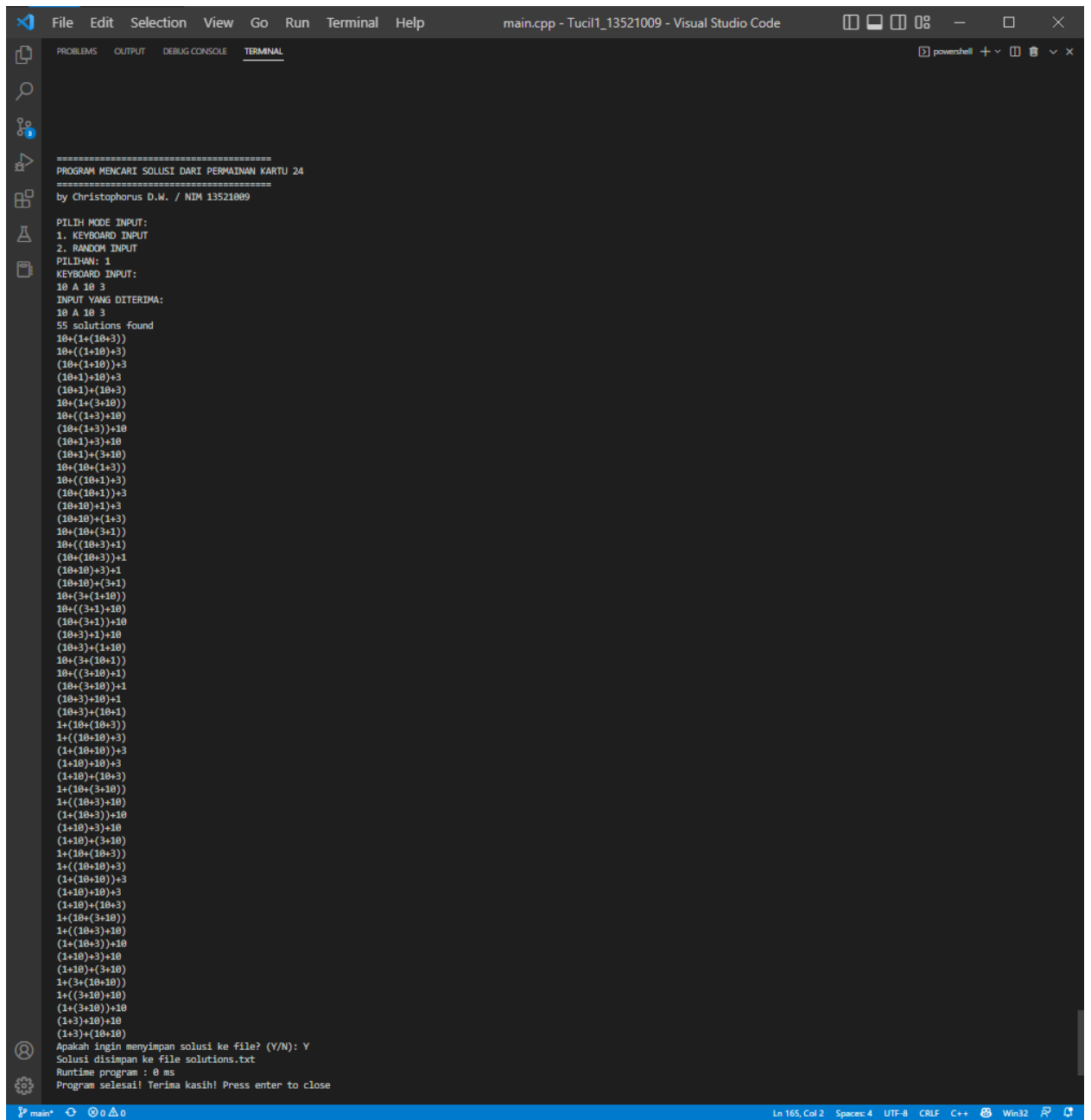
src > bruteforce.cpp x
src > bruteforce.cpp > op_permutation(int i4, vector<string> &0, int &0)

88 //X1 op1 X2) op2 (X3 op3 X4)
89 for (int i = 0; i < 4; i++)
90 {
91     for (int j = 0; j < 4; j++)
92     {
93         for (int k = 0; k < 4; k++)
94         {
95             temp = operate(operate(numbers[0], numbers[1], j), operate(numbers[2], numbers[3], k), i);
96             if (temp == 24) {
97                 count++;
98                 result.push_back("(" + to_string(numbers[0]) + translate_toOperator(j) + to_string(numbers[1]) + ")" + translate_toOperator(i) + "(" + to_string(numbers[2]) + translate_toOperator(k) + to_string(numbers[3]) + ")"");
99             }
100         }
101     }
102 }
103
104 //mencari semua urutan bilangan
105 void bruteforce(int numbers[4], vector<string> &result, int &count){
106     int temp[4];
107     //X1 X2 X3 X4
108     op_permutation(numbers, result, count);
109     //X1 X2 X4 X3
110     temp[0] = numbers[0];
111     temp[1] = numbers[1];
112     temp[2] = numbers[3];
113     temp[3] = numbers[2];
114     op_permutation(temp, result, count);
115     //X1 X3 X2 X4
116     temp[0] = numbers[0];
117     temp[1] = numbers[2];
118     temp[2] = numbers[1];
119     temp[3] = numbers[3];
120     op_permutation(temp, result, count);
121     //X1 X3 X4 X2
122     temp[0] = numbers[0];
123     temp[1] = numbers[2];
124     temp[2] = numbers[3];
125     temp[3] = numbers[1];
126     op_permutation(temp, result, count);
127     //X1 X4 X2 X3
128     temp[0] = numbers[0];
129     temp[1] = numbers[3];
130     temp[2] = numbers[1];
131     temp[3] = numbers[2];
132     op_permutation(temp, result, count);
133     //X1 X4 X3 X2
134     temp[0] = numbers[0];
135     temp[1] = numbers[3];
136 }
```

```
src> C: bruteforce.cpp > op_permutation(int (&vector<string> &result, int &count){
105 //mencari semua urutan bilangan
106 void bruteforce(int numbers[4], vector<string> &result, int &count){
107     int temp[4];
108     //X1 X2 X3 X4
109     op_permutation(numbers, result, count);
110     //X1 X2 X4 X3
111     temp[0] = numbers[0];
112     temp[1] = numbers[1];
113     temp[2] = numbers[3];
114     temp[3] = numbers[2];
115     op_permutation(temp, result, count);
116     //X1 X3 X2 X4
117     temp[0] = numbers[0];
118     temp[1] = numbers[2];
119     temp[2] = numbers[1];
120     temp[3] = numbers[3];
121     op_permutation(temp, result, count);
122     //X1 X3 X4 X2
123     temp[0] = numbers[0];
124     temp[1] = numbers[2];
125     temp[2] = numbers[3];
126     temp[3] = numbers[1];
127     op_permutation(temp, result, count);
128     //X4 X2 X3 X1
129     temp[0] = numbers[0];
130     temp[1] = numbers[3];
131     temp[2] = numbers[1];
132     temp[3] = numbers[2];
133     op_permutation(temp, result, count);
134     //X1 X4 X3 X2
135     temp[0] = numbers[0];
136     temp[1] = numbers[3];
137     temp[2] = numbers[2];
138     temp[3] = numbers[1];
139     op_permutation(temp, result, count);
140     //X2 X1 X3 X4
141     temp[0] = numbers[1];
142     temp[1] = numbers[0];
143     temp[2] = numbers[2];
144     temp[3] = numbers[3];
145     op_permutation(temp, result, count);
146     //X2 X1 X4 X3
147     temp[0] = numbers[1];
148     temp[1] = numbers[0];
149     temp[2] = numbers[3];
150     temp[3] = numbers[2];
151     op_permutation(temp, result, count);
152     //X2 X3 X1 X4
153     temp[0] = numbers[1];
154     temp[1] = numbers[2];
155     temp[2] = numbers[0];
156     temp[3] = numbers[3];
157     op_permutation(temp, result, count);
158     //X2 X3 X4 X1
159     temp[0] = numbers[1];
160     temp[1] = numbers[2];
161     temp[2] = numbers[3];
162     temp[3] = numbers[0];
163     op_permutation(temp, result, count);
164     //X2 X4 X1 X3
165     temp[0] = numbers[1];
166     temp[1] = numbers[3];
167     temp[2] = numbers[0];
168     temp[3] = numbers[2];
169     op_permutation(temp, result, count);
170 }
```

Link dari repository source code juga dilampirkan pada belakang dokumen ini.

4 Test



```
File Edit Selection View Go Run Terminal Help main.cpp - Tucil1_13521009 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
=====
PROGRAM MENCAIRI SOLUSI DARI PERMAINAN KARTU 24
=====
by Christophorus D.W. / NIM 13521009

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHAN: 1
KEYBOARD INPUT:
10 A 10 3
INPUT YANG DITERIMA:
10 A 10 3
55 solutions found
10+(1+(10+3))
10+((1+10)+3)
(10+(1+10))+3
(10+1)+10+3
(10+1)+(10+3)
10+(1+(3+10))
10+((1+3)+10)
(10+(1+3))+10
(10+1)+3+10
(10+1)+(3+10)
10+(10+(1+3))
10+((10+1)+3)
(10+(10+1))+3
(10+10)+1+3
(10+10)+(1+3)
10+(10+(3+1))
10+((10+3)+1)
(10+(10+3))+1
(10+10)+3+1
(10+10)+(3+1)
10+(3+(1+10))
10+((3+1)+10)
(10+(3+1))+10
(10+3)+1+10
(10+3)+(1+10)
10+(3+(10+1))
10+((3+10)+1)
(10+(3+10))+1
(10+3)+10+1
(10+3)+(10+1)
1+(10+(10+3))
1+((10+10)+3)
(1+(10+10))+3
(1+10)+10+3
(1+10)+(10+3)
1+(10+(3+10))
1+((10+3)+10)
(1+(10+3))+10
(1+10)+3+10
(1+10)+(3+10)
1+(10+(10+3))
1+((10+10)+3)
(1+(10+10))+3
(1+10)+10+3
(1+10)+(10+3)
1+(10+(3+10))
1+((10+3)+10)
(1+(10+3))+10
(1+10)+3+10
(1+10)+(3+10)
1+(3+(10+10))
1+((3+10)+10)
(1+(3+10))+10
(1+3)+10+10
(1+3)+(10+10)
Apakah ingin menyimpan solusi ke file? (Y/N): Y
Solusi disimpan ke file solutions.txt
Runtime program : 0 ms
Program selesai! Terima kasih! Press enter to close
```

4.1

```
C:\Users\Christophorus Dharma\OneDrive - Institut Teknologi Bandung\Stima\Tucil 1 Stima\Tucil1_13521009\bin\main.exe
=====
PROGRAM MENCARI SOLUSI DARI PERMAINAN KARTU 24
=====
by Christophorus D.W. / NIM 13521009

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHAN: 1
KEYBOARD INPUT:
K K K K
INPUT YANG DITERIMA:
K K K K
0 solutions found
Apakah ingin menyimpan solusi ke file? (Y/N): N
Solusi tidak disimpan
Runtime program : 0 ms
Program selesai! Terima kasih! Press enter to close
-
```

4.2

4.3

```

118 srand(time(NULL));
119 //Randomly generate single integer 0
120 for (int i = 0; i < d2; i++)
121 {
122     arr[i] = 0;
123 }
124 cout << "===== PENANGKAP SOLUSI DARI PERMAINAN KARTU 24 =====>>> endl;
125 cout << "===== PILIHAN MODE INPUT =====>>> endl;
126 cout << "By Christopherus D.M. / NIM 13521009" << endl << endl;
127 cout << "PILIH MODE INPUT :>>> endl;
128 cout << "1. KEYBOARD INPUT" << endl;
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
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1 18 solutions found
2 (3-(4-5))*6
3 (3-4)+5)*6
4 (3+(5-4))*6
5 (3/5)+4)*6
6 (3+5)-4)*6
7 (3/5)+(4*6
8 (3/5)+6)*4
9 (3/5)+(6*4
10 4*((3/5)+6
11 (4+(3/5))*6
12 (4-(3/5))*6
13 4*((5/3)+6
14 (4+(5/3))*6
15 (4-(5/3))*6
16 4*(6+(3/5))
17 4*(6-(3/5))
18 (4*6)+(3/5)
19 (4*6)-(3/5)
20

```

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHAN: 1
KEYBOARD INPUT:

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure for 'TUCIL1_13521009'. The main.cpp file is open in the editor. The terminal window at the bottom shows the output of the program, which includes a prompt to save solutions to a file, followed by a list of solutions found and a confirmation to save them.

```
src> g++ main.cpp > ...  
130     if (vector<string>::const_iterator i = solutions.begin(); i !=  
131         cout << *i << endl;  
140     }  
141     cout << "Apakah ingin menyimpan solusi ke file? (Y/N): ";  
142     do  
143     {  
144         cin >> save;  
145         if (save == "Y" || save == "y")  
146             save = "y";  
147     } while (save != "Y" && save != "y");  
148     if (save == "Y" || save == "y")  
149     {  
150         ofstream file("solutions.txt");  
151         if (!file.is_open())  
152             cout << "Gagal membuka file!" << endl;  
153         for (vector<string>::const_iterator i = solutions.begin(); i !=  
154             file << *i << endl;  
155         file.close();  
156     }  
157 }
```

Solusi disimpan ke file solutions.txt
Runtime program : 1 ms
Program selesai! Terima kasih! Press enter to close

PS C:\Users\Christophorus.Dharma\OneDrive - Institut Teknologi Bandung\Stima\Tucil 1 Stima\Tucil1_13521009\main
=====

PROGRAM MENCAIRI SOLUSI DARI PERMAINAN KARTU 24

=====

by Christophorus D.M. / NIM 13521009

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHAN: 2
RANDOM INPUT:
3 4 7 Q
6 solutions found
3+(7*(12/4))
3+((7*(12/4))
(3*7)+(12/4)
3+((12/4)*7)
3+((12*7)/4)
(4*7)-(12/3)
Apakah ingin menyimpan solusi ke file? (Y/N): y
Solusi disimpan ke file solutions.txt
Runtime program : 1 ms
Program selesai! Terima kasih! Press enter to close

4.4

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure for 'TUCIL1_13521009'. The solutions.txt file is open in the editor. The terminal window at the bottom shows the output of the program, which includes a list of solutions found and a confirmation to save them.

```
test> g++ solutions.txt > ...  
1 6 solutions found  
2 3+(7*(12/4))  
3 3+((7*(12/4))  
4 (3*7)+(12/4)  
5 3+((12/4)*7)  
6 3+((12*7)/4)  
7 (4*7)-(12/3)  
8
```

3 4 7 Q
6 solutions found
3+(7*(12/4))
3+((7*(12/4))
(3*7)+(12/4)
3+((12/4)*7)
3+((12*7)/4)
(4*7)-(12/3)
Apakah ingin menyimpan solusi ke file? (Y/N): y
Solusi disimpan ke file solutions.txt
Runtime program : 1 ms
Program selesai! Terima kasih! Press enter to close

```
C:\Users\Christophorus Dharma\OneDrive - Institut Teknologi Bandung\Stima\Tucil 1 Stima\Tucil1_13521009\bin\m...
=====
PROGRAM MENCARI SOLUSI DARI PERMAINAN KARTU 24
=====
by Christophorus D.W. / NIM 13521009

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHAN: 2
RANDOM INPUT:
A 9 5 6
17 solutions found
1*((9-5)*6)
(1*(9-5))*6
(1*9)-5)*6
1*(6*(9-5))
(1*6)*(9-5)
(9-(1*5))*6
(9*1)-5)*6
(9/1)-5)*6
(9-(5*1))*6
(9-(5/1))*6
(9-5)*1)*6
(9-5)/1)*6
(9-5)*(1*6)
(9-5)*6)*1
(9-5)*6)/1
(9-5)*(6*1)
(9-5)*(6/1)
Apakah ingin menyimpan solusi ke file? (Y/N): n
Solusi tidak disimpan
Runtime program : 1 ms
Program selesai! Terima kasih! Press enter to close
```

4.5

4.6

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure for 'TUCIL1_13521009'. The main.cpp file is open in the editor, showing C++ code for a program that finds solutions to a 24-point game. The terminal window at the bottom shows the program's output, including the title 'PROGRAM Mencari Solusi dari Permainan Kartu 24', the user's name 'by Christophorus D.W. / NIM 13521009', and the list of 14 solutions found for the input '5 6 A 3'. The solutions are mathematical expressions that evaluate to 24, such as $5 + ((6 * 3) + 1)$.

```

src > C:\main.cpp > ...
138 for (vector<string>::const_iterator i = solutions.begin(); i !=
139 cout << *i << endl;
140 }
141 cout << "Apakah ingin menyimpan solusi ke file? (Y/N): ";
142 do
143 {
144     cin >> save;
145     if (save == 'Y' || save == 'y'){
146         ofstream myfile("test/solutions.txt");
147         myfile << count << " solutions found" << endl;
148         for (vector<string>::const_iterator i = solutions.begin(); i !=
149             myfile << *i << endl;
150         }
151         cout << "Solusi disimpan ke file solutions.txt" << endl;
152     } else if (save == 'N' || save == 'n'){
153         cout << "Solusi tidak disimpan" << endl;
154     }
155     else {
156         cout << "Masukan tidak sesuai (input harus Y atau N)" << endl;
157     }
158 } while (true);
159 }
160
=====
PROGRAM Mencari Solusi dari Permainan Kartu 24
=====
by Christophorus D.W. / NIM 13521009

PILIH MODE INPUT:
1. KEYBOARD INPUT
2. RANDOM INPUT
PILIHMODE: 2
RANDOM INPUT:
5 6 A 3
14 solutions found
5+((6*3)+1)
(5+((6*3)+1))
5+(1+(6*3))
(5+1)+(6*3)
5+(1+(3*6))
(5+1)*3+6
(5+1)*(3*6)
5+((3*6)+1)
6+((5+1)*3)
6+((1+5)*3)
6+(3*(5+1))
(6*3)+5+1
(6*3)*(5+1)
Apakah ingin menyimpan solusi ke file? (Y/N): y
Solusi disimpan ke file solutions.txt
Runtime program : 0 ms
Program selesai! Terima kasih! Press enter to close

```

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure. The solutions.txt file is open in the editor, showing the list of 14 solutions found for the input '5 6 A 3'. The solutions are mathematical expressions that evaluate to 24, such as $5 + ((6 * 3) + 1)$.

```

test > C:\main.cpp > ...
1 14 solutions found
2 5+((6*3)+1)
3 (5+((6*3)+1))
4 5+(1+(6*3))
5 (5+1)+(6*3)
6 5+(1+(3*6))
7 (5+1)*3+6
8 (5+1)*(3*6)
9 5+((3*6)+1)
10 5+((3*6)+1)
11 6+((5+1)*3)
12 6+((1+5)*3)
13 6+(3*(5+1))
14 (6*3)+5+1
15 (6*3)*(5+1)
16

```


5 Lampiran

Berikut link repository GitHub source code yang dikerjakan : [Tucil1_13521009](#)

Ceklist untuk asisten:

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan		
2. Program berhasil running		
3. Program dapat membaca input / generate sendiri dan memberikan luaran		
4. Solusi yang diberikan memenuhi (berhasil mencapai 24)		
5. Program dapat menyimpan solusi dalam file teks		