

Laporan Tugas Kecil IF2211 Strategi Algoritma

## Penyelesaian Permainan Kartu 24 dengan Algoritma Brute-Force

Ditujukan untuk memenuhi salah satu tugas kecil mata kuliah IF2211 Strategi Algoritma pada Semester II Tahun Akademik 2022/2023

Disusun oleh :  
Aulia Mey Diva Annandya      K1/13521103



PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2023

## A. Algoritma Brute-Force

Algoritma Brute-Force adalah algoritma dengan pendekatan yang lempang atau pendekatan yang straightforward untuk memecahkan suatu persoalan. Maksudnya adalah menyelesaikan suatu masalah dengan cara yang mudah atau terpampang dengan jelas. Algoritma Brute-Force ini cenderung lebih menggunakan tenaga daripada otak. Contoh-contoh dari algoritma ini, antara lain: Mencari elemen terbesar maupun terkecil, pencarian beruntun, mngitung matriks, dan banyak lainnya.

Pada tugas kecil 1 Strategi Algoritma ini, mahasiswa diminta untuk membuat sebuah *game* kartu 24 dengan menggunakan algoritma Brute-Force. Secara garis besar, program akan meminta input dari user ataupun program dapat men-generate sendiri. Input yang diterima akan di-concat menjadi satu dan di permutasi agar menghasilkan semua kemungkinan. Setelahnya, setiap kemungkinan permutasi akan diuji satu per-satu dengan setiap kemungkinan operator dan kurung. Jika pengujian tersebut menghasilkan 24, maka kemungkinan tadi akan di push ke dalam vektor hasil.

Alur Brute-Force untuk penyelesaian *game* Kartu 24 :

1. Menerima input 4 string dari user maupun random-generator.
2. Mengecek apakah keempat string ada yang berupa dua digit yaitu “10”
3. Jika ada string yang berupa dua digit, string diubah menjadi string satu digit “Z” dengan prosedur tenToZ.
4. Setelah semua string berupa satu digit, semua string di concat menjadi sebuah string kesatuan yang nantinya akan dipermutasi dengan prosedur permutation.
5. Setelah mendapatkan semua kemungkinan permutasi, setiap string kesatuan akan dipecah kembali dan string “Z” diubah kembali menjadi string “10” dengan menggunakan prosedur singleString.
6. Untuk setiap kemungkinan, keempat string tadi akan diubah menjadi integer dan dilakukan operasi dengan semua kemungkinan operator dan kurung dengan menggunakan prosedur is24.

Maka dari itu, saya mengimplementasikan alur tersebut dalam program saya sebagai berikut:

1. double operations

Fungsi ini memiliki 3 parameter, yaitu string op, double num1, dan double num2. Fungsi akan mengembalikan sebuah hasil dimana kedua angka telah dioperasikan berdasarkan parameter op yang diterima.

```

double operations (string op, double num1, double num2){
    //operasi matematika
    double hasil;

    if (op == "+"){
        hasil = num1 + num2;
    } else if (op == "-"){
        hasil = num1 - num2;
    } else if (op == "*"){
        hasil = num1 * num2;
    } else if (op == "/"){
        hasil = num1 / num2;
    }

    return hasil;
}

```

## 2. void tenToZ

Prosedur ini memiliki 5 parameter, yaitu string k1, string k2, string k3, string k4, dan int count. Prosedur akan mengubah string input yang berbunyi “10” menjadi “Z”. Maksudnya string “10” memiliki 2 digit yang berbeda. Apabila nanti semua input di concat maka akan memiliki hasil permutasi yang salah sehingga input diubah terlebih dahulu menjadi string 1 digit. Count berguna untuk UI dalam mencetak solusi nantinya.

```

void tenToZ(string& k1, string& k2, string& k3, string& k4, int& count){
    //Mengubah string 10 yang berupa dua digit menjadi string 1 huruf untuk di permutasi
    if (k1 == "10"){
        k1 = "Z";
        count++;
    }
    if (k2 == "10"){
        k2 = "Z";
        count++;
    }
    if (k3 == "10"){
        k3 = "Z";
        count++;
    }
    if (k4 == "10"){
        k4 = "Z";
        count++;
    }
}

```

## 3. void permutation

Prosedur ini memiliki 3 parameter, yaitu string kartu, int n, dan vector<string> hasil. Prosedur digunakan untuk mencari semua kemungkinan dari permutasi sebuah string input kartu. Nantinya, semua kemungkinan tersebut akan di pushback ke dalam vektor string hasil. Untuk int n, digunakan sebagai awal start. Perlu diperhatikan untuk kemungkinan input yang sama misal A A 4 8. Jika A pertama dan A kedua ditukar, akan menghasilkan sebuah string yang sama. Kemungkinan ganda ini harus di-handle sehingga hasil yang dikeluarkan merupakan hasil yang benar.

```

void permutation(string& kartu, int n, vector<string>& hasil){
    double temp;
    int cardLength = kartu.length();

    if(n == cardLength){
        bool found = false;
        int i = 0;
        // cek in case kalo input nya ada yang sama
        while (!found and i < hasil.size()){
            if (kartu == hasil.at(i)){
                found = true;
            }
            i++;
        }
        //kalau berbeda, di pushback ke vector hasil
        if (!found){
            hasil.push_back(kartu);
            return;
        }
    }

    for (int i=n; kartu[i]; ++i){
        temp = kartu[n];
        kartu[n] = kartu[i];
        kartu[i] = temp;
        permutation(kartu, n+1, hasil);
        temp = kartu[n];
        kartu[n] = kartu[i];
        kartu[i] = temp;
    }
}

```

#### 4. void is24

Prosedur ini memiliki 6 parameter, yaitu string a, string b, string c, string d, vector<string> arrayhasil, int count. Prosedur akan memfilter semua kemungkinan permutasi tadi apakah jika dioperasikan secara matematika akan menghasilkan 24. Prosedur juga menghitung semua kemungkinan permutasi dengan semua kemungkinan kurung-kurung yang ada. Bila kemungkinan tadi berjumlah 24, maka kemungkinan tersebut akan di push back ke dalam vector<string> array hasil. Int count berguna untuk menghitung jumlah semua kemungkinan yang memenuhi syarat.

```

void is24(string a, string b, string c, string d, vector<string>& arrayhasil, int& count){
    //Menghitung dengan operasi matematika dan kemungkinan kurung
    double num1 = strTodouble(a), num2 = strTodouble(b), num3 = strTodouble(c), num4 = strTodouble(d);
    double ab, bc, cd;
    string hasil;
    string operators[4] = {"+", "-", "*", "/"};

    for (int i=0; i<4; i++){
        for (int j=0; j<4; j++){
            for (int k=0; k<4; k++){
                ab = operations(operators[i], num1, num2);
                bc = operations(operators[j], num2, num3);
                cd = operations(operators[k], num3, num4);

                //((ab)(cd)
                //jadi (a operators[i] b) operators[j] (c operators[k] d)
                if (operations(operators[i], ab, cd) == 24){
                    hasil = "(" + a + operators[i] + b + ")" + operators[j] + "(" + c + operators[k] + d + ")";
                    arrayhasil.push_back(hasil);
                    count++;
                }
                if (operations(operators[k], operations(operators[j], ab, num3), num4) == 24){
                    //((ab)c)d
                    //((a operators[i] b) operators[j] c) operators[k] d
                    hasil = "(" + a + operators[i] + b + ")" + operators[j] + c + ")" + operators[k] + d;
                    arrayhasil.push_back(hasil);
                    count++;
                }
                if (operations(operators[k], operations(operators[i], num1, bc), num4) == 24){
                    //((abc))d
                    //((a operators[i] (b operators[j] c)) operators[k] d
                    hasil = "(" + a + operators[i] + "(" + b + operators[j] + c + ")" + operators[k] + d;
                    arrayhasil.push_back(hasil);
                    count++;
                }
            }
        }
    }
}

```

```

        if (operations(operators[i], num1, operations(operators[k], bc, num4)) == 24){
            //a((bc)d)
            //a operators[i] ((b operators[j] c) operators[k] d)
            hasil = a + operators[i] + "(" + b + operators[j] + c + ")" + operators[k] + d + ")";
            arrayhasil.push_back(hasil);
            count++;
        }
        if (operations(operators[i], num1, operations(operators[j], num2, cd)) == 24){
            //a(b(cd))
            //a operators[i] (b operators[j] (c operators[k] d))
            hasil = a + operators[i] + "(" + b + operators[j] + "(" + c + operators[k] + d + "))";
            arrayhasil.push_back(hasil);
            count++;
        }
    }
}

```

## 5. void singleString

Prosedur ini memiliki 7 parameter, yaitu string k1, string k2, string k3, string k4, int count, vector<string> hasil, dan vector<string> hasil permutasi. Prosedur ini berguna untuk memisahkan string kemungkinan permutasi menjadi vector string yang elemen nya tersiri dari 1 digit.

```

void singleString(string& k1, string& k2, string& k3, string& k4, int& count, vector<string>& hasilpermutasi, vector<string>& hasil){
    //Memisahkan hasil permutasi menjadi perdigit
    for (int i=0; i<hasilpermutasi.size(); i++){
        string temp = hasilpermutasi.at(i);
        for (int j=0; temp[j]; ++j){
            //string permutasi dipisah satu satu misal "ABCD" jadi{"A", "B", "C", "D"}
            if (j == 0){
                if (temp[j] == 'Z'){
                    k1 = "10";
                } else {
                    k1 = temp[j];
                }
            } else if (j == 1){
                if (temp[j] == 'Z'){
                    k2 = "10";
                } else {
                    k2 = temp[j];
                }
            } else if (j == 2){
                if (temp[j] == 'Z'){
                    k3 = "10";
                } else {
                    k3 = temp[j];
                }
            } else if (j == 3) {
                if (temp[j] == 'Z'){
                    k4 = "10";
                } else {
                    k4 = temp[j];
                }
            }
        }
        is24(k1, k2, k3, k4, hasil, count);
    }
}

```

## B. Source Code Program

### 1. game.cpp (as main)

```
src > C* game.cpp > M main()
 1  #include <iostream>
 2  #include <iomanip>
 3  #include <vector>
 4  #include <time.h>
 5  #include "ui.cpp"
 6  #include "utility.cpp"
 7  #include "bruteforce.cpp"
 8  #include "output.cpp"
 9  using namespace std;
10
11 int main() {
12     string card1, card2, card3, card4;
13     string w1, w2, w3, w4;
14     string input;
15     bool exit = false;
16
17     asciiarts();
18     while (exit != true){
19         inputGenerate(card1, card2, card3, card4, exit);
20
21         if (exit == false){
22             int count = 0;
23             int ctr = 0;
24             vector<string> arrpermute, arrayofhasil;
25             tenToZ(card1, card2, card3, card4, count);
26
27             string fullcard = card1 + card2 + card3 + card4;
28             clock_t start, end;
29
30             //hitung waktu eksekusi program
31             start = clock();
32             permutation(fullcard, 0, arrpermute);
33             singleString(w1, w2, w3, w4, ctr, arrpermute, arrayofhasil);
34             end = clock();
35
36             double duration_sec = double(end-start)/CLOCKS_PER_SEC;
37             double duration_ms = duration_sec * 1000;
38
39             boldcyan();
40             if (ctr == 0){
41                 cout << "\n\n\n"                                     no solution found.";
42             } else if (ctr == 1){
43                 cout << "\n\n\n"                                     " << ctr << " solution found.";
44             } else {
45                 cout << "\n\n\n"                                     " << ctr << " solutions found.";
46             }
47             boldred();
48             cout << "\n"                                         execution time : " << duration_ms << " ms\n"; defclr();
49
50             if (ctr != 0){
51                 showSolve(arrayofhasil, count);
52                 saveFile(arrayofhasil, ctr);
53             }
54         }
55     }
56     return 0;
57 }
```

### 2. utility.cpp

```

rc > c utility.cpp > ui uicards(string &, string &, string &, string &)
1 #include <string>
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 void uicards(string& k1, string& k2, string& k3, string& k4){
7     // string symbol[4] = {"*", "#", "♥", "♦"}; only on mac
8     // char symbol[4] = {'*','&','♥','♦'}; only on windows
9
10    // int randsymbols1 = rand() % 4;
11    // int randsymbols2 = rand() % 4;
12    // int randsymbols3 = rand() % 4;
13    // int randsymbols4 = rand() % 4;
14    string symbol = "$";
15    string n1, n2, n3, n4;
16    int count = 0;
17
18    if (k1 != "10"){
19        n1 = k1 + " ";
20        count++;
21    } else {
22        n1 = "10";
23    }
24    if (k2 != "10"){
25        n2 = k2 + " ";
26        count++;
27    }else {
28        n2 = "10";
29    }
30    if (k3 != "10"){
31        n3 = k3 + " ";
32        count++;
33    }else {
34        n3 = "10";
35    }
36    if (k4 != "10"){
37        n4 = k4 + " ";
38        count++;
39    } else {
40        n4 = "10";
41    }
42
43
44    boldblue();
45    cout << "\n\n";
46    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2; boldblue();
47    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
48    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
49    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
50    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
51    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
52    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
53    cout << " _____"; boldyellow(); cout << symbol << n1; boldblue(); cout << " _____"; boldyellow(); cout << symbol << n2;
54
55    if (k1 == "10"){
56        cout << " _____"; boldyellow(); cout << symbol << k1;
57    } else {
58        cout << " _____"; boldyellow(); cout << symbol << k1;
59    }
60    if (k2 == "10"){
61        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k2;
62    } else {
63        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k2;
64    }
65    if (k3 == "10"){
66        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k3;
67    } else {
68        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k3;
69    }
70    if (k4 == "10"){
71        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k4;boldblue(); cout << "\n"; defclr();
72    } else{
73        boldblue(); cout << " _____"; boldyellow(); cout << symbol << k4;boldblue(); cout << "\n"; defclr();
74    }
75
76    cout << "\n\n";
77    boldpurple();
78    cout << " _____"; Your Cards : \n";
79    boldyellow();
80    string hasil;
81    if (count == 0){
82        hasil = " ";

```

```

src > utility.cpp > checkInput(string)
82     hasil = " ";
83 } else if (count == 1){
84     hasil = " ";
85 } else if (count == 2){
86     hasil = " ";
87 } else if (count == 3){
88     hasil = " ";
89 } else if (count == 4){
90     hasil = " ";
91 }
92 cout << "
93 "
94 }
95
96 bool checkInput(string Input){
97     bool found = false;
98     string input[14] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
99
100    for (int i=0; i<14; i++){
101        if (Input == input[i]){
102            found = true;
103        }
104    }
105
106    return found;
107 }
108
109 void wrongInput(){
110     boldred(); cout << "
111     cout << "\n                                         Wrong Input!";
112                                         Re-Enter a Number!";
113 }
114
115 void enterNumber(){
116     boldpurple(); cout << "\n                                         Enter a number : ";
117 }
118
119 void inputGenerate (string& k1, string& k2, string& k3, string& k4, bool& exit){
120     void inputGenerate (string& k1, string& k2, string& k3, string& k4, bool& exit){
121         string input;
122         boldcyan();
123         cout << "\n\n                                         ****";
124         cout << "*****\n                                         ";
125         cout << "*****\n                                         ";
126         cout << "*****\n                                         ";
127         boldcyan();
128         cout << "*****\n                                         ";
129         cout << "*****\n                                         ";
130         boldcyan();
131         cout << "*****\n                                         ";
132         cout << "*****\n                                         ";
133         boldcyan();
134         cout << "*****\n                                         ";
135         cout << "*****\n                                         ";
136         enterNumber();
137         cin >> input; defclr();
138
139     while ((input != "1") && (input != "2") && (input != "3")){
140         wrongInput();
141         enterNumber();
142         cin >> input; defclr();
143     }
144
145     if (input == "1"){
146         //input random/self-generate
147         boldcyan();
148         cout << "\n\n                                         ****";
149         cout << "*****\n                                         ";
150         cout << "*****\n                                         ";
151         srand(time(NULL));
152         string arrofcards[14] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};

```

```

src > C:\utility.cpp > ⊖ inputGenerate(string &, string &, string &, string &, bool &)
155     int randIndex1 = rand() % 13;
156     int randIndex2 = rand() % 13;
157     int randIndex3 = rand() % 13;
158     int randIndex4 = rand() % 13;
159     K1 = arrOfCards[randIndex1];
160     K2 = arrOfCards[randIndex2];
161     K3 = arrOfCards[randIndex3];
162     K4 = arrOfCards[randIndex4];
163
164     uicards(k1,k2,k3,k4);
165 } else if (input == "2"){
166     //User Input
167     boldpurple(); cout << "\n";
168     cin >> k1;
169     bool cek = checkInput(k1);
170     while (cek == false){
171         wrongInput();
172         boldpurple(); cout << "\n";
173         cin >> k1;
174         cek = checkInput(k1);
175     }
176     boldpurple(); cout << "\n";
177     cin >> k2;
178     cek = checkInput(k2);
179     while (cek == false){
180         wrongInput();
181         boldpurple(); cout << "\n";
182         cin >> k2;
183         cek = checkInput(k2);
184     }
185     boldpurple(); cout << "\n";
186     cin >> k3;
187     cek = checkInput(k3);
188     while (cek == false){
189         wrongInput();
190         boldpurple(); cout << "\n";
191         cin >> k3;
192         cek = checkInput(k3);
193     }
194     boldpurple(); cout << "\n";
195     cin >> k4;
196     cek = checkInput(k4);
197     while (cek == false){
198         wrongInput();
199         boldpurple(); cout << "\n";
200         _cin >> _k4;
201         cek = checkInput(k4);
202     }
203
204     uicards(k1, k2, k3, k4);
205 } else if (input == "3"){
206     boldpurple(); cout << "\n";
207     exit = true;
208 }
209 }
210
211 double strToDouble (string X){
212     string input[14] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
213     vector<int> inputsystem;
214
215     for(int i=0; i<13; i++){
216         inputsystem.push_back(i+1);
217     }
218
219     bool found = false;
220     int it = 0;
221     double val;
222
223     while (!found){
224         if (X == input[it]){
225             found = true;
226             val = inputsystem.at(it);
227         }
228         it++;
229     }
230
231     return val;
232 }

```

### 3. ui.cpp

```

src > ui.cpp > defclr()
1 #include <iostream>
2 #include <stdio.h>
3 using namespace std;
4
5 void defclr(){
6     cout << "\033[0m";
7 }
8
9 void boldblue(){
10    cout << "\033[1;34m";
11 }
12
13 void boldgreen(){
14    cout << "\033[1;32m";
15 }
16
17 void boldcyan(){
18    cout << "\033[1;36m";
19 }
20
21 void boldred(){
22    cout << "\033[1;31m";
23 }
24
25 void boldpurple(){
26    cout << "\033[1;35m";
27 }
28
29 void boldyellow(){
30    cout << "\033[1;33m";
31 }
32
33 void asciarts() {
34     boldcyan();
35     cout << "\n*****\n";
36     cout << "*****\n";
37     cout << "*****\n";
38     boldpurple(); cout << "*****\n";
39     boldcyan(); cout << "*****\n";
40     boldpurple(); cout << "*****\n";
41     boldcyan(); cout << "*****\n";
42     boldpurple(); cout << "*****\n";
43     boldcyan(); cout << "*****\n";
44     boldpurple(); cout << "*****\n";
45     boldcyan(); cout << "*****\n";
46     boldpurple(); cout << "*****\n";
47     boldcyan(); cout << "*****\n";
48     boldpurple(); cout << "*****\n";
49     boldcyan(); cout << "*****\n";
50     boldcyan(); cout << "*****\n";
51     boldyellow(); cout << "*****\n";
52     boldcyan(); cout << "*****\n";
53     boldyellow(); cout << "*****\n";
54     boldcyan(); cout << "*****\n";
55     boldyellow(); cout << "*****\n";
56     boldcyan(); cout << "*****\n";
57     boldyellow(); cout << "*****\n";
58     boldcyan(); cout << "*****\n";
59     boldyellow(); cout << "*****\n";
60     boldcyan(); cout << "*****\n";
61     boldyellow(); cout << "*****\n";
62     boldcyan(); cout << "*****\n";
63     cout << "*****\n";
64     cout << "*****\n";
65     cout << "*****\n";

```

#### 4. bruteForce.cpp

```

void tenToZ(string& k1, string& k2, string& k3, string& k4, int& count){
    // Mengubah string 10 yang berupa dua digit menjadi string 1 huruf untuk di permutasi
    if (k1 == "10"){
        k1 = "Z";
        count++;
    }
    if (k2 == "10"){
        k2 = "Z";
        count++;
    }
    if (k3 == "10"){
        k3 = "Z";
        count++;
    }
    if (k4 == "10"){
        k4 = "Z";
        count++;
    }
}

```

```

void permutation(string& kartu, int n, vector<string>& hasil){
    double temp;
    int cardLength = kartu.length();

    if(n == cardLength){
        bool found = false;
        int i = 0;
        // cek in case kalo input nya ada yang sama
        while (!found and i < hasil.size()){
            if (kartu == hasil.at(i)){
                found = true;
            }
            i++;
        }
        //kalau berbeda, di pushback ke vector hasil
        if (!found){
            hasil.push_back(kartu);
            return;
        }
    }

    for (int i=n; kartu[i]; ++i){
        temp = kartu[n];
        kartu[n] = kartu[i];
        kartu[i] = temp;
        permutation(kartu, n+1, hasil);
        temp = kartu[n];
        kartu[n] = kartu[i];
        kartu[i] = temp;
    }
}

```

```

double operations (string op, double num1, double num2){
    //operasi matematika
    double hasil;

    if (op == "+"){
        hasil = num1 + num2;
    } else if (op == "-"){
        hasil = num1 - num2;
    } else if (op == "*"){
        hasil = num1 * num2;
    } else if (op == "/"){
        hasil = num1 / num2;
    }

    return hasil;
}

```

```

void is24(string a, string b, string c, string d, vector<string>& arrayhasil, int& count){
    //Menghitung dengan operasi matematika dan kemungkinan kurung
    double num1 = strToDouble(a), num2 = strToDouble(b), num3 = strToDouble(c), num4 = strToDouble(d);
    double ab, bc, cd;
    string hasil;
    string operators[4] = {"+", "-", "*", "/"};

    for (int i=0; i<4; i++){
        for (int j=0; j<4; j++){
            for (int k=0; k<4; k++){
                ab = operations(operators[i], num1, num2);
                bc = operations(operators[j], num2, num3);
                cd = operations(operators[k], num3, num4);

                //((ab)(cd))
                //jadi (a operators[i] b) operators[j] (c operators[k] d)
                if (operations(operators[j], ab, cd) == 24){
                    hasil = "(" + a + operators[i] + b + ")" + operators[j] + "(" + c + operators[k] + d + ")";
                    arrayhasil.push_back(hasil);
                    count++;
                }
                if (operations(operators[k], operations(operators[j], ab, num3), num4) == 24){
                    //((ab)c)d
                    //((a operators[i] b) operators[j] c) operators[k] d
                    hasil = "(" + a + operators[i] + b + ")" + operators[j] + "(" + c + ")" + operators[k] + d;
                    arrayhasil.push_back(hasil);
                    count++;
                }
                if (operations(operators[k], operations(operators[i], num1, bc), num4) == 24){
                    //((a(bc))d
                    //((a operators[i] (b operators[j] c)) operators[k] d
                    hasil = "(" + a + operators[i] + "(" + b + operators[j] + c + ")" + operators[k] + d;
                    arrayhasil.push_back(hasil);
                    count++;
                }
            }
        }
    }
}

```

```
        if (operations(operators[i], num1, operations(operators[k], bc, num4)) == 24){
            //a((bc)d)
            //a operators[i] ((b operators[j] c) operators[k] d)
            hasil = a + operators[i] + "(" + b + operators[j] + c + ")" + operators[k] + d + ")";
            arrayhasil.push_back(hasil);
            count++;
        }
        if (operations(operators[i], num1, operations(operators[j], num2, cd)) == 24){
            //a(b(cd))
            //a operators[i] (b operators[j] (c operators[k] d))
            hasil = a + operators[i] + "(" + b + operators[j] + "(" + c + operators[k] + d + "))";
            arrayhasil.push_back(hasil);
            count++;
        }
    }
}
}
```

```
void singleString(string& k1, string& k2, string& k3, string& k4, int& count, vector<string>& hasilpermutasi, vector<string>& hasil){  
    //Memisahkan hasil permutasi menjadi perdigit  
    for (int i=0; i<hasilpermutasi.size(); i++){  
        string temp = hasilpermutasi.at(i);  
        for (int j=0; temp[j]; ++j){  
            //string permutasi dipisah satu satunya misal "ABCD" jadi{"A", "B", "C", "D"}  
            if (j == 0){  
                if (temp[j] == 'Z'){  
                    k1 = "10";  
                } else {  
                    k1 = temp[j];  
                }  
            } else if (j == 1){  
                if (temp[j] == 'Z'){  
                    k2 = "10";  
                } else {  
                    k2 = temp[j];  
                }  
            } else if (j == 2){  
                if (temp[j] == 'Z'){  
                    k3 = "10";  
                } else {  
                    k3 = temp[j];  
                }  
            } else if (j == 3){  
                if (temp[j] == 'Z'){  
                    k4 = "10";  
                } else {  
                    k4 = temp[j];  
                }  
            }  
        }  
        is24(k1, k2, k3, k4, hasil, count);  
    }  
}
```

## 5. output.cpp

```

1 #include <iostream>
2 #include <string>
3 #include <iostream>
4 #include <vector>
5
6 void showSolve(vector<string>& hasil, int count){
7     string solve;
8     string space1 = " ", space2 = " ";
9     string awal = "*****      , akhir =      ";
10    boldcyan();
11    cout << "\n\n";
12    cout << "*****"; boldpurple(); cout << "SHOW THE SOLUTIONS? ";
13    boldcyan();
14    cout << "****\n";
15    cout << " ";
16    cout << "*****"; boldyellow(); cout << " 1. YES ";
17    boldcyan();
18    cout << "****\n";
19    cout << "*****"; boldyellow(); cout << " 2. NO ";
20    boldcyan();
21    cout << "****\n";
22    cout << "*****"; boldyellow(); cout << "*****\n";
23    enterNumber();
24    cin >> solve; defclr();
25
26    while ((solve != "1") & (solve != "2")){
27        wrongInput();
28        enterNumber();
29        cin >> solve; defclr();
30    }
31
32    if (solve == "1"){
33        boldcyan();
34        cout << "\n\n";
35        cout << "*****"; boldpurple(); cout << "THE SOLUTIONS ";
36        boldcyan();
37        cout << "****\n";
38        cout << "*****"; boldyellow(); cout << "*****\n";
39        for (int i=0; i<hasil.size(); i++){
40            if (count == 4){
41                cout << awal; boldyellow(); cout << hasil.at(i) << akhir;
42            } else if (count == 3){
43                cout << awal << space1; boldyellow(); cout << hasil.at(i) << akhir;
44            } else if (count == 2){
45                cout << awal << space1 << boldyellow(); cout << hasil.at(i) << akhir << space1;
46            } else if (count == 1){
47                cout << awal << space1 << boldyellow(); cout << hasil.at(i) << akhir;
48            }
49        }
50    }
51}
```

```

46         cout << awal << space2; boldyellow(); cout<< hasil.at(i) << akhir << space1;
47     } else if (count == 0){
48         cout << awal << space2; boldyellow(); cout<< hasil.at(i) << akhir << space2;
49     }
50     boldcyan();
51     cout << "****\n"                                     ****                         ****\n";
52 }
53 cout << "*****\n";                                *****\n";
54 }
55 }
56
57 void saveFile(vector<string>& hasil, int ctr){
58     string savefile;
59     boldcyan();
60     cout << "\n\n"                                     *****\n";
61     cout << "                                         ****"; boldpurple(); cout << " SAVE TO FILE?      ";
62     boldcyan();
63     cout << "****\n"                                     *****\n";
64     cout << "                                         ****\n";
65     cout << "                                         ****\n";
66     boldcyan();
67     cout << "****\n"                                     ****                         ****\n";
68     cout << "                                         ****"; boldyellow(); cout << "    1. YES      ";
69     boldcyan();
70     cout << "****\n"                                     ****                         ****\n";
71     cout << "                                         ****\n";
72     enterNumber();
73     cin >> savefile; defclr();
74     while ((savefile != "1") and (savefile != "2")){
75         wrongInput();
76         enterNumber();
77         cin >> savefile; defclr();
78     }
79
80     if (savefile == "1"){
81         string name;
82         boldpurple(); cout << "                               Input file name : ";
83         cin >> name; defclr();
84
85         string filename = name + ".txt";
86         ofstream output("test/" + filename);
87
88         if (output.is_open()){
89             if (ctr == 0){
90                 output << "no solution found.\n";
91             } else if (ctr == 1){
92                 output << ctr << " solution found\n";
93             } else {
94                 output << ctr << " solutions found\n";
95             }
96             //simpan vektor ke dalam array
97             for (int i=0; i<hasil.size(); i++){
98                 output << hasil.at(i) << "\n";
99             }
100            output.close();
101            boldblue();
102            cout << "                               saved in " + filename + "\n";
103        } else {
104            boldred();
105            cout << "                               There was a problem while opening the file\n";
106        }
107        defclr();
108    }
109 }

```

## C. Tampilan Program dan Contoh Input-Output

## 1. Tampilan Menu

## 2. Tampilan Menu Random Card

### 3. Tampilan Input User

```

***** MENU *****
**** 1. RANDOM CARDS ****
**** 2. INPUT USER ****
**** 3. EXIT ****
*****
Enter a number : 2
Enter 1st card number : 3
Enter 2nd card number : 4
Enter 3rd card number : 5
Enter 4th card number : 6

 $\begin{array}{cccc} \heartsuit 3 & \clubsuit 4 & \spadesuit 5 & \diamondsuit 6 \\ & \clubsuit 4 & \spadesuit 5 & \diamondsuit 6 \\ & & \spadesuit 5 & \diamondsuit 6 \\ & & & \diamondsuit 6 \end{array}$ 

Your Cards :
3 4 5 6

16 solutions found.
execution time : 2.816 ms

```

#### 4. Tampilan Solusi

```

***** SHOW THE SOLUTIONS? ****
**** 1. YES ****
**** 2. NO ****
*****
Enter a number : 1

***** THE SOLUTIONS ****
*****
 $(Q*(7+K))/10$ 
 $Q*((7+K)/10)$ 
 $(Q*(K+7))/10$ 
 $Q*((K+7)/10)$ 
 $Q/(10/(K+7))$ 
 $(Q/10)*(7+K)$ 
 $Q/(10*(7+K))$ 
 $(7+K)*(Q/10)$ 
 $((7+K)*Q)/10$ 
 $((7+K)/10)*Q$ 
 $(7+K)/(10/Q)$ 
 $(K+7)*(0/10)$ 
 $((K+7)*0)/10$ 
 $((K+7)/10)*0$ 
 $(K+7)/(10/Q)$ 
*****
```

#### 5. Tampilan Save File

Terlihat bahwa file tersimpan di folder test dengan nama test1.txt.

```

*** 0/(10/(7+K)) ***
*** (7+K)*(Q/10) ***
*** ((7+K)*Q)/10 ***
*** ((7+K)/10)*Q ***
*** (7+K)/(10/Q) ***
*** (K+7)*(Q/10) ***
*** ((K+7)*Q)/10 ***
*** ((K+7)/10)*Q ***
*** (K+7)/(10/Q) ***

*** SAVE TO FILE? ***
1. YES
2. NO

Enter a number : 1
Input file name : test1
saved in test1.txt

*** MENU ***
1. RANDOM CARDS
2. INPUT USER
3. EXIT

```

## 6. Tampilan solusi yang tersimpan dalam file

```

test > test1.txt
1 16 solutions found
2 (Q*(7+K))/10
3 Q*((7+K)/10)
4 (Q*(K+7))/10
5 Q*((K+7)/10)
6 (Q/10)*(K+7)
7 Q/(10/(K+7))
8 (Q/10)*(7+K)
9 Q/(10/(7+K))
10 (7+K)*(Q/10)
11 ((7+K)*Q)/10
12 ((7+K)/10)*Q
13 (7+K)/(10/Q)
14 (K+7)*(Q/10)
15 ((K+7)*Q)/10
16 ((K+7)/10)*Q
17 (K+7)/(10/Q)
18

```

## 7. Tampilan jika input yang dimasukkan salah

```
*****
*****          MENU          *****
*****
*****          1. RANDOM CARDS *****
*****
*****          2. INPUT USER   *****
*****
*****          3. EXIT        *****
*****
*****          *****
Enter a number : 9
Wrong Input!
Re-Enter a Number!
Enter a number : d
Wrong Input!
Re-Enter a Number!
Enter a number : 
```

#### 8. Contoh Input User Tidak Ada Solusi (4 6 K 7)

```
*****
*****          MENU          *****
*****
*****          1. RANDOM CARDS *****
*****
*****          2. INPUT USER   *****
*****
*****          3. EXIT        *****
*****
*****          *****
Enter a number : 2
Enter 1st card number : 4
Enter 2nd card number : 6
Enter 3rd card number : K
Enter 4th card number : 7



$4



$6



$K



$7


Your Cards :
4 6 K 7

no solution found.
execution time : 2.952 ms
```

#### 9. Contoh Input User Dengan 1 Solusi (5 5 5 5)

```

Enter 1st card number : 5
Enter 2nd card number : 5
Enter 3rd card number : 5
Enter 4th card number : 5

$5 $5 $5 $5
$5 $5 $5 $5
$5 $5 $5 $5
$5 $5 $5 $5

Your Cards :
5 5 5 5

1 solution found.
execution time : 0.273 ms

*****
**** SHOW THE SOLUTIONS ****
*****
**** 1. YES ****
**** 2. NO ****
*****
Enter a number : 1

*****
**** THE SOLUTIONS ****
*****
**** (5*5)-(5/5) ****
*****

```

```

*****
**** THE SOLUTIONS ****
*****
**** (5*5)-(5/5) ****
*****

```

```

*****
**** SAVE TO FILE? ****
*****
**** 1. YES ****
**** 2. NO ****
*****

```

```

Enter a number : 1
Input file name : test2
saved in test2.txt

```

$\equiv$  test2.txt M X

```

test >  $\equiv$  test2.txt
1 1 solution found
2 (5*5)-(5/5)
3

```

## 10. Contoh Input User Dengan Banyak Solusi (10 8 9 6)

```
Enter 1st card number : 10
Enter 2nd card number : 8
Enter 3rd card number : 9
Enter 4th card number : 6

$10
$10
$10

$8
$8
$8

$9
$9
$9

$6
$6
$6

Your Cards :
10 8 9 6

6 solutions found.
execution time : 3.561 ms
```

```
Code File Edit Selection View Go Run Terminal Window Help
test3.txt -- Tucll1_13521103
OPEN EDITORS test > test3.txt
TUCLL... M
bin M
game M
game.exe M
doc M
src M
bruteforce.cpp M
game.cpp M
output.cpp M
ui.cpp M
utility.cpp M
test M
test1.txt M
test2.txt M
test3.txt M
README.md

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
game + ...
```

```
*****
**** SAVE TO FILE? ****
**** 1. YES ****
**** 2. NO ****
****

Enter a number : 1
Input file name : test3
saved in test3.txt
```

## 11. Contoh Random Input Dengan Banyak Solusi (K 3 4 Q)

```
*****
**** RANDOM CARDS ****
*****
```

```
$K
$K
$K

$3
$3
$3

$4
$4
$4

$Q
$Q
$Q

Your Cards :
K 3 4 Q

90 solutions found.
execution time : 3.512 ms
```

```

test > test4.txt
1 90 solutions found
2 ((K+3)-4)+Q
3 ((K+(3-4))+Q
4 K+(3-4)+Q
5 (K+3)-(4-Q)
6 K+(3-(4-Q))
7 (K+3)+(Q-4)
8 ((K+3)+Q)-4
9 ((K+(3+Q))-4
10 K+(3+Q)-4
11 K+(3+(Q-4))
12 (K-4)+(3+Q)
13 ((K-4)+3)+Q
14 (K-(4-3))+Q
15 K-(4-(3+Q))

***** SAVE TO FILE? *****
**** 1. YES *****
**** 2. NO *****
***** Enter a number : 1
Input file name : test4
saved in test4.txt

```

## 12. Contoh Random Input Tidak ada solusi (A 5 A K)

```

*****
**** RANDOM CARDS ****
*****
$A $5 $A $K
$A $5 $A $K

Your Cards :
A 5 A K

no solution found.
execution time : 2.035 ms

```

## 13. Contoh Random Input Dengan 1 Solusi (3 3 5 5)

```
***** RANDOM CARDS ****
$3 $3 $5 $5
$3 $3 $5 $5
$3 $3 $5 $5
$3 $3 $5 $5

Your Cards :
3 3 5 5

1 solution found.
execution time : 1.065 ms

***** SHOW THE SOLUTIONS? ****
1. YES
2. NO

Enter a number : 1

Code File Edit Selection View Go Run Terminal Window Help
test5.txt — Tucil1_13521103

EXPLORER      ...
OPEN EDITORS   ...
test > test5.txt M
  1 1 solution found
  2 (5*5)-(3/3)
  3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

*****
*** THE SOLUTIONS ***
*** (5*5)-(3/3) ***
****

*****
*** SAVE TO FILE? ***
*** YES ***
*** NO ***
****

Enter a number : 1
Input file name : test5
saved in test5.txt
```

#### *D. Link To Repository*

[https://github.com/auliamey/Tucill\\_13521103](https://github.com/auliamey/Tucill_13521103)

## E. Tabel

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 program memenuhi (berhasil mencapai 24)	✓	
5. Program dapat menyimpan solusi dalam file teks	✓	