

**Tugas Kecil I IF2211 Strategi Algoritma**

**Semester II**

**Tahun Ajaran 2022/2023**

# **Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force**



**Disusun oleh:**

**M. Malik I. Baharsyah**

**13521029**

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

**2023**

## 1. Penjelasan Algoritma Brute Force

Algoritma *brute force* merupakan algoritma untuk menemukan solusi dengan mencoba segala kemungkinan yang ada. Program ini dibuat untuk mencari seluruh solusi yang mungkin dari 4 input kartu berupa A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K dengan memanipulasi operasi (penjumlahan, pengurangan, perkalian, pembagian), letak angka, dan prioritas perhitungan (menggunakan kurung) sehingga didapatkan nilai angka dari operasi tersebut adalah 24. Oleh sebab itulah permainan ini disebut 24 game.

## 2. Source Code (C++)

```
3. #include <iostream>
4. #include <string>
5. #include <time.h>
6. #include <fstream>
7. using namespace std;
8.
9. int olahOperasi(int a, int b, int i) {
10.     int ans = 9999;
11.     switch(i) {
12.         case 0:
13.             ans = a + b;
14.             break;
15.         case 1:
16.             ans = a - b;
17.             break;
18.         case 2:
19.             ans = a * b;
20.             break;
21.         case 3:
22.             if (b!=0) {
23.                 ans = a/b;
24.             }
25.             break;
26.     }
27.     return ans;
28. }
29.
30. char charOperator(int i) {
31.     switch(i) {
32.         case 0:
33.             return '+';
34.             break;
35.         case 1:
36.             return '-';
37.             break;
38.         case 2:
39.             return '*';
40.             break;
41.         case 3:
```

```

42.         return '/';
43.         break;
44.     default:
45.         return 'i';
46.     }
47.}
48.
49.// bentuk solusi
50.
51.// bentuk ((a op b) op c) op d
52.void sol0(int listAngka[4], int* n, string* txt) {
53.    int A, B, C;
54.    for (int i=0; i<4; i++) {
55.        A = olahOperasi(listAngka[0], listAngka[1], i);
56.        for (int j=0; j<4; j++) {
57.            B = olahOperasi(A, listAngka[2], j);
58.            for (int k=0; k<4; k++) {
59.                C = olahOperasi(B, listAngka[3], k);
60.                if (C == 24) {
61.                    (*n)++;
62.                    (*txt) = (*txt) + "(" + to_string(listAngka[0]) +
charOperator(i) + to_string(listAngka[1]) + ")"
63.                        + charOperator(j) + to_string(listAngka[2]) + ")" +
charOperator(k) + to_string(listAngka[3]) + "\n";
64.                }
65.            }
66.        }
67.    }
68.}
69.
70.// bentuk (a op (b op c)) op d
71.void sol1(int listAngka[4], int* n, string* txt) {
72.    int A, B, C;
73.    for (int i=0; i<4; i++) {
74.        A = olahOperasi(listAngka[1], listAngka[2], i);
75.        for (int j=0; j<4; j++) {
76.            B = olahOperasi(listAngka[0], A, j);
77.            for (int k=0; k<4; k++) {
78.                C = olahOperasi(B, listAngka[3], k);
79.                if (C == 24) {
80.                    (*n)++;
81.                    (*txt) = (*txt) + "(" + to_string(listAngka[0]) +
charOperator(j) + "(" + to_string(listAngka[1])
82.                        + charOperator(i) + to_string(listAngka[2]) + ")")
+ charOperator(k) + to_string(listAngka[3]) + "\n";
83.                }
84.            }
85.        }

```

```

86.     }
87. }
88.
89. // bentuk (a op b) op (c op d)
90. void sol2(int listAngka[4], int* n, string* txt) {
91.     int A, B, C;
92.     for (int i=0; i<4; i++) {
93.         A = olahOperasi(listAngka[0], listAngka[1], i);
94.         for (int j=0; j<4; j++) {
95.             B = olahOperasi(listAngka[2], listAngka[3], j);
96.             for (int k=0; k<4; k++) {
97.                 C = olahOperasi(A, B, k);
98.                 if (C == 24) {
99.                     (*n)++;
100.                    (*txt) = (*txt) + "(" +
to_string(listAngka[0]) + charOperator(i) + to_string(listAngka[1]) +
                    ")"
101.                    + charOperator(k) + "("
+ to_string(listAngka[2]) + charOperator(j) + to_string(listAngka[3])
+ ")" + "\n";
102.                }
103.            }
104.        }
105.    }
106. }
107.
108. // bentuk a op ((b op c) op d)
109. void sol3(int listAngka[4], int* n, string* txt) {
110.     int A, B, C;
111.     for (int i=0; i<4; i++) {
112.         A = olahOperasi(listAngka[1], listAngka[2], i);
113.         for (int j=0; j<4; j++) {
114.             B = olahOperasi(A, listAngka[3], j);
115.             for (int k=0; k<4; k++) {
116.                 C = olahOperasi(listAngka[0], B, k);
117.                 if (C == 24) {
118.                     (*n)++;
119.                     (*txt) = (*txt) + "(" +
to_string(listAngka[0]) + charOperator(k) + "(" +
to_string(listAngka[1])
120.                    + charOperator(i)
+ to_string(listAngka[2]) + ")" + charOperator(j) +
to_string(listAngka[3]) + ")" + "\n";
121.                }
122.            }
123.        }
124.    }
125. }

```

```

126.
127.     void solusi24(int listAngka[4], int* n, string* txt) {
128.         sol0(listAngka, n, txt);
129.         sol1(listAngka, n, txt);
130.         sol2(listAngka, n, txt);
131.         sol3(listAngka, n, txt);
132.     }
133.
134.     void replaceList(int listAngka[4], int a, int b, int c, int d) {
135.         listAngka[0] = a;
136.         listAngka[1] = b;
137.         listAngka[2] = c;
138.         listAngka[3] = d;
139.     }
140.
141.     void userInputList(int listAngka[4]) {
142.         string input;
143.         int iNum;
144.         bool flag = true;
145.         while (flag) {
146.             flag = false;
147.             for(int i=0; i<4; i++) {
148.                 cin >> input;
149.                 if (input == "A") {
150.                     listAngka[i] = 1;
151.                 } else if (input == "J") {
152.                     listAngka[i] = 11;
153.                 } else if (input == "Q") {
154.                     listAngka[i] = 12;
155.                 } else if (input == "K") {
156.                     listAngka[i] = 13;
157.                 } else if (stoi(input) >= 2 && stoi(input) <= 10) {
158.                     listAngka[i] = stoi(input);
159.                 } else {
160.                     printf("Masukan tidak sesuai, silakan masukkan
kembali.\n");
161.                     flag = true;
162.                     break;
163.                 }
164.             }
165.         }
166.     }
167. }
168.
169.     void printList(int listAngka[4]) {
170.         for (int i=0; i<4; i++) {
171.             printf("%d ", listAngka[i]);
172.         } printf("\n");

```

```

173.     }
174.
175. void randomizer(int listAngka[4]) {
176.     srand(time(0));
177.     for (int i = 0; i < 4; i++) {
178.         int a = rand() % 13 + 1;
179.         listAngka[i] = a;
180.     }
181.     printf("Berhasil mengacak dengan hasil:\n");
182.     printList(listAngka);
183. }
184.
185. void splash() { cout
186.     <<" .----- .----- .----- \n"
187.     .----- .----- .----- \n"
188.     <<"| .----- || .----- | | .----- ||
189.     -----|| .-----|| .-----| \n"
190.     <<"| |      | || |   _   | || |   _   | || |
191.     |    | || |   _   | || |   _   | || |   _   | || | \n"
192.     <<"| | / ___`_   | || |   _   | || |   _   | || |   _   | || | \n"
193.     <<"| | |_/_)_ | | || |   _   | || |   _   | || |   _   | || | \n"
194.     <<"| | .'_._.' | || |   _   | || |   _   | || |   _   | || | \n"
195.     <<"| | / ___\_\_| || |   _   | || |   _   | || |   _   | || | \n"
196.     <<"| | / /___ | || |   _   | || |   _   | || |   _   | || | \n"
197.     <<"| | _____ | || |   _   | || |   _   | || |   _   | || | \n"
198.     <<"| |       | || |   _   | || |   _   | || |   _   | || | \n"
199.     <<"| '-----' || '-----' | | '-----' | |
200.     '-----' || '-----' || '-----' | \n"
201.     <<" '-----' '-----' '-----' \n\n";
202. }
203.
204. void menuUtama() {
205.     printf("Pilih input yang Anda inginkan:\n");
206.     printf("1. Input sendiri\n");
207.     printf("2. Diinputin (random)\n");
208. }
209.
210. void menuInputSendiri() {
211.     printf("Silakan masukkan 4 kartu Anda\n");
212.     printf("Input yang available: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A\n");
213. }
```

```
209.
210.     void menuOutput() {
211.         printf("Apakah Anda ingin menyimpan solusi dalam file?\n");
212.         printf("1. Ya\n");
213.         printf("2. Tidak\n");
214.     }
215.
216.     void exit() {
217.         printf("Terima kasih telah menggunakan program\n");
218.         printf("Masukkan input sembarang untuk keluar.\n");
219.         string ex;
220.         cin >> ex;
221.     }
222.
223.     int main() {
224.         splash();
225.         menuUtama();
226.         int inp;
227.         bool flag = true;
228.         int listAngka[4];
229.         int n = 0;
230.         string output;
231.
232.         while(flag) {
233.             flag = false;
234.             cin >> inp;
235.             if (inp == 1) {
236.                 menuInputSendiri();
237.                 userInputList(listAngka);
238.             } else if (inp == 2) {
239.                 randomizer(listAngka);
240.             } else {
241.                 flag = true;
242.             }
243.         }
244.
245.         clock_t start = clock();
246.
247.         int a = listAngka[0];
248.         int b = listAngka[1];
249.         int c = listAngka[2];
250.         int d = listAngka[3];
251.
252.         int list[4] = {a,b,c,d};
253.         replaceList(list, a,b,c,d);solusi24(list, &n, &output);
254.         replaceList(list, a,b,d,c);solusi24(list, &n, &output);
255.         replaceList(list, a,c,b,d);solusi24(list, &n, &output);
256.         replaceList(list, a,c,d,b);solusi24(list, &n, &output);
```

```

257.         replaceList(list, a,d,b,c);solusi24(list, &n, &output);
258.         replaceList(list, a,d,c,b);solusi24(list, &n, &output);
259.         replaceList(list, b,a,c,d);solusi24(list, &n, &output);
260.         replaceList(list, b,a,d,c);solusi24(list, &n, &output);
261.         replaceList(list, b,c,a,d);solusi24(list, &n, &output);
262.         replaceList(list, b,c,d,a);solusi24(list, &n, &output);
263.         replaceList(list, b,d,a,c);solusi24(list, &n, &output);
264.         replaceList(list, b,d,c,a);solusi24(list, &n, &output);
265.         replaceList(list, c,a,b,d);solusi24(list, &n, &output);
266.         replaceList(list, c,a,d,b);solusi24(list, &n, &output);
267.         replaceList(list, c,b,a,d);solusi24(list, &n, &output);
268.         replaceList(list, c,b,d,a);solusi24(list, &n, &output);
269.         replaceList(list, c,d,a,b);solusi24(list, &n, &output);
270.         replaceList(list, c,d,b,a);solusi24(list, &n, &output);
271.         replaceList(list, d,a,b,c);solusi24(list, &n, &output);
272.         replaceList(list, d,a,c,b);solusi24(list, &n, &output);
273.         replaceList(list, d,b,a,c);solusi24(list, &n, &output);
274.         replaceList(list, d,b,c,a);solusi24(list, &n, &output);
275.         replaceList(list, d,c,a,b);solusi24(list, &n, &output);
276.         replaceList(list, d,c,b,a);solusi24(list, &n, &output);
277.
278.         string nSol;
279.         if (n>0) {
280.             nSol = to_string(n) + " solutions found\n";
281.         } else {
282.             nSol = "Tidak ada solusi\n";
283.         }
284.
285.         cout << nSol;
286.         cout << output;
287.         cout.precision(10);
288.         cout << "Execution time " << float(clock() -
start)/CLOCKS_PER_SEC << "s" << endl;
289.         menuOutput();
290.
291.         flag = true;
292.         while (flag) {
293.             flag = false;
294.             cin >> inp;
295.             if (inp == 1) {
296.                 ofstream out("output.txt");
297.                 out << n << " solutions found\n";
298.                 out << output;
299.                 out.close();
300.                 printf("Berhasil menyimpan dengan nama file
output.txt\n");
301.                 exit();
302.             } else if (inp == 2) {

```



```

303.             exit();
304.         } else {
305.             flag = true;
306.         }
307.     }
308. }

```

### 3. Screenshot Program

```

Pilih input yang Anda inginkan:
1. Input sendiri
2. Diinputin (random)
1
Silakan masukkan 4 kartu Anda
Input yang available: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A
A 5 6 3
26 solutions found
(1+5)+(6*3)
((1+5)*3)+6
(1+5)+(3*6)
(1+(6*3))+5
(1+((6*3)+5)
(1+(3*6))+5
(1+((3*6)+5)
(5+1)+(6*3)
((5+1)*3)+6
(5+1)+(3*6)
(5+(6*3))+1
(5+((6*3)+1)
(5+(3*6))+1
(5+((3*6)+1)
(6+((1+5)*3)
(6+((5+1)*3)
((6*3)+1)+5
(6*3)+(1+5)
((6*3)+5)+1
(6*3)+(5+1)
(3*(1+5))+6
(3*(5+1))+6
((3*6)+1)+5
(3*6)+(1+5)

```



Pilih input yang Anda inginkan:

1. Input sendiri
2. Diinputin (random)

2

Berhasil mengacak dengan hasil:

8 1 8 5

54 solutions found

$(8*1)*(8-5)$   
 $(8/1)*(8-5)$   
 $(8*((1*8)-5))$   
 $((8*1)-5)*8$   
 $((8/1)-5)*8$   
 $(8-(1*5))*8$   
 $(8*((8*1)-5))$   
 $(8*((8/1)-5))$   
 $(8*(8-5))*1$   
 $(8*(8-5))/1$   
 $(8*((8-5)*1))$   
 $(8*((8-5)/1))$   
 $((8-5)*1)*8$   
 $((8-5)/1)*8$   
 $(8*(5-1))-8$   
 $(8-(5*1))*8$   
 $(8-(5/1))*8$   
 $(8-5)*(1*8)$   
 $((8-5)*8)*1$   
 $((8-5)*8)/1$   
 $(8-5)*(8*1)$   
 $(8-5)*(8/1)$   
 $(1*8)*(8-5)$   
 $((1*8)-5)*8$   
 $(1*(8-5))*8$



Pilih input yang Anda inginkan:

1. Input sendiri
2. Diinputin (random)

1

Silakan masukkan 4 kartu Anda

Input yang available: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A  
J Q K 10

140 solutions found

$(11+12)+(13/10)$   
 $(11*(12/10))+13$   
 $(11/(12/10))+13$   
 $(11+((12/10)*13))$   
 $(11+13)*(12/10)$   
 $(11+13)/(12/10)$   
 $((11+13)/10)*12$   
 $(11+(13/10))+12$   
 $(11+13)+(10/12)$   
 $(11+13)-(10/12)$   
 $(11+((13/10)+12))$   
 $(11+(10/12))+13$   
 $(11-(10/12))+13$   
 $(11+((10/12)+13))$   
 $(11-((10/12)-13))$   
 $((12-11)+13)+10$   
 $((12/11)+13)+10$   
 $(12-(11-13))+10$   
 $(12+11)+(13/10)$   
 $(12-11)+(13+10)$   
 $(12/11)+(13+10)$   
 $(12*((11+13)/10))$   
 $(12-((11-13)-10))$   
 $((12-11)+10)+13$

```

  9  7  1 10
4 solutions found
(9-1)*(10-7)
(7-10)*(1-9)
(1-9)*(7-10)
(10-7)*(9-1)
Execution time 0s
Apakah Anda ingin menyimpan solusi dalam file?
1. Ya
2. Tidak
1
Berhasil menyimpan dengan nama file output.txt
Terima kasih telah menggunakan program
Masukkan input sembarang untuk keluar.
```

```

 10 13 10 13
40 solutions found
(10+(13/10))+13
(10+((13/10)+13))
(10+13)+(13/10)
(10+13)+(13/10)
(10+(13/10))+13
(10+((13/10)+13))
((13/10)+10)+13
(13/10)+(10+13)
((13/10)+13)+10
(13+10)+(13/10)
(13/10)+(13+10)
((13/10)+10)+13
(13/10)+(10+13)
((13/10)+13)+10
(13+10)+(13/10)
(13/10)+(13+10)
(13+(13/10))+10
(13+((13/10)+10))
(13+(13/10))+10
(13+((13/10)+10))
(10+(13/10))+13
(10+((13/10)+13))
(10+13)+(13/10)
(10+(13/10))+13
(10+((13/10)+13))
```

```

  24
  5
  9
  12
  11
  10

Pilih input yang Anda inginkan:
1. Input sendiri
2. Diinputin (random)
1
Silakan masukkan 4 kartu Anda
Input yang available: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A
A J 9 Q
28 solutions found
((1*11)-9)*12
(1*(11-9))*12
(1+(11/9))*12
(1*((11-9)*12))
(1*12)*(11-9)
((11*1)-9)*12
((11/1)-9)*12
(11-(1*9))*12
((11-9)*1)*12
((11-9)/1)*12
((11/9)+1)*12
(11-(9*1))*12
(11-(9/1))*12
(11-9)*(1*12)
((11-9)*12)*1
((11-9)*12)/1
(11-9)*(12*1)
(11-9)*(12/1)
(12*1)*(11-9)
(12/1)*(11-9)
(12*((11*11)-9))
(12*((11*1)-9))
(12*((11/1)-9))
(12*(11-9))*1

```

#### 4. Link Repository

[https://github.com/Darkfirm/Tucil1\\_13521029](https://github.com/Darkfirm/Tucil1_13521029)

#### 5. Ceklis Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil <i>running</i>	✓	
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	✓	