

Tugas Kecil IF2211 Strategi Algoritma

**Laporan Solusi Permainan 24 dengan
Algoritma *Brute Force***



Disusun oleh:

Vanessa Rebecca Wiyono

K01 / 13521151

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

2023

1. Algoritma *Brute Force*

Algoritma *brute force* merupakan suatu pendekatan yang sangat sering digunakan dalam memecahkan suatu masalah karena metodenya yang praktis dan dapat digunakan untuk memecahkan hampir seluruh permasalahan. Algoritma ini memecahkan masalah dengan secara langsung (*straightforward*) dengan cara yang jelas dan sering kali disertai dengan pengulangan karena meninjau seluruh kemungkinan yang dapat dihasilkan dan kemudian memilih kemungkinan mana yang sesuai dengan permintaan pengguna. Dalam beberapa kasus, sebagian perhitungan dapat dibuang karena bersifat sama persis dengan yang lainnya sehingga tidak diperlukan lagi..

Konsep algoritma *Brute Force* yang diterapkan pada program pencari solusi permainan 24 antara lain adalah:

1. Program menerima input berupa 4 kartu yang didapat melalui masukkan user secara manual maupun dari hasil generator kartu random
2. Program melakukan operasi $\{+, -, *, /\}$ sesuai dengan posisi angka awal secara berurutan, sehingga array yang awalnya berisi 4 angka menjadi berisi 3 angka, kemudian 2 angka, dan terakhir 1 angka.
3. Ketika array hanya berisi 1 angka, maka dilakukan pengecekan apakah angka tersebut bernilai 24.
4. Jika angka terakhir bernilai 24, maka seluruh proses operasi yang telah dilakukan akan dicatat sebagai salah satu solusi.
5. Langkah-langkah diatas akan diulang melalui permutasi yang dilakukan sehingga program akan meninjau kemungkinan dari seluruh permutasi yang terjadi pada kartu dan pada operasi.
6. Program akan berhenti saat seluruh permutasi leksikografis telah dilakukan. (permutasi leksikografis dilakukan dengan menukar dua angka secara berturut-turut, sebagai contoh $(1, 2, \dots, n)$ akan berakhir pada $(n, n-1, \dots, 2, 1)$).

Berdasarkan kompleksitasnya, algoritma ini menghasilkan kompleksitas waktu yang tinggi karena menghasilkan notasi $O(n * n!)$. Namun, pada kasus ini nilai n terbatas pada $n = 4$, sebagaimana program menerima masukkan berupa 4 kartu.

2. Source Program

```
1  // ---- HEADERS ----
2  #include <iostream>
3  #include <string>
4  #include <cstring>
5  #include <vector>
6  #include <fstream>
7  #include <ctime>
8  #include <algorithm>
9  using namespace std;
10 using std::vector;
11
12 int bykSolusi = 0;
13 int arrInt[4]; //array isi angka actualnya
14
15 // ---- PROTOTYPES ----
16 void combinations(const string &s, vector<int> &arr, int n);
17 void StartInput(int n, int k, string userInput);
18 int calculate(char a, int b, int c);
19 void lexicograph(char *inputStr, char *data, int terakhir, int idx);
20 void GameSolver(string arrStr[], int arrNum[]);
21 void printRec(char set[], string inputStr, int n, int k, int arrNum[]);
22 void printAll(char set[], int k, int n, int arrNum[]);
23 int inputRandom();
```

```
// ---- MAIN ----
int main() {
    string inputUser;
    string isSave, isManual;
    float timer;

    cout << " _____ " << endl;
    cout << "| _ _ _ _ _ |" << endl;
    cout << "| \ \ \ \ / / | | " << endl;
    cout << "| \ \ \ / \ / _ / _ | | _ _ _ _ _ |" << endl;
    cout << "| \ \ \ / \ / _ / _ / _ \ \ ' _ _ \ \ / _ |" << endl;
    cout << "| \ \ / \ / _ / | ( _ ( ) | | | | _ /" << endl;
    cout << "| \ \ \ \ \ _ | \ \ \ \ / | | | _ \ \ _" << endl;
    cout << "| _____ |" << endl;
    cout << "===== 24 GAME SOLVER =====" << endl;
    cout << "apakah anda ingin meng-input kartu secara manual [Y/N]?: ";
    cin >> isManual;

    if (isManual == "Y"){
        cout << "Masukkan kartu ^^ : ";
        cin >> inputUser;
        timer = clock();
        do{
            StartInput(4, 3, inputUser);
        } while (next_permutation(inputUser.begin(), inputUser.end()));
        timer = (clock() - timer) / CLOCKS_PER_SEC;
    }
    else if (isManual == "N"){
        timer = clock();
        inputRandom();
        timer = (clock() - timer) / CLOCKS_PER_SEC;
    }
    else {
        cout << "perintah tidak valid !" << endl;
    }
}
```

```

cout << "Apakah anda ingin menyimpan solusi [Y / N] ? : ";
cin >> isSave;

if (isSave == "Y"){
    ofstream myFile;
    myFile.open("solusi.txt", ofstream::app);
    myFile << "Jadi, terdapat " << bykSolusi << " solusi" << endl;
    cout << "solusi berhasil disimpan :)" << endl;
}
else if (isSave == "N"){
    char filename[] = "solusi.txt";
    int result = remove(filename);
    cout << result;
    cout << "baik, sampai jumpa !" << endl;
}
else {
    cout << "perintah salah !" << endl;
    char filename[] = "solusi.txt";
    int result = remove(filename);
    cout << result;
}

printf(" --- EXECUTION TIME: %.5f s ---\n", timer);

if (bykSolusi == 0){
    cout << "Tidak ada solusi" << endl;
}
return 0;
}

```

```

89 // ---- FUNCTIONS ----
90 void combinations(const string &s, vector<int> &arr, int n) {
91     if (n == s.size()) {
92         for (int i = 0; i != n; i++) {
93             cout << s[arr[i]] << endl;
94         }
95         return;
96     }
97     for (int i = 0; i < s.size(); i++) {
98         arr[n] = i;
99         combinations(s, arr, n + 1);
100     }
101 }
102
103 void StartInput(int n, int k, string userInput) {
104     char arrayChar[5];
105     int arrayNum[4];
106     bool valid = true;
107     string cc;
108     strcpy(arrayChar, userInput.c_str());
109
110     for (int i = 0; i < n; i++) {
111         cc = arrayChar[i];
112
113         //klo inputnya huruf
114         if (cc == "J"){
115             arrInt[i] = 11;
116         }
117         else if (cc == "Q"){
118             arrInt[i] = 12;
119         }
120         else if (cc == "K"){
121             arrInt[i] = 13;
122         }
123         else if(cc == "A"){
124             arrInt[i] = 1;
125         }

```

```
129         else {
130             arrayNum[i] = stoi(cc); //ubah jd int
131             arrInt[i] = arrayNum[i];
132             if (arrInt[i] < 2 || arrInt[i] > 9){
133                 valid = false;
134             }
135         }
136     }
137     char Operators[] = {'+', '-', '*', '/'};
138     if (!valid || userInput.length() >= 5){
139         cout << "Input tidak valid!" << endl;
140     }
141     else {
142         printAll(Operators, k, n, arrInt);
143     }
144 }
145
146 void inputRandom() {
147     int random, res;
148     string toStr, toStr2;
149
150     int i = 1;
151     srand(time(0));
152     for (int i = 0; i < 4; i++){
153         random = 1 + (rand() % 9);
154         arrInt[i] = random;
155     }
156
157     char Operators[] = {'+', '-', '*', '/'};
158     printAll(Operators, 3, 4, arrInt);
159 }
```

```
161 int calculate(char a, int b, int c) {
162     int hasil = 0;
163
164     switch (a) {
165         case '+':
166             hasil = b + c;
167             break;
168
169         case '*':
170             hasil = b * c;
171             break;
172
173         case '-':
174             if(b > c){
175                 hasil = b-c;
176             }
177             else {
178                 hasil = 0;
179                 c = 0;
180             }
181             break;
182
183         case '/':
184             if (b < c) {
185                 if (b != 0) {
186                     if ((c % b) == 0)
187                         hasil = c / b;
188                     else
189                         hasil = 0;
190                     c = 0;
191                 } else
192                     hasil = 0;
193                 c = 0;
194             } else {
195                 if (c != 0) {
196                     if ((b % c) == 0)
197                         hasil = b / c;
```

```

198         else
199             hasil = 0;
200             c = 0;
201     }
202     else
203         hasil = 0;
204         c = 0;
205     }
206     break;
207 }
208 return hasil;
209 }
210
211 void lexicograph(char *inputStr, char *data, int terakhir, int idx) {
212     int i;
213     for (i = 0; i < 4; i++) {
214         data[idx] = inputStr[i];
215         if (idx == terakhir)
216             cout << data << endl;
217         else
218             lexicograph(inputStr, data, terakhir, idx + 1);
219     }
220 }
221
222 void GameSolver(string arrStr[], int arrNum[]) {
223     char a, b, cc[1];
224     int total = 0, total2 = 0, num = 0, num2 = 0;
225     int numAA = 0, numBB = 0, numCC = 0, numDD = 0, numAB = 0, numCD = 0;
226
227     for (int i = 0; i < 3; i++) {
228         strcpy(cc, arrStr[i].c_str());
229         a = cc[0];
230
231         if (i == 0) {
232             total = arrNum[0];
233             num = arrNum[1];
234             total = calculate(a, total, num);

```



```
235     }
236     if (i == 1) {
237         num = arrNum[2];
238         total = calculate(a, total, num);
239     }
240     if (i == 2) {
241         num = arrNum[3];
242         total = calculate(a, total, num);
243     }
244 }
245
246 if (total == 24) {
247     bykSolusi++;
248     cout << "solusi " << bykSolusi << ": ";
249     cout << "(" << arrNum[0] << " ";
250     cout << arrStr[0];
251     cout << " " << arrNum[1] << ")";
252     cout << arrStr[1];
253     cout << " " << arrNum[2] << "]";
254     cout << arrStr[2];
255     cout << " " << arrNum[3] << " ";
256     cout << " = " << total << endl;
257
258     // save ke txt
259     ofstream myFile;
260     myFile.open("solusi.txt", ofstream::app);
261     myFile << "(" << arrNum[0] << " ";
262     myFile << arrStr[0];
263     myFile << " " << arrNum[1] << ")";
264     myFile << arrStr[1];
265     myFile << " " << arrNum[2] << "]";
266     myFile << arrStr[2];
267     myFile << " " << arrNum[3] << " ";
268     myFile << " = " << total << endl;
269     myFile.close();
270 }
```

```

271
272     for (int i = 0; i < 3; i++){
273         strcpy(cc, arrStr[i].c_str());
274         b = cc[0];
275
276         if (i == 0){
277             numAA = arrNum[0];
278             numBB = arrNum[1];
279             numAB = calculate(b, numAA, numBB);
280         }
281         if (i == 1){
282             numCC = arrNum[2];
283             numDD = arrNum[3];
284             numCD = calculate(b, numCC, numDD);
285         }
286         if (i == 2){
287             total2 = calculate(b, numAB, numCD);
288         }
289     }
290     if (total2 == 24){
291         bykSolusi++;
292         cout << "solusi " << bykSolusi << ": ";
293         cout << "[" << arrNum[0] << " ";
294         cout << arrStr[0];
295         cout << " " << arrNum[1] << ")";
296         cout << arrStr[1];
297         cout << " " << arrNum[2] << "]";
298         cout << arrStr[2];
299         cout << " " << arrNum[3] << " ";
300         cout << " = " << total2 << endl;
301
302         // save ke txt LAGI
303         ofstream myFile;
304         myFile.open("solusi.txt", ofstream::app);
305         myFile << "[" << arrNum[0] << " ";
306         myFile << arrStr[0];
307         myFile << " " << arrNum[1] << ")";

```




```
===== 24 GAME SOLVER =====
apakah anda ingin meng-input kartu secara manual [Y/N]? : Y
Masukkan kartu ^^ : 562K
solusi 1: [(5+13)-6]*2 = 24
solusi 2: [(6/2)-13]*5 = 24
solusi 3: [(13-5)/2]*6 = 24
solusi 4: [(13-5)/2]*6 = 24
solusi 5: [(13+5)-6]*2 = 24
solusi 6: [(13-5)*6]/2 = 24
solusi 7: [(13-5)/6]*2 = 24
solusi 8: [(13-6)+5]*2 = 24
Apakah anda ingin menyimpan solusi [Y / N] ? : N
0baik, sampai jumpa !
--- EXECUTION TIME: 0.08700 s ---
```

```
PS C:\Vanessa\ITB\tingkat 2\sem 4\stima> ./main
```



```
===== 24 GAME SOLVER =====
apakah anda ingin meng-input kartu secara manual [Y/N]? : Y
Masukkan kartu ^^ : QQQQ
solusi 1: [(12+12)+12]-12 = 24
solusi 2: [(12+12)-12]+12 = 24
solusi 3: [(12+12)-12]+12 = 24
solusi 4: [(12+12)-12]-12 = 24
solusi 5: [(12+12)*12]/12 = 24
solusi 6: [(12+12)/12]*12 = 24
solusi 7: [(12+12)/12]*12 = 24
solusi 8: [(12+12)/12]/12 = 24
solusi 9: [(12-12)+12]+12 = 24
solusi 10: [(12-12)+12]+12 = 24
solusi 11: [(12*12)/12]+12 = 24
solusi 12: [(12/12)+12]*12 = 24
solusi 13: [(12/12)+12]/12 = 24
solusi 14: [(12/12)*12]+12 = 24
solusi 15: [(12/12)/12]+12 = 24
Apakah anda ingin menyimpan solusi [Y / N] ? : Y
solusi berhasil disimpan :)
--- EXECUTION TIME: 0.08300 s ---
PS C:\Vanessa\ITB\tingkat 2\sem 4\stima> █
```

```

===== 24 GAME SOLVER =====
apakah anda ingin meng-input kartu secara manual [Y/N]? : N
solusi 1: [(6+4)+8]+6 = 24
solusi 2: [(6+4)+8]+6 = 24
solusi 3: [(6-4)*8]/6 = 24
solusi 4: [(6-4)/8]*6 = 24
solusi 5: [(6*4)/8]+6 = 24
solusi 6: [(6*4)/8]-6 = 24
Apakah anda ingin menyimpan solusi [Y / N] ? : Y
solusi berhasil disimpan :)
--- EXECUTION TIME: 0.05300 s ---

```

```

PS C:\Vanessa\ITB\tingkat 2\sem 4\stima> ./main

===== 24 GAME SOLVER =====
apakah anda ingin meng-input kartu secara manual [Y/N]? : Y
Masukkan kartu ^^ : AJQK
solusi 1: [(1*13)-11]*12 = 24
solusi 2: [(1/13)-11]*12 = 24
solusi 3: [(1-12)+11]+13 = 24
solusi 4: [(1-12)+11]+13 = 24
solusi 5: [(1-12)+13]+11 = 24
solusi 6: [(1-12)+13]+11 = 24
solusi 7: [(1*12)-13]*11 = 24
solusi 8: [(1/12)-13]*11 = 24
solusi 9: [(11+13)-1]+12 = 24
solusi 10: [(11+13)-1]-12 = 24
solusi 11: [(13*1)-11]*12 = 24
solusi 12: [(13/1)-11]*12 = 24
solusi 13: [(13+11)-1]+12 = 24
solusi 14: [(13+11)-1]-12 = 24
solusi 15: [(13-11)*1]*12 = 24
solusi 16: [(13-11)*1]*12 = 24
solusi 17: [(13-11)/1]*12 = 24
solusi 18: [(13-11)/1]*12 = 24
solusi 19: [(13-11)*12]*1 = 24
solusi 20: [(13-11)*12]*1 = 24
solusi 21: [(13-11)*12]/1 = 24
solusi 22: [(13-11)/12]*1 = 24
solusi 23: [(12*1)-13]*11 = 24
solusi 24: [(12/1)-13]*11 = 24
Apakah anda ingin menyimpan solusi [Y / N] ? : N
Obaik, sampai jumpa !
--- EXECUTION TIME: 0.10600 s ---

```



===== 24 GAME SOLVER =====

apakah anda ingin meng-input kartu secara manual [Y/N]?: Y

Masukkan kartu ^^ : AJ49

solusi 1: $[(1*11)+4]+9 = 24$

solusi 2: $[(1*11)+4]+9 = 24$

solusi 3: $[(1/11)+4]+9 = 24$

solusi 4: $[(1/11)+4]+9 = 24$

solusi 5: $[(1*11)+9]+4 = 24$

solusi 6: $[(1*11)+9]+4 = 24$

solusi 7: $[(1/11)+9]+4 = 24$

solusi 8: $[(1/11)+9]+4 = 24$

solusi 9: $[(11+4)+9]*1 = 24$

solusi 10: $[(11+4)+9]/1 = 24$

solusi 11: $[(11+4)*9]+1 = 24$

solusi 12: $[(11+4)/9]+1 = 24$

solusi 13: $[(11+4)*1]+9 = 24$

solusi 14: $[(11+4)*1]+9 = 24$

solusi 15: $[(11+4)/1]+9 = 24$

solusi 16: $[(11+4)/1]+9 = 24$

solusi 17: $[(11+9)+4]*1 = 24$

solusi 18: $[(11+9)+4]/1 = 24$

solusi 19: $[(11+9)*4]+1 = 24$

solusi 20: $[(11+9)/4]+1 = 24$

solusi 21: $[(11+9)*1]+4 = 24$

solusi 22: $[(11+9)*1]+4 = 24$

solusi 23: $[(11+9)/1]+4 = 24$

solusi 24: $[(11+9)/1]+4 = 24$

solusi 25: $[(11*1)+4]+9 = 24$

solusi 26: $[(11*1)+4]+9 = 24$

solusi 27: $[(11/1)+4]+9 = 24$

solusi 28: $[(11/1)+4]+9 = 24$

solusi 29: $[(11*1)+9]+4 = 24$

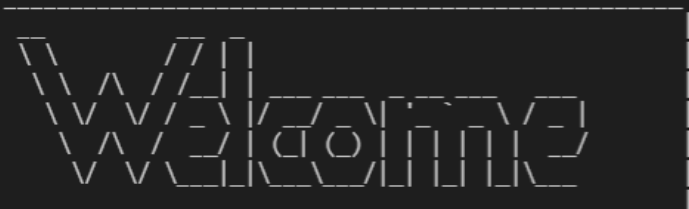
solusi 30: $[(11*1)+9]+4 = 24$

solusi 31: $[(11/1)+9]+4 = 24$

solusi 32: $[(11/1)+9]+4 = 24$

Apakah anda ingin menyimpan solusi [Y / N] ? : Y

solusi berhasil disimpan :)



===== 24 GAME SOLVER =====

apakah anda ingin meng-input kartu secara manual [Y/N]?: N

solusi 1: $[(5-8)+4]*6 = 24$

solusi 2: $[(5-8)*4]+6 = 24$

solusi 3: $[(5/8)+4]*6 = 24$

solusi 4: $[(5/8)*4]+6 = 24$

Apakah anda ingin menyimpan solusi [Y / N] ? : N

0baik, sampai jumpa !

--- EXECUTION TIME: 0.03700 s ---

3. *Link Repository*

https://github.com/vanessrw/Tucil1_13521151.git

4. To Do List

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