Tugas Kecil I IF2211 Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force*

Disusun Oleh :

Laila Bilbina Khoiru Nisa (13521016)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2023

A. Algoritma *Brute Force*

Program menggunakan algoritma Brute Force dengan langkah-langkah sebagai berikut.

1. Melakukan 4 pengulangan bersarang dengan setiap pengulangan melakukan iterasi terhadap variabel indeks array.
2. Untuk setiap pengulangan setiap variabel tidak boleh memiliki nilai yang sama dengan variabel yang lain sehingga tidak terjadi double.
3. Pengecekan apakah kombinasi keempat angka tersebut belum pernah digunakan sebelumnya.
4. Melakukan iterasi dari 1 sampai 5 dengan kelima angka menunjukan peletakan tanda kurung.
5. Melakukan tiga iterasi lagi dengan pengulangan 1 sampai 4 untuk peletakan symbol operasi.
6. Pengecrekan apakah hasil dari operasi merupakan 24 atau bukan.
7. Program juga dapat menghitung berapa banyak kombinasi yang menghasilkan 24.

B. Source Program

Program dibuat dengan bahasa C++ dalam 1 file bernama Tucil1_13521016.cpp dengan isi sebagai berikut.

src > Tucil_13521016.cpp > main()

```cpp
#include <iostream>
#include <vector>
#include <sstream>
#include <istream>
#include <fstream>
#include <cstdlib>
#include <chrono>

using namespace std;

// variabel global
void inputanUser(vector<string> *kartuu);
bool iskartu(string kartu);
vector<string> pengubahKartu(vector<string> kartuu);
void kartuKeAngka(vector<string> strkartu, vector<double> *dblkartu);
void bruteForceSolver24(vector<double> kartuList, string *hasil, int *counthasil);
bool isSame(vector<vector<double>> permutasi, double a, double b, double c, double d);
void permutasiKurung(double a, double b, double c, double d, string *hasil, int *counthasil, bool *is24);
void permutasiOperan(double a, double b, double c, double d, string *hasil, int *counthasil, bool *is24, int brType);
double operan(double a, double b, char op);
string angkaKeKartu(double a);



void inputanUser(vector<string> *kartuu) {
    string tipeInputan;
    string lineInputan, kartu, namaFile, stemp;
    int i;
    int M;
    bool baik;
    ifstream ok;

    cout << "Masukkan tipe inputan (file atau keyboard): " << endl;
    cout << "1. Keyboard" << endl;
    cout << "2. File" << endl;
    cout << "3. Random" << endl;
    cout << "Pilihan: ";
    getline(cin, tipeInputan);
```

```
    if (tipeInputan == "1") {
        baik = false;
        M = 0;
        do {
            cout << "Masukkan 4 kartu : ";
            getline(cin, lineInputan);
            stringstream ss(lineInputan);

            M = 0; kartu.clear();
            while (ss >> kartu){
                if (iskartu(kartu)) {
                    kartuu->push_back(kartu);
                    M++;
                }
                else {
                    break;
                }
            }

            if (M == 4) {
                baik = true;
            }
            else {
                cout << "Input tidak valid, Ulangi!" << endl;
            }
        } while (!baik);
    }
    else if (tipeInputan == "2") {
        cout << "Masukkan nama file : ";
        getline(cin, stemp);
        namaFile = "test/" + stemp;

        ok.open(namaFile);
        if (ok.fail()) {
            ok.clear();
            cout << "File tidak ditemukan\n" << "exiting program..." << endl;
            exit(1);
        }
        getline(ok, lineInputan);
```

```
        ok.close();

        stringstream ss(lineInputan);

        M = 0;
        while(ss >> kartu) {
            if (iskartu(kartu)) {
                kartuu->push_back(kartu);
                M++;
            }
            else {
                break;
            }
        }

        if (M != 4) {
            cout << "Isi file tidak valid!" << endl;
            exit(1);
        }
    }
    else if (tipeInputan == "3") {
        srand(time(NULL));

        for (i = 0; i < 4; i++) {
            kartuu->push_back(to_string(rand() % 13 + 1));
        }
        *kartuu = pengubahKartu(*kartuu);
    }
    else {
        cout << "Input tidak valid, exiting program..." << endl;
        exit(1);
    }
}

bool iskartu(string kartu) {
    if (kartu == "A" || kartu == "J" || kartu == "Q" || kartu == "K")
    {
        return true;
    }
```

```
    }
    else if (kartu == "1" || kartu == "2" || kartu == "3" || kartu == "4" || kartu == "5" || kartu == "6" || kartu == "7" || kartu == "8" || kartu == "9" || kartu == "10"
    {
        return true;
    }
    else
    {
        return false;
    }
}

void kartuKeAngka(vector<string> strkartu, vector<double> *dblkartu) {
    int i;
    string kartu;

    i = 0;
    do
    {
        if (strkartu[i] == "A")
        {
            dblkartu->push_back(1);
        }
        else if (strkartu[i] == "J")
        {
            dblkartu->push_back(11);
        }
        else if (strkartu[i] == "Q")
        {
            dblkartu->push_back(12);
        }
        else if (strkartu[i] == "K")
        {
            dblkartu->push_back(13);
        }
        else
        {
            dblkartu->push_back(stod(strkartu[i]));
        }
        i++;
```

```
    } while (i < 4);
}

vector<string> pengubahKartu(vector<string> strkartu) {
    int i;
    string kartu;

    for (i = 0; i < 4; i++)
    {
        if (strkartu[i] == "1")
        {
            strkartu[i] = "A";
        }
        else if (strkartu[i] == "11")
        {
            strkartu[i] = "J";
        }
        else if (strkartu[i] == "12")
        {
            strkartu[i] = "Q";
        }
        else if (strkartu[i] == "13")
        {
            strkartu[i] = "K";
        }
    }

    return strkartu;
}

// Proses
void bruteForceSolver24(vector<double> kartuList, string *hasil, int *counthasil) {
    vector<vector<double>> dbkartuuPermutation;
    double a, b, c, d;
    int i, j, k, l;
    bool is24;
    int M = 0;

    is24 = false;
```

```
195         *counthasil = 0;
196
197         // Permutasi
198         for (i = 0; i < 4; i++)
199         {
200             for (j = 0; j < 4; j++)
201             {
202                 for (k = 0; k < 4; k++)
203                 {
204                     for (l = 0; l < 4; l++)
205                     {
206                         if (i != j && i != k && i != l && j != k && j != l && k != l)
207                         {
208                             a = kartuList[i];
209                             b = kartuList[j];
210                             c = kartuList[k];
211                             d = kartuList[l];
212
213                             if (!isSame(dbkartuuPermutation, a, b, c, d))
214                             {
215                                 dbkartuuPermutation.push_back({ a, b, c, d });
216                                 permutasiKurung(a, b, c, d, hasil, counthasil, &is24);
217                             }
218                         }
219                     }
220                 }
221             }
222         }
223     }
224
225     bool isSame(vector<vector<double>> permutasi, double a, double b, double c, double d) {
226         int i, j;
227         bool isSame = false;
228
229         for (i = 0; i < permutasi.size(); i++)
230         {
231             if (permutasi[i][0] == a && permutasi[i][1] == b && permutasi[i][2] == c && permutasi[i][3] == d)
232             {
233                 isSame = true;
```

```
234             }
235         }
236
237         return isSame;
238     }
239
240     void permutasiKurung(double a, double b, double c, double d, string *hasil, int *counthasil, bool *is24) {
241         int i = 1;
242
243         for (i = 1; i <= 5; i++)
244         {
245             permutasiOperan(a, b, c, d, hasil, counthasil, is24, i);
246         }
247     }
248
249     void permutasiOperan(double a, double b, double c, double d, string *hasil, int *counthasil, bool *is24, int Tipe) {
250         string op = "+-*/";
251         string temphasil;
252         int i, j, k;
253         double aa, bb, cc, dblhasil;
254
255         for (i = 0; i < 4; i++)
256         {
257             for (j = 0; j < 4; j++)
258             {
259                 for (k = 0; k < 4; k++)
260                 {
261                     dblhasil = 0;
262                     aa = operan(a, b, op[i]);
263                     bb = operan(b, c, op[j]);
264                     cc = operan(c, d, op[k]);
265
266                     switch (Tipe)
267                     {
268                     case 1:
269                         dblhasil = operan(aa, cc, op[j]);
270                         temphasil = "(" + angkaKeKartu(a) + " " + op[i] + " " + angkaKeKartu(b) + ") " + op[j] + " (" + angkaKeKartu(c) + " " + op[k] + " " + angkaKeKartu(d)
271                         break;
272                     case 2:
```

```
                case 2:
                    dblhasil = operan(operan(aa, c, op[j]), d, op[k]);
                    temphasil = "((" + angkaKeKartu(a) + " " + op[i] + " " + angkaKeKartu(b) + ") " + op[j] + " " + angkaKeKartu(c) + ") " + op[k] + " " + angkaKeKartu(d)
                    break;
                case 3:
                    dblhasil = operan(operan(a, bb, op[i]), d, op[k]);
                    temphasil = "(" + angkaKeKartu(a) + " " + op[i] + " (" + angkaKeKartu(b) + " " + op[j] + " " + angkaKeKartu(c) + ")) " + op[k] + " " + angkaKeKartu(d)
                    break;
                case 4:
                    dblhasil = operan(a, operan(bb, d, op[k]), op[i]);
                    temphasil = angkaKeKartu(a) + " " + op[i] + " ((" + angkaKeKartu(b) + " " + op[j] + " " + angkaKeKartu(c) + ") " + op[k] + " " + angkaKeKartu(d) + ")"
                    break;
                case 5:
                    dblhasil = operan(a, operan(b, cc, op[j]), op[i]);
                    temphasil = angkaKeKartu(a) + " " + op[i] + " (" + angkaKeKartu(b) + " " + op[j] + " (" + angkaKeKartu(c) + " " + op[k] + " " + angkaKeKartu(d) + "))"
                    break;
                }

                if (dblhasil == 24)
                {
                    *hasil += temphasil + "\n";
                    *counthasil += 1;
                    *is24 = true;
                }
            }
        }
    }
}

double operan(double a, double b, char op) {
    double hasil;

    switch (op)
    {
    case '+':
        hasil = a + b;
        break;
    case '-':
        hasil = a - b;
        break;
    case '*':
        hasil = a * b;
        break;
    case '/':
        if (b == 0)
        {
            hasil = 0;
        }
        else
        {
            hasil = a / b;
        }
        break;
    }
    return hasil;
}

string angkaKeKartu(double a) {
    string kartu;

    if (a == 1)
    {
        kartu = "A";
    }
    else if (a == 11)
    {
        kartu = "J";
    }
    else if (a == 12)
    {
        kartu = "Q";
    }
    else if (a == 13)
    {
        kartu = "K";
    }
    else
    {
        kartu = to_string((int)a);
    }
```

```
352     return kartu;
353 }
354
355 void displaykartu(vector<string> h) {
356     int i;
357
358     for (i = 0; i < h.size(); i++) {
359         cout << h[i] << " ";
360     }
361     cout << endl;
362 }
363
364 void simpanFile(string hasil, vector<string> kartuu) {
365     ofstream file;
366     string namafile;
367     int i;
368
369     namafile = "";
370     for (i = 0; i < kartuu.size(); i++) {
371         namafile += kartuu[i];
372     }
373     namafile += ".txt";
374     file.open(namafile);
375     file << hasil;
376     file.close();
377     cout << "Hasil telah disimpan di file " << namafile << endl;
378 }
379
380 int main() {
381     vector<string> kartuu;
382     vector<double> kartuu1;
383     string hasil;
384     double executionTime;
385     int counthasil;
386
387     cout << "==================== 24 Solver : Card Game <3 ====================" << endl;
388     inputanUser(&kartuu);
389     kartuKeAngka(kartuu, &kartuu1);
390     cout << "============================================================" << endl;
```

```
390     cout << "============================================================" << endl;
391     displaykartu(kartuu);
392
393     counthasil = 0;
394     auto start = chrono::high_resolution_clock::now();
395     bruteForceSolver24(kartuu1, &hasil, &counthasil);
396     auto end = chrono::high_resolution_clock::now();
397     executionTime = chrono::duration_cast<chrono::microseconds>(end - start).count();
398
399     cout << counthasil<< "jumlah solusi :" << endl;
400     cout << hasil << endl;
401     cout << "Execution time : " << executionTime << " microseconds" << endl;
402     simpanFile(hasil, kartuu);
403
404     return 0;
405 }
```

C. Hasil Pengujian

1. Tampilan awal

```
PS D:\TUBES\Tucil1_13521016> cd "d:\TUBES\Tucil1_13521016\src\" ; if ($?) { g++ Tucil_13521016
.cpp -o Tucil_13521016 } ; if ($?) { .\Tucil_13521016 }
==================== 24 Solver : Card Game <3 ====================
Masukkan tipe inputan (file atau keyboard):
1. Keyboard
2. File
3. Random
Pilihan:
```

2. Input dari keyboard
   a. Masukan 2 5 8 4

```
Pilihan: 1
Masukkan 4 kartu : 2 5 8 4
================================================================
2 5 8 4
58jumlah solusi :
((2 + 5) - 4) * 8
(2 + (5 - 4)) * 8
2 * ((5 * 4) - 8)
(2 * (8 - 5)) * 4
2 * ((8 - 5) * 4)
((2 - 4) + 5) * 8
((2 * 4) - 5) * 8
(2 - (4 - 5)) * 8
2 * ((4 * 5) - 8)
(2 * 4) * (8 - 5)
2 * (4 * (8 - 5))
((5 / 2) * 8) + 4
(5 / (2 / 8)) + 4
((5 + 2) - 4) * 8
(5 + (2 - 4)) * 8
((5 * 8) / 2) + 4
(5 * (8 / 2)) + 4
((5 - 4) + 2) * 8
(5 - (4 - 2)) * 8
(5 - (4 / 2)) * 8
(5 * 4) + (8 / 2)
4 + (5 / (2 / 8))
(4 * 5) + (8 / 2)
((4 * 5) - 8) * 2
4 + ((5 * 8) / 2)
4 + (5 * (8 / 2))
4 + ((8 / 2) * 5)
4 + (8 / (2 / 5))
(4 * (8 - 5)) * 2
4 + ((8 * 5) / 2)
4 * ((8 - 5) * 2)
4 + (8 * (5 / 2))

Execution time : 3907 microseconds
Hasil telah disimpan di file 2584.txt
```

   b. Masukan 1 2 3 4

```
Pilihan: 1
Masukkan 4 kartu : 1 2 3 4
Input tidak valid, Ulangi!
Masukkan 4 kartu :
```

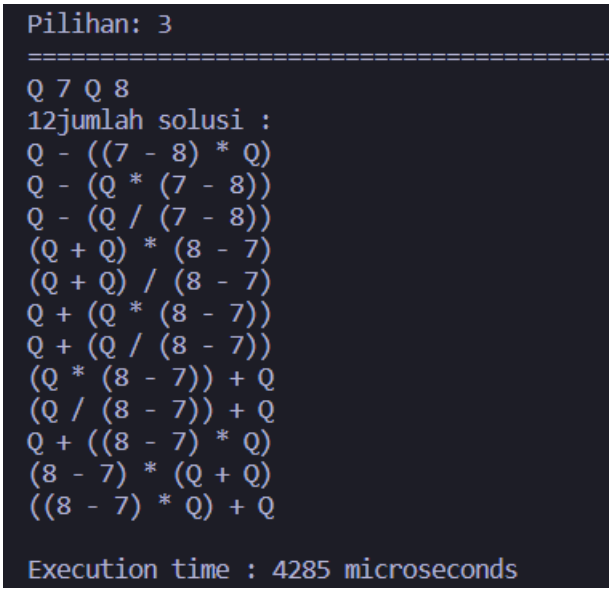
c. Masukan K 4 6 2

```
Masukkan 4 kartu : K 4 6 2
==========================================
K 4 6 2
24jumlah solusi :
((K - 4) * 2) + 6
(K * (6 - 4)) - 2
(K * 2) + (4 - 6)
((K * 2) + 4) - 6
(K * 2) - (6 - 4)
((K * 2) - 6) + 4
(4 + (K * 2)) - 6
4 + ((K * 2) - 6)
(4 - 6) + (K * 2)
4 - (6 - (K * 2))
(4 - 6) + (2 * K)
4 - (6 - (2 * K))
(4 + (2 * K)) - 6
4 + ((2 * K) - 6)
```

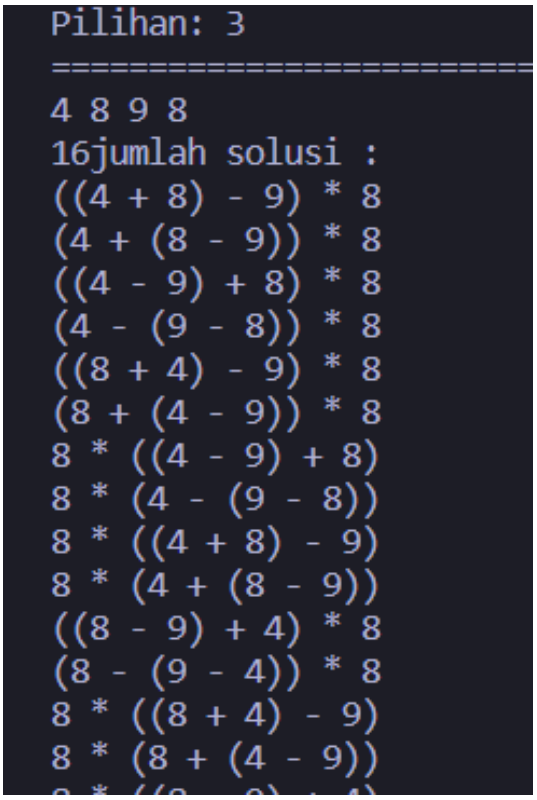
3. Input Random
   a. Masukan 2 8 10 9

```
Pilihan: 3
============================
2 8 10 9
20jumlah solusi :
(2 * (8 + 9)) - 10
((2 + 10) - 9) * 8
(2 + (10 - 9)) * 8
(2 * (9 + 8)) - 10
((2 - 9) + 10) * 8
(2 - (9 - 10)) * 8
8 * ((2 + 10) - 9)
8 * (2 + (10 - 9))
8 * ((2 - 9) + 10)
8 * (2 - (9 - 10))
8 * ((10 + 2) - 9)
8 * (10 + (2 - 9))
```

b. Masukan Q 7 Q 8



c. Masukan 4 8 9 8



D. Lampiran

Link Github : https://github.com/Lailabkn/Tucil1_13521016.git