

LAPORAN TUGAS KECIL
IF2211 STRATEGI ALGORITMA
24 PUZZLE SOLVER MENGGUNAKAN ALGORITMA
BRUTE FORCE



Disusun Oleh :

Febryan Arota Hia 13521120

1. 24 Puzzle

Permainan kartu 24 adalah permainan kartu aritmatika dengan tujuan mencari cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24. Permainan ini menarik cukup banyak peminat dikarenakan dapat meningkatkan kemampuan berhitung serta mengasah otak agar dapat berpikir dengan cepat dan akurat. Permainan Kartu 24 biasa dimainkan dengan menggunakan kartu remi. Kartu remi terdiri dari 52 kartu yang terbagi menjadi empat suit (sekop, hati, keriting, dan wajik) yang masing-masing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Yang perlu diperhatikan hanyalah nilai kartu yang didapat (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). As bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri. Pada awal permainan moderator atau salah satu pemain mengambil 4 kartu dari dek yang sudah dikocok secara random. Permainan berakhir ketika pemain berhasil menemukan solusi untuk membuat kumpulan nilainya menjadi 24. Pengubahan nilai tersebut dapat dilakukan menggunakan operasi dasar matematika penjumlahan (+), pengurangan (-), perkalian (\times), divisi (/) dan tanda kurung (). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas. (Paragraf di atas dikutip dari [sini](#))

2. Algoritma Brute Force pada 24 Puzzle Solver

Strategi *brute force* merupakan salah satu strategi pemecahan masalah yang sederhana. Dalam pengaplikasiannya, algoritma ini akan mencoba semua kemungkinan yang ada hingga suatu permasalahan tersebut terselesaikan. Algoritma ini mungkin bukan merupakan algoritma yang paling efisien tetapi relatif lebih mudah diaplikasikan dan dapat digunakan untuk memecahkan hampir sebagian besar masalah.

Pada tugas kecil ini, penulis menggunakan algoritma *brute force* untuk mencari semua kemungkinan solusi yang ada dalam penyelesaian 24 puzzle. Adapun langkah-langkah yang diimplementasikan sebagai berikut.

1. Program akan menerima input 4 kartu dengan validasi hingga input sesuai, atau program dapat men-*generate* 4 kartu secara random.
2. Keempat nilai kartu tersebut akan dicari setiap kemungkinan permutasinya dengan menggunakan 4 buah loop, lalu disimpan pada suatu array. Pada looping ini semua kemungkinan akan selalu berjumlah 24 karena tidak ada validasi apakah keempat kartu pasti berbeda. Contoh, kartu mungkin saja 4 Q 6 4, tetapi pada proses ini duplicate dari kemungkinan yang sama akan tetap disimpan.
 - a. 4 Q 6 4;
 - b. 4 Q 4 6;
 - c. 4 4 Q 6;
 - d. 4 Q 6 4; dst. (a dan d dianggap berbeda)
3. Array kemungkinan permutasi akan dicek satu per-satu, apabila terdapat urutan yang sama (duplicate), elemen tersebut akan diubah menjadi 0, menandakan duplicate dan tidak akan digunakan untuk proses selanjutnya.
4. Selanjutnya program akan me-looping semua kemungkinan semua operasi (+, -, /, *) pada setiap kemungkinan permutasi yang sudah ada pada array sebelumnya dengan lima kombinasi urutan tanda kurung.
 - a. ((a op b) op c) op d
 - b. (a op (b op c)) op d
 - c. (a op b) op (c op d)
 - d. a op ((b op c) op d)
 - e. a op (b op (c op d))
5. Lalu program akan menampilkan seluruh kemungkinan solusi yang menghasilkan nilai 24.

3. Source Code

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <vector>
using namespace std;

float calc(int ops, float a, float b) {
    /*Mengembalikan nilai operasi a terhadap b*/
    /*operator list:
    0 = sum, 1 = sub, 2 = mul, else = div*/
    if (ops == 0) {
        return a + b;
    } else if (ops == 1) {
        return a - b;
    } else if (ops == 2) {
        return a * b;
    } else {
        return a / b ;
    }
}

string ToString(int num){
    /*Mengubah integer menjadi string*/
    string str = "";
    while (num > 0){
        str = (char)(num % 10 + 48) + str;
        num /= 10;
    }
    return str;
}

int main() {
    cout << R"(
        .-----..-----.
    .-----..-----..-----..-----..-----..-----.
        |2.--. ||4.--. |.-.  |P.--. ||U.--. ||Z.--. ||Z.--. ||L.--.
    ||E.--. |
        | (\/) || :/\: |      | :/\: || (\/) || :(): || :(): || :/\: ||
    (\/) |
```

```

| :\\/: || :\\/: |'-.-.| (__) || :\\/: || () () || () () || (__) ||
:\\/: |
| '---'2|| '---'4| | '---'P|| '---'U|| '---'Z|| '---'Z|| '---'L||
'---'E|
'-----'-----'
|'-----'-----'-----'-----'-----'-----'
2 4 P U Z Z L E
)";
cout << "\\n\\n1. Manual Input" << endl;
cout << "2. Randomize" << endl;
int input;

do {
    cout << "Pilih menu: ";
    cin >> input;
} while (!(input == 1 || input == 2));

int values[4];
int n;
int count = 0;
char card[4];

if (input == 1) { //input manual
    cout << "\\nMasukkan 4 kartu (2-10, A, J, Q, K)!" << endl ;
    for (int i=0; i< 4; i++) {
        cout << "kartu ke-" << i+1 << ": ";
        cin >> card;
        if (card[0] == 'A' && card[1] == '\\0') {
            values[i] = 1;
        } else if (card[0] == 'J' && card[1] == '\\0') {
            values[i] = 11;
        } else if (card[0] == 'Q' && card[1] == '\\0') {
            values[i] = 12;
        } else if (card[0] == 'K' && card[1] == '\\0') {
            values[i] = 13;
        } else if (int(card[0]) >= 50 && int(card[0]) <= 57 && card[1]
== '\\0') {
            values[i] = int(card[0]) - 48;
        } else if (card[0] == '1' && card[1] =='0' && card[2] == '\\0')

```

```

{
    values[i] = 10;
} else {
    cout << "Masukan salah. Ulangi. \n\n";
    i-=1;
}
}
} else if (input == 2) { //input random
    srand(time(0)); //random seed
    cout << "\nKartu: \n";
    for (int i = 0; i < 4; i++) {
        values[i] = 1 + rand() % 13;
        if (values[i] == 1) {
            cout << "A ";
        } else if (values[i] == 11) {
            cout << "J ";
        } else if (values[i] == 12) {
            cout << "Q ";
        } else if (values[i] == 13) {
            cout << "K ";
        } else {
            cout << values[i] << " ";
        }
    }
}
}

clock_t time_req; //start execution timer
time_req = clock();
vector<string> vec;
string str;

//permutasi urutan 4 bilangan (max 4! = 24)
/*kemungkinan
a # b # c # d
a # c # b # d
a # d # b # c
dst
*/
int perm[24][4];

```

```

int p = 0;
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        for (int k = 0; k < 4; k++) {
            for (int l = 0; l < 4; l++) {
                if (!(i == j || i == k || i == l || j == k || j == l || k
== l)) {
                    perm[p][0] = values[i];
                    perm[p][1] = values[j];
                    perm[p][2] = values[k];
                    perm[p][3] = values[l];
                    p++;
                }
            }
        }
    }
}

//Memeriksa apakah terdapat permutasi yang sama
for (int i = 0; i<24; i++) {
    for (int j = i +1 ; j < 24; j++) {
        if (perm[j][0] == perm[i][0] && perm[j][1]== perm[i][1] &&
perm[j][2] == perm[i][2] && perm[j][3] == perm[i][3]) {
            perm[i][0] = 0; //tanda duplicate
        }
    }
}

char ops[4] = {'+', '-', '*', '/'};
for (int i = 0; i<24; i++) {
    if (perm[i][0] != 0) {
        int a = perm[i][0]; int b = perm[i][1]; int c = perm[i][2]; int
d = perm[i][3];
        for (int i = 0; i<4; i++) {
            for (int j = 0; j<4; j++) {
                for (int k = 0; k<4; k++) {

                    // ((a #i b) #j c) #k d
                    if (calc(k, calc(j, calc(i, a, b), c), d) == 24) {

```

```

        count += 1;
        str = ToString(count) + ". (" + ToString(a) + ops[i] +
ToString(b) + ")" + ops[j] + ToString(c) + ")" + ops[k] + ToString(d)
+ "\n";

        vec.push_back(str);
    }

    // (a #i (b #j c)) #k d
    if (calc(k, calc(i, a, calc(j, b, c)), d) == 24) {
        count += 1;
        str = ToString(count) + ". (" + ToString(a) + ops[i] +
"(" + ToString(b) + ops[j] + ToString(c) + "))" + ops[k] +
ToString(d) + "\n";
        vec.push_back(str);
    }

    // (a #i b) #j (c #k d)
    if (calc(j, calc(i, a, b), calc(k, c, d)) == 24) {
        count += 1;
        str = ToString(count) + ". (" + ToString(a) + ops[i] +
ToString(b) + ")" + ops[j] + "(" + ToString(c) + ops[k] + ToString(d)
+ ")" + "\n";
        vec.push_back(str);
    }

    // a #i ((b #j c) #k d)
    if (calc(i, a, calc(k, calc(j, b, c), d)) == 24) {
        count += 1;
        str = ToString(count) + ". " + ToString(a) + ops[i] +
"(" + ToString(b) + ops[j] + ToString(c) + ")" + ops[k] +
ToString(d) + ")" + "\n";
        vec.push_back(str);
    }

    // a #i (b #j (c #k d))
    if (calc(i, a, calc(j, b, calc(k, c, d))) == 24) {
        count += 1;
        str = ToString(count) + ". " + ToString(a) + ops[i] +
"(" + ToString(b) + ops[j] + "(" + ToString(c) + ops[k] + ToString(d)

```



```

+ "}))" + "\n";
        vec.push_back(str);
    }
}
}

}

}

if (count == 0) {
    cout << "\n\nTidak ada solusi.\n";
} else {
    cout << "\n\nTotal solusi: " << count << endl;
    for (auto i : vec) {
        cout << i;
    }
}

time_req = clock() - time_req;
cout << "Waktu eksekusi: " << (float)time_req/CLOCKS_PER_SEC << "
detik" << endl;

char save;
do {
    cout << "\nSimpan solusi? (y/n) " ;
    cin >> save;
} while (!(save == 'y' || save == 'n'));

if (save == 'y') {
    string filename;
    cout << "Masukkan nama file: ";
    cin >> filename;
    filename = "../test/" + filename + ".txt";
    ofstream file(filename);

    file << "Kartu:\n";
    for (int i = 0; i < 4; i++){
        if (values[i] == 1) {
            file << "A ";
        } else if (values[i] == 11) {

```

```

        file << "J ";
    } else if (values[i] == 12) {
        file << "Q ";
    } else if (values[i] == 13) {
        file << "K ";
    } else {
        file << values[i] << " ";
    }
}

if (count == 0) {
    file << "\n\nTidak ada solusi.\n";
} else {
    file << "\n\nTotal solusi: " << count << endl;
    for (auto i : vec) {
        file << i;
    }
}

file << "Waktu eksekusi: " << (float)time_req/CLOCKS_PER_SEC << "
detik" << endl;
cout << "\nSolusi berhasil disimpan!";
}
}

```

4. Test Case

Input	Output
<pre>1. Manual Input 2. Randomize Pilih menu: 2 Kartu: 10 2 3 10</pre>	<pre>Total solusi: 6 1. (2*(10-3))+10 2. 10+((10-3)*2) 3. 10+(2*(10-3)) 4. 10-(2*(3-10)) 5. 10-((3-10)*2) 6. ((10-3)*2)+10 Waktu eksekusi: 0.002 detik Simpan solusi? (y/n) n</pre>
<pre>1. Manual Input 2. Randomize Pilih menu: 2 Kartu: 8 3 7 9</pre>	<pre>Tidak ada solusi. Waktu eksekusi: 0 detik Simpan solusi? (y/n) █</pre>

```
1. Manual Input
2. Randomize
Pilih menu: 2
```

```
Kartu:
2 6 K 5
```

Total solusi: 23

1. $2 * ((13 - 6) + 5)$
2. $2 * (13 - (6 - 5))$
3. $2 * ((13 + 5) - 6)$
4. $2 * (13 + (5 - 6))$
5. $2 * ((5 - 6) + 13)$
6. $2 * (5 - (6 - 13))$
7. $2 * ((5 + 13) - 6)$
8. $2 * (5 + (13 - 6))$
9. $(6 / 2) * (13 - 5)$
10. $6 / (2 / (13 - 5))$
11. $(6 * (13 - 5)) / 2$
12. $6 * ((13 - 5) / 2)$
13. $((13 - 6) + 5) * 2$
14. $(13 - (6 - 5)) * 2$
15. $((13 - 5) / 2) * 6$
16. $((13 + 5) - 6) * 2$
17. $(13 + (5 - 6)) * 2$
18. $((13 - 5) * 6) / 2$
19. $(13 - 5) * (6 / 2)$
20. $((5 - 6) + 13) * 2$
21. $(5 - (6 - 13)) * 2$
22. $((5 + 13) - 6) * 2$
23. $(5 + (13 - 6)) * 2$

Waktu eksekusi: 0.005 detik

```
1. Manual Input
2. Randomize
Pilih menu: 1
```

```
Masukkan 4 kartu (2-10, A, J, Q, K)!
kartu ke-1: 4
kartu ke-2: 4
kartu ke-3: 4
kartu ke-4: 4
```

Total solusi: 6

1. $(4 + 4) + (4 * 4)$
2. $4 + (4 + (4 * 4))$
3. $(4 + (4 * 4)) + 4$
4. $4 + ((4 * 4) + 4)$
5. $((4 * 4) + 4) + 4$
6. $(4 * 4) + (4 + 4)$

Waktu eksekusi: 0.002 detik

Simpan solusi? (y/n) ☐

1. Manual Input
2. Randomize
Pilih menu: 1

Masukkan 4 kartu (2-10, A, J, Q, K)!
kartu ke-1: 1
Masukan salah. Ulangi.

kartu ke-1: A
kartu ke-2: A
kartu ke-3: A
kartu ke-4: Q

Total solusi: 27

1. $((1+1)*1)*12$
2. $(1+(1*1))*12$
3. $(1+1)*(1*12)$
4. $((1+1)/1)*12$
5. $(1+(1/1))*12$
6. $((1*1)+1)*12$
7. $(1*(1+1))*12$
8. $1*((1+1)*12)$
9. $((1/1)+1)*12$
10. $((1+1)*12)*1$
11. $(1+1)*(12*1)$
12. $((1+1)*12)/1$
13. $(1+1)*(12/1)$
14. $(1*12)*(1+1)$
15. $1*(12*(1+1))$
16. $(12*(1+1))*1$
17. $12*((1+1)*1)$
18. $12*(1+(1*1))$
19. $(12*(1+1))/1$
20. $12*((1+1)/1)$
21. $12*(1+(1/1))$
22. $(12*1)*(1+1)$
23. $12*((1*1)+1)$
24. $12*(1*(1+1))$
25. $12*((1/1)+1)$
26. $(12/1)*(1+1)$
27. $12/(1/(1+1))$

Waktu eksekusi: 0.006 detik

Simpan solusi? (y/n) y
Masukkan nama file: test

Solusi berhasil disimpan!

1. Manual Input
2. Randomize
Pilih menu: 1

Masukkan 4 kartu (2-10, A, J, Q, K)!
kartu ke-1: 4
kartu ke-2: 4
kartu ke-3: 4
kartu ke-4: 4

Total solusi: 6

1. $(4+4)+(4*4)$
2. $4+(4+(4*4))$
3. $(4+(4*4))+4$
4. $4+((4*4)+4)$
5. $((4*4)+4)+4$
6. $(4*4)+(4+4)$

Waktu eksekusi: 0.002 detik

5. Lampiran

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	✓	

Repository github: https://github.com/febryanarota/Tucil1_13521120