

**LAPORAN
TUGAS KECIL 1 IF2211
STRATEGI ALGORITMA**

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force



oleh

Go Dillon Audris

13521062

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2022 / 2023**

DAFTAR ISI

DAFTAR ISI	2
BAB I PENJELASAN ALGORITMA PROGRAM	3
1.1 Permainan Kartu 24 dan Pencarian Solusi dengan Algoritma Brute Force	3
1.2 Implementasi Algoritma Brute Force untuk Mencari Solusi	3
BAB II SOURCE CODE PROGRAM	5
2.1 File converter.cpp	5
2.2 File io.cpp	8
2.3 File solver.cpp	14
2.4 File main.cpp	19
BAB III EKSPERIMEN DAN TESTING BERBAGAI SUSUNAN KARTU	21
3.1 Susunan Kartu 10, K, 2, 6	21
3.2 Susunan Kartu 2, 3, 5, 7	22
3.3 Susunan Kartu 6, 6, 6, 6	23
3.4 Susunan Kartu 7, 8, 5, 3	24
3.5 Susunan Kartu 9, 4, 4, 2	25
3.6 Susunan Kartu 10, 6, 4, 2	25
3.7 Susunan Kartu 2, 9, 10, 10	28
3.8 Susunan Kartu A, K, Q, J	29
3.9 Susunan Kartu K, A, 4, 9	30
3.10 Susunan Kartu Q, 5, A, A	31
LAMPIRAN	32
Link To Repository	33

BAB I PENJELASAN ALGORITMA PROGRAM

1.1 Permainan Kartu 24 dan Pencarian Solusi dengan Algoritma Brute Force

Permainan Kartu 24 merupakan permainan yang diselesaikan dengan mendapatkan angka 24 dari kombinasi perhitungan di antara empat angka yang diberikan. Hal ini menyebabkan terdapat berbagai kemungkinan penyusunan angka dan operasi yang mungkin dilakukan. Dalam algoritma Brute Force yang dipakai oleh penulis, seluruh solusi dari permainan ini dicari dengan mencoba setiap kemungkinan susunan empat angka, peletakan tanda kurung dan kemungkinan susunan operasi yang mungkin. Terdapat:

- 1) 24 kemungkinan susunan yang mungkin untuk keempat angka
- 2) 5 jenis peletakan tanda kurung di antara keempat angka, yaitu untuk sembarang angka p, q, r , dan s , peletakan tanda kurung dapat berupa::
 - a) $(p + q) + (r + s)$
 - b) $((p + q) + r) + s$
 - c) $(p + (q + r)) + s$
 - d) $p + ((q + r) + s)$
 - e) $p + (q + (r + s))$
- 3) 64 kemungkinan susunan untuk keempat tanda operasi (+, -, x, /)

1.2 Implementasi Algoritma Brute Force untuk Mencari Solusi

Algoritma Brute Force untuk mencari seluruh solusi Permainan Kartu 24 dimulai dengan melakukan empat pengulangan bersarang dimana setiap pengulangan akan melakukan iterasi terhadap variabel (i, j, k, l) dari angka 0 sampai 3. Variabel tersebut akan menandai indeks array untuk mengambil angka yang telah disimpan di dalam array. Untuk setiap pengulangan diberikan

syarat bahwa setiap variabel tidak boleh sama nilainya dengan variabel pada kalang di atasnya. Hal ini untuk mencegah pengambilan angka dari tempat atau indeks yang sama sehingga kemungkinan susunan keempat angka tetap unik dan tidak ada angka yang diduplikasi dan dipakai dua kali. Setelah keempat angka diambil dari array, dilakukan pengecekan untuk memastikan bahwa susunan empat angka yang terbentuk belum pernah digunakan sebelumnya. Hal ini dapat terjadi jika dari empat angka yang diberikan, ada dua atau lebih angka yang sama. Hal ini kemudian menyebabkan susunan angka yang identik, kedua angka yang sama hanya bertukar posisi saja. Setelah dilakukan pengecekan, kita akan mendapatkan susunan angka yang unik yang akan dipakai untuk menentukan kemungkinan operasi.

Selanjutnya, algoritma akan melakukan iterasi dari angka 1 sampai 5, kelima angka ini akan menandai kelima jenis peletakan tanda kurung yang akan digunakan dalam operasi seperti yang dijelaskan pada subbab 1.1. Algoritma lalu akan melakukan lagi tiga pengulangan bersarang, dimana setiap pengulangan melakukan iterasi dari angka 1 sampai 4 untuk menentukan susunan tanda operasi yang akan digunakan. Angka 1 sampai 4 akan menandai keempat tanda operasi yang dapat digunakan.

Setelah didapat susunan angka, peletakan tanda kurung, dan susunan tanda operasi yang unik, algoritma akan mengevaluasi operasi aritmetika tersebut. Jika didapat bahwa operasi aritmetika menghasilkan 24, maka operasi tersebut tidak akan langsung ditampilkan ke layar, namun disimpan sebagai suatu string ke dalam suatu variabel. Algoritma juga menaikkan nilai banyaknya solusi yang telah ditemukan. Algoritma lalu mengulang pengulangan untuk menemukan susunan operasi lainnya hingga seluruh susunan operasi yang mungkin sudah dievaluasi. Seluruh solusi dan banyaknya lalu ditampilkan ke layar.

BAB II SOURCE CODE PROGRAM

Program ditulis dengan menggunakan bahasa C++ dan terbagi-bagi menjadi beberapa file.

2.1 File converter.cpp

```
// File: converter.cpp
// Berisi kumpulan fungsi-fungsi antara untuk memudahkan konversi
antara string, integer, dan operand

#include <iostream>
using namespace std;

int charToNum (string CC) {
// Mengembalikan angka yang bersesuaian dengan A, 2-10, J, Q, K
sesuai dengan input user
// Mengembalikan -1 jika karakter tidak valid

// KAMUS LOKAL

// ALGORITMA
    if (CC == "A") {
        return 1;
    }
    else if (CC == "J") {
        return 11;
    }
    else if (CC == "Q") {
        return 12;
    }
    else if (CC == "K") {
        return 13;
    }
    else if (CC == "2") {
        return 2;
    }
    else if (CC == "3") {
        return 3;
    }
}
```

```

        else if (CC == "4") {
            return 4;
        }
        else if (CC == "5") {
            return 5;
        }
        else if (CC == "6") {
            return 6;
        }
        else if (CC == "7") {
            return 7;
        }
        else if (CC == "8") {
            return 8;
        }
        else if (CC == "9") {
            return 9;
        }
        else if (CC == "10") {
            return 10;
        }
        else {
            return -1;
        }
    }

string numToChar (int num) {
    // Mengembalikan karakter yang bersesuaian dengan angka num

    // KAMUS LOKAL

    // ALGORITMA
    if (num == 13) {
        return "K";
    }
    else if (num == 12) {
        return "Q";
    }
    else if (num == 11) {

```

```

        return "J";
    }
    else if (num == 10){
        return "10";
    }
    else if (num == 9) {
        return "9";
    }
    else if (num == 8) {
        return "8";
    }
    else if (num == 7) {
        return "7";
    }
    else if (num == 6) {
        return "6";
    }
    else if (num == 5) {
        return "5";
    }
    else if (num == 4) {
        return "4";
    }
    else if (num == 3) {
        return "3";
    }
    else if (num == 2) {
        return "2";
    }
    else {
        return "A";
    }
}

string numToOperand (int num) {
// Mengembalikan karakter operan yang bersesuaian dengan num
// 1 = +
// 2 = -
// 3 = *

```

```
// 4 = /

// KAMUS LOKAL

// ALGORITMA
if (num == 1) {
    return " + ";
}
else if (num == 2) {
    return " - ";
}
else if (num == 3) {
    return " * ";
}
else {
    return " / ";
}
}
```

2.2 File io.cpp

```
// File: io.cpp
// Berisi kumpulan fungsi-fungsi untuk menyelesaikan permasalahan
// terkait input/output
// dan yang akan digunakan pada file main.cpp

#include "converter.cpp"

#include <iostream>
#include <fstream>
#include <random>
using namespace std;

int countWords (string input) {
    // Mengembalikan jumlah kata dalam suatu strings

    // KAMUS LOKAL
    int count = 0;
```



```

    bool inWord = false;

// ALGORITMA
    for (char CC : input) {
        if (not(CC == ' ' || CC == '\n' || CC == '\0')) {
            inWord = true;
        }
        else if ((CC == ' ' || CC == '\n' || CC == '\0') && inWord)
        {
            count++;
            inWord = false;
        }
    }

    if (inWord) {
        count++;
        inWord = false;
    }

    return count;
}

bool getNum(int arr[], string input) {
// Mengembalikan true jika fungsi berhasil mengisi arr dengan angka
yang ada pada input
// Mengembalikan false jika tidak

// KAMUS LOKAL
    int count = 0;
    string temp = "";
    bool done = true, inWord = false;

// ALGORITMA
    for (char CC : input) {
        if (not(CC == ' ' || CC == '\n' || CC == '\0')) {
            temp += CC;
            inWord = true;
        }
        else if ((CC == ' ' || CC == '\n' || CC == '\0') && inWord)

```

```

{
    if (charToNum(temp) != -1) {
        arr[count] = charToNum(temp);
        count++;
        temp = "";
        inWord = false;
    }
    else {
        done = false;
        inWord = false;
        break;
    }
}

if (inWord) {
    if (charToNum(temp) != -1) {
        arr[count] = charToNum(temp);
        count++;
        temp = "";
        inWord = false;
    }
    else {
        done = false;
        inWord = false;
    }
}

return done;
}

string solve24mode () {
// Menampilkan pilihan mode untuk memasukkan angka kepada pengguna
(random atau input sendiri).
// Mengembalikan pilihan mode yang dipilih (1-2)

// KAMUS LOKAL
int len;
string temp;

```

```

    bool valid = false;

// ALGORITMA
    cout << "=====\n";
    cout << "Selamat datang di 24 Solver. Silakan pilih mode yang
anda inginkan: ";
    cout << "\n1. Input angka kartu oleh pengguna";
    cout << "\n2. Generate angka kartu secara random";

    while (!valid) {
        cout << "\nMasukkan pilihan mode (1-2): ";
        getline(cin, temp);
        len = countWords(temp);
        if (len == 1) {
            if (temp == "1" || temp == "2") {
                valid = true;
            }
            else {
                cout << "Input tidak valid";
            }
        }
        else {
            cout << "Input tidak valid";
        }
    }

    return temp;
}

void inputNum (int arr[]) {
// I.S Array arr berukuran 4 dan semua elemen bernilai 0
// F.S Array arr akan terisi dengan angka-angka yang bersesuaian
dengan karakter yang dimasukkan oleh user

// KAMUS LOKAL
    string val;
    int len;
    bool done = false;

```

```

// ALGORITMA
cout <<
"\n=====\\n\\n";
    cout << "Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10,
J, Q, K)\\n\\n";

    while (!done) {
        cout << "Input: ";
        getline(cin, val);
        len = countWords(val);

        if (len != 4) {
            cout << "Input tidak valid\\n";
        }
        else {
            done = getNum(arr, val);
            if (!done) {
                cout << "Input tidak valid\\n";
            }
        }
    }
}

void generateNum (int arr[]) {
// I.S Array arr berukuran empat dan semua nilainya 0
// F.S Array arr terisi dengan angka 1 sampai 13 yang digenerate
secara random

    srand((unsigned)time(0));
    for (int i = 0; i < 4; i++) {
        arr[i] = (rand() % 13) + 1;
    }
}

void saveOpt (string text) {
// I.S solutions berisi solusi dari Permainan Kartu 24
// F.S solutions akan disimpan pada suatu file atau tidak disimpan

```

```

// KAMUS LOKAL
int len;
string fileName, opt;
bool valid = false;
ofstream file;

// ALGORITMA
while (!valid) {
    cout << "\nApakah solusi ingin disimpan ke file (Y/n) ? ";
    getline(cin, opt);
    len = countWords(opt);
    if (len == 1) {
        valid = (opt == "y" or opt == "Y" or opt == "n" or opt
== "N");
        if (!valid) {
            cout << "Input tidak valid";
        }
    }
    else {
        cout << "Input tidak valid";
    }
}

if (opt == "Y" or opt == "y") {
    valid = false;
    while (!valid) {
        cout << "\nMasukkan nama file: ";
        getline(cin, fileName);
        valid = (countWords(fileName) != 0);
        if (!valid) {
            cout << "Input tidak valid";
        }
    }

    fileName = "../test/" + fileName + ".txt";
    file.open(fileName);
    file << text;
    file.close();
}

```

```

        cout << "\nSolusi berhasil disimpan";
    }

    cout << "\nTerima kasih sudah menggunakan 24 Solver";
}

```

2.3 File solver.cpp

```

// File: solver.cpp
// Berisi kumpulan fungsi-fungsi untuk menyelesaikan Permainan
Kartu 24
// dan yang akan digunakan pada file main.cpp

#include "io.cpp"

#include <iostream>
#include <chrono>
using namespace std;
using namespace std::chrono;

bool isAlreadyPermute (int mat[][4], int arr[], int neff) {
    // Mengembalikan true jika arr merupakan salah satu elemen array
    pada mat
    // Mengembalikan false jika tidak

    // KAMUS LOKAL
    bool thereIs = false;
    int i = 0;

    // ALGORITMA
    while (i < neff && !thereIs) {
        if (mat[i][0] == arr[0] && mat[i][1] == arr[1] && mat[i][2]
== arr[2] && mat[i][3] == arr[3]) {
            thereIs = true;
        }
        else {
            i++;
        }
    }
}

```

```

    }

    return thereIs;
}

void addElement (int mat[][4], int arr[], int neff) {
// I.S mat, arr, dan neff terdefinisi
// F.S Menambahkan arr sebagai salah satu elemen mat

// KAMUS LOKAL

// ALGORITMA
    for (int i = 0; i < 4; i++) {
        mat[neff][i] = arr[i];
    }
}

float operation (float num1, int operand, float num2) {
// Mengembalikan hasil operasi antara num1 dan num2 dengan operand
yang sesuai

// KAMUS LOKAL

// ALGORITMA
    if (operand == 1) {
        return num1 + num2;
    }
    else if (operand == 2) {
        return num1 - num2;
    }
    else if (operand == 3) {
        return num1 * num2;
    }
    else {
        return num1 / num2;
    }
}

bool isIt24 (int tempArr[], int groupType, int operandArr[]) {

```

```

// Mengembalikan true jika hasil operasi dengan susunan angka,
operand, dan jenis grouping menghasilkan 24
// Mengembalikan false jika hasil operasi tidak sama dengan 24

// KAMUS LOKAL
float result;

// ALGORITMA
if (groupType == 1) {
    result = operation(operation(tempArr[0], operandArr[0],
tempArr[1]), operandArr[1], operation(tempArr[2], operandArr[2],
tempArr[3]));
}
else if (groupType == 2) {
    result = operation(operation(operation(tempArr[0],
operandArr[0], tempArr[1]), operandArr[1], tempArr[2]),
operandArr[2], tempArr[3]);
}
else if (groupType == 3) {
    result = operation(operation(tempArr[0], operandArr[0],
operation(tempArr[1], operandArr[1], tempArr[2])), operandArr[2],
tempArr[3]);
}
else if (groupType == 4) {
    result = operation(tempArr[0], operandArr[0],
operation(operation(tempArr[1], operandArr[1], tempArr[2]),
operandArr[2], tempArr[3]));
}
else {
    result = operation(tempArr[0], operandArr[0],
operation(tempArr[1], operandArr[1], operation(tempArr[2],
operandArr[2], tempArr[3])));
}

return result == 24.0;
}

string solutionStringGenerator (int arr[], int groupType, int
operandArr[]) {

```



```

// Mengembalikan string berupa teks yang menjadi salah satu solusi
untuk Permainan Kartu 24

// KAMUS LOKAL
    string tempArr[4] = {numToChar(arr[0]), numToChar(arr[1]),
numToChar(arr[2]), numToChar(arr[3])};
// ALGORITMA
    if (groupType == 1) {
        return "(" + (tempArr[0]) + numToOperand(operandArr[0]) +
(tempArr[1]) + ")" +
            numToOperand(operandArr[1]) +
            "(" + (tempArr[2]) + numToOperand(operandArr[2]) +
(tempArr[3]) + ")";
    }
    else if (groupType == 2) {
        return "(" + (tempArr[0]) + numToOperand(operandArr[0]) +
(tempArr[1]) + ")" +
            numToOperand(operandArr[1]) + (tempArr[2]) + ")" +
            numToOperand(operandArr[2]) + (tempArr[3]);
    }
    else if (groupType == 3) {
        return "(" + (tempArr[0]) + numToOperand(operandArr[0]) +
"(" +
            (tempArr[1]) + numToOperand(operandArr[1]) +
(tempArr[2]) + ") +
            numToOperand(operandArr[2]) + (tempArr[3]);
    }
    else if (groupType == 4) {
        return (tempArr[0]) + numToOperand(operandArr[0]) + "(" +
            (tempArr[1]) + numToOperand(operandArr[1]) +
(tempArr[2]) + ")" +
            numToOperand(operandArr[2]) + (tempArr[3]) + ")";
    }
    else {
        return (tempArr[0]) + numToOperand(operandArr[0]) + "(" +
            (tempArr[1]) + numToOperand(operandArr[1]) + "(" +
            (tempArr[2]) + numToOperand(operandArr[2]) +
(tempArr[3]) + ") +
    }
}

```



```
string solutions;

pilihan = solve24mode();
if (pilihan == "1") {
    inputNum(numArr);
}
else {
    generateNum(numArr);
}

solutions = solve24(numArr);
saveOpt(solutions);
return 0;
}
```

BAB III EKSPERIMEN DAN TESTING BERBAGAI SUSUNAN KARTU

3.1 Susunan Kartu 10, K, 2, 6

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 10 K 2 6

=====
```

Untuk susunan kartu: 10, K, 2, 6

Banyak solusi: 22

Solusi Permainan Kartu 24:

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

Waktu eksekusi: 117 mikrosekond

Gambar 3.1 Solusi untuk susunan kartu 10, K, 2, 6

3.2 Susunan Kartu 2, 3, 5, 7

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 2 3 5 7

=====
```

Untuk susunan kartu: 2, 3, 5, 7

Banyak solusi: 46

Solusi Permainan Kartu 24:

1. $(2 + (3 * 5)) + 7$
2. $2 + ((3 * 5) + 7)$
3. $(2 + (5 * 3)) + 7$
4. $2 + ((5 * 3) + 7)$
5. $(2 + 7) + (3 * 5)$
6. $2 + (7 + (3 * 5))$
7. $(2 + 7) + (5 * 3)$
8. $2 + (7 + (5 * 3))$
9. $3 - ((2 - 5) * 7)$
10. $(3 * 5) + (2 + 7)$
11. $((3 * 5) + 2) + 7$
12. $3 + ((5 - 2) * 7)$
13. $(3 * 5) + (7 + 2)$
14. $((3 * 5) + 7) + 2$
15. $(3 * 7) - (2 - 5)$
16. $((3 * 7) - 2) + 5$
17. $3 - (7 * (2 - 5))$
18. $(3 * 7) + (5 - 2)$
19. $((3 * 7) + 5) - 2$
20. $3 + (7 * (5 - 2))$
21. $(5 - 2) + (3 * 7)$
22. $5 - (2 - (3 * 7))$
23. $(5 - 2) + (7 * 3)$
24. $((5 - 2) * 7) + 3$
25. $5 - (2 - (7 * 3))$
26. $(5 * 3) + (2 + 7)$
27. $((5 * 3) + 2) + 7$
28. $(5 * 3) + (7 + 2)$
29. $((5 * 3) + 7) + 2$
30. $(5 + (3 * 7)) - 2$
31. $5 + ((3 * 7) - 2)$
32. $(5 + (7 * 3)) - 2$
33. $5 + ((7 * 3) - 2)$
34. $(7 + 2) + (3 * 5)$

```

35.  $7 + (2 + (3 * 5))$ 
36.  $(7 + 2) + (5 * 3)$ 
37.  $7 + (2 + (5 * 3))$ 
38.  $(7 * 3) - (2 - 5)$ 
39.  $((7 * 3) - 2) + 5$ 
40.  $(7 * 3) + (5 - 2)$ 
41.  $((7 * 3) + 5) - 2$ 
42.  $(7 + (3 * 5)) + 2$ 
43.  $7 + ((3 * 5) + 2)$ 
44.  $(7 * (5 - 2)) + 3$ 
45.  $(7 + (5 * 3)) + 2$ 
46.  $7 + ((5 * 3) + 2)$ 

Waktu eksekusi: 141 mikrosekon

```

Gambar 3.2 Solusi untuk susunan kartu 2, 3, 5, 7

3.3 Susunan Kartu 6, 6, 6, 6

```

=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 6 6 6 6

=====

Untuk susunan kartu: 6, 6, 6, 6

Banyak solusi: 7

Solusi Permainan Kartu 24:
1.  $(6 + 6) + (6 + 6)$ 
2.  $(6 * 6) - (6 + 6)$ 
3.  $((6 + 6) + 6) + 6$ 
4.  $((6 * 6) - 6) - 6$ 
5.  $(6 + (6 + 6)) + 6$ 
6.  $6 + ((6 + 6) + 6)$ 
7.  $6 + (6 + (6 + 6))$ 

Waktu eksekusi: 49 mikrosekon

```

Gambar 3.3 Solusi untuk susunan kartu 6, 6, 6, 6

3.4 Susunan Kartu 7, 8, 5, 3

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 7 8 5 3

=====
```

Untuk susunan kartu: 7, 8, 5, 3

Banyak solusi: 30

Solusi Permainan Kartu 24:

1. $(7 * (8 - 5)) + 3$
2. $(7 * 5) - (8 + 3)$
3. $((7 * 5) - 8) - 3$
4. $(7 * 5) - (3 + 8)$
5. $((7 * 5) - 3) - 8$
6. $(7 * 3) + (8 - 5)$
7. $((7 * 3) + 8) - 5$
8. $(7 * 3) - (5 - 8)$
9. $((7 * 3) - 5) + 8$
10. $(8 + (7 * 3)) - 5$
11. $8 + ((7 * 3) - 5)$
12. $(8 - 5) + (7 * 3)$
13. $((8 - 5) * 7) + 3$
14. $8 - (5 - (7 * 3))$
15. $(8 - 5) + (3 * 7)$
16. $8 - (5 - (3 * 7))$
17. $(8 + (3 * 7)) - 5$
18. $8 + ((3 * 7) - 5)$
19. $(5 * 7) - (8 + 3)$
20. $((5 * 7) - 8) - 3$
21. $(5 * 7) - (3 + 8)$
22. $((5 * 7) - 3) - 8$
23. $(3 * 7) + (8 - 5)$
24. $((3 * 7) + 8) - 5$
25. $3 + (7 * (8 - 5))$
26. $(3 * 7) - (5 - 8)$
27. $((3 * 7) - 5) + 8$
28. $3 - (7 * (5 - 8))$
29. $3 + ((8 - 5) * 7)$
30. $3 - ((5 - 8) * 7)$

Waktu eksekusi: 129 mikrosekon

Gambar 3.4 Solusi untuk susunan kartu 7, 8, 5, 3

3.5 Susunan Kartu 9, 4, 4, 2

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 9 4 4 2

=====

Untuk susunan kartu: 9, 4, 4, 2

Banyak solusi: 2

Solusi Permainan Kartu 24:
1.  $((9 - 2) * 4) - 4$ 
2.  $(4 * (9 - 2)) - 4$ 

Waktu eksekusi: 65 mikrosekond
```

Gambar 3.5 Solusi untuk susunan kartu 9, 4, 4, 2

3.6 Susunan Kartu 10, 6, 4, 2

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 10 6 4 2

=====
```

Untuk susunan kartu: 10, 6, 4, 2

Banyak solusi: 76

Solusi Permainan Kartu 24:

1. $(10 + 6) + (4 * 2)$
2. $(10 - 6) * (4 + 2)$
3. $((10 + 6) - 4) * 2$
4. $(10 + (6 - 4)) * 2$
5. $10 + (6 + (4 * 2))$
6. $(10 + 6) + (2 * 4)$
7. $(10 - 6) * (2 + 4)$
8. $((10 - 6) + 2) * 4$
9. $(10 - (6 - 2)) * 4$
10. $10 + (6 + (2 * 4))$
11. $(10 - 4) * (6 - 2)$
12. $((10 - 4) + 6) * 2$
13. $(10 - (4 - 6)) * 2$
14. $((10 - 4) - 2) * 6$
15. $(10 + (4 * 2)) + 6$
16. $(10 - (4 + 2)) * 6$
17. $10 + ((4 * 2) + 6)$
18. $(10 + 2) * (6 - 4)$
19. $((10 + 2) - 6) * 4$
20. $(10 + (2 - 6)) * 4$
21. $((10 - 2) - 4) * 6$
22. $(10 + (2 * 4)) + 6$
23. $(10 - (2 + 4)) * 6$
24. $10 + ((2 * 4) + 6)$
25. $(6 + 10) + (4 * 2)$
26. $((6 + 10) - 4) * 2$
27. $(6 + (10 - 4)) * 2$
28. $6 * ((10 - 4) - 2)$
29. $6 + (10 + (4 * 2))$
30. $6 * (10 - (4 + 2))$
31. $(6 + 10) + (2 * 4)$
32. $6 * ((10 - 2) - 4)$
33. $6 + (10 + (2 * 4))$
34. $6 * (10 - (2 + 4))$

```

35. (6 - 4) * (10 + 2)
36. ((6 - 4) + 10) * 2
37. (6 - (4 - 10)) * 2
38. (6 - 4) * (2 + 10)
39. (6 + (4 * 2)) + 10
40. 6 + ((4 * 2) + 10)
41. (6 - 2) * (10 - 4)
42. (6 + (2 * 4)) + 10
43. 6 + ((2 * 4) + 10)
44. 4 * ((10 - 6) + 2)
45. 4 * (10 - (6 - 2))
46. (4 - 10) * (2 - 6)
47. 4 * ((10 + 2) - 6)
48. 4 * (10 + (2 - 6))
49. (4 + 2) * (10 - 6)
50. (4 * 2) + (10 + 6)
51. ((4 * 2) + 10) + 6
52. 4 * ((2 + 10) - 6)
53. 4 * (2 + (10 - 6))
54. (4 * 2) + (6 + 10)
55. ((4 * 2) + 6) + 10
56. 4 * ((2 - 6) + 10)
57. 4 * (2 - (6 - 10))
58. (2 + 10) * (6 - 4)
59. ((2 + 10) - 6) * 4
60. (2 + (10 - 6)) * 4

```

```

61. 2 * ((10 + 6) - 4)
62. 2 * (10 + (6 - 4))
63. 2 * ((10 - 4) + 6)
64. 2 * (10 - (4 - 6))
65. ((2 - 6) + 10) * 4
66. (2 - (6 - 10)) * 4
67. 2 * ((6 + 10) - 4)
68. 2 * (6 + (10 - 4))
69. (2 - 6) * (4 - 10)
70. 2 * ((6 - 4) + 10)
71. 2 * (6 - (4 - 10))
72. (2 + 4) * (10 - 6)
73. (2 * 4) + (10 + 6)
74. ((2 * 4) + 10) + 6
75. (2 * 4) + (6 + 10)
76. ((2 * 4) + 6) + 10

```

Waktu eksekusi: 173 mikrosekon

Gambar 3.6 Solusi untuk susunan kartu 10, 6, 4, 2

3.7 Susunan Kartu 2, 9, 10, 10

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: 2 9 10 10

=====

Untuk susunan kartu: 2, 9, 10, 10

Banyak solusi: 12

Solusi Permainan Kartu 24:
1.  $(9 + (10 / 2)) + 10$ 
2.  $9 + ((10 / 2) + 10)$ 
3.  $(9 + 10) + (10 / 2)$ 
4.  $9 + (10 + (10 / 2))$ 
5.  $(10 / 2) + (9 + 10)$ 
6.  $((10 / 2) + 9) + 10$ 
7.  $(10 / 2) + (10 + 9)$ 
8.  $((10 / 2) + 10) + 9$ 
9.  $(10 + 9) + (10 / 2)$ 
10.  $10 + (9 + (10 / 2))$ 
11.  $(10 + (10 / 2)) + 9$ 
12.  $10 + ((10 / 2) + 9)$ 

Waktu eksekusi: 76 mikrosekond
```

Gambar 3.7 Solusi untuk susunan kartu 2, 9, 10, 10

3.8 Susunan Kartu A, K, Q, J

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: A K Q J

=====

Untuk susunan kartu: A, K, Q, J

Banyak solusi: 32

Solusi Permainan Kartu 24:
1. ((A * K) - J) * Q
2. (A * (K - J)) * Q
3. A * ((K - J) * Q)
4. (A * Q) * (K - J)
5. A * (Q * (K - J))
6. ((K * A) - J) * Q
7. ((K / A) - J) * Q
8. (K - (A * J)) * Q
9. (K - J) * (A * Q)
10. (K - J) / (A / Q)
11. ((K - J) * A) * Q
12. ((K - J) / A) * Q
13. (K - (J * A)) * Q
14. (K - (J / A)) * Q
15. (K - J) * (Q * A)
16. (K - J) * (Q / A)
17. ((K - J) * Q) * A
18. ((K - J) * Q) / A
19. (Q * A) * (K - J)
20. (Q / A) * (K - J)
21. Q * ((A * K) - J)
22. Q * (A * (K - J))
23. Q / (A / (K - J))
24. Q * ((K * A) - J)
25. Q * ((K / A) - J)
26. Q * (K - (A * J))
27. (Q * (K - J)) * A
28. (Q * (K - J)) / A
29. Q * ((K - J) * A)
30. Q * ((K - J) / A)
31. Q * (K - (J * A))
32. Q * (K - (J / A))

Waktu eksekusi: 244 mikrosekond
```

Gambar 3.8 Solusi untuk susunan kartu A, K, Q, J

3.9 Susunan Kartu K, A, 4, 9

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: K A 4 9

=====

Untuk susunan kartu: K, A, 4, 9

Banyak solusi: 16

Solusi Permainan Kartu 24:
1. (A - K) + (4 * 9)
2. A - (K - (4 * 9))
3. (A - K) + (9 * 4)
4. A - (K - (9 * 4))
5. (A + (4 * 9)) - K
6. A + ((4 * 9) - K)
7. (A + (9 * 4)) - K
8. A + ((9 * 4) - K)
9. (4 * 9) - (K - A)
10. ((4 * 9) - K) + A
11. (4 * 9) + (A - K)
12. ((4 * 9) + A) - K
13. (9 * 4) - (K - A)
14. ((9 * 4) - K) + A
15. (9 * 4) + (A - K)
16. ((9 * 4) + A) - K

Waktu eksekusi: 123 mikrosekond
```

Gambar 3.9 Solusi untuk susunan kartu K, A, 4, 9

3.10 Susunan Kartu Q, 5, A, A

```
=====
Selamat datang di 24 Solver. Silakan pilih mode yang anda inginkan:
1. Input angka kartu oleh pengguna
2. Generate angka kartu secara random
Masukkan pilihan mode (1-2): 1

=====

Mohon masukkan 4 angka atau huruf yang sesuai (A, 2-10, J, Q, K)

Input: Q 5 A A

=====

Untuk susunan kartu: Q, 5, A, A

Banyak solusi: 0

Tidak ada solusi yang mungkin

Waktu eksekusi: 96 mikrosekond
```

Gambar 3.10 Solusi untuk susunan kartu Q, 5, A, A

LAMPIRAN

Link To Repository:

[GoDillonAudris512/Tucil1_13521062: Repository Tugas Kecil 1 Strategi Algoritma Go Dillon Audris \(github.com\)](https://github.com/GoDillonAudris512/Tucil1_13521062)