

**IF2211 – STRATEGI ALGORITMA**  
**LAPORAN TUGAS KECIL I**



**YOBEL DEAN CHRISTOPHER**

**13521067**

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

## DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
ALGORITMA BRUTE FORCE.....	3
BAB II.....	4
SOURCE CODE .....	4
BAB III .....	13
HASIL MENJALANKAN PROGRAM .....	13
BAB IV .....	18
LINK TO REPOSITORY .....	18

# BAB I

## ALGORITMA BRUTE FORCE

Algoritma Brute Force adalah metode pencarian atau pemecahan masalah dengan mencoba semua kemungkinan yang ada secara berurutan sampai ditemukan solusi yang sesuai (*straightforward*).

Program 24 Card Game Solver menggunakan Algoritma Brute Force. Deskripsi langkah-langkah Algoritma Brute Force yang digunakan pada program tersebut adalah sebagai berikut:

- Setiap ekspresi matematika yang akan dievaluasi memiliki 4 angka dan 3 operator (+, -, \*, /).
- Setiap kemungkinan angka akan dicari menggunakan permutasi.
- Setiap kemungkinan operator akan dicari menggunakan kombinasi.
- Terdapat 5 kemungkinan penggunaan tanda kurung. Yakni,
  - $(a \text{ op } b) \text{ op } (c \text{ op } d)$
  - $(a \text{ op } (b \text{ op } c)) \text{ op } d$
  - $a \text{ op } (b \text{ op } (c \text{ op } d))$
  - $a \text{ op } ((b \text{ op } c) \text{ op } d)$
  - $((a \text{ op } b) \text{ op } c) \text{ op } d$
- Setiap kemungkinan angka dan operator diletakkan sesuai dengan posisinya.
- Setiap kemungkinan tersebut akan dievaluasi untuk mencari hasil 24.

Untuk contoh pelaksanaan program adalah sebagai berikut,

1. Salah satu kemungkinan permutasi dari 4 angka adalah [5 7 3 4].
2. Salah satu kemungkinan permutasi dari 3 operator adalah [ + + \* ].
3. Salah satu kemungkinan penggunaan tanda kurung adalah  $(a \text{ op } b) \text{ op } (c \text{ op } d)$ .  
Kemungkinan permutasi angka dan kemungkinan permutasi operator ditempatkan pada variabel yang sesuai sehingga menghasilkan suatu ekspresi matematika yaitu  $(5 + 7) + (3 * 4)$ .
4. Dari hasil ekspresi matematika tersebut akan dievaluasi. Apabila hasilnya adalah 24, akan ditambahkan pada solusi.
5. Program akan mengganti kemungkinan penggunaan tanda kurung terlebih dahulu, kemudian kemungkinan operator, dan yang terakhir kemungkinan angka (program berjalan mundur).

## BAB II

### SOURCE CODE

```
#include <iostream>
#include <vector>
#include <string>
#include <stack>
#include <chrono>
#include <fstream>

using namespace std;

// count the number of solutions
int countify = 0;

// Function to determine if a string is a valid card
int validate(string a, string b, string c, string d) {
    if ((a == "A" || a == "2" || a == "3" || a == "4" || a == "5" || a == "6"
    || a == "7" || a == "8" || a == "9" || a == "10" || a == "J" || a == "Q" || a
    == "K")
        && (b == "A" || b == "2" || b == "3" || b == "4" || b == "5" || b ==
    "6" || b == "7" || b == "8" || b == "9" || b == "10" || b == "J" || b == "Q"
    || b == "K")
        && (c == "A" || c == "2" || c == "3" || c == "4" || c == "5" || c ==
    "6" || c == "7" || c == "8" || c == "9" || c == "10" || c == "J" || c == "Q"
    || c == "K")
        && (d == "A" || d == "2" || d == "3" || d == "4" || d == "5" || d ==
    "6" || d == "7" || d == "8" || d == "9" || d == "10" || d == "J" || d == "Q"
    || d == "K")) {
        return 1;
    }
    else {
        return 0;
    }
}

// Function to determine user's choice (input themselves, randomly generated,
or exit)
int choosify(){
    int choice;
    while(true){
        cout << "Choose one of the below!" << endl;
        cout << "1. MYSELF, I'll input my numbers myself." << endl;
        cout << "2. RANDOMLY, let the machine generate randomly." << endl;
        cout << "3. BYEBYE, I'm done." << endl;
        cout << "Your choice : ";
        cin >> choice;
```

```

        cout << endl;
        if (choice == 1 || choice == 2 || choice == 3){
            break;
        }
        cout << endl;
        cout << "Please enter 1, 2, or 3." << endl;
    }
    return choice;
}

// Function to evaluate an expression in postfix notation
double evalify(std::string expression) {
    std::stack<double> values;
    std::stack<char> operators;

    // check if the input string is a valid mathematical expression
    for (int i = 0; i < expression.length(); i++) {
        // check if current character is a digit
        if (isdigit(expression[i])) {
            // get the complete number
            double num = 0;
            while (i < expression.length() && isdigit(expression[i])) {
                num = num * 10 + (expression[i] - '0');
                i++;
            }
            i--;
            values.push(num);
        }
        // check if current character is an operator or a parenthesis
        else if (expression[i] == '+' || expression[i] == '-' || expression[i]
== '*' || expression[i] == '/' || expression[i] == '(' || expression[i] ==
')') {
            if (expression[i] == '(') {
                operators.push(expression[i]);
            } else if (expression[i] == ')') {
                while (operators.top() != '(') {
                    double num2 = values.top();
                    if (values.empty()) {
                        return -1;
                    }
                    values.pop();
                    double num1 = values.top();
                    if (values.empty()) {
                        return -1;
                    }
                    values.pop();
                    char op = operators.top();
                    if (operators.empty()) {

```

```

        return -1;
    }
    operators.pop();
    double res = 0;
    if (op == '+') {
        res = num1 + num2;
    } else if (op == '-') {
        res = num1 - num2;
    } else if (op == '*') {
        res = num1 * num2;
    } else if (op == '/') {
        if(num2 == 0){
            return -1;
        }
        res = num1 / num2;
    }
    values.push(res);
    if (operators.empty()) {
        return -1;
    }
}
if (operators.empty()) {
    return -1;
}
operators.pop();
} else {
    while (!operators.empty() && operators.top() != '(') {
        double num2 = values.top();
        double num1 = values.top();
        if (values.empty()) {
            return -1;
        }
        values.pop();
        char op = operators.top();
        if (operators.empty()) {
            return -1;
        }
        operators.pop();
        double res = 0;
        if (op == '+') {
            res = num1 + num2;
        } else if (op == '-') {
            res = num1 - num2;
        } else if (op == '*') {
            res = num1 * num2;
        } else if (op == '/') {
            if(num2 == 0){

```

```

        return -1;
    }
    cout << "YES 5" << endl;
    res = num1 / num2;
}
values.push(res);
}
operators.push(expression[i]);
}
}
}
while (!operators.empty()) {
    if (values.empty()) {
        return -1;
    }
    double num2 = values.top();
    if (values.empty()) {
        return -1;
    }
    values.pop();
    if (values.empty()) {
        return -1;
    }
    double num1 = values.top();
    values.pop();
    char op = operators.top();
    if (operators.empty()) {
        return -1;
    }
    operators.pop();
    double res = 0;
    if (op == '+') {
        res = num1 + num2;
    } else if (op == '-') {
        res = num1 - num2;
    } else if (op == '*') {
        res = num1 * num2;
    } else if (op == '/') {
        if(num2 == 0){
            return -1;
        }
        res = num1 / num2;
    } else {
        return -1;
    }
    values.push(res);
}
return values.top();

```

```

}

// Function to swap two elements in a vector
void swapify(string &a, string &b) {
    string temp = a;
    a = b;
    b = temp;
}

vector<string> theSolution;
// The Brute Force Algorithm
void permutify(vector<string> &vec, int l, int r) {
    // Base condition: if left index is equal to right index, print the vector
    if (l == r) {
        vector<string> permuteNum;

        for (int i = 0; i < vec.size(); i++) {
            permuteNum.push_back(vec[i]);
        }

        vector<string> operator1 = {"+", "-", "*", "/"};
        vector<string> operator2 = {"+", "-", "*", "/"};
        vector<string> operator3 = {"+", "-", "*", "/"};

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                for (int k = 0; k < 4; k++) {

                    // Parentheses position 1 (a op b) op (c op d)
                    string parentheses1 = "(" + permuteNum[0] + operator1[i] +
permuteNum[1] + ")" + operator2[j] + "(" + permuteNum[2] + operator3[k] +
permuteNum[3] + ")";

                    // Parentheses position 2 a op (b op (c op d))
                    string parentheses2 = permuteNum[0] + operator1[i] + "(" +
permuteNum[1] + operator2[j] + "(" + permuteNum[2] + operator3[k] +
permuteNum[3] + "))";

                    // Parentheses position 3 a op ((b op c) op d)
                    string parentheses3 = permuteNum[0] + operator1[i] + "((("
+ permuteNum[1] + operator2[j] + permuteNum[2] + ")" + operator3[k] +
permuteNum[3] + ")";

                    // Parentheses position 4 (a op (b op c)) op d
                    string parentheses4 = "(" + permuteNum[0] + operator1[i] +
 "(" + permuteNum[1] + operator2[j] + permuteNum[2] + ")") + operator3[k] +
permuteNum[3];

                    // Parentheses position 5 ((a op b) op c) op d
                    string parentheses5 = "((((" + permuteNum[0] + operator1[i]
+ permuteNum[1] + ")" + operator2[j] + permuteNum[2] + ")") + operator3[k] +
permuteNum[3];

```



```

        for (int l = 0; l < 5; l++) {
            string parentheses;
            if (l == 0) {
                parentheses = parentheses1;
            } else if (l == 1) {
                parentheses = parentheses2;
            } else if (l == 2) {
                parentheses = parentheses3;
            } else if (l == 3) {
                parentheses = parentheses4;
            } else {
                parentheses = parentheses5;
            }
            double result = evalify(parentheses);
            if (result == 24) {
                theSolution.push_back(parentheses);
                countify += 1;
            }
        }
    }
} else {
    // Loop to swap elements at the left index with other elements
    for (int i = 1; i <= r; i++) {
        swapify(vec[l], vec[i]);
        permutify(vec, l+1, r);
        swapify(vec[l], vec[i]);
    }
}
}

// MAIN PROGRAM
int main() {
    cout << "Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!" <<
endl;
    cout << endl;

    vector<string> cards;
    vector<string> card0;
    string card;

    // User's choice to input themselves, randomly generated, or exit
    int choice = choosify();
    if (choice == 1) { // User inputs cards

```

```

        cout << "Remember, your cards are [A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J,
Q, K]" << endl;
        cout << "Enter your cards, Player: ";
        while(true) {
            for(int i = 0; i < 4; i++){
                cin >> card;
                card0.push_back(card);
            }
            if (validify(card0[0], card0[1], card0[2], card0[3]) == 1) {
                break;
            } else {
                card0.clear();
                cout << "Invalid input, please try again: ";
            }
        }

        for (int i = 0; i < 4; i++) {
            if (card0[card0.size() - i - 1] == "A") {
                card0[card0.size() - i - 1] = "1";
            } else if (card0[card0.size() - i - 1] == "J") {
                card0[card0.size() - i - 1] = "11";
            } else if (card0[card0.size() - i - 1] == "Q") {
                card0[card0.size() - i - 1] = "12";
            } else if (card0[card0.size() - i - 1] == "K") {
                card0[card0.size() - i - 1] = "13";
            }
            cards.push_back(card0[card0.size() - i - 1]);
        }

    } else if (choice == 2){ // Randomly generated cards
        cout << endl;
        cout << "The Machine have chosen 4 cards for you." << endl;
        unsigned seed =
chrono::system_clock::now().time_since_epoch().count();
        srand(seed);
        cout << "The numbers are : ";
        for(int i = 0; i < 4; i++){
            card = to_string(1 + (rand() % 13));
            cout << card << " ";
            cards.push_back(card);
        }
        cout << endl;
    } else { // Exit
        cout << endl;
        cout << "Program ended, bye-bye player." << endl;
        return 0;
    }
}

```

```

// Begin timing
auto begin = std::chrono::high_resolution_clock::now();

// The Brute Force Algorithm
permotify(cards, 0, cards.size()-1);

// End timing
auto end = std::chrono::high_resolution_clock::now();
auto diff = std::chrono::duration_cast<std::chrono::nanoseconds>(end -
begin);

cout << endl;
// Print out the number of solutions found
if (countify == 0) {
    cout << "NO solutions found." << endl;
} else {
    cout << countify << " solutions found." << endl;
}

// Print out the solutions
if (countify > 0){
    cout << "Here you go, the solutions to MAKE IT 24 are: " << endl;
    cout << endl;
    for (int i = 0; i < theSolution.size(); i++) {
        cout << theSolution[i] << endl;
    }
}
cout << endl;
// Print out the time needed
cout << diff.count() * 1e-9 << " seconds needed." << endl;

// Save the solutions to a file
while(countify> 0){
    cout << "Do you want to save the solution? (y/n) : ";
    char saveChoice;
    cin >> saveChoice;
    cout << endl;
    if(saveChoice == 'y' || saveChoice == 'Y'){
        string fileName;
        cout << "Enter file name : ";
        cin >> fileName;
        cout << endl;
        ofstream myFile;
        myFile.open("test/" + fileName + ".txt");
        for (int i = 0; i < theSolution.size(); i++) {
            myFile << theSolution[i] << endl;
        }
        myFile.close();
    }
}

```

```
        cout << "File " << fileName << ".txt is saved." << endl;
        cout << endl;
        break;
    } else if (saveChoice == 'n' || saveChoice == 'N'){
        cout << "Program ended, bye-bye player." << endl;
        break;
    }
}

return 0;
}
```

## BAB III

### HASIL MENJALANKAN PROGRAM

#### 1) User Input sendiri dan File disave

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Choose one of the below!
1. MYSELF, I'll input my numbers myself.
2. RANDOMLY, let the machine generate randomly.
3. BYEBYE, I'm done.
Your choice : 1

Remember, your cards are [A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K]
Enter your cards, Player: 9 7 10 8

8 solutions found.
Here you go, the solutions to MAKE IT 24 are:

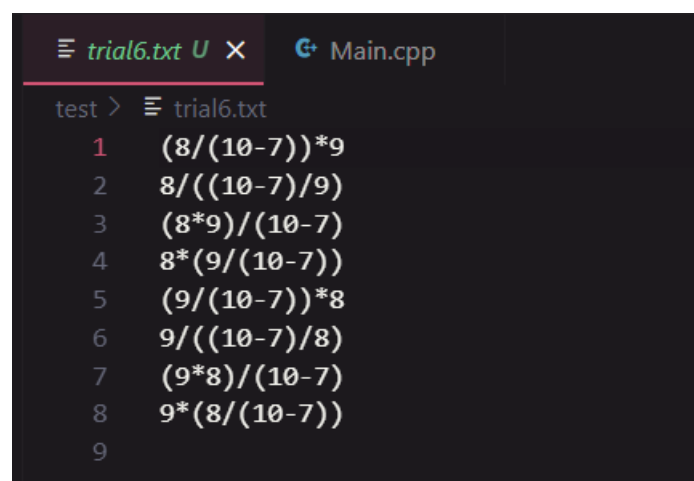
(8/(10-7))*9
8/((10-7)/9)
(8*9)/(10-7)
8*(9/(10-7))
(9/(10-7))*8
9/((10-7)/8)
(9*8)/(10-7)
9*(8/(10-7))

0.0165744 seconds needed.
Do you want to save the solution? (y/n) : y

Enter file name : trial6

File trial6.txt is saved.
```

#### 2) File berhasil disave.



The screenshot shows a code editor with two tabs: 'trial6.txt' (active) and 'Main.cpp'. The 'trial6.txt' tab contains a list of 8 mathematical expressions, each on a new line, numbered 1 through 8. The expressions are the same as those shown in the previous terminal output. The editor has a dark theme and a light-colored border around the code area.

```
test > trial6.txt
1 (8/(10-7))*9
2 8/((10-7)/9)
3 (8*9)/(10-7)
4 8*(9/(10-7))
5 (9/(10-7))*8
6 9/((10-7)/8)
7 (9*8)/(10-7)
8 9*(8/(10-7))
9
```

#### 3) User Input Sendiri dan File TIDAK di save.

Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Remember, your cards are [A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K]

Enter your cards, Player: J 9 Q 7

18 solutions found.

Here you go, the solutions to MAKE IT 24 are:

$((7+11)/9)*12$   
 $(7+11)/(9/12)$   
 $(7+11)*(12/9)$   
 $((7+11)*12)/9$   
 $12*((7+11)/9)$   
 $(12*(7+11))/9$   
 $(12/9)*(7+11)$   
 $12/(9/(7+11))$   
 $(12/9)*(11+7)$   
 $12/(9/(11+7))$   
 $12*((11+7)/9)$   
 $(12*(11+7))/9$   
 $(9*(11-7))-12$   
 $((11+7)/9)*12$   
 $(11+7)/(9/12)$   
 $((11-7)*9)-12$   
 $(11+7)*(12/9)$   
 $((11+7)*12)/9$

0.0590802 seconds needed.

Do you want to save the solution? (y/n) : n

Program ended, bye-bye player.

#### 4) Program meng-generate Angka secara RANDOM

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Choose one of the below!
1. MYSELF, I'll input my numbers myself.
2. RANDOMLY, let the machine generate randomly.
3. BYEBYE, I'm done.
Your choice : 2

The Machine have chosen 4 cards for you.
The numbers are : 11 6 6 7

8 solutions found.
Here you go, the solutions to MAKE IT 24 are:

(11*6)-(6*7)
(11*6)-(7*6)
(11*6)-(6*7)
(11*6)-(7*6)
(6*11)-(6*7)
(6*11)-(7*6)
(6*11)-(6*7)
(6*11)-(7*6)

0.0207024 seconds needed.
Do you want to save the solution? (y/n) : y

Enter file name : trial7

File trial7.txt is saved.
```

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Choose one of the below!
1. MYSELF, I'll input my numbers myself.
The numbers are : 8 3 5 6

30 solutions found.
Here you go, the solutions to MAKE IT 24 are:

(8*3)*(6-5)
8*(3*(6-5))
(8*3)/(6-5)
8*(3/(6-5))
(8/(5-3))*6
8/((5-3)/6)
8*(5-(6/3))
8*((6-5)*3)
(8*(6-5))^3
(8*6)/(5-3)
8*(6/(5-3))
(8/(6-5))^3
8/((6-5)/3)
(3*8)*(6-5)
3*(8*(6-5))
(3*8)/(6-5)
3*(8/(6-5))
3*((6-5)*8)
(3*(6-5))*8
(3/(6-5))*8
3/((6-5)/8)
(5-(6/3))*8
(6-5)*(3*8)
((6-5)*3)*8
(6/(5-3))*8
6/((5-3)/8)
(6-5)*(8*3)
((6-5)*8)^3
(6*8)/(5-3)
6*(8/(5-3))

0.0195426 seconds needed.
Do you want to save the solution? (y/n) : n

Program ended, bye-bye player.
```

## 5) Tidak ada solusi

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Choose one of the below!
1. MYSELF, I'll input my numbers myself.
2. RANDOMLY, let the machine generate randomly.
3. BYEBYE, I'm done.
Your choice : 1

Remember, your cards are [A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K]
Enter your cards, Player: 7 7 7 7

NO solutions found.

0.0178687 seconds needed.
```

## 6) Input Tidak Valid

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!

Choose one of the below!
1. MYSELF, I'll input my numbers myself.
2. RANDOMLY, let the machine generate randomly.
3. BYEBYE, I'm done.
Your choice : 1

Remember, your cards are [A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K]
Enter your cards, Player: 14 3 5 6
Invalid input, please try again: 15 3 5 6
Invalid input, please try again: 16 3 5 6
Invalid input, please try again: 100 3 5 7
Invalid input, please try again: K 3 5 8

8 solutions found.
Here you go, the solutions to MAKE IT 24 are:

(8*5)-(3+13)
((8*5)-3)-13
(8*5)-(13+3)
((8*5)-13)-3
(5*8)-(3+13)
((5*8)-3)-13
(5*8)-(13+3)
((5*8)-13)-3

0.0175597 seconds needed.
Do you want to save the solution? (y/n) : y

Enter file name : trial8

File trial8.txt is saved.
```



## 7) Keluar Program

```
Welcome to MAKE IT 24, guaranteed to solve your 24 Card Game!
```

```
Choose one of the below!
```

1. MYSELF, I'll input my numbers myself.
2. RANDOMLY, let the machine generate randomly.
3. BYEBYE, I'm done.

```
Your choice : 3
```

```
Program ended, bye-bye player.
```

Semua fitur pada tabel telah terpenuhi.

Poin	Judul Fitur	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	✓	
2	Program berhasil running	✓	
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	✓	

## **BAB IV**

### **LINK TO REPOSITORY**

[https://github.com/yobeldc/Tucil1\\_13521067.git](https://github.com/yobeldc/Tucil1_13521067.git)